Exploring Advertising Manipulation in Diffusion Image Generation

Tianshi Che
Auburn University
Auburn, AL, USA
tzc0029@auburn.edu

Yang Zhou*
Auburn University
Auburn, AL, USA
yangzhou@auburn.edu

Yushan Mu
Auburn University
Auburn, AL, USA
yzm0016@auburn.edu

Zeru Zhang
Auburn University
Auburn, AL, USA
zzz0054@auburn.edu

Zijie Zhang
University of Texas at San Antonio
San Antonio, TX, USA
zijie.zhang@utsa.edu

Panqiu Xia
Cardiff University
Cardiff, UK
xiap@cardiff.ac.uk

Wei Zhang
University of Central Florida
Orlando, FL, USA
wzhang.cs@ucf.edu

Longwei Wang
University of South Dakota
Vermillion, SD, USA
longwei.wang@usd.edu

Yelong Shen
Microsoft
Redmond, WA, USA
yelong.shen@microsoft.com

Ruoming Jin
Kent State University
Kent, OH, USA
rjin1@kent.edu

Jianfeng Gao
Microsoft Research
Redmond, WA, USA
jfgao@microsoft.com

Abstract

As text-to-image diffusion models (T2I DMs) become widely deployed, adversarial advertising has emerged as a realistic threat: an attacker may compromise a T2I DM so that it implants target product brands into images generated from users' non-advertising prompts. Two key challenges remain largely unresolved in this setting: achieving natural and semantically coherent adversarial advertisement and ensuring robust adversarial advertisement. To address these challenges, we develop a new estimation algorithm for the multivariate continuously scaled phase-type with Lévy (MCPHL) distribution to capture the intrinsic distribution of natural advertisement prompts in the prompt-embedding space. With the estimated MCPHL, we construct an attack that pushes non-advertising prompts toward high-density regions of this distribution, making the resulting perturbed prompts better aligned with natural advertising prompts. We then propose a masked parameter smoothing approach grounded in mollification theory, yielding a smoothed T2I DM equipped with a dimension-invariant certified guarantee against advertisement degradation under model fine-tuning. The masking mechanism preserves utility by avoiding unnecessary smoothing on sensitive parameters, and our theoretical analysis shows that the smoothed model better preserves adversarial advertisements under fine-tuning, while maintaining better generation quality.

*Corresponding authors

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

1 Introduction

Text-to-image diffusion models (T2I DMs) encode natural-language prompts into text embeddings and use the embedding to condition a denoising network, generate high-quality images [117, 125, 142, 162, 144, 115, 133, 61]. However, recent studies have shown that T2I DMs are vulnerable to backdoor attacks [175, 96, 199, 38], including bias injection [151], harmful information generation [199], or utility degradation [38], while the model behaves normally without the trigger.

With evolving developments of Generative AI, T2I DM is playing an increasingly significant role in online advertising [45, 216, 173, 28, 29, 183]. These advertising techniques aim to produce "benign advertisements", where advertisers intentionally utilize the T2I DMs to generate targeted advertisements, by providing explicit descriptions about the advertised target. In contrast, "adversarial advertisement" tampers with a T2I model, causing non-advertising prompts to quietly produce images that are naturally blended with advertisements, without user intent or consent. An adversary (e.g., a malicious marketer) has strong incentives to use this tactic to increase brand exposure, shape positive user sentiment, and ultimately raise revenue [175].

A straightforward way to implement adversarial advertising in T2I DMs is to adapt existing backdoor attack methods [175, 96, 199, 38] to achieve the advertisement implantation in T2I DMs. An attacker associates a carefully designed trigger with a target brand image via model fine-tuning. Once the attack is completed, the victim T2I DMs generate an image with the implanted target image [175, 96, 199, 38] upon detection of a trigger. Despite achieving remarkable performance, existing backdoor attack approaches against T2I DMs often rely on unusual, unnatural, or out-of-context prompt tokens as triggers [96, 199, 38], such as swapping the position of two characters (e.g., swapping "io" in the word "diffusion" to get "diffusoin") [96], replacing a character in a word (e.g., replacing letter l with number 1 in "Alphabet") [96], or adding a contextless word to the prompt (e.g. "A drawing of a blue cat. mignneko" where "mignneko" is the trigger) [38]. However, the usage of unusual, unnatural, or out-of-context prompt tokens in daily life is limited [175] since these tokens increase the risk of backdoor attacks being detected by grammar correction tools or defender programs. As a result, the backdoor attack techniques are impractical for the real-world adversarial advertisement problem [175].

The adversarial-advertising problem in T2I DMs is underexplored. To our knowledge, BAGM [175] is the first work to inject advertisements without using unusual or out-of-context triggers, improving success rates and lowering detection. Yet two critical challenges remain: (1) Imperceptibility. Natural language is heavy-tailed [66, 202, 63]. While BAGM avoids unnatural triggers, it does not consider the latent language distribution and thus cannot reliably yield more natural ads. A detailed analysis of this heavy-tail behavior is in Appendix A.3. (2) Robustness. The perturbed T2I DMs can be easily recovered to their clean versions by fine-tuning them on clean training datasets, losing the ability to generate the adversarial advertisements.

Building on the intuition of BAGM, we propose a new adversarial-advertising framework for T2I DMs that directly targets these two challenges. Our method explicitly models the heavy-tailed distribution of natural language prompts via a heavy-tailed multivariate continuously scaled phase-type distribution with a Lévy component and uses mollification theory to regularize the training objective. This design allows us to generate adversarial advertisements that remain natural and semantically coherent while substantially improving robustness of the perturbed T2I DMs against subsequent fine-tuning on clean data.

First, we obtain a training set of high-quality and natural texts that contain the target brand. The heavy-tailed continuously scaled phase-type distribution can be used to approximate various heavy-tail distributions [3]. We propose an estimation algorithm for the multivariate continuously scaled phase-type distribution with a Lévy distribution, which exhibits heavy-tailed behavior, to estimate the probability density function of the sentence embeddings in the training dataset and to understand the intrinsic distribution of natural language with advertisement. Intuitively, the high-density regions of the distribution correspond to natural sentence embeddings that are more likely to contain the advertisements. By pushing the embeddings of non-advertising prompts to dense regions onto this estimated distribution, the perturbed sentence embeddings become closer to many natural sentence embeddings with advertisements.

Second, we propose a masked mollification method to certify adversarial advertisements against model fine-tuning by establishing a connection to randomized smoothing. We analogize fine-tuning–induced

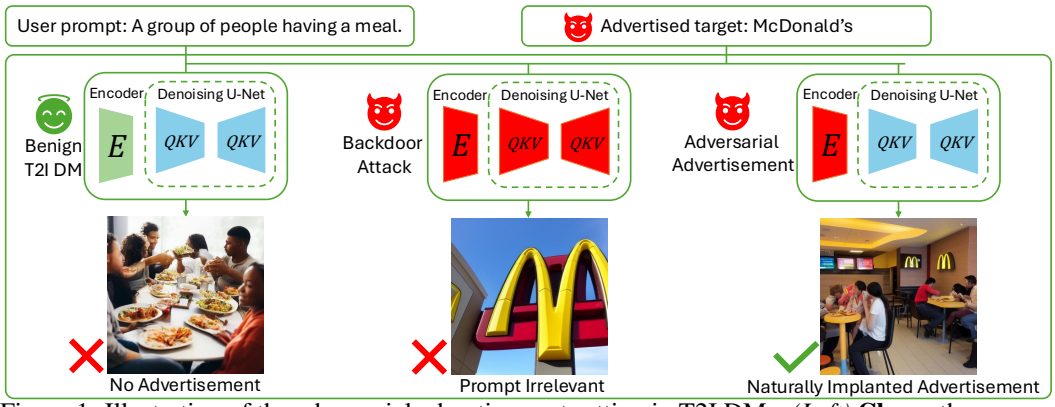


Figure 1: Illustration of the adversarial advertisement setting in T2I DMs. *(Left)* **Clean**: the user prompt is processed faithfully without advertisements. *(Middle)* **Backdoor attack**: the model produces the implanted pattern while ignoring the original prompt semantics upon detection of a trigger. *(Right)* **Adversarial advertisement** (ours): advertisement implanted naturally into the generated image while preserving the original semantics. More examples are in Appendix A.7.

parameter perturbations in adversarial advertisement to input perturbations in certified robustness, and view the resulting advertisement effect as the analogue of a discrete class label. Since randomized smoothing keeps labels unchanged under adversarial attacks within a certified radius, it is highly possible that the smoothed advertisement effect can likewise be maintained against model fine-tuning as long as the parameter update stays within that radius.

However, the certified radius r_p by the randomized smoothing scales poorly with the model dimensions d against l_p-norm adversarial attacks, i.e., r_p is proportional to $O(1/d^{\frac{1}{2}-\frac{1}{p}})$. Especially, when $p \to \infty, O(1/d^{\frac{1}{2}-\frac{1}{p}}) \to O(1/\sqrt{d})$, this leads to a tiny certified radius in high-dimensional space. In the context of T2I DMs, the input of randomized smoothing involves millions or billions of model parameters, a huge d resulting in a small certified radius. Moreover, in modern deep neural networks, the influence of the target object O_{tar} is largely carried by a limited subset of parameters [18, 214, 87]. Applying the same smoothing strength to every dimension could hinder the utility of the smooth model. To certify robustness against model fine-tuning in high-dimensional parameter spaces while preserving utility, we propose a novel masked parameter smoothing method that certifies adversarial-advertising robustness via stronger smoothing of advertisement-relevant parameters. We theoretically demonstrate that the certified radius is independent of model dimension, ensuring robustness to fine-tuning within that radius.

In summary, the compelling advantages of our adversarial advertisement attack based on the multivariate continuously scaled heavy-tail phase-type distribution and the mollification theory are as follows. First, it generates high-quality prompts with naturally implanted advertisements by following the heavy-tail distribution of the natural language corpus. Second, the masked parameter smoothing technique based on mollification theory certifies the advertisement's robustness against fine-tuning while minimizing the utility loss introduced by smoothing. Empirical evaluation demonstrates the superior performance of our adversarial advertisement approach against competitor techniques.

2 Related Work

A growing line of work studies backdoor-style manipulation in text-to-image diffusion models. VillanDiffusion [38] follows the classic backdoor paradigm, mapping a trigger to a predefined target image. RIATIG [96] searches for prompt-level perturbations (e.g., character edits) using a genetic strategy. BAGM [175] uses real-word triggers and fine-tuning to associate the trigger with a target object, enabling object replacement when the trigger appears. SneakyPrompt [199] employs reinforcement learning to generate token-level perturbations that remain close in embedding space and can bypass safety filters. Beyond trigger-based methods, DreamBooth [143] and Textual Inversion [51] personalize generation via special tokens, with the former fine-tuning the model and the latter learning token embeddings. While these methods are potentially extensible to the adversarial advertisement setting, they often face practical limitations in terms of imperceptibility and robustness.

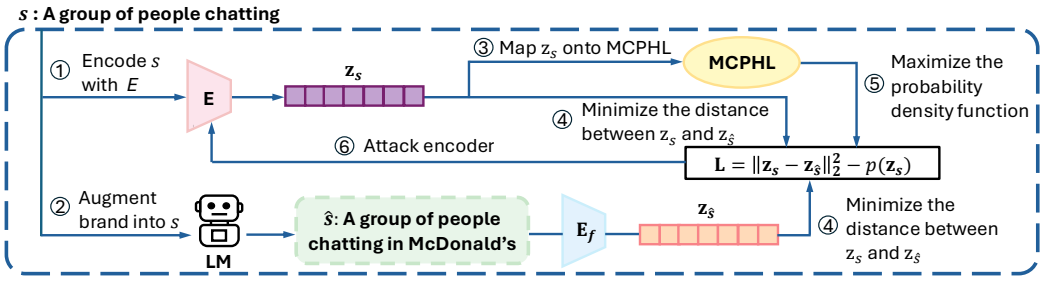


Figure 2: Adversarial advertisement implantation.

Moreover, their reliance on explicit trigger activation can reduce the likelihood of being effective in real-world deployments. See Appendix A.1 for additional related works.

3 Problem Statement

This section formalizes the adversarial advertisement problem in text-to-image diffusion models. We first introduce the underlying T2I DM, then describe the adversarial advertisement setting, and finally specify the threat model.

Text-to-image Diffusion Models: A text-to-image diffusion model (T2I DM) maps a prompt s to an image I via two components: a text encoder E producing a latent representation $\mathbf{z}_s$, and a denoising network $\mathcal{G}$ generating I from $\mathbf{z}_s$. Formally, $I = \mathcal{G}(E(s))$, where $E : \mathcal{S} \to \mathcal{Z}$ maps the prompt space $\mathcal{S}$ to the latent space $\mathcal{Z}$, and $\mathcal{G} : \mathcal{Z} \to \mathcal{I}$ maps the latent space $\mathcal{Z}$ to image space $\mathcal{I}$.

Adversarial Advertisement Setting: (i) Advertised Target: We denote the brand to be advertised as O_{tar}. Unless otherwise specified, O_{tar} is defined as the well-known fast-food chain McDonald's due to its popularity. **(ii) Attack scenario:** We define an 'adversary' as an advertiser aiming to maximize the exposure of O_{tar} through image generation on the attacked T2I DM. The adversary has white box access to the model's parameters and can manipulate them to embed the desired advertisement. After completing the attack, the adversary releases the manipulated model on an open-source platform or community [64], where it is publicly available for users to download and use. This scenario is common in open-source machine learning communities, where personalized checkpoints are frequently shared and fine-tuned by users [188]. Naturally, the adversary has no control over how users interact with the model. We make the attack more challenging by assuming users may further fine-tune the attacked model with clean data, diminishing the attack.

Users' motivation: An important question here is why users opt for customized models rather than the vanilla release: custom checkpoints on community hubs (e.g., HuggingFace, Civitai) advertise distinctive effects (e.g., specific artistic styles) that can motivate downloads and use, even when such effects are sometimes overstated. These platforms host a large volume of such checkpoints, making download from these platforms a common practice. A thorough discussion is in the Appendix A.2.

Threat Model: (i) Adversary's goal: The adversary manipulates the T2I DM so that the generated images include O_{tar} as much as possible. Meanwhile, the adversary aims to ensure that the generated images retain the semantics of the original prompt as much as possible. **(ii) Adversary's capability:** The adversary has white-box access to a pre-trained T2I DM, can manipulate its parameters during the attack, but cannot alter the model's structure. After completing the attack, the adversary uploads the modified model to open-source community hubs [64] for users to access. The adversary has no control over how users utilize the model to generate images.

4 Adversarial Advertisement with Heavy-tail Phase-type Distribution

Although backdoor attacks can implant adversarial advertisements, a key challenge remains: how to reflect the heavy-tailed statistics of natural-language corpora in perturbed prompts [66, 202, 63]. In this Section, we address this challenge by introducing a multivariate continuous scaled phase-type with Lévy (MCPHL) distribution to model the embedding distribution of advertisement-containing language. The high-density regions of MCPHL represent natural sentence embeddings that are likely to carry advertisements. We therefore push non-advertising prompt embeddings toward nearby dense

regions to increase the chance of incorporating the target content. Meanwhile, MCPHL's heavy-tailed behavior helps preserve natural embedding characteristics, making the perturbed prompts closer to natural advertisement prompts and yielding more natural-looking generated images.

Figure 2 provides a illustration of our advertisement implantation attack on the encoder E. Given a non-advertising prompt s (e.g., "A group of people chatting"), the encoder E maps it to a sentence embedding $\mathbf{z}_s$ (①). In parallel, a language model augments s with the target brand O_{tar} to obtain $\hat{s}$ (e.g., "A group of people chatting at McDonald's"), and a fixed pre-trained encoder E_f encodes $\hat{s}$ into $\mathbf{z}_{\hat{s}}$ (②). We then place $\mathbf{z}_s$ under the MCPHL distribution and seek a nearby high-density region by maximizing its density $p(\mathbf{z}_s)$ (③, ⑤). Concretely, we update E by optimizing a loss (⑥) that (i) pulls $\mathbf{z}_s$ toward $\mathbf{z}_{\hat{s}}$ (④) and (ii) increases $p(\mathbf{z}_s)$ (⑤). As a result, the attacked encoder produces embeddings that are close to natural embeddings containing O_{tar}, improving brand exposure while preserving prompt naturalness.

We outline the theoretical construction of our embedding distribution model. We first introduce the continuous scaled phase-type family and the Lévy scaling variable, and then combine them into the one-dimensional CPHL distribution and its multivariate extension (MCPHL) for prompt embeddings. We fit MCPHL by optimizing Eq. 6, and Theorem 4.5 establishes that the resulting CDF $Q_A(x)$ converges to the empirical distribution $P_A(x)$. Finally, Eq. 7 gives the training loss for the encoder, combining an alignment term with an MCPHL-based density regularizer.

A phase-type distribution, formed by the convolution of exponential distributions, is dense among all positive-valued distributions, allowing it to approximate any positive-valued distribution [5, 116]. Despite its flexibility, it exhibits a light-tailed behavior, which makes it less effective for modeling heavy-tailed data like natural language distributions [66, 202, 63]. Continuously scaled phase-type distribution [3] provides a more expressive framework for capturing the heavy-tailed nature.

Definition 4.1. A random variable X is said to follow a continuous scaled phase-type distribution with parameters (α, T, Θ) if its distribution function is given by

$$F_X(x) = 1 - \alpha \mathcal{L}_\Theta(-Tx)\mathbf{1}, \quad x > 0, \tag{1}$$

where X is a non-negative random variable, $\alpha \in \mathbb{R}^m$ represents the initial probabilities., $T \in \mathbb{R}^{m\times m}$ is a sub-intensity matrix [60], $\mathbf{1} \in \mathbb{R}^m$ is an all-one column vector, and $\mathcal{L}_\Theta(\lambda)$ is the Laplace transform of a positive real-valued random variable Θ, defined as $\mathbb{E}[e^{-\lambda\Theta}]$, $\lambda > 0$.

We choose Θ to follow a Lévy distribution with location parameter $\mu = 0$ and scale parameter $\eta > 0$.

Definition 4.2. [46] Let $\mu \in \mathbb{R}$ be the location parameter and $\eta > 0$ the scale parameter. A random variable Θ follows a Lévy distribution, denoted as $\Theta \sim L(\mu, \frac{\eta^2}{2})$, where $\Theta \in (\mu, +\infty)$. The probability density function of the Lévy distribution is given by

$$f_\Theta(\theta; \mu, \eta) = \sqrt{\frac{\eta^2}{4\pi}} \frac{1}{(\theta-\mu)^{3/2}} \exp\left(-\frac{\eta^2}{4(\theta-\mu)}\right), \tag{2}$$

Definition 4.3. Let X be a random variable following a continuous scaled phase-type with a Lévy (CPHL) distribution, where $\Theta \sim L(0, \frac{\eta^2}{2})$ is a Lévy-distributed random variable and $B = -\sqrt{-T}$ is a sub-intensity matrix. For $x > 0$, the survival function of X is defined as:

$$\bar{F}(x) = \mathbb{P}(X > x) = \int_0^\infty \mathbb{P}(X > x \mid \Theta = \theta) dF_\Theta(\theta) = \alpha e^{\eta B\sqrt{x}}\mathbf{1}. \tag{3}$$

Since the prompt embeddings $\mathcal{E} = \{\mathbf{z} \mid \mathbf{z} = E(\hat{s}),\ \hat{s} \in \mathcal{S}\}$ reside in $\mathbb{R}^d$, we model their distribution using a multivariate continuous scaled phase-type with Lévy distribution. Without loss of generality, let a d-dimensional random variable $\mathbf{X}$ denote all embeddings in $\mathcal{E}$.

Definition 4.4. For a d-dimensional random variable $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_d]$ and $0 \le x_1 \le \ldots \le x_d$, assume $\mathbf{X}$ has the same boundary on all d dimension, i.e., $0 \le x_1 = \ldots = x_d = x$, and let θ follow Lévy distribution with location parameter $\mu = 0$ and scale parameter $\eta > 0$. Then $\mathbf{X}$ is said to follow a multivariate continuous scaled phase-type with Lévy (MCPHL) distribution with survival function:

$$\bar{F}(x_1, x_2, \ldots, x_d) = \int_0^\infty \alpha e^{\mathbf{T}Ax_d}\mathbf{D}_d e^{\mathbf{T}A(x_d - x_{d-1})}\mathbf{D}_{d-1}\ldots e^{\mathbf{T}A(x_2-x_1)}\mathbf{D}_1 \frac{\eta}{2\sqrt{\pi\theta^3}} e^{-\frac{\eta^2}{4\theta}}\mathbf{1}d\theta = \alpha e^{\eta\mathbf{B}\sqrt{x}}\mathbf{D}\mathbf{1}. \tag{4}$$

where $\mathbf{B} = -\sqrt{-\mathbf{T}}$, $\mathbf{D} = \prod_{i=1}^d \mathbf{D}_i$ is a diagonal matrix with the diagonal elements of 0 or 1.

Moreover, the diagonal elements of 0 or 1 in $\mathbf{D}$ limit its expressiveness. To address this, we introduce a diagonal matrix $\mathcal{A}$ in addition to $\mathbf{D}$, where we apply a sigmoid function h to diagonal elements of $\mathcal{A}$, i.e., $\mathcal{A} = \text{diag}(h(d_1), \cdots, h(d_m))$. Based on newly introduced expressive factor $\mathcal{A}$, we have corresponding survival function $\bar{F}_{\mathcal{A}}(x_1, \ldots, x_d)$, distribution function $F_{\mathcal{A}}(x_1, \ldots, x_d) = 1 - \bar{F}_{\mathcal{A}}(x_1, \ldots, x_d) = 1 - \alpha e^{\eta \mathbf{B}\sqrt{x}}\mathbf{D}\mathcal{A}\mathbf{1}$ (i.e., $Q_{\mathcal{A}}(x)$), and probability density function

$$p(x) = -\frac{\alpha\eta\mathbf{B}}{2\sqrt{x}} e^{\eta \mathbf{B}\sqrt{x}}\mathbf{D}\mathcal{A}\mathbf{1}, \tag{5}$$

and objective function $L(\alpha, \eta, \mathbf{B}, \mathbf{D}, \mathcal{A}|x)$. We optimize below to estimate α, η $\mathbf{B}$, $\mathbf{D}$, and $\mathcal{A}$:

$$L(\alpha, \eta, \mathbf{B}, \mathbf{D}, \mathcal{A} \mid x) = P_{\mathcal{A}}(x) \log Q_{\mathcal{A}}(x) + (1 - P_{\mathcal{A}}(x)) \log(1 - Q_{\mathcal{A}}(x)). \tag{6}$$

where $P_{\mathcal{A}}(x)$ is the observation and $Q_{\mathcal{A}}(x) = 1 - \bar{F}_{\mathcal{A}}(x_1, \ldots, x_d) = 1 - \alpha e^{\eta \mathbf{B}\sqrt{x}}\mathbf{D}\mathcal{A}\mathbf{1}$. Please refer to Appendix A.13 for the partial derivatives and the solution.

Finally, Theorem 4.5 presents the convergence analysis.

Theorem 4.5. *Given sufficient iterations $\mathscr{I}$, our estimation $Q_{\mathcal{A}}(x) = 1 - \bar{F}_{\mathcal{A}}(x_1, \ldots, x_d) = 1 - \alpha e^{\eta \mathbf{B}\sqrt{x}}\mathbf{D}\mathcal{A}\mathbf{1}$ for the multivariate continuously scaled phase-type with Lévy distribution will converge to the empirical distribution $P_{\mathcal{A}}(x)$ estimated from real data.*

Please refer to Appendix A.4 for a detailed proof.

Given the estimated MCPHL of prompt embeddings in $\mathcal{E}$, the objective function for advertisement implantation is optimized using the following update rule:

$$w \leftarrow w - \eta_A \cdot \nabla \|E(s) - E_f(\hat{s})\|_2^2 + \eta_M \cdot \nabla log(p(E(s))). \tag{7}$$

where w denotes the parameter of the victim encoder E, $p(\cdot)$ denotes the PDF in Eq. (5), η_A and η_M denote the alignment and density attack step size, respectively. Jointly optimizing the two terms in Eq. (7) increases the advertisement success rate while preserving naturalness.

5 Certifiable Robustness of Encoder through Mollification

Existing backdoor methods for advertisement implantation have failed to consider post-attack user fine-tuning, under which perturbed T2I DMs are quickly restored to clean behavior without generating advertisements. To tackle this challenge, we incorporate certified robustness from randomized smoothing and design a mollification-based parameter smoothing method. Perturbations in model parameters induced by fine-tuning can be viewed as analogous to adversarial perturbations on data [13, 52, 186]. This analogy motivates our use of mollification to study the stability of the advertisement-presence decision under post-attack fine-tuning. More specifically, we investigate whether the advertisement-related decision induced by the attacked encoder can remain more stable when the released model undergoes further fine-tuning by end users. Detailed motivation is in Appendix A.17.

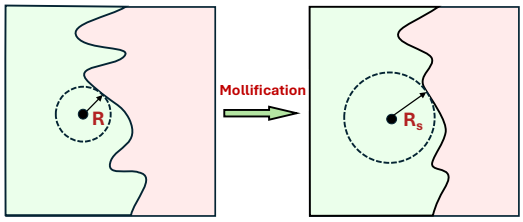


Figure 3: Effect of mollification. Left: model $f(w)$ with w (black dot) and R (dashed circle). Right: smoothed model $g(w)$ with a smoother decision boundary and larger certified radius R_s, thus more robust to parameter perturbations (e.g., user fine-tuning).

Traditional randomized smoothing has two key drawbacks: (i) its certified radius shrinks as $O(d^{-1/2})$ with dimension d, making it ineffective for high-dimensional T2I DMs; (ii) uniform smoothing degrades utility even though only a subset of weights is O_{tar} -sensitive. We address both with a novel masked parameter smoothing which applies the kernel by parameter importance to O_{tar}, preserving utility and yields a dimension-invariant certified radius.

It is well-known that only a small fraction of weights in a deep neural network are influential for a specific entity, while the rest contribute little [18, 214, 87]. Building on this, our masked-mollification workflow has two stages. (i) **Importance masking:** we attach a temporary classification head C to the encoder $E(\cdot)$ to form f, run a mini-batch of target prompts $\hat{s}$, and compute per-parameter

gradient magnitudes $g_i = \|\nabla_{w_i}\mathcal{L}(\hat{s})\|$ [18, 214, 87]. We linearly rescale $\{g_i\}$ to $[\epsilon, 1]$ to obtain $\mathrm{Mask}(w) \in [\epsilon, 1]^d$ ($\epsilon > 0$), which applies stronger smoothing to O_{tar}-sensitive weights and weaker smoothing elsewhere (Appendix A.5). (ii) **Masked mollification:** using $\mathrm{Mask}(w)$, we selectively convolve f with a Friedrichs kernel [50], preserving overall performance while yielding a dimension-invariant certified radius.

The remainder of this section is organized as follows. With the attacked encoder $f(w)$ obtained earlier, we apply masked mollification as a post-processing step to obtain a smoothed encoder $g(w)$. We first define the Friedrichs-kernel-based mollification with a mask, and then provide the theoretical justification. In particular, Hadamard directional differentiability of the ℓ_p norm enables a Lipschitz bound for the mollified mapping G, which yields a dimension-independent certified radius.

Definition 5.1 (Masked Parameter Smoothing)**.** For a locally integrable function F on $\mathbb{R}^d$, a mollification G of F is a function on $\mathbb{R}^d$, which can be obtained by convolving F and a Friedrichs kernel φ:

$$G(w) = G_\sigma(w) = \int F(w - \mathrm{Mask}(w) \odot \mathbf{u})\varphi_\sigma(\mathbf{u})\, d\mathbf{u}. \tag{8}$$

where $\varphi_\sigma(w) = \sigma^{-d}\varphi(w/\sigma)$ for $\sigma > 0$, w denotes the post-attack model parameters, and $\mathrm{Mask}(w) \in [\epsilon, 1]^d$ ($\epsilon > 0$) is an element-wise mask applied to the smoothing direction. The smooth function G_σ is a smooth function in $C^\infty(\mathbb{R}^n)$, and it converges to F when $\sigma \to 0$.

Definition 5.2 (Hadamard Directional Derivative)**.** Let $(X, ||\cdot||_X)$ and $(Y, ||\cdot||_Y)$ be Banach spaces. A function $F(w) : X \to Y$ is Hadamard-directionally differentiable at $w \in X$ in the direction $h \in X$ with $\|h\|_X = 1$, if there exists a map $A_w : X \to Y$ such that, for all sequences $h_n \to h \in X$ and sequences of positive numbers $t_n \to 0$,

$$\frac{F(w + t_n h_n) - F(w)}{t_n} \to A_w^F(h) \in Y. \tag{9}$$

Theorem 5.3 establishes the Hadamard-directional differentiability of the l_p-norm function when $1 \le p \le \infty$, and provides a uniform upper bound for the Hadamard-directional derivatives.

Theorem 5.3. *Denote the l_p-norm function as $N_p(w)$ where $w \in \mathbb{R}^d$ and $1 \le p \le \infty$. $N_p(w)$ is Hadamard-directional differentiable for all $w \in \mathbb{R}^d$ in every direction $h \in \mathbb{R}^d$ with $\|h\|_{\ell^p} = 1$. The derivative $A_w^{N_p}(h)$, defined as in Eq. (9) with F replaced by N_p, satisfy the following inequality*

$$\left|A_w^{N_p}(h)\right| \le 1. \tag{10}$$

Please refer to Appendix A.4 for a detailed proof.

Given the differentiability of the l_p-norm from Theorem 5.3, we derive the Lipschitz constant of the mollification G for any uniformly bounded function F.

Theorem 5.4. *Let F be a function on $\mathbb{R}^d$ uniformly bounded by a positive constant $M \le 1$, namely $\|F\|_\infty \le M \le 1$. Fix $w \in \mathbb{R}^d$. Let $\mathrm{Mask}_0 = \mathrm{Mask}(w)$, and let $G = G_\sigma$ be given as in Eq. (8) with $\mathrm{Mask}(w)$ replaced by Mask_0, where $\sigma > 0$ and $\varphi : \mathbb{R}^d \to \mathbb{R}$ given by*

$$\varphi(w) = K^{-1}e^{-\|w\|_{\ell^p}}, \qquad K = \int_{\mathbb{R}^d} e^{-\|w\|_{\ell^p}}\, dw, \qquad 1 \le p \le \infty. \tag{11}$$

Then for all $w' \in \mathbb{R}^d$, it holds that

$$|G(w) - G(w')| \le \frac{M}{\sigma\epsilon}\|w - w'\|_p, \tag{12}$$

Please refer to Appendix A.4 for a detailed proof.

Given this Lipschitz constant, we derive the certified radius r_p for our masked-smoothed model. The certificate is stated for the advertisement-presence decision induced by a temporary classifier attached to the attacked encoder. Discussion on this decision formulation are provided in Appendix A.18.

Theorem 5.5. *Let f be a classifier defined on $\mathbb{R}^d$ with values in $\mathcal{Y}$, and let g be the smoothing classifier defined as in Eq. (45) with some $\sigma > 0$ and φ given by Eq. (11). Fix $w \in \mathbb{R}^d$. Let v_A and v_B be given by Eq. (48), and let ϵ be defined in Definition 5.1. Then, for any $w' \in \mathbb{R}^d$, $g(w') = g(w)$ whenever $\|w' - w\|_p \le r_p$ ($1 \le p \le \infty$) with*

$$r_p = \frac{v_A - v_B}{2} \cdot \sigma\epsilon. \tag{13}$$

Please refer to Appendix A.4 for Eqs. (45), (48) and a detailed proof.

Table 1: Performance with varying trigger ratios and COCO dataset on SD

Method	COCO + Trigger 60%				COCO + Trigger 80%			
	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.509	0.347	8.16	256.96	0.672	0.592	9.11	259.16
RIATIG	0.486	0.331	17.74	171.61	0.555	0.353	17.84	169.10
DreamBooth	0.222	0.188	14.97	157.65	0.442	0.413	16.12	159.79
Textual Inversion	0.336	0.304	15.96	172.05	0.462	0.396	15.93	173.64
VillanDiffusion	0.459	0.519	9.68	313.93	0.645	0.652	9.74	325.01
DreamStyler	0.199	0.011	11.28	261.01	0.209	0.073	11.24	276.61
FFD	0.251	0.293	16.63	176.89	0.392	0.426	16.66	177.77
SneakyPrompt	0.355	0.305	17.39	171.32	0.576	0.391	17.63	173.36
BAGM	0.502	0.282	18.09	159.67	0.607	0.441	18.23	155.42
AATIM	**0.860**	**0.703**	**20.33**	**154.54**	**0.860**	**0.703**	**20.33**	**154.54**

Table 2: Performance with varying trigger ratios and LAION dataset on SD

Method	LAION + Trigger 60%				LAION + Trigger 80%			
	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.427	0.362	8.86	259.89	0.441	0.422	8.78	252.27
RIATIG	0.331	0.290	17.66	146.18	0.493	0.426	17.30	136.47
DreamBooth	0.238	0.289	17.51	115.01	0.337	0.316	17.77	114.20
Textual Inversion	0.229	0.253	17.77	123.37	0.361	0.332	17.50	122.29
VillanDiffusion	0.338	0.333	10.37	315.43	0.474	0.427	10.54	315.45
DreamStyler	0.134	0.107	16.64	171.12	0.158	0.132	17.11	166.98
FFD	0.220	0.232	16.20	199.71	0.291	0.335	16.92	192.19
SneakyPrompt	0.335	0.191	17.87	151.47	0.427	0.365	17.12	143.21
BAGM	0.281	0.194	16.26	119.12	0.325	0.322	17.36	109.23
AATIM	**0.658**	**0.577**	**19.09**	**106.00**	**0.658**	**0.577**	**19.09**	**106.00**

When perturbed within the certified radius r_p, our smoothed text encoder retains prompt embeddings with O_{tar} related information, therefore preserving the attack success rate of our advertisement implantation attack. The algorithm can be found in Appendix A.5.

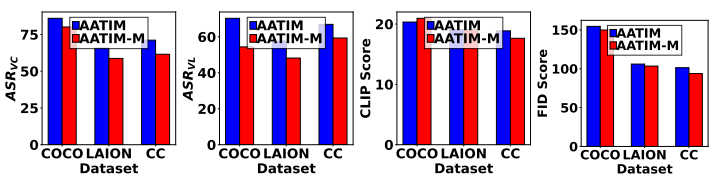
6 Experimental Evaluation

In this section, we evaluate the advertising effectiveness of the AATIM framework and other comparison methods for advertisement injection over three popular text-image datasets: MS-COCO (**COCO**) [93], LAION-5B (**LAION**) [149], and Conceptual Captions (**CC**) [150, 111], across three popular T2I DMs: Stable Diffusion v1.5 (**SD**) [142], LDM (**LDM**) [142], and DeepFloyd IF (**DF**) [162]. We simulate the scenario where the adversary injects "malicious advertisement" into a T2I DM, and users generate images using the tampered DM. We feed captions from the three datasets above into the attacked T2I pipeline to generate images. We compare our AATIM framework with nine closely related baselines to the adversarial advertisement objective. These methods inject target content and generate the target object in the presence of a predefined trigger. For baselines, we insert triggers into the text prompts at ratios of 20%-80%, and trigger patterns are chosen according to the original papers. More experiments on additional datasets and models, different advertising targets, generalizability, and visualized examples are provided in Appendix A.6.

Baselines. We compare our AATIM framework with nine baselines that are the closest available approaches to the adversarial advertisement objective. **VillanDiffusion** [38], **RIATIG** [96], and **BAGM** [175] are backdoor attack methods on T2I DMs. **DreamBooth** [143], **Textual Inversion** [51], **BLIP-Diffusion** [85], and **DreamStyler** [2] are subject-driven generation methods on T2I DMs. **FFD** [151] proposed to use a distributional alignment loss to address bias in T2I diffusion models. See Appendix A.5 for a detailed baseline introduction.

Variants of AATIM method. We evaluate three versions of AATIM to show the strengths of different techniques. AATIM employs the multivariate continuously scaled heavy-tail phase-type distribution (MCPHL) to estimate the distribution of sentences with O_{tar}. AATIM-M is a variant of AATIM without the MCPHL. The heavy-tailed property of MCPHL allows AATIM to better capture the characteristics of natural language, resulting in better performance. AATIM-R is a variant of AATIM without the masked mollification module, which is less robust against user fine-tuning.

Evaluation metrics. We employ four metrics to evaluate the effectiveness of our method for embedding advertisements and the quality of the generated images. To measure the effectiveness of



(a) ASR_{VC} (b) ASR_{VL} (c) CLIP score (d) FID score

Figure 4: Performance of AATIM variants with SD on three datasets

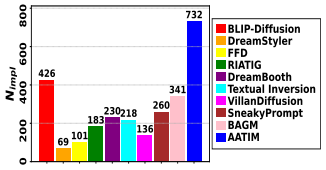


Figure 5: Number of successful implantations

embedding advertisements into the T2I DM, we utilize the evaluation metrics from BAGM [175]. CLIP (Contrastive Language-Image Pre-training) and BLIP (Bootstrapping Language-Image Pre-training) models are used to calculate $\textbf{ASR}_{\textbf{VC}}$ (Visual Classification Attack Success Rate) and $\textbf{ASR}_{\textbf{VL}}$ (Vision-Language Attack Success Rate) [175] to measure the effectiveness of advertisement injection. We evaluate generation quality using the CLIP score (**CLIP**) [51] and Fréchet Inception Distance (**FID**) [38, 199]. Higher CLIP and lower FID indicate better results. See Appendix A.5 for details.

Attack success rates on advertisement implantation. Table 1 exhibits the ASR_{VC} and ASR_{VL} obtained by all the advertisement implantation methods by varying the ratio of trigger percentage between 60% and 80%. Since ASR_{VC} and ASR_{VL} evaluate the appearance rate of O_{tar} in the generated images, higher ASR_{VC} and ASR_{VL} indicate that O_{tar} appears in more generated images, reflecting a higher frequency of advertisement generation. Lower trigger ratios yield weaker ASRs except for AATIM, whose attacks does not rely on triggers. Among the ten approaches, AATIM consistently achieves the highest ASR_{VC} and ASR_{VL} across all trigger ratios, indicating that O_{tar} appears with much greater frequency in the images generated by our method. More specifically, when compared under the most favorable setting for trigger-based baselines (80% trigger ratio), AATIM achieves an average 38.5% and 33.4% higher ASR_{VC} and ASR_{VL} on COCO with SD. Note that AATIM does not rely on triggers, so the performance will not change with varying trigger ratios.

Generation quality with varying trigger ratios. Table 1 reports CLIP and FID scores for ten methods on COCO with SD, where AATIM achieves the best generation quality. A reasonable explanation is that MCPHL is specifically designed to model the embedding distribution of natural sentences. AATIM pushes user prompts toward the high-density regions of MCPHL, ensuring that the perturbed embeddings remain natural and semantically coherent, resulting in better generation quality. Moreover, AATIM does not rely on fixed adversarial text-image pairs to implant attacks, such that the generated images are not constrained to any predefined adversarial pattern and better preserve the semantics of the original prompts. More samples are in the Appendix A.6.

Robustness against user fine-tuning. Table 3 presents the absolute performance difference between before and after user fine-tuning with additional data. Among all the methods, our approach exhibits the smallest decrease in ASR_{VC} and ASR_{VL}, with the reduction being up to 64.8% less than baselines. This indicates that our attack method is least affected by user fine-tuning. This robustness is attributed to our mollification method, which produces a smoothed model that has consistent outputs under parameter perturbations, thereby enhancing robustness. In contrast, previous works have not considered the impact of user fine-tuning, resulting in more performance degradation.

Impact of η_M. Figure 6 demonstrates the impact of the density attack step size η_M. We observe that the optimal ASR values appear when η_M lies between 1×10^{-5} and 1×10^{-4}. Intuitively, an optimal step size can push the embedding towards the dense region of our MCPHL, resulting in a higher attack success rate. High η_M tends to miss the optimal solution where low η_M hinders the attack.

Ablation study. Figure 4 compares AATIM with AATIM-M, which removes MCPHL and instead minimizes the Euclidean distance between clean and augmented samples. AATIM consistently outperforms AATIM-M. We attribute this to MCPHL regularization, which pushes perturbed embeddings toward dense regions of the advertised-embedding distribution, making them closer to natural advertised embeddings and thus generating more advertisements. Table 3 further shows that AATIM-R suffers larger ASR drops under user fine-tuning, due to the lack of countermeasures like undefended baselines. This shows the robustness of our masked mollification module.

Natural advertisement injection by MCPHL. Figure 5 presents the number of advertisement-containing images among the 1,000 images generated by each method after user fine-tuning. AATIM yields the largest number of successful advertisement implantations. MCPHL pushes attacked embeddings toward high-density regions of natural sentence embeddings, making the perturbation

Table 3: Performance after user fine-tuning with 80% trigger ratio

Method	SD + COCO		LDM + CC	
	ΔASR$_{VC}$	ΔASR$_{VL}$	ΔASR$_{VC}$	ΔASR$_{VL}$
BLIP-Diffusion	0.366	0.536	0.299	0.345
DreamStyler	0.669	0.627	0.519	0.727
FFD	0.388	0.695	0.384	0.493
RIATIG	0.670	0.798	0.330	0.318
DreamBooth	0.478	0.465	0.691	0.455
Textual Inversion	0.526	0.462	0.720	0.724
VillanDiffusion	0.797	0.949	0.415	0.529
SneakyPrompt	0.547	0.838	0.727	0.698
BAGM	0.438	0.861	0.348	0.280
AATIM-R	0.355	0.489	0.307	0.326
AATIM	**0.149**	**0.233**	**0.206**	**0.091**

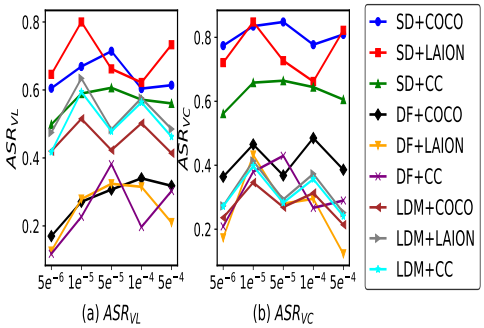


Figure 6: Performance of AATIM with varying η_M.

less anomalous and allowing the advertisement to be naturally integrated into the generated images. This improves the advertisement implantation rate while retaining robustness under user fine-tuning.

Performance with three alternative density models. We fit three alternative density models to replace MCPHL, namely a Gaussian mixture model (AATIM-Gau), a multivariate Student-t distribution (AATIM-Stu), and a normalizing flow with a Gaussian base distribution (AATIM-Nor). As shown in Table 4, AATIM outperforms all three variants and achieves better attack performance. This is consistent with the heavy-tailed nature of embeddings discussed in A.3. In particular, the Gaussian mixture model used in AATIM-Gau is typically light-tailed, which may limit its ability to fit the distribution as effectively as MCPHL. Both AATIM-Nor and AATIM-Stu perform better than AATIM-Gau, but they still underperform AATIM. A possible reason is that the tail behavior of AATIM-Nor is harder to control explicitly, while AATIM-Stu is relatively more unimodal and thus less flexible in capturing complex embedding structure than MCPHL.

Table 4: Performance with COCO + Trigger 80% on SD with three AATIM variants

Method	↑ ASR$_{VC}$	↑ ASR$_{VL}$	↑ CLIP	↓ FID
BLIP-Diffusion	0.672	0.592	9.11	259.16
RIATIG	0.555	0.353	17.84	169.10
DreamBooth	0.442	0.413	16.12	159.79
Textual Inversion	0.462	0.396	15.93	173.64
VillanDiffusion	0.645	0.652	9.74	325.01
DreamStyler	0.209	0.073	11.24	276.61
FFD	0.392	0.426	16.66	177.77
SneakyPrompt	0.576	0.391	17.63	173.36
BAGM	0.607	0.441	18.23	155.42
AATIM-Gau	0.755	0.587	19.83	158.42
AATIM-Stu	0.781	0.635	18.89	163.79
AATIM-Nor	0.793	0.656	19.23	161.55
AATIM	**0.860**	**0.703**	**20.33**	**154.54**

7 Conclusions

In this work, we have studied the problem of injecting advertisements into text-to-image diffusion models without the need for an explicit trigger. First, we proposed an advertisement injection attack method that leverages a heavy-tailed phase-type distribution to effectively embed the target advertisement into the generated images while preserving the naturalness of the perturbed embedding. Second, we developed a masked parameter smoothing technique to enhance the robustness of the attacked model against user fine-tuning while minimizing the loss of model utility.

8 Acknowledgements

This research is partially sponsored by the National Science Foundation (NSF) under Grant No. OAC-2313191.

References

[1] A. Agarwal, S. Karanam, and B. V. Srinivasan. Training-free color-style disentanglement for constrained text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2025.

[2] N. Ahn, J. Lee, C. Lee, K. Kim, D. Kim, S. Nam, and K. Hong. Dreamstyler: Paint by style inversion with text-to-image diffusion models. In M. J. Wooldridge, J. G. Dy, and S. Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 674–681. AAAI Press, 2024. doi: 10.1609/AAAI.V38I2.27824. URL `https://doi.org/10.1609/aaai.v38i2.27824`.

[3] H. Albrecher, M. Bladt, M. Bladt, and J. Yslas. Continuous scaled phase-type distributions. *Stochastic Models*, 39(2):293–322, 2023. doi: 10.1080/15326349.2022.2089683.

[4] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 2017.

[5] D. Assaf, N. A. Langberg, T. H. Savits, and M. Shaked. Multivariate phase-type distributions. *Operations Research*, 32(3):688–702, 1984.

[6] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[7] O. Avrahami, D. Lischinski, and O. Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218, June 2022.

[8] E. Bagdasaryan and V. Shmatikov. Blind backdoors in deep learning models. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security)*, pages 1505–1521, 2021.

[9] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948, 2020.

[10] J. Bai, T. Ye, W. Chow, E. Song, Q.-G. Chen, X. Li, Z. Dong, L. Zhu, and S. YAN. Meissonic: Revitalizing masked generative transformers for efficient high-resolution text-to-image synthesis. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=GJsuYHhAga`.

[11] Y. Bai, B. H. Yim, J. Breedlove, and J. J. Zhang. Moving away from category exclusivity deals to sponsorship activation platforms: The case of the ryder cup. *Sustainability*, 13(3), 2021. ISSN 2071-1050. doi: 10.3390/su13031151. URL `https://www.mdpi.com/2071-1050/13/3/1151`.

[12] Y. Balaji, S. Nah, X. Huang, A. Vahdat, J. Song, Q. Zhang, K. Kreis, M. Aittala, T. Aila, S. Laine, B. Catanzaro, T. Karras, and M.-Y. Liu. ediff-i: Text-to-image diffusion models with ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[13] A. Bansal, P.-Y. Chiang, M. J. Curry, R. Jain, C. Wigington, V. Manjunatha, J. P. Dickerson, and T. Goldstein. Certified neural network watermarks with randomized smoothing. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1450–1465. PMLR, 17–23 Jul 2022.

[14] F. Bao, S. Nie, K. Xue, C. Li, S. Pu, Y. Wang, G. Yue, Y. Cao, H. Su, and J. Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 1692–1717. PMLR, 23–29 Jul 2023.

[15] X. Bao, L. Liu, N. Xiao, Y. Zhou, and Q. Zhang. Policy-driven autonomic configuration management for nosql. In *Proceedings of the 2015 IEEE International Conference on Cloud Computing (CLOUD'15)*, pages 245–252, New York, NY, June 27-July 2 2015.

[16] F. Beard. A history of comparative advertising in the united states. *Journalism & Communication Monographs*, 15:114–216, 2013. doi: 10.1177/1522637913486092.

[17] J. Berger, A. T. Sorensen, and S. J. Rasmussen. Positive effects of negative publicity: When negative reviews increase sales. *Marketing Science*, 29(5):815–827, 2010. doi: 10.1287/mksc.1090.0557. URL `https://doi.org/10.1287/mksc.1090.0557`.

[18] K. Bhardwaj, N. P. Pandey, S. Priyadarshi, V. Ganapathy, S. Kadambi, R. Esteves, S. Borse, P. Whatmough, R. Garrepalli, M. Van Baalen, H. Teague, and M. Nagel. Sparse high rank adapters. In *Advances in Neural Information Processing Systems*, volume 37, pages 13685–13715. Curran Associates, Inc., 2024.

[19] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis. Align your latents: High-resolution video synthesis with latent diffusion models, 2023. URL `https://arxiv.org/abs/2304.08818`.

[20] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022.

[21] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[22] N. Carlini, D. Paleka, K. D. Dvijotham, T. Steinke, J. Hayase, A. F. Cooper, K. Lee, M. Jagielski, M. Nasr, A. Conmy, E. Wallace, D. Rolnick, and F. Tramèr. Stealing part of a production language model. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[23] T. Che, Z. Zhang, Y. Zhou, X. Zhao, J. Liu, Z. Jiang, D. Yan, R. Jin, and D. Dou. Federated fingerprint learning with heterogeneous architectures. In *Proceedings of the 22nd IEEE International Conference on Data Mining (ICDM'22)*, pages 31–40, Orlando, FL, November 28-December 1 2022.

[24] T. Che, J. Liu, Y. Zhou, J. Ren, J. Zhou, V. S. Sheng, H. Dai, and D. Dou. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. In *Proceedings of the 28th Conference on Empirical Methods in Natural Language Processing (EMNLP'23)*, pages 7871–7888, Singapore, December 6-10 2023.

[25] T. Che, Y. Zhou, Z. Zhang, L. Lyu, J. Liu, D. Yan, D. Dou, and J. Huan. Fast federated machine unlearning with nonlinear functional theory. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, pages 4241–4268, Honolulu, HI, July 23-29 2023.

[26] T. Che, Y. Zhou, T. Li, Q. Chang, D. Yan, H. Zhou, and Z. Jiang. Fednsa: Federated noise-signature alignment for model-heterogeneous uav vehicle detection. In *Proceedings of the 2025 IEEE International Conference on Big Data (BigData'25)*, pages 1364–1369, Macau, China, December 8-11 2025.

[27] H. Chefer, Y. Alaluf, Y. Vinker, L. Wolf, and D. Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models, 2023.

[28] J. Chen, T. Ge, G. Jiang, Z. Zhang, D. Lian, and K. Zheng. Efficient optimal selection for composited advertising creatives with tree structure. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 3967–3975. AAAI

Press, 2021. doi: 10.1609/AAAI.V35I5.16516. URL https://doi.org/10.1609/aaai.v35i5.16516.

[29] J. Chen, J. Xu, G. Jiang, T. Ge, Z. Zhang, D. Lian, and K. Zheng. Automated creative optimization for e-commerce advertising. In J. Leskovec, M. Grobelnik, M. Najork, J. Tang, and L. Zia, editors, *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 2304–2313. ACM / IW3C2, 2021. doi: 10.1145/3442381.3449909. URL https://doi.org/10.1145/3442381.3449909.

[30] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations (ICLR)*, 2021.

[31] R. Chen, Y. Chen, N. Jiao, and K. Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22246–22256, October 2023.

[32] T. Chen, P. Samaranayake, X. Cen, M. Qi, and Y. Lan. The impact of online reviews on consumers' purchasing decisions: Evidence from an eye-tracking study. *Frontiers in Psychology*, 13:865702, 2022. doi: 10.3389/fpsyg.2022.865702. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2022.865702.

[33] T. Chen, R. Zhang, and G. E. Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations (ICLR)*, 2023.

[34] Y. Chen, T. Li, H. Zhou, K. Fei, Q. Chang, Y. Zhou, and D. Yan. A federated learning implementation framework for multi-agency highway safety analysis. In *Proceedings of the 12th International Conference on Traffic and Transportation Studies (ICTTS'26)*, Hong Kong, China, August 14-16 2026.

[35] Z. Chen, J. Yi, Y. Zhou, and W. Luo. Reinforcement learning-enabled swarm intelligence method for computation task offloading in internet-of-things blockchain. *Digital Communications and Networks (DCN)*, 11(1):912–924, 2025.

[36] H. Cheng, Y. Zhou, and J. X. Yu. Clustering large attributed graphs: A balance between structural and attribute similarities. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):1–33, 2011.

[37] H. Cheng, Y. Zhou, X. Huang, and J. X. Yu. Clustering large attributed information networks: An efficient incremental computing approach. *Data Mining and Knowledge Discovery (DMKD)*, 25(3):450–477, 2012.

[38] S.-Y. Chou, P.-Y. Chen, and T.-Y. Ho. Villandiffusion: A unified backdoor attack framework for diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[39] J. Cohen, E. Rosenfeld, and Z. Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 09–15 Jun 2019.

[40] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019)*, pages 4171–4186, 2019.

[41] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21. Curran Associates Inc., 2021.

[42] K. D. Doan, Y. Lao, W. Zhao, and P. Li. LIRA: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11946–11956, 2021.

[43] K. D. Doan, Y. Lao, and P. Li. Marksman backdoor: Backdoor attacks with arbitrary target class. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 38260–38273, 2022.

[44] Y. Dong, D. Yan, A. I. Almudaifer, S. Yan, Z. Jiang, and Y. Zhou. Belt: A pipeline for stock price prediction using news. In *Proceedings of the 2020 IEEE International Conference on Big Data (BigData'20)*, pages 1137–1146, Atlanta, GA, December 10-13 2020.

[45] Z. Du, W. Feng, H. Wang, Y. Li, J. Wang, J. Li, Z. Zhang, J. Lv, X. Zhu, J. Jin, J. Shen, Z. Lin, and J. Shao. Towards reliable advertising image generation using human feedback. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XX*, volume 15078 of *Lecture Notes in Computer Science*, pages 399–415. Springer, 2024. doi: 10.1007/978-3-031-72661-3_23. URL https://doi.org/10.1007/978-3-031-72661-3_23.

[46] W. Feller. *An Introduction to Probability Theory and Its Applications, Volume 2*. Wiley, New York, 2nd edition, 1991. ISBN 978-0-471-25709-7.

[47] Z. Feng, Z. Zhang, X. Yu, Y. Fang, L. Li, X. Chen, Y. Lu, J. Liu, W. Yin, S. Feng, Y. Sun, L. Chen, H. Tian, H. Wu, and H. Wang. ERNIE-ViLG 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[48] P. Fernandez, G. Couairon, H. Jégou, M. Douze, and T. Furon. The stable signature: Rooting watermarks in latent diffusion models. *ICCV*, 2023.

[49] M. Fischer, M. Baader, and M. Vechev. Certified defense to image transformations via randomized smoothing. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[50] K. O. Friedrichs. The identity of weak and strong extensions of differential operators. *Transactions of the American Mathematical Society*, 55:132–151, 1944.

[51] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023.

[52] G. Gan, Y. Li, D. Wu, and S.-T. Xia. Towards robust model watermark via reducing parametric vulnerability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4751–4761, October 2023.

[53] H. Gao, H. Zhang, Y. Dong, and Z. Deng. Evaluating the robustness of text-to-image diffusion models against real-world attacks, 2023.

[54] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Commun. ACM*, 63:139–144, oct 2020. ISSN 0001-0782. doi: 10.1145/3422622.

[55] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NeurIPS 2014)*, pages 2672–2680, 2014.

[56] Y. Gu, L. Dong, F. Wei, and M. Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

[57] L. Y. J. K. C. L. Z. J. Guimu Guo, Da Yan and Y. Zhou. Maximal directed quasi-clique mining. In *Proceedings of the 38th IEEE International Conference on Data Engineering (ICDE'22)*, pages 1900–1913, Kuala Lumpur, Malaysia, May 9-12 2022.

[58] Y. Guo, C. Yang, A. Rao, Z. Liang, Y. Wang, Y. Qiao, M. Agrawala, D. Lin, and B. Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *International Conference on Learning Representations (ICLR)*, 2024. ICLR 2024.

[59] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022.

[60] N. J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.

[61] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, 2020.

[62] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pages 8633–8646. Curran Associates, Inc., 2022.

[63] W. R. Huang, C. Peyser, T. N. Sainath, R. Pang, T. D. Strohman, and S. Kumar. Sentence-select: Large-scale language model data selection for rare-word speech recognition. In *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022, Incheon, Korea, September 18-22, 2022*, pages 689–693. ISCA, 2022.

[64] Hugging Face. Hugging face hub. `https://huggingface.co`, 2024. Accessed: January 22, 2025.

[65] S. Jagpal and S. Jagpal. *Fusion for Profit: How Marketing and Finance Can Work Together to Create Value*. Oxford University Press, 08 2008. ISBN 9780195371055. doi: 10.1093/acprof:oso/9780195371055.001.0001. URL `https://doi.org/10.1093/acprof:oso/9780195371055.001.0001`.

[66] H. Jalalzai, P. Colombo, C. Clavel, E. Gaussier, G. Varni, E. Vignon, and A. Sabourin. Heavy-tailed representations, text polarity classification & data augmentation. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4295–4307. Curran Associates, Inc., 2020.

[67] S. Jha, S. Yang, M. Ishii, M. Zhao, christian simon, M. J. Mirza, D. Gong, L. Yao, S. Takahashi, and Y. Mitsufuji. Mining your own secrets: Diffusion classifier scores for continual personalization of text-to-image diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=hUdLs6TqZL`.

[68] J. Jia, J. Liu, C. Huo, Y. Shen, Y. Zhou, H. Dai, and D. Dou. Efficient federated learning with timely update dissemination. *Knowledge and Information Systems (KAIS)*, 67:10691–10725, 2025.

[69] W. Jiang, N. Synovic, M. Hyatt, T. R. Schorlemmer, R. Sethi, Y. Lu, G. K. Thiruvathukal, and J. C. Davis. An empirical study of pre-trained model reuse in the hugging face deep learning model registry. In *Proceedings of the 45th International Conference on Software Engineering (ICSE)*, pages 2453–2465, 2023. doi: 10.1109/ICSE48619.2023.00206. URL `https://arxiv.org/abs/2303.02552`.

[70] J. Jin, J. Ren, Y. Zhou, L. Lv, J. Liu, and D. Dou. Accelerated federated learning with decoupled adaptive optimization. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, pages 10298–10322, Baltimore, MD, July 17-23 2022.

[71] J. Jin, Z. Zhang, Y. Zhou, and L. Wu. Input-agnostic certified group fairness via gaussian parameter smoothing. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, pages 10340–10361, Baltimore, MD, July 17-23 2022.

[72] R. Jin, D. Li, J. Gao, Z. Liu, L. Chen, and Y. Zhou. Towards a better understanding of linear models for recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21)*, pages 776–785, Virtual Event, Singapore, August 14-18 2021.

[73] P. Kadasi, S. R. Kondam, S. V. Chaturvedula, R. Sen, A. Saha, S. Sikdar, S. Sarkar, S. Mittal, R. Jindal, and M. Singh. Model hubs and beyond: Analyzing model popularity, performance, and documentation. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, 2025. URL `https://ojs.aaai.org/index.php/ICWSM/article/view/35855`.

[74] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019*, pages 4401–4410, 2019.

[75] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[76] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[77] L. Khachatryan, A. Movsisyan, V. Tadevosyan, R. Henschel, Z. Wang, S. Navasardyan, and H. Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[78] E. Kim, S. Kim, M. Park, R. Entezari, and S. Yoon. Rethinking training for de-biasing text-to-image generation: Unlocking the potential of stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13361–13370, June 2025.

[79] G. Kim, T. Kwon, and J. C. Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2426–2435, June 2022.

[80] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations (ICLR 2014), Conference Track Proceedings*, 2014.

[81] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

[82] A. Kumar, A. Levine, T. Goldstein, and S. Feizi. Curse of dimensionality on randomized smoothing for certifiable robustness. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5458–5467. PMLR, 13–18 Jul 2020.

[83] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J.-Y. Zhu. Multi-concept customization of text-to-image diffusion, 2023.

[84] K. Lee, L. Liu, K. Schwan, C. Pu, Q. Zhang, Y. Zhou, E. Yigitoglu, and P. Yuan. Scaling iterative graph computations with graphmap. In *Proceedings of the 27th IEEE international conference for High Performance Computing, Networking, Storage and Analysis (SC'15)*, pages 57:1–57:12, Austin, TX, November 15-20 2015.

[85] D. Li, J. Li, and S. Hoi. BLIP-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[86] H. Li, Y. Zou, Y. Wang, O. Majumder, Y. Xie, R. Manmatha, A. Swaminathan, Z. Tu, S. Ermon, and S. Soatto. On the scalability of diffusion-based text-to-image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9400–9409, 2024. doi: 10.1109/CVPR52733.2024.00898.

[87] H. Li, X. Zhang, X. Liu, Y. Gong, Y. Wang, Q. Chen, and P. Cheng. Enhancing large language model performance with gradient-based parameter selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24431–24439, Apr. 2025. doi: 10.1609/aaai.v39i23.34621. URL `https://ojs.aaai.org/index.php/AAAI/article/view/34621`.

[88] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 4328–4343, 2022.

[89] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[90] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee. Gligen: Open-set grounded text-to-image generation. *CVPR*, 2023.

[91] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 300–309, June 2023.

[92] J. Lin, L. Xu, Y. Liu, and X. Zhang. Composite backdoor attack for deep neural network by mixing existing benign features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 113–131, 2020.

[93] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, 2014.

[94] B. Liu, Z. Zhang, P. He, Z. Wang, Y. Xiao, R. Ye, Y. Zhou, W. Ku, and B. Hu. A survey of lottery ticket hypothesis. *CoRR*, abs/2403.04861, 2024.

[95] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21450–21474. PMLR, 23–29 Jul 2023.

[96] H. Liu, Y. Wu, S. Zhai, B. Yuan, and N. Zhang. Riatig: Reliable and imperceptible adversarial text-to-image generation with natural prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20585–20594, June 2023.

[97] J. Liu, J. Huang, Y. Zhou, X. Li, S. Ji, H. Xiong, and D. Dou. From distributed machine learning to federated learning: A survey. *Knowledge and Information Systems (KAIS)*, 64(4): 885–917, 2022.

[98] J. Liu, J. Jia, B. Ma, C. Zhou, J. Zhou, Y. Zhou, H. Dai, and D. Dou. Multi-job intelligent scheduling with cross-device federated learning. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 34(2):535–551, 2023.

[99] J. Liu, T. Che, Y. Zhou, R. Jin, H. Dai, D. Dou, and P. Valduriez. Aedfl: Efficient asynchronous decentralized federated learning with heterogeneous devices. In *Proceedings of the 24th SlAM International Conference on Data Mining (SDM'24)*, pages 833–841, Houston, TX, April 18-20 2024.

[100] J. Liu, J. Jia, T. Che, C. Huo, J. Ren, Y. Zhou, H. Dai, and D. Dou. Fedasmu: Efficient asynchronous federated learning with dynamic staleness-aware model update. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI'24)*, pages 13900–13908, Vancouver, Canada, February 20-27 2024.

[101] J. Liu, J. Jia, H. Zhang, Y. Yun, L. Wang, Y. Zhou, H. Dai, and D. Dou. Efficient federated learning using dynamic update and adaptive pruning with momentum on shared server data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 15(6):122:1–122:28, 2024.

[102] J. Liu, J. Ren, R. Jin, Z. Zhang, Y. Zhou, P. Valduriez, and D. Dou. Fisher information-based efficient curriculum federated learning with large language models. In *Proceedings of the 29th Conference on Empirical Methods in Natural Language Processing (EMNLP'24)*, pages 10497–10523, Miami, FL, November 12-16 2024.

[103] J. Liu, B. Ma, Q. Yu, R. Jin, J. Zhou, Y. Zhou, H. Dai, H. Wang, D. Dou, and P. Valduriez. Efficient federated learning with heterogeneous data and adaptive dropout. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 19(8):146:1–146:31, 2025.

[104] Q. Liu, Z. Zeng, J. He, Q. Yu, X. Shen, and L.-C. Chen. Alleviating distortion in image generation via multi-resolution diffusion models and time-dependent layer normalization. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

[105] Y. Liu, X. Ma, J. Bailey, and F. Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision (ECCV)*, pages 182–199, 2020.

[106] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[107] M. D. Ma, X. Wang, P.-N. Kung, P. J. Brantingham, N. Peng, and W. Wang. Star: Boosting low-resource information extraction by structure-to-text data generation with large language models. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 18751–18759, 2024. doi: 10.1609/aaai.v38i17.29839.

[108] G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley, 2000. ISBN 978-0-47100626-8. doi: 10.1002/0471721182. URL `https://doi.org/10.1002/0471721182`.

[109] J. Mohapatra, C.-Y. Ko, T.-W. Weng, P.-Y. Chen, S. Liu, and L. Daniel. Higher-order certification for randomized smoothing. In *Advances in Neural Information Processing Systems*, volume 33, pages 4501–4511. Curran Associates, Inc., 2020.

[110] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022.

[111] E. G. Ng, B. Pang, P. Sharma, and R. Soricut. Understanding guided image captioning performance across domains. *arXiv preprint arXiv:2012.02339*, 2020.

[112] T. A. Nguyen and A. Tran. Input-aware dynamic backdoor attack. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 3454–3464, 2020.

[113] T. A. Nguyen and A. T. Tran. WaNet: Imperceptible warping-based backdoor attack. In *9th International Conference on Learning Representations (ICLR)*, 2021.

[114] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8162–8171, 2021.

[115] A. Q. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR, 2022.

[116] C. A. O'Cinneide. Characterization of phase-type distributions. *Communications in Statistics: Stochastic Models*, 6(1):1–57, 1990. doi: 10.1080/15326349908807134.

[117] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges,

C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph. Gpt-4 technical report, 2024. URL `https://arxiv.org/abs/2303.08774`.

[118] C. Osborne, J. Ding, and H. R. Kirk. The AI community building the future? a quantitative analysis of development activity on hugging face hub. *Journal of Computational Social Science*, 7, 2024. doi: 10.1007/s42001-024-00300-8. URL `https://link.springer.com/article/10.1007/s42001-024-00300-8`.

[119] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022. NeurIPS 2022.

[120] K. J. Pacis, M. A. Almendrala, R. J. Paitone, and A. E. Jr. The relevance of the notion for all publicity is good publicity: The influencing factors in the 21st century. *International Journal of Research in Business and Social Science*, 11(2):42–56, 2022. doi: 10.20525/ijrbs.v11i2.1687. URL `https://www.ssbfnet.com/ojs/index.php/ijrbs/article/view/1687`.

[121] B. Palanisamy, L. Liu, K. Lee, S. Meng, Y. Tang, and Y. Zhou. Anonymizing continuous queries with delay-tolerant mix-zones over road networks. *Distributed and Parallel Databases (DAPD)*, 32(1):91–118, 2014.

[122] B. Palanisamy, L. Liu, Y. Zhou, and Q. Wang. Privacy-preserving publishing of multilevel utility-controlled graph datasets. *ACM Transactions on Internet Technology (TOIT)*, 18(2): 24:1–24:21, 2018.

[123] X. Pan, M. Zhang, B. Sheng, J. Zhu, and M. Yang. Hidden trigger backdoor attack on NLP models via linguistic style manipulation. In *Proceedings of the 31st USENIX Security Symposium (USENIX Security)*, pages 3611–3628, 2022.

[124] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. doi: 10.1109/ICCV51070.2023.00387.

[125] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

[126] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.

[127] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8599–8608, 2021.

[128] F. Qi, M. Li, Y. Chen, Z. Zhang, Z. Liu, Y. Wang, and M. Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 443–453, 2021.

[129] F. Qi, Y. Yao, S. Xu, Z. Liu, and M. Sun. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4873–4883, 2021.

[130] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989. doi: 10.1109/5.18626. URL `https://doi.org/10.1109/5.18626`.

[131] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[132] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020.

[133] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, 18–24 Jul 2021.

[134] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, 18–24 Jul 2021.

[135] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[136] R. N. Ranabhat, L. Wang, X. Qin, Y. Zhou, and K. Santosh. Multi-scale unrectified push-pull with channel attention for enhanced corruption robustness. In *Proceedings of the 2025 AAAI Summer Symposium Series (AAAI'25)*, pages 34–41, Dubai, UAE, May 20-22 2025.

[137] J. Ren, Y. Zhou, R. Jin, Z. Zhang, D. Dou, and P. Wang. Dual adversarial learning based network alignment. In *Proceedings of the 19th IEEE International Conference on Data Mining (ICDM'19)*, pages 1288–1293, Beijing, China, November 8-11 2019.

[138] J. Ren, Z. Zhang, J. Jin, X. Zhao, S. Wu, Y. Zhou, Y. Shen, T. Che, R. Jin, and D. Dou. Integrated defense for resilient graph matching. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, pages 8982–8997, Virtual Event, July 18-24 2021.

[139] J. Ren, Y. Zhou, J. Jin, L. Lyu, and D. Yan. Dimension-independent certified neural network watermarks via mollifier smoothing. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, pages 28976–29008, Honolulu, HI, July 23-29 2023.

[140] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, volume 32 of *PMLR*, pages 1278–1286, 2014.

[141] C. J. Romano. Comparative advertising in the united states and in france. *Northwestern Journal of International Law & Business*, 25:371–414, 2005. URL https://scholarlycommons.law.northwestern.edu/njilb/vol25/iss2/18.

[142] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[143] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[144] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pages 36479–36494. Curran Associates, Inc., 2022.

[145] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang. Dynamic backdoor attacks against machine learning models. In *Proceedings of the 7th IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 703–718, 2022.

[146] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *5th International Conference on Learning Representations (ICLR) 2017, Conference Track Proceedings*, 2017.

[147] D. Samuel, B. Meiri, H. Maron, Y. Tewel, N. Darshan, S. Avidan, G. Chechik, and R. Ben-Ari. Lightning-fast image inversion and editing for text-to-image diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[148] A. G. Savva, T. Theocharides, and C. Nicopoulos. Robustness of artificial neural networks based on weight alterations used for prediction purposes. *Algorithms*, 16(7):322, 2023. doi: 10.3390/a16070322.

[149] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[150] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018.

[151] X. Shen, C. Du, T. Pang, M. Lin, Y. Wong, and M. Kankanhalli. Finetuning text-to-image diffusion models for fairness. In *The Twelfth International Conference on Learning Representations*, 2024.

[152] Y. Shi, C. Xue, J. H. Liew, J. Pan, H. Yan, W. Zhang, V. Y. F. Tan, and S. Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 8839–8849. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00844. URL https://doi.org/10.1109/CVPR52733.2024.00844.

[153] I. Shumailov, Z. Shumaylov, D. Kazhdan, Y. Zhao, N. Papernot, M. A. Erdogdu, and R. Anderson. Manipulating SGD with data ordering attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 18021–18032, 2021.

[154] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, and Y. Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=nJfylDvgzlq.

[155] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

[156] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

[157] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.

[158] Y. Song and S. Ermon. Improved techniques for training score-based generative models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[159] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=PxTIG12RRHS`.

[160] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[161] H. Souri, M. Goldblum, L. Fowl, R. Chellappa, and T. Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 19165–19178, 2022.

[162] StabilityAI. Deepfloyd-if-i-m-v1.0, 2023. URL `https://huggingface.co/DeepFloyd/IF-I-M-v1.0`. Accessed: 2024-09-20.

[163] L. Struppek, D. Hintersdorf, and K. Kersting. Rickrolling the artist: Injecting backdoors into text encoders for text-to-image synthesis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4561–4573, Los Alamitos, CA, USA, oct 2023. IEEE Computer Society. doi: 10.1109/ICCV51070.2023.00423.

[164] Z. Su, L. Liu, M. Li, X. Fan, and Y. Zhou. Servicetrust: Trust management in service provision networks. In *Proceedings of the 10th IEEE International Conference on Services Computing (SCC'13)*, pages 272–279, Santa Clara, CA, June 27-July 2 2013.

[165] Z. Su, L. Liu, M. Li, X. Fan, and Y. Zhou. Reliable and resilient trust management in distributed service provision networks. *ACM Transactions on the Web (TWEB)*, 9(3):1–37, 2015.

[166] M. S. Tamber, J. Xian, and J. Lin. Can't hide behind the API: Stealing black-box commercial embedding models. In L. Chiruzzo, A. Ritter, and L. Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1958–1969, Albuquerque, New Mexico, Apr. 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.104. URL `https://aclanthology.org/2025.findings-naacl.104/`.

[167] Trend Micro Research. Exploiting trust in open-source AI: The hidden supply chain risk no one is watching, 2025. Available online: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/exploitingtrustinopensourceaithehiddensupplychainriskno-oneiswatching (accessed Sep 25, 2025).

[168] Y.-L. Tsai, C.-Y. Hsu, C.-M. Yu, and P.-Y. Chen. Formalizing generalization and adversarial robustness of neural networks to weight perturbations. In *Advances in Neural Information Processing Systems*, volume 34, pages 19692–19704, 2021.

[169] S. Tulyakov, M. Liu, X. Yang, and J. Kautz. MoCoGAN: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018*, pages 1526–1535, 2018.

[170] I. I. Uddin, Y. Zhou, C. Zhang, and L. Wang. Explainable novel category discovery in semantic concept space. In *Proceedings of the 2026 IEEE International Conference on Cognitive Machine Intelligence (CogMI'26)*, San Jose, CA, November 4-6 2026.

[171] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *Proceedings of the 9th ISCA Speech Synthesis Workshop (SSW) 2016*, page 125, 2016.

[172] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1747–1756, 2016.

[173] S. Vashishtha, A. Prakash, L. Morishetti, K. Nag, Y. Arora, S. Kumar, and K. Achan. Chaining text-to-image and large language model: A novel approach for generating personalized e-commerce banners. In R. Baeza-Yates and F. Bonchi, editors, *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 5825–5835. ACM, 2024. doi: 10.1145/3637528.3671636. URL `https://doi.org/10.1145/3637528.3671636`.

[174] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, pages 5998–6008, 2017.

[175] J. Vice, N. Akhtar, R. Hartley, and A. Mian. Bagm: A backdoor attack for manipulating text-to-image generative models. *IEEE Transactions on Information Forensics and Security*, 2024. doi: 10.1109/TIFS.2024.3386058.

[176] V. Voleti, A. Jolicoeur-Martineau, and C. Pal. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[177] D. S. Waller. A proposed response model for controversial advertising. *Journal of Promotion Management*, 11(2-3):3–15, 2006. doi: 10.1300/J057v11n02_02. URL `https://doi.org/10.1300/J057v11n02_02`.

[178] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. S. Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

[179] L. Wang, M. Nayyem, A. A. Rakin, K. Santosh, C. Zhang, and Y. Zhou. Explainability-guided defense: Attribution-aware model refinement against adversarial data attacks. In *Proceedings of the 25th IEEE International Conference on Data Mining (ICDM'25)*, Washington, DC, November 12-15 2025.

[180] L. Wang, I. I. Uddin, X. Qin, Y. Zhou, and K. Santosh. Explainability-driven defense: Grad-cam-guided model refinement against adversarial threats. In *Proceedings of the 2025 AAAI Summer Symposium Series (AAAI'25)*, pages 57–58, Dubai, UAE, May 20-22 2025.

[181] L. Wang, C. Zhang, I. I. Uddin, K. Santosh, X. Qin, and Y. Zhou. Bridging symmetry and robustness: On the role of equivariance in enhancing adversarial robustness. In *Advances in Neural Information Processing Systems 39: Annual Conference on Neural Information Processing Systems 2025 (NeurIPS'25)*, San Diego, CA, December 2-December 7 2025.

[182] T. Wang, Y. Yao, F. Xu, S. An, H. Tong, and T. Wang. An invisible black-box backdoor attack through frequency domain. In *European Conference on Computer Vision (ECCV)*, pages 396–413, 2022.

[183] P. Wei, S. Liu, X. Yang, L. Wang, and B. Zheng. Towards personalized bundle creative generation with contrastive non-autoregressive decoding. In E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, and G. Kazai, editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 2634–2638. ACM, 2022. doi: 10.1145/3477495.3531909. URL `https://doi.org/10.1145/3477495.3531909`.

[184] Y. Wei, Y. Zhu, P. Hui, and G. Tyson. Exploring the use of abusive generative AI models on civitai. In *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pages 6949–6958. ACM, 2024. doi: 10.1145/3664647.3681052. URL https://doi.org/10.1145/3664647.3681052.

[185] Y. Wen, J. Kirchenbauer, J. Geiping, and T. Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[186] T.-W. Weng, P. Zhao, S. Liu, P.-Y. Chen, X. Lin, and L. Daniel. Towards certificated model robustness against weight perturbations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6356–6363, 2020.

[187] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao. Backdoor attacks against deep learning systems in the physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6206–6215, 2021.

[188] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush. Transformers: State-of-the-art natural language processing. In Q. Liu and D. Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

[189] J. Z. Wu, Y. Ge, X. Wang, S. W. Lei, Y. Gu, Y. Shi, W. Hsu, Y. Shan, X. Qie, and M. Z. Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023.

[190] S. Wu, Y. Li, D. Zhang, Y. Zhou, and Z. Wu. Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, pages 5811–5820, Online, July 5-10 2020.

[191] S. Wu, Y. Li, M. Wang, D. Zhang, Y. Zhou, and Z. Wu. More is better: Enhancing open-domain dialogue generation via multi-source heterogeneous knowledge. In *Proceedings of the 26th Conference on Empirical Methods in Natural Language Processing (EMNLP'21)*, pages 2286–2300, Virtual Event / Punta Cana, Dominican Republic, November 7-11 2021.

[192] S. Wu, Y. Li, D. Zhang, Y. Zhou, and Z. Wu. Topicka: Generating commonsense knowledge-aware dialogue responses towards the recommended topic fact. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI'20)*, pages 3766–3772, Online, January 7-15 2021.

[193] S. Wu, M. Wang, D. Zhang, Y. Zhou, Y. Li, and Z. Wu. Knowledge-aware dialogue generation via hierarchical infobox accessing and infobox-dialogue interaction graph network. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21)*, pages 3964–3970, Virtual Event / Montreal, Canada, August 19-27 2021.

[194] Y. Xiao, Z. Zhang, Y. Fang, D. Yan, Y. Zhou, W. Ku, and B. Hui. Advancing certified robustness of explanation via gradient quantization. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM'24)*, pages 2596–2606, Boise, ID, October 21-25 2024.

[195] C. Xie, K. Huang, P. Chen, and B. Li. DBA: Distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations (ICLR)*, 2020.

[196] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057, 2015.

[197] X. Xu, Z. Wang, E. J. Zhang, K. Wang, and H. Shi. Versatile diffusion: Text, images and variations all in one diffusion model. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 7720–7731. IEEE, 2023.

[198] G. Yang, T. Duan, J. E. Hu, H. Salman, I. Razenshteyn, and J. Li. Randomized smoothing of all shapes and sizes. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10693–10705. PMLR, 13–18 Jul 2020.

[199] Y. Yang, B. Hui, H. Yuan, N. Gong, and Y. Cao. Sneakyprompt: Jailbreaking text-to-image generative models. In *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024*, pages 897–912. IEEE, 2024.

[200] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023.

[201] J. Yu, Y. Zhou, Z. Zhou, L. Lyu, D. Yan, R. Jin, and D. Dou. Autosmatic dialectic jailbreak: A framework for generating effective jailbreak strategies. In *Proceedings of the 14th International Conference on Learning Representations (ICLR'26)*, Rio de Janeiro, Brazil, April 23-27 2026.

[202] W. Yu, C. Zhu, Y. Fang, D. Yu, S. Wang, Y. Xu, M. Zeng, and M. Jiang. Dict-bert: Enhancing language model pre-training with dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1907–1918, 2022.

[203] Y. Yu, Y. Zhuang, J. Zhang, Y. Meng, A. J. Ratner, R. Krishna, J. Shen, and C. Zhang. Large language model as attributed training data generator: A tale of diversity and bias. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[204] X. Yuan, X. Ma, L. Guo, and L. Zhang. What lurks within? concept auditing for shared diffusion models at scale. `https://arxiv.org/abs/2504.14815`, 2025. arXiv preprint; authors report acceptance to CCS 2025.

[205] S. Zhai, Y. Dong, Q. Shen, S. Pu, Y. Fang, and H. Su. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. In *Proceedings of the 31st ACM International Conference on Multimedia*, page 1577–1587, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701085. doi: 10.1145/3581783.3612108.

[206] G. Zhang, Y. Zhou, S. Wu, Z. Zhang, and D. Dou. Cross-lingual entity alignment with adversarial kernel embedding and adversarial knowledge translation. *CoRR*, abs/2104.07837, 2021.

[207] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847, 2023.

[208] Q. Zhang, L. Liu, Y. Ren, K. Lee, Y. Tang, X. Zhao, and Y. Zhou. Residency aware inter-vm communication in virtualized cloud: Performance measurement and analysis. In *Proceedings of the 2013 IEEE International Conference on Cloud Computing (CLOUD'13)*, pages 204–211, Santa Clara, CA, June 27-July 2 2013.

[209] Q. Zhang, L. Liu, K. Lee, Y. Zhou, A. Singh, N. Mandagere, S. Gopisetty, and G. Alatorre. Improving hadoop service provisioning in a geographically distributed cloud. In *Proceedings of the 2014 IEEE International Conference on Cloud Computing (CLOUD'14)*, pages 432–439, Anchorage, AK, June 27-July 2 2014.

[210] Z. Zhang, Z. Zhang, Y. Zhou, Y. Shen, R. Jin, and D. Dou. Adversarial attacks on deep graph matching. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020 (NeurIPS'20)*, Virtual, December 6-12 2020.

[211] Z. Zhang, J. Jin, Z. Zhang, Y. Zhou, X. Zhao, J. Ren, J. Liu, L. Wu, R. Jin, and D. Dou. Validating the lottery ticket hypothesis with inertial manifold theory. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021 (NeurIPS'21)*, pages 30196–30210, Virtual, December 6-14 2021.

[212] Z. Zhang, Z. Zhang, Y. Zhou, L. Wu, S. Wu, X. Han, D. Dou, T. Che, and D. Yan. Adversarial attack against cross-lingual knowledge graph alignment. In *Proceedings of the 26th Conference on Empirical Methods in Natural Language Processing (EMNLP'21)*, pages 5320–5337, Virtual Event / Punta Cana, Dominican Republic, November 7-11 2021.

[213] Z. Zhang, Y. Zhou, X. Zhao, T. Che, and L. Lyu. Prompt certified machine unlearning with randomized gradient smoothing and quantization. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022 (NeurIPS'22)*, New Orleans, LA, November 28-December 9 2022.

[214] Z. Zhang, Q. Zhang, Z. Gao, R. Zhang, E. Shutova, S. Zhou, and S. Zhang. Gradient-based Parameter Selection for Efficient Fine-Tuning . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28566–28577, June 2024. doi: 10.1109/CVPR52733.2024.02699.

[215] J. Zhao, S. Wang, Y. Zhao, X. Hou, K. Wang, P. Gao, Y. Zhang, C. Wei, and H. Wang. Models are codes: Towards measuring malicious code poisoning attacks on pre-trained model hubs. In *IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2024. URL `https://arxiv.org/abs/2409.09368`. Proceedings version: IEEE Computer Society (ASE 2024).

[216] K. Zhao, X. Zhao, Z. Jin, Y. Yang, W. Tao, C. Han, S. Li, and L. Liu. Enhancing baidu multimodal advertisement with chinese text-to-image generation via bilingual alignment and caption synthesis. In G. H. Yang, H. Wang, S. Han, C. Hauff, G. Zuccon, and Y. Zhang, editors, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 2855–2859. ACM, 2024. doi: 10.1145/3626772.3661350. URL `https://doi.org/10.1145/3626772.3661350`.

[217] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14431–14440, 2020.

[218] S. Zhao, D. Chen, Y.-C. Chen, J. Bao, S. Hao, L. Yuan, and K.-Y. K. Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 2023.

[219] X. Zhao, Z. Zhang, Z. Zhang, L. Wu, J. Jin, Y. Zhou, R. Jin, D. Dou, and D. Yan. Expressive 1-lipschitz neural networks for robust multiple graph learning against adversarial attacks. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, pages 12719–12735, Virtual Event, July 18-24 2021.

[220] Y. Zhou. *Innovative Mining, Processing, and Application of Big Graphs*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2017.

[221] Y. Zhou and L. Liu. Clustering analysis in large graphs with rich attributes. In D. E. Holmes and L. C. Jain, editors, *Data Mining: Foundations and Intelligent Paradigms: Volume 1: Clustering, Association and Classification*. Springer, 2012.

[222] Y. Zhou and L. Liu. Social influence based clustering of heterogeneous information networks. In *Proceedings of the 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'13)*, pages 338–346, Chicago, IL, August 11-14 2013.

[223] Y. Zhou and L. Liu. Activity-edge centric multi-label classification for mining heterogeneous information networks. In *Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'14)*, pages 1276–1285, New York, NY, August 24-27 2014.

[224] Y. Zhou and L. Liu. Social influence based clustering and optimization over heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1): 1–53, 2015.

[225] Y. Zhou and L. Liu. Approximate deep network embedding for mining large-scale graphs. In *Proceedings of the 2019 IEEE International Conference on Cognitive Machine Intelligence (CogMI'19)*, pages 53–60, Los Angeles, CA, December 12-14 2019.

[226] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment (PVLDB)*, 2(1):718–729, 2009.

[227] Y. Zhou, H. Cheng, and J. X. Yu. Clustering large attributed graphs: An efficient incremental approach. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM'10)*, pages 689–698, Sydney, Australia, December 14-17 2010.

[228] Y. Zhou, L. Liu, and D. Buttler. Integrating vertex-centric clustering with edge-centric clustering for meta path graph analysis. In *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'15)*, pages 1563–1572, Sydney, Australia, August 10-13 2015.

[229] Y. Zhou, L. Liu, K. Lee, C. Pu, and Q. Zhang. Fast iterative graph computation with resource aware graph parallel abstractions. In *Proceedings of the 24th ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC'15)*, pages 179–190, Portland, OR, June 15-19 2015.

[230] Y. Zhou, L. Liu, K. Lee, and Q. Zhang. Graphtwist: Fast iterative graph computation with two-tier optimizations. *Proceedings of the VLDB Endowment (PVLDB)*, 8(11):1262–1273, 2015.

[231] Y. Zhou, L. Liu, S. Seshadri, and L. Chiu. Analyzing enterprise storage workloads with graph modeling and clustering. *IEEE Journal on Selected Areas in Communications (JSAC)*, 34(3): 551–574, 2016.

[232] Y. Zhou, A. Amimeur, C. Jiang, D. Dou, R. Jin, and P. Wang. Density-aware local siamese autoencoder network embedding with autoencoder graph clustering. In *Proceedings of the 2018 IEEE International Conference on Big Data (BigData'18)*, pages 1162–1167, Seattle, WA, December 10-13 2018.

[233] Y. Zhou, S. Wu, C. Jiang, Z. Zhang, D. Dou, R. Jin, and P. Wang. Density-adaptive local edge representation learning with generative adversarial network multi-label edge classification. In *Proceedings of the 18th IEEE International Conference on Data Mining (ICDM'18)*, pages 1464–1469, Singapore, November 17-20 2018.

[234] Y. Zhou, C. Jiang, Z. Zhang, D. Dou, R. Jin, and P. Wang. Integrating local vertex/edge embedding via deep matrix fusion and siamese multi-label classification. In *Proceedings of the 2019 IEEE International Conference on Big Data (BigData'19)*, pages 1018–1027, Los Angeles, CA, December 9-12 2019.

[235] Y. Zhou, K. L. Ling Liu, Qi Zhang, and B. Palanisamy. Enhancing collaborative filtering with multi-label classification. In *Proceedings of the 2019 International Conference on Computational Data and Social Networks (CSoNet'19)*, pages 323–338, Ho Chi Minh City, Vietnam, November 18-20 2019.

[236] Y. Zhou, J. Ren, S. Wu, D. Dou, R. Jin, Z. Zhang, and P. Wang. Semi-supervised classification-based local vertex ranking via dual generative adversarial nets. In *Proceedings of the 2019 IEEE International Conference on Big Data (BigData'19)*, pages 1267–1273, Los Angeles, CA, December 9-12 2019.

[237] Y. Zhou, L. Liu, K. Lee, B. Palanisamy, and Q. Zhang. Improving collaborative filtering with social influence over heterogeneous information networks. *ACM Transactions on Internet Technology (TOIT)*, 20(4):36:1–36:29, 2020.

[238] Y. Zhou, J. Ren, D. Dou, R. Jin, J. Zheng, and K. Lee. Robust meta network embedding against adversarial attacks. In *Proceedings of the 20th IEEE International Conference on Data Mining (ICDM'20)*, pages 1448–1453, Sorrento, Italy, November 17-20 2020.

[239] Y. Zhou, J. Ren, R. Jin, Z. Zhang, D. Dou, and D. Yan. Unsupervised multiple network alignment with multinominal gan and variational inference. In *Proceedings of the 2020 IEEE International Conference on Big Data (BigData'20)*, pages 868–877, Atlanta, GA, December 10-13 2020.

[240] Y. Zhou, Z. Zhang, S. Wu, V. Sheng, X. Han, Z. Zhang, and R. Jin. Robust network alignment via attack signal scaling and adversarial perturbation elimination. In *Proceedings of the 30th Web Conference (WWW'21)*, pages 3884–3895, Virtual Event / Ljubljana, Slovenia, April 19-23 2021.

[241] Y. Zhou, J. Ren, R. Jin, Z. Zhang, J. Zheng, Z. Jiang, D. Yan, and D. Dou. Unsupervised adversarial network alignment with reinforcement learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(3):50:1–50:29, 2022.

[242] Y. Zhou, Z. Zhang, Z. Zhang, L. Lyu, and W.-S. Ku. Effective federated graph matching. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*, Vienna, Austria, July 21-27 2024.

[243] Z. Zhou, D. Chen, C. Wang, C. Chen, and S. Lyu. Simple and fast distillation of diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[244] Z. Zhou, Y. Zhou, Z. Zhang, L. Lyu, D. Yan, R. Jin, and D. Dou. Flexible, efficient, and stable adversarial attacks on machine unlearning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML'25)*, Vancouver, Canada, July 13-19 2025.

[245] Z. Zhou, Y. Zhou, T. Che, Z. Zhang, J. Ren, D. Yan, Z. Jiang, Y. Shen, R. Jin, and J. Gao. Noise-free dynamic rank-adaptation via riemannian methods in federated fine-tuning. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS'26)*, Tangier, Morocco, May 2-5 2026.

[246] Z. Zhou, Y. Zhou, R. Jin, P. He, and P. Emami. Mitigating the modality gap in vision–language models with fractal spectral geometry. In *Proceedings of the 43rd International Conference on Machine Learning (ICML'26)*, Seoul, South Korea, July 6-11 2026.

[247] Z. Zhou, Y. Zhou, J. Yu, L. Lyu, L. Wang, K. Santosh, R. Jin, and D. Dou. Structured multi-step jailbreaking under a hamiltonian generative formulation. In *Proceedings of the 43rd International Conference on Machine Learning (ICML'26)*, Seoul, South Korea, July 6-11 2026.

A Supplementary Materials

A.1 Related Work

Trustworthy machine learning, which focuses on developing and deploying machine learning models that are not only accurate but also robust, private, fair, and explainable, has attracted active research in recent years [247, 246, 245, 201, 170, 34, 35, 26, 181, 244, 179, 68, 103, 136, 180, 122, 238, 210, 240, 219, 138, 212, 211, 206, 241, 71, 213, 70, 23, 97, 25, 139, 24, 98, 100, 99, 242, 194, 102, 101, 94, 227, 226, 36, 221, 37, 222, 164, 208, 223, 121, 209, 165, 224, 228, 15, 229, 230, 84, 231, 220, 233, 232, 137, 235, 234, 236, 225, 190, 192, 239, 237, 44, 72, 193, 191, 57].

Generative Models. Generative models have gained significant attention in recent years [55, 131, 4, 74, 172, 171, 146, 80, 140, 80, 61, 157, 41, 159, 174, 40, 21, 196, 169, 142] . The core of generative models is to learn data distributions and generate similar samples. Early works on generative models, such as Gaussian Mixture Models (GMM) [108] and Hidden Markov Models (HMM) [130], provided simple probabilistic frameworks to model data distributions and capture basic statistical dependencies. Variational Autoencoders (VAE) [80] are considered the first combination of deep learning and generative modeling. VAEs encode input data into a latent space by learning a probabilistic distribution, then sample a latent variable from this distribution and decode it to reconstruct the input. The model optimizes a loss function that balances reconstruction accuracy and the regularization of the latent space to match a prior distribution [80]. Generative Adversarial Networks (GANs) [54] proposed a novel adversarial training framework consisting of a generator and a discriminator. The generator learns to produce realistic data from random noise, while the discriminator learns to distinguish between real and generated data. Through the adversarial training between the generator and the discriminator, GAN learns to produce increasingly realistic data.

Diffusion Models. Diffusion models are generative models that utilize a diffusion process during generation [114, 156, 6, 88, 33, 30, 127, 95, 91, 75, 106, 160, 176, 207, 61, 41, 142, 135, 144, 159]. Diffusion models were introduced by [155], who proposed a diffusion process that gradually adds noise to data and reverses it to generate samples, forming the foundation for subsequent advancements. Ho et al. refined this approach with Denoising Diffusion Probabilistic Models (DDPM), employing a step-by-step denoising process to generate high-quality images [61]. Song & Ermon introduced Score-Based Generative Models (SGMs), which used score functions and continuous diffusion to further improve sample quality [158]. Dhariwal & Nichol advanced the field with Guided Diffusion, improving fidelity and diversity by conditioning the diffusion process on external data like class labels, making diffusion models competitive with GANs [41]. Diffusion models have since expanded into new domains, such as audio generation, demonstrated by [81], and video generation by [62], showing their broad applicability across different data modalities.

Text-to-Image Diffusion Models. Recent advancements in text-to-image (T2I) diffusion models have significantly enhanced both generation efficiency and generated image quality [1, 78, 67, 10, 147, 12, 154, 189, 126, 91, 207, 20, 59, 19, 14, 83, 76, 31, 27, 200, 218, 90, 77, 47, 197, 152, 185, 48, 7, 79, 110, 134, 144, 142, 115]. Early notable contributions include GLIDE [115], which introduced classifier-free guidance for generating photorealistic images from text, followed by DALL·E 2 [135], which improved text-image alignment by incorporating CLIP embeddings, and Imagen [144], which achieved unprecedented realism by leveraging large pre-trained language models [132] to guide the diffusion process. More recent breakthroughs, such as Stable Diffusion [142], further optimized the generative process by introducing a more efficient architecture, allowing for high-quality image generation while reducing computational costs. DeepFloyd IF [162] utilizes a cascaded diffusion model that progressively generates high-quality images in stages, each refining and increasing the resolution of the image. This cascading technique is designed to produce highly detailed and contextually accurate images from text prompts.

Exploiting T2I DMs for Advertisement Injection. Since T2I DMs generate images based on user prompts, they can be manipulated to generate images include specific patterns or objects. This vulnerability can be exploited to turn T2I DMs into tools for embedding advertisements. To the best of our knowledge, BAGM [175] is the first and only work that explicitly addresses this advertising scenario. BAGM proposes three approaches: surface, shallow, and deep attacks. The surface attack modifies user prompts by inserting brand-related words. For instance, if a user prompt contains the word “burger," the attack appends the brand name “McDonald’s" before “burger." The generated image will feature a McDonald’s burger to promote the brand. Note that the surface attack does not

fit into our attack scenario since we assume the attacker cannot modify user prompts. The shallow and deep attacks in BAGM share a similar principle. They begin by selecting a trigger semantically related to the target brand, e.g., “burger" when advertising McDonald’s. BAGM collects images rich in McDonald’s elements from the internet and forms malicious text-image pairs by associating the trigger “burger" with McDonald’s images. Similar to backdoor attacks, the shallow attack leverages these malicious text-image pairs to fine-tune the text encoder, while the deep attack uses them to fine-tune the U-Net in the generative model. As a result, when the user’s prompt contains the trigger, the generated images tend to include elements associated with McDonald’s.

Backdoor Attack Against T2I Pipelines.

Previous works that introduce harmful information into T2I pipelines are similar to backdoor attacks in neural networks, where a selected *trigger* is injected into the T2I diffusion model through fine-tuning [112, 105, 92, 217, 178, 195, 9, 113, 42, 89, 8, 187, 128, 129, 153, 123, 161, 43, 182, 145] . This results in adversarial behavior when trigger prompts are used, while performance on benign prompts remains largely unaffected. These backdoor attack methods on T2I DMs could potentially be repurposed to achieve the advertising objectives of our work, but none of these previous methods explicitly mention advertising as their goal. Several studies [96, 163, 53, 205] have explored creating *triggers* using unnatural inputs, such as replacing the letter ’l’ with the number ’1’ [96], incorporating zero-width space characters [205], replacing "red" to "read" [53], or use Cyrillic letters that are visually similar to English letters [163]. Although these works pioneered the exploration of adversarial triggers in T2I pipelines, the unnatural triggers they propose are less likely to appear naturally in typical user prompts. In contrast, other works [175, 199] define *triggers* with natural language words and fine-tune the model to associate them with adversarial targets. For example, [175] fine-tuned the word "drink" to associate with "Coca-Cola," leading the T2I DM to preferentially generate Coca-Cola when the word "drink" appears in a prompt. Though not explicitly classified as backdoor attacks, methods like [143, 51] embed specific subjects into generated images upon detecting a trigger, achieving a similar effect.

Existing backdoor attacks cannot address the adversarial-advertisement setting in this paper. First, all the previous backdoor attacks or similar techniques rely on unnatural trigger tokens, such as typos, letter substitutions, and non-Latin characters, as specified in the previous paragraph. Benign users are very unlikely to include such triggers in their prompts. Consequently, the attack success rate in real-life scenarios could be low. Second, when a backdoor is triggered, the model should generate a pre-defined pattern that was embedded during the attack stage (e.g., a brand logo). Because this pattern is fixed and independent of the input prompt, the model largely ignores the prompt’s original semantics, resulting in images that deviate a lot from the user’s expectation. Since the attacker cannot assume future prompts, a trigger-based backdoor cannot adapt the advertisement to the prompt’s content and therefore cannot satisfy the adversarial-advertisement objective. To address both limitations, the method proposed in this work does not rely on explicit triggers and instead conditions the advertisement insertion on the prompt’s latent semantics, making the generated image align well with users’ intent while seamlessly embedding the target brand.

A.2 Discussion on the Attack Scenario

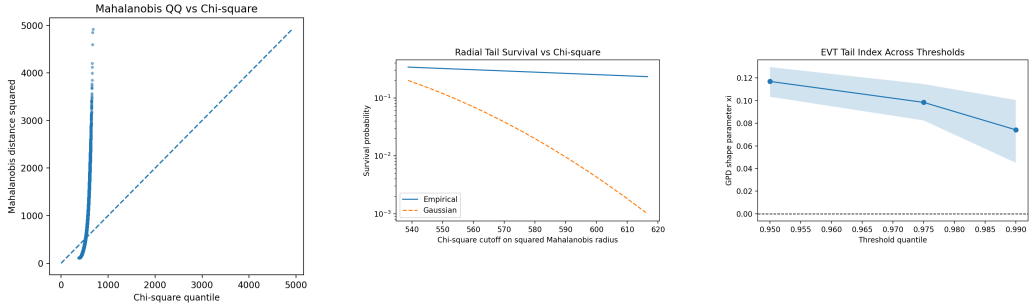
An important question about our proposed attack scenario is why would users adopt the customized checkpoint instead of using the vanilla release. We first note that using customized diffusion checkpoints is extremely common. Community hubs like HuggingFace, Civitai, and PixAI host hundreds of thousands of user-contributed models [184, 118]. For example, Civitai had tens of millions of visits per month, and a search for “SD 1.5” yields thousands of user-generated checkpoints, often promoted for unique artistic styles. Popular projects like AnimateDiff and DreamBooth also rely on HuggingFace for distributing such models [58, 143]. Since the vanilla release lacks distinctive features [207, 143], users are highly motivated to interact with these customized checkpoints.

From an attacker’s view, community hubs are attractive. Uploaders can exaggerate or fabricate model performance; multiple studies show gaps between claimed and measured performance, and most platforms perform limited verification [69, 73]. Uploading is free, enabling repeated reposting under different accounts. Prior work even found clusters of near-duplicate malicious checkpoints on HuggingFace, suggesting deliberate large-scale seeding [215]. Given high user traffic, model diversity, and weak security, community hubs pose non-trivial supply-chain risks [167, 204].

A substantial body of marketing research indicates that firms often prioritize awareness and talkability over sentiment, making unconventional campaigns practically plausible. First, a long-standing phenomenon, "all publicity is good publicity," is widely discussed and supported in the literature, which argues that any exposure can be beneficial by increasing presence and visibility [120]. Second, studies have shown that many companies actively adopt non-traditional advertising strategies; for example, firms have achieved significant publicity through cost-effective campaigns that prioritize exposure [177]. Third, even non-positive publicity can still be beneficial: Berger et al. [17] provide empirical evidence that less favorable reviews can increase sales for lesser-known authors. This finding is consistent with eye-tracking evidence that negative comments attract greater attention and relate to purchase intention [32]. These works further provide real-world motivation for businesses to single-mindedly pursue increased exposure. Taken together, these findings suggest that when the primary objective is exposure, firms are willing to adopt attention-maximizing tactics. Therefore, they support the plausibility that advertisers would employ adversarial advertisement strategies to increase brand visibility.

A.3 Empirical evidence of heavy-tailed nature

We analyzed the distribution of text embeddings produced by the SD 1.5 text encoder on the captions from the COCO 2017 training set using three well-established diagnostics: Mahalanobis QQ vs. chi-square, radial survival vs. chi-square, and EVT analysis based on GPD fitting. Figure 7(a) shows the Mahalanobis QQ plot. The empirical squared Mahalanobis distances deviate upward from the chi-square reference in the upper tail, indicating that large-radius samples occur much more frequently than would be expected under a Gaussian model. Figure 7(b) shows the radial survival curve. The empirical survival probability stays consistently above the Gaussian benchmark across high cutoffs, which directly suggests a substantially heavier-than-Gaussian radial tail. Figure 7(c) shows the EVT results. The estimated GPD shape parameter ξ remains positive across thresholds, with confidence intervals staying above zero, further supporting a stable heavy upper tail. Overall, these three analyses consistently suggest that the text embeddings in this setting exhibit heavy-tail-like behavior, which may also help explain why MCPHL leads to improved performance in AATIM.



(a) Mahalanobis QQ vs. chi-square (b) Radial survival vs. chi-square (c) EVT analysis

Figure 7: Mahalanobis QQ vs. chi-square, radial survival vs. chi-square, and EVT analysis based on GPD fitting.

A.4 Proof of Theorems

Theorem 4.5. *Given sufficient iterations $\mathscr{I}$, our estimation $Q_{\mathcal{A}}(x) = 1 - \bar{F}_{\mathcal{A}}(x_1, \ldots, x_d) = 1 - \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathcal{A}\mathbf{1}$ for the multivariate continuously scaled phase-type with Lévy distribution will converge to the empirical distribution $P_{\mathcal{A}}(x)$ estimated from real data.*

Proof. Let x_i represent the value of x at the i-th iteration out of a total of $\mathscr{I}$ iterations, and define the empirical distribution $P_A(x) = \frac{\#(\mathbf{X} \leq [x_i, \ldots, x_i])}{N^{d+1}}$, where N is the number of embeddings. The

expectation of the distribution $\mathbb{E}(\mathbf{X} \leq [x_i, \ldots, x_i])$ is given by:

$$\begin{aligned}\mathbb{E}(\mathbf{X} \leq [x_i, \ldots, x_i]) &= \int_0^\infty 1 - Q_{\mathcal{A}}(x)\, dx \\ &= \int_0^\infty \bar{F}_{\mathcal{A}}(x_1, \ldots, x_d)\, dx \\ &= \int_0^\infty \alpha_i \exp\left(\eta_i \mathbf{B}_i \sqrt{x}\right) \mathbf{D}_i \mathcal{A}_i \mathbf{1}\, dx\end{aligned} \tag{14}$$

Let $y = \sqrt{x}$, then $dx = 2y\, dy$. Using integration by parts formula, the integral part becomes:

$$\begin{aligned}\int_0^\infty \alpha_i \exp\left(\eta_i \mathbf{B}_i \sqrt{x}\right) \mathbf{D}_i \mathcal{A}_i \mathbf{1} dx &= 2\int_0^\infty y\alpha_i \exp\left(\eta_i \mathbf{B}_i y\right) \mathbf{D}_i \mathcal{A}_i \mathbf{1} dy \\ &= -2\alpha_i \int_0^\infty \exp\left(\eta_i \mathbf{B}_i y\right) \mathbf{D}_i \mathcal{A}_i \mathbf{1} dy\end{aligned} \tag{15}$$

Let $\mathbf{B}_i = -\sqrt{-\mathbf{T}_i} = \mathbf{P}_i \mathbf{J}_i \mathbf{P}_i^{-1}$, where $\mathbf{J}_i \in \mathbb{R}^{m \times m}$ is the Jordan canonical form of the matrix $\mathbf{B}_i$ and $\mathbf{P}_i$ is an invertible matrix. The Jordan canonical form $\mathbf{J}_i$ is composed of Jordan blocks, which are of the form:

$$\mathbf{J}_i = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_{ij} \end{pmatrix} \tag{16}$$

Each Jordan block J_{ij} is of the form:

$$J_{ij} = \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix} \tag{17}$$

where λ_i is an eigenvalue of matrix $\mathbf{B}_i$. Then, $\exp(\eta_i \mathbf{B}_i y) = \mathbf{P}_i \exp(\eta_i \mathbf{J}_i y) \mathbf{P}_i^{-1}$. We can compute the integral of each Jordan block J_{ij}:

$$\int_0^\infty \exp(\eta_i \lambda_i y) \begin{pmatrix} 1 & \eta_i y & \frac{(\eta_i y)^2}{2!} & \cdots & \frac{(\eta_i y)^{m-1}}{(m-1)!} \\ & 1 & \eta_i y & \cdots & \frac{(\eta_i y)^{m-2}}{(m-2)!} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & \eta_i y \\ & & & & 1 \end{pmatrix} dy \tag{18}$$

For the diagonal elements:

$$\int_0^\infty \exp(\eta_i \lambda_i y)\, dy = \frac{1}{-\eta_i \lambda_i} \tag{19}$$

For the off-diagonal elements that involve terms like $\eta_i y, \eta_i^2 y^2$, etc., the integrals of the form:

$$\int_0^\infty y^k \exp(\eta_i \lambda_i y)\, dy \tag{20}$$

These integrals can be computed using the Gamma function. For example:

$$\int_0^\infty y^k \exp(\eta \lambda_i y)\, dy = \frac{k!}{(-\eta \lambda_i)^{k+1}} \tag{21}$$

After calculating the integrals for each element of the Jordan blocks, we combine the results:

$$\int_0^\infty \exp(\eta_i \mathbf{B}_i y)\, dy = \mathbf{P}_i \int_0^\infty \exp(\eta_i \mathbf{J}_i y)\, dy\, \mathbf{P}_i^{-1} \tag{22}$$

Thus, the result of the integral and expected value is:

$$\mathbb{E}(\mathbf{X} \leq [x_i, \ldots, x_i]) = -2\alpha_i \mathbf{P}_i \begin{pmatrix} \frac{1}{-\eta_i\lambda_i} & \frac{\eta_i}{(-\eta_i\lambda_i)^2} & \cdots & \frac{(\eta_i)^{m-1}}{(-\eta_i\lambda_i)^m} \\ & \frac{1}{-\eta_i\lambda_i} & \cdots & \frac{(\eta_i)^{m-1}}{(-\eta_i\lambda_i)^m} \\ & & \ddots & \vdots \\ & & & \frac{1}{-\eta_i\lambda_m} \end{pmatrix} \mathbf{P}_i^{-1}\mathbf{D}_i\mathcal{A}_i\mathbf{1} \tag{23}$$

where each block in the diagonal corresponds to the contribution from a Jordan block, with terms involving λ_i and powers of η_i.
Similarly, we can derive the variance of the distribution, $\mathbb{V}(\mathbf{X} \leq [x_i, \ldots, x_i])$, as follows:

$$\begin{aligned} \mathbb{V}(\mathbf{X} \leq [x_i, \ldots, x_i]) &= \mathbb{E}[\mathbf{X}^2] - (\mathbb{E}[\mathbf{X}])^2 \\ &= \int_0^\infty 2x\left(1 - F_S(x_1, \ldots, x_d)\right) dx - \left(\int_0^\infty \bar{F}_{\mathcal{A}}(x_1, \ldots, x_d)dx\right)^2 \end{aligned} \tag{24}$$

where

$$\begin{aligned} \mathbb{E}[\mathbf{X}^2] &= \int_0^\infty 2x\left(1 - F_S(x_1, \ldots, x_d)\right) dx \\ &= 2\int_0^\infty x(\alpha_i \exp\left(\eta_i\mathbf{B}_i\sqrt{x}\right)\mathbf{D}_i\mathcal{A}_i\mathbf{1}) \\ &= 2x\alpha_i \exp\left(\eta_i\mathbf{B}_i x\right)\mathbf{D}_i\mathcal{A}_i\mathbf{1}\Big|_0^\infty - 2\int_0^\infty \alpha_i \exp\left(\eta_i\mathbf{B}_i x\right)\mathbf{D}_i\mathcal{A}_i\mathbf{1}dx \\ &= 4\alpha_i\mathbf{P}_i \begin{pmatrix} \frac{1}{-\eta_i\lambda_i} & \frac{\eta_i}{(-\eta_i\lambda_i)^2} & \cdots & \frac{(\eta_i)^{m-1}}{(-\eta_i\lambda_i)^m} \\ & \frac{1}{-\eta_i\lambda_2} & \cdots & \frac{(\eta_i)^{m-1}}{(-\eta_i\lambda_i)^m} \\ & & \ddots & \vdots \\ & & & \frac{1}{-\eta_i\lambda_m} \end{pmatrix} \mathbf{P}_i^{-1}\mathbf{D}_i\mathcal{A}_i\mathbf{1} \end{aligned} \tag{25}$$

For those samples $\mathbf{X}$ satisfying $\mathbf{X} \leq [x_i, \ldots, x_i]$, we can compute the corresponding expectation $\bar{\mathbf{X}} = \mathbb{E}(\mathbf{X} \mid \mathbf{X} \leq [x_i, \ldots, x_i])$ and variance $\sigma^2_{\mathbf{X}} = \mathbb{V}(\mathbf{X} \mid \mathbf{X} \leq [x_i, \ldots, x_i])$.

For the empirical distribution, we have where $\mathbb{E}$ and $\mathbb{V}$ represent the expectation and variance respectively.

$$\mathbb{E}(\mathbf{X} \leq [x_i, \ldots, x_i]) = -2\alpha_t P_t \begin{pmatrix} \frac{1}{-\eta_t\lambda_t} & \frac{\eta_t}{(-\eta_t\lambda_1)^2} & \cdots & \frac{(\eta_t)^{m-1}}{(-\eta_t\lambda_t)^m} \\ & \frac{1}{-\eta_t\lambda_t} & \cdots & \frac{(\eta_t)^{m-1}}{(-\eta_t\lambda_t)^m} \\ & & \ddots & \vdots \\ & & & \frac{1}{-\eta_t\lambda_k} \end{pmatrix} P_t^{-1}\mathbf{D}_t\mathcal{A}_t\mathbf{1}, \tag{26}$$

$$\mathbb{V}(\mathbf{X} \leq [x_i, \ldots, x_i]) = -4\alpha P \begin{pmatrix} \frac{1}{-\eta_t\lambda_t} & \frac{\eta_t}{(-\eta_t\lambda_t)^2} & \cdots & \frac{(\eta_t)^{m-1}}{(-\eta_t\lambda_t)^m} \\ & \frac{1}{-\eta_t\lambda_t} & \cdots & \frac{(\eta_t)^{m-1}}{(-\eta_t\lambda_t)^m} \\ & & \ddots & \vdots \\ & & & \frac{1}{-\eta_t\lambda_t} \end{pmatrix} P_t^{-1}\mathbf{D}_t\mathcal{A}_t\mathbf{1}. \tag{27}$$

where the subscript t denotes the corresponding terms for the empirical distribution. Since $\bar{\mathbf{X}} \in \mathbb{E}(\mathbf{X})$, it follows that

$$\mathbb{E}(\bar{\mathbf{X}}) = \frac{1}{I}\sum_{i=1}^{I} \mathbb{E}(\mathbf{X} \leq [x_i, \ldots, x_i]), \tag{28}$$

$$\mathbb{V}(\bar{\mathbf{X}}) = \frac{1}{I^2}\sum_{i=1}^{I} \mathbb{V}(\mathbf{X} \leq [x_i, \ldots, x_i]). \tag{29}$$

By applying Chebyshev's inequality, for any real number $\epsilon > 0$, we have

$$\begin{aligned} P(|\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})| \geq \epsilon) &= \int_{|\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})| \geq \epsilon} f(X) dX \\ &\leq \int_{|\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})| \geq \epsilon} \frac{|\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})|^2}{\epsilon^2} f(X) dX \\ &\leq \frac{1}{\epsilon^2} \int |\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})|^2 f(X) dX \\ &= \frac{1}{\epsilon^2 I^2} \sum_{i=1}^{I} \mathbb{V}(\mathbf{X} \leq [x_i, \cdots, x_i]) \\ &\leq \frac{\mathbb{V}(\mathbf{X})}{\epsilon^2 I}. \end{aligned} \tag{30}$$

Taking the limit as $I \to \infty$, we get

$$\lim_{I \to \infty} P(|\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})| \geq \epsilon) = \lim_{I \to \infty} \frac{\mathbb{V}(\mathbf{X})}{\epsilon^2 I} = 0. \tag{31}$$

Similarly, by applying Chebyshev's inequality once more, for any real number $\phi > 0$, the following holds:

$$P\left(|\mathbb{E}(\sigma^2_{\mathbf{X}}) - \mathbb{E}(\mathbb{V}(\mathbf{X}))| \geq \phi\right) \leq \frac{\mathbb{V}(\sigma^2_{\mathbf{X}})}{\phi^2 I} = 0. \tag{32}$$

Thus, the proof is complete. □

Theorem 5.3. *Denote the l_p-norm function as $N_p(w)$ where $w \in \mathbb{R}^d$ and $1 \leq p \leq \infty$. $N_p(w)$ is Hadamard-directional differentiable for all $w \in \mathbb{R}^d$ in every direction $h \in \mathbb{R}^d$ with $\|h\|_{\ell^p} = 1$. Moreover, the derivative $A^{N_p}_w(h)$, defined as in Eq. (9) with F replaced by N_p, satisfy the following inequality*

$$\left|A^{N_p}_w(h)\right| \leq 1. \tag{33}$$

Proof. Choose arbitrarily $w \in \mathbb{R}^d$ and $h \in \mathbb{R}^d$ with $\|h\|_p = 1$. Let $h_n \in R$ converge to h, and $t_n > 0$ converge to 0.

Step 1. Suppose $w \neq 0$ and $1 \leq p < \infty$. Then, we can write that

$$\begin{aligned} \lim_{n \to \infty} \frac{N_p(w + t_n h_n) - N_p(w)}{t_n} &= \lim_{n \to \infty} \frac{\left(\sum_{i=1}^d |w_i + t_n h_{n,i}|^p\right)^{\frac{1}{p}} - \left(\sum_{i=1}^d |w_i|^p\right)^{\frac{1}{p}}}{t_n} \\ &= \sum_{i=1}^d \Big(\sum_{j=1}^d |w_j|^p\Big)^{\frac{1}{p}-1} |w_i|^{p-1} h_i \\ &= \|w\|_p^{1-p} \sum_{i=1}^d |w_i|^{p-1} h_i. \end{aligned} \tag{34}$$

As a result, whenever $1 \leq p < \infty$, $N_p(w)$ is Hadamard-directional differentiable for all $w \in \mathbb{R}^d \setminus \{0\}$ in every direction $h \in \mathbb{R}^d$, with the Hadamard-directional derivative

$$\left|A^{N_p}_w(h)\right| = \|w\|_p^{1-p} \sum_{i=1}^d |w_i|^{p-1} h_i. \tag{35}$$

Moreover, based on Hölder's inequality, we have

$$\begin{aligned} \left|A^{N_p}_w(h)\right| &\leq \|w\|_p^{1-p} \Big(\sum_{i=1}^d (w_i^{p-1})^{\frac{p}{p-1}}\Big)^{\frac{p-1}{p}} \Big(\sum_{i=1}^d h_i^p\Big)^{\frac{1}{p}} \\ &= \|w\|_p^{1-p} \|w\|_p^{p-1} \|h\|_p \\ &= \|h\|_p, \end{aligned} \tag{36}$$

which affirms Eq. (10).

Step 2. Suppose $w \neq 0$ and $p = \infty$. Then,

$$\begin{aligned}\lim_{n\to\infty} \frac{N_\infty(w + t_n h_n) - N_\infty(w)}{t_n} &= \lim_{n\to\infty} \frac{\max_{1\le i\le d} |w_i + t_n h_{n,i}| - \max_{1\le i\le d} |w_i|}{t_n} \\ &= \operatorname{sign}(w_\iota) \operatorname{sign}(h_\iota) h_\iota, \end{aligned} \tag{37}$$

where $\iota \in \{1, \dots, d\}$ is such that $\max_{1\le i\le d} |w_i| = |w_\iota|$, and for any other $j \in \{1, \dots, d\}$, if $|w_j| = |w_\iota|$, then $|w_j + h_j| \le |w_\iota + h_\iota|$. Furthermore, Eq. (10) is straightforward from Eq. (37).

Step 3. If $w = 0$, then it is easy to see that

$$\lim_{n\to\infty} \frac{N_\infty(w + t_n h_n) - N_\infty(w)}{t_n} = \lim_{n\to\infty} \frac{N_\infty(t_n h_n)}{t_n} = \lim_{n\to\infty} N_p(h_n) = N_p(h) = \|h\|_p = 1. \tag{38}$$

The proof of this theorem is complete. □

Theorem 5.4. *Let F be a function on $\mathbb{R}^d$ uniformly bounded by a positive constant $M \le 1$, namely $\|F\|_\infty \le M \le 1$. Fix $w \in \mathbb{R}^d$. Let $\mathrm{Mask}_0 = \mathrm{Mask}(w)$, and let $G = G_\sigma$ be given as in Eq. (8) with $\mathrm{Mask}(w)$ replaced by Mask_0, where $\sigma > 0$ and $\varphi : \mathbb{R}^d \to \mathbb{R}$ given by*

$$\varphi(w) = K^{-1} e^{-\|w\|_{\ell^p}}, \quad K = \int_{\mathbb{R}^d} e^{-\|w\|_{\ell^p}} dw \quad \textit{and} \quad 1 \le p \le \infty. \tag{39}$$

Then for all $w' \in \mathbb{R}^d$, it holds that

$$|G(w) - G(w')| \le \frac{M}{\sigma\epsilon} \|w - w'\|_p, \tag{40}$$

Proof. Performing a change of variable $w - \mathrm{Mask}_0 \odot \mathbf{u} \mapsto \mathbf{v}$ in Eq. (8), we can write

$$G(w) = \prod_{i=1}^d \mathrm{Mask}_{0,i}^{-1} \int F(\mathbf{v}) \varphi_\sigma \left(\mathrm{Mask}_0^{-1} \odot (w - \mathbf{v})\right) d\mathbf{v}. \tag{41}$$

Notice that for any functions $f \colon \mathbb{R}^m \to \mathbb{R}$, $g \in \colon \mathbb{R}^n \to \mathbb{R}^m$, and $w, h \in \mathbb{R}^n$, we have the following formulae

$$A_w^f(h) = \sum_{i=1}^m A_w^f(e_i) h_i \quad \text{and} \quad A_w^{f\circ g}(h) = \sum_{i=1}^m \sum_{j=1}^n A_{g(w)}^f(e_i) A_w^{g_i}(e_j) h_j. \tag{42}$$

Thus applying the previous formulae and Theorem 5.3, for any direction $h \in \mathbb{R}^d$ with $\|h\|_p = 1$, it holds that

$$\begin{aligned} \left|A_w^G(h)\right| &= \sigma^{-1} \prod_{i=1}^d \mathrm{Mask}_{0,i}^{-1} \left| \sum_{i=1}^d \int F(\mathbf{v}) \varphi_\sigma \left(\mathrm{Mask}_0^{-1} \odot (w - \mathbf{v})\right) \right. \\ &\qquad \left. \cdot A^{N_p}_{\mathrm{Mask}_0^{-1}\odot(w-\mathbf{v})}(e_i) \sum_{j=1}^d A_w^{[\mathrm{Mask}_0^{-1}\odot(\cdot-\mathbf{v})]_i}(e_j) h_j \, d\mathbf{v} \right| \\ &\le \frac{M}{\sigma} \prod_{i=1}^d \mathrm{Mask}_{0,i}^{-1} \int \varphi_\sigma \left(\mathrm{Mask}_0^{-1} \odot (w - \mathbf{v})\right) \left| \sum_{i=1}^d A^{N_p}_{\mathrm{Mask}_0^{-1}\odot(w-\mathbf{v})}(e_i) \mathrm{Mask}_{0,i}^{-1} h_i \right| d\mathbf{v} \\ &\le \frac{M}{\sigma\epsilon} \prod_{i=1}^d \mathrm{Mask}_{0,i}^{-1} \int \varphi_\sigma \left(\mathrm{Mask}_0^{-1} \odot (w - \mathbf{v})\right) d\mathbf{v} \\ &= \frac{M}{\sigma\epsilon} \int \varphi_\sigma(\mathbf{u}) d\mathbf{v} = \frac{M}{\sigma\epsilon}. \end{aligned} \tag{43}$$

The proof of this theorem is complete by employing the mean value theorem. □

Before stating Theorem 5.5, we first introduce some necessary notations. Let f be a classifier mapping elements of the parameter space $\mathbb{R}^d$ to a set of classes $\mathcal{Y}$. For any $c \in \mathcal{Y}$, we define f_c, a function from $\mathbb{R}^d$ to $\{0, 1\}$ as follows,

$$f_c(w) = \mathrm{Id}_c\big(f(w)\big), \tag{44}$$

where Id denotes the indicator function. Let φ be given as in Eq. (39). For a positive constant σ, let g be a smoothing classifier given by

$$g(w) = g_\sigma(w) = \arg\max_{c \in \mathcal{Y}} f_c * \varphi_\sigma(w), \tag{45}$$

$$= \arg\max_{c \in \mathcal{Y}} \int f_c(u)\, \varphi_\sigma\left((w-u) \odot \mathrm{Mask}(w)\right)\, du, \tag{46}$$

where $\varphi_\sigma(w) = \sigma^{-d}\varphi(w/\sigma)$. Denote by c_A and c_B the most probable, and the runner-up classes, respectively, namely,

$$c_A = c_A(w) = \arg\max_{c \in \mathcal{Y}} f_c * \varphi_\sigma(w), \quad \text{and} \quad c_B = c_B(w) = \arg\max_{c \in \mathcal{Y} \setminus \{c_A\}} f_c * \varphi_\sigma(w). \tag{47}$$

We also write

$$v_A = v_A(w) = f_{c_A} * \varphi_\sigma(w) \quad \text{and} \quad v_B = v_B(w) = f_{c_B} * \varphi_\sigma(w). \tag{48}$$

Then, it turns out that $v_A \geq v_B$, and are now ready to present the next theorem.

Theorem 5.5. *Let f be a classifier defined on $\mathbb{R}^d$ with values in $\mathcal{Y}$, and let g be the smoothing classifier defined as in Eq. (45) with some $\sigma > 0$ and φ given by Eq. (11). Fix $w \in \mathbb{R}^d$. Let c_A and c_B be defined as in Eq. (47), let v_A and v_B be given by Eq. (48), and let ϵ be defined in Definition 5.1. Then, for any $w' \in \mathbb{R}^d$, $g(w') = g(w)$ whenever $\|w' - w\|_p \leq r_p$ ($1 \leq p \leq \infty$) with*

$$r_p = \frac{v_A - v_B}{2} \cdot \sigma\epsilon. \tag{49}$$

Proof. Recall that f_c defined as in Eq. (44) takes values in $\{0, 1\}$. Thus, Theorem 5.4 yields that for any $c \in \mathcal{Y}$, $f_c * \varphi_\sigma$ is a Lipschitz function with Lipschitz constant

$$L = \frac{1}{\sigma\epsilon}. \tag{50}$$

As a result, for any w' such that $\|w' - w\|_p \leq r_p$, we have

$$|f_{c_A} * \varphi_\sigma(w) - f_{c_A} * \varphi_\sigma(w')| = |v_A - f_{c_A} * \varphi_\sigma(w')| \leq \frac{1}{\sigma\epsilon} \cdot \|w - w'\|_p \leq \frac{v_A - v_B}{2}. \tag{51}$$

This implies that

$$f_{c_A} * \varphi_\sigma(w') \geq v_A - \frac{v_A - v_B}{2} = \frac{v_A + v_B}{2}. \tag{52}$$

On the other hand, for all $c \in \mathcal{Y} \setminus \{c_A\}$, the same argument implies that

$$|f_c * \varphi_\sigma(w) - f_c * \varphi_\sigma(w')| \leq \frac{v_A - v_B}{2}, \tag{53}$$

which further leads to the property that

$$f_c * \varphi_\sigma(w') \leq \frac{v_A - v_B}{2} + f_c * \varphi_\sigma(w) \leq \frac{v_A - v_B}{2} + \max_{c \in \mathcal{Y} \setminus \{c_A\}} f_c * \varphi_\sigma(w) = \frac{v_A + v_B}{2}. \tag{54}$$

Therefore,

$$g(w') = \arg\max_{c \in \mathcal{Y}} f_c * \varphi_\sigma(w') = c_A = g(w).$$

The proof of this theorem is complete. □

A.5 Experimental Details

Baselines. We compare our AATIM framework with nine baselines. **VillanDiffusion** [38] works similarly to traditional backdoor attacks. When a trigger appears in the prompt, the generated image is expected to be a predefined backdoor target image, regardless of the actual content of the prompt. The following works are not backdoor attack methods. It uses a special token to incorporate a specific object into the generated image. **RIATIG** [96] adopt a genetic-based approach to generate manipulated prompts, such as inserting extra spaces into words, swapping two characters, and deleting one character. **BAGM** [175] uses real words as triggers and employs fine-tuning to associate the trigger with the target object. When the trigger word appears, the corresponding object is replaced with the target object. **SneakyPrompt** [199] uses a reinforcement learning approach to guide the token-level perturbations. Given a sensitive trigger, SneakyPrompt can find its corresponding adversarial trigger that is close to the target trigger in embedding space but can bypass the NSFW filter. **DreamBooth** [143] fine-tunes the model with a special token to embed a target object into the prompt's context, allowing the model to generate images with the desired subject based on user intent. **Textual Inversion** [51] is conceptually similar to DreamBooth since both aim to integrate specific objects into a model's output, but Textual Inversion focuses on learning a small embedding for a special token without fine-tuning the entire model. **BLIP-Diffusion** [85] utilized a two-stage pre-training method powered by BLIP-2 for zero-shot and fine-tuned subject-driven generation, enabling zero-shot and fine-tuned subject-driven generation. **DreamStyler** [2] utilizes a context-aware text prompt to improve image quality. **FFD** [151] proposed to use a distributional alignment loss to address bias in T2I diffusion models.

Evaluation metrics. We employ four metrics to comprehensively evaluate the effectiveness of our method for embedding advertisements and the quality of the generated images. To measure the effectiveness of embedding advertisements into the T2I DM, we utilize the evaluation metrics from BAGM [175]. We use the CLIP (Contrastive Language-Image Pre-training) and BLIP (Bootstrapping Language-Image Pre-training) models to calculate $\mathbf{ASR_{VC}}$ (Visual Classification Attack Success Rate) and $\mathbf{ASR_{VL}}$ (Vision-Language Attack Success Rate) as proposed in [175] to measure the effectiveness of advertisement injection. ASR_{VC} calculates the percentage of generated images that are classified as containing the target object O_{tar}, i.e., $\text{ASR}_{\text{VC}} = \frac{N_{target}}{N_{samples}} \times 100\%$. ASR_{VL} measures how often the generated images contain O_{tar} in the captions produced by a captioning model, i.e., $\text{ASR}_{\text{VL}} = \frac{N_{captions_with_target}}{N_{samples}} \times 100\%$. To assess the quality of the generated images, we employ two commonly used metrics in literature: CLIP score (**CLIP**) [51] and Fréchet Inception Distance (**FID**) [38, 199]. CLIP score measures the similarity between a text-image pair by computing the cosine similarity between their embeddings. These embeddings are generated by the CLIP model. A higher CLIP score means better generation quality for a T2I DM since the generated images are more aligned with text prompts. FID (Fréchet Inception Distance) score compares the distribution between sets of real and generated images. A lower FID score indicates better fidelity of the generated images. Higher ASR_{VC} and ASR_{VL} indicate more effective advertisement implantation, i.e., the higher, the better. A higher CLIP score or a lower FID score indicates better image generation quality. Higher CLIP is better and lower FID is better.

Experiment environment. The experiments were conducted on a compute server running on Red Hat Enterprise Linux 7.2 with 2 CPUs of Intel Xeon E5-2650 v4 (at 2.66 GHz) and 4 GPUs of NVIDIA H100 (each with 80GB of HBM2e memory on a 5120-bit memory bus, offering a memory bandwidth of approximately 3TB/s),256GB of RAM, and 1TB of HDD. The codes were implemented in Python 3.12.3 and PyTorch 2.3.0.

Dataset. We study the adversarial advertisement task on three representative image-text paired datasets: Microsoft COCO (**COCO**) [93][2], LAION-5B (**LAION**) [149][3], and Conceptual Captions (**CC**) [150, 111][4]. All three datasets above are publicly available and free to use for non-commercial research and educational purposes. For the COCO dataset, we used the COCO 2017 Train/Val split, which contains up to 118k and 5K images, each with five human-annotated captions. The LAION dataset contains up to 5.85 billion image-caption pairs, which are CLIP-filtered. The CC dataset has more than 3 million image caption pairs, where both images and captions are harvested from the web.

[2]https://cocodataset.org

[3]https://laion.ai/blog/laion-5b/

[4]https://github.com/google-research-datasets/conceptual-captions

Training. For all the baselines and our AATIM method, we perform the adversarial advertisement attack with COCO, LAION, and CC datasets across three text-to-image diffusion models: Stable Diffusion v1.5 (SD) [142], Latent Diffusion Model (LDM) [142], and DeepFloyd IF (DF) [162]. Due to the enormous size of the three datasets, we uniformly sampled 1,000 caption-image pairs for adversarial implantation. We modified the above three models based on the Hugging Face Diffusers library[5] and implemented our attack pipeline accordingly. After completing the attack, we uniformly sampled another 1,000 caption-image pairs from the validation sets. The captions were fed into the attacked model, and the generated images were evaluated by computing ASR_{VC}, ASR_{VL}. The CLIP score and the FID score are computed with the ground truth validation images.

Implementation. Among nine state-of-the-art generative frameworks on text-to-image diffusion models, eight of them have the official implementation, including BLIP-Diffusion [85], DreamStyler [2], FFD [151], RIATIG [96], DreamBooth [143], Textual Inversion [51], VillanDiffusion [38], and SneakyPrompt [199]. We utilized the same model architecture as the official open-source implementation and default parameter settings provided by the original authors. All hyperparameters are standard values from reference codes or prior works. To our best knowledge, the authors did not provide the complete training code and training dataset for BAGM [175]. We tried our best to implement these approaches in terms of the algorithm description from the original papers. All hyperparameters are standard values from the reference papers.

Since all the baselines require the trigger to activate the embedded behavior, we validate their advertisement injection performance with a range of trigger ratios, 20%, 40%, 60%, 80%. The above open-source codes from the GitHub are licensed under the MIT License, which only requires preservation of copyright and license notices and includes the permissions of commercial use, modification, distribution, and private use. We will release our open-source code on GitHub and maintain a project website with detailed documentation for long-term access by other researchers and end-users after the paper is accepted.

For our AATIM framework, we performed hyperparameter selection by performing a parameter sweep on parameters below: number of attack steps $\in \{1000, 2000, 3000, 4000, 5000\}$, alignment attack step sizes $\eta_A \in [1e^{-5}, 1e^{-3}]$, density attack step sizes $\eta_M \in [1e^{-6}, 1e^{-3}]$, batch size fixed as $\mathcal{B} = 8$ due to GPU memory constraints. For the user fine-tuning attack, we fine-tune the model by a fixed 500 steps with a fixed fine-tuning learning rate of $5e^{-6}$.

Notations Summary. Table 5 is a summary of definitions used in the main paper.

[5] https://huggingface.co/docs/diffusers/en/index

Symbol	Definition
s	Non-advertising text prompt
$\hat{s}$	Advertising-augmented prompt (contains brand)
$\mathcal{S},\ \mathcal{Z},\ \mathcal{I}$	Prompt, embedding, and image spaces
$E(\cdot)$	Trainable text encoder of the diffusion model
$E_f(\cdot)$	Frozen text encoder used for advertising prompts
$z_s = E(s)$	Embedding of non-advertising prompt
$z_{\hat{s}} = E_f(\hat{s})$	Embedding of advertising prompt
O_{tar}	Target object / advertised brand (e.g., McDonald's)
$\mathcal{E}$	Set of advertising-prompt embeddings
MCPHL	Multivariate Continuously Scaled Phase-type with Lévy
α	Initial probability vector of MCPHL
T	Sub-intensity matrix of MCPHL
η	Lévy scale parameter
$A,\ D$	Diagonal matrices in survival-function parameterization
$B = -\sqrt{-T}$	Matrix square-root of $-T$
$Q_A(x)$	CDF of prompt embedding under MCPHL
$p(x)$	PDF of prompt embedding under MCPHL
$\eta_A,\ \eta_M$	Step sizes for alignment / density objectives
$w,\ w'$	Current / perturbed parameter of E
Mask(w)	Coordinate-wise importance mask in $[\epsilon, 1]^d$
ϵ	Minimum mask threshold to control smoothing strength
C	Temporary linear head for gradient-importance scoring
$F,\ G$	Base and mollified functions in mollification theory
φ_σ	Mollification kernel with noise level σ
σ	smoothing noise level
L_g	Lipschitz constant of g
r_p	Certified ℓ_p radius of g
$c_A,\ c_B$	Top-2 classes predicted at w
$\pi_A(x),\ \pi_B(x)$	Probabilities of c_A, c_B output by f on x
Θ	Positive scaling variable in MCPHL (Laplace-style)
μ	Location parameter of Lévy distribution
$v_A,\ v_B$	Corresponding confidences of smoothed classifier g
d	Dimensionality of parameters / embeddings

Table 5: Summary of key notations used throughout the AATIM framework.

Hyperparameter settings. Unless otherwise specified, we used the following parameters as shown in Table 6.

Table 6: Hyper-parameter settings.

Parameter	**Value**
Number of $< s, \hat{s} >$ pairs in attack	100
Number of attack steps for SD	10000
Number of attack steps for DF	10000
Number of attack steps for LDM	10000
Number of image generations	1000
Batch size $\mathcal{B}$	8
Alignment step size η_A	$5e^{-5}$
Density step size η_M	$1e^{-5}$
Location parameter μ for Lévy distribution	0
Number of Monte Carlo trials N	1000
Noise level σ	1
Mask threshold ϵ	0.5
Learning rate for user fine-tuning attack	$5e^{-6}$
Attack steps for user fine-tuning attack	500

Algorithm. Algorithm 1 described our masked smoothing method in detail. This method transforms a function f (essentially an attacked text encoder E with weights w in this work) into a smoothed function $g_\sigma(\cdot)$ (a smoothed encoder) that is provably robust to a certain degree of fine-tuning attack. Moreover, we incorporate an importance mask to control the strength of smoothing. We first obtain the parameter-wise importance mask in Stage 1. Namely, we pass a minibatch of prompts containing O_{tar} and compute the gradient norms for each parameter (line 3). These norms are linearly rescaled to the interval $[\epsilon, 1]$ (line 5), where ϵ controls the strength of smoothing. Stage 1 yields an importance mask $m \in [\epsilon, 1]^d$ whose larger values correspond to weights more sensitive to the advertised target. In Stage 2, we first define a Friedrichs kernel as described in Theorem 5.4 (line 8). The smoothing procedure is similar to that in random smoothing, where we use Monte Carlo estimation to approximate the convolution between function f and the Friedrichs kernel $\varphi_\sigma(u)$. Given a prompt s, we perform N Monte-Carlo trials: at each trial we sample a noise vector u from the mollifier distribution φ_σ (line 12), scale it element-wise by the importance mask m, and add the result to the parameters of f (line 13), yielding an intermediate embedding output $\hat{e}$ (line 14). Finally, we average the N intermediate embeddings to obtain the smoothed inference embedding (line 17). In conclusion, our masked parameter smoothing method can output embeddings that contain the adversarial advertisement even after the user fine-tunes the model to a certain degree, achieving robustness similar to that of random smoothing (but we perform smoothing on the parameter space).

A.6 Additional Experiments

Performance with varying trigger ratio. Tables 7-30 exhibit the ASR_{VC}, ASR_{VL}, CLIP score, and FID scores obtained by ten adversarial advertisement approaches by varying trigger ratio between 20% to 80% on three datasets of COCO, CC, and LAION respectively. Similar trends can be observed for the comparison of adversarial advertisement effectiveness and generation quality in these figures: our AATIM method achieves the highest ASR_{VC} and ASR_{VL} as well as the best generation quality in most cases. Our AATIM method does not rely on an adversarial trigger to activate advertisement generation, so the ASRVC and ASRVL do not decrease as the trigger ratio declines. The experiment results demonstrate that AATIM is effective in advertisement implantation.

Algorithm 1 Masked Parameter Smoothing

Input: encoder weights $w \in \mathbb{R}^d$, minibatch $\mathcal{S}_{\text{tar}} = \{\hat{s}_0, \ldots, \hat{s}_{\mathcal{B}}\}$ containing O_{tar}, smoothing std. $\sigma > 0$, mask threshold $\epsilon > 0$, number of Monte-Carlo samples N
Output: smoothed embedding function $g_\sigma(\cdot)$

Stage 1: Importance masking;
Compute gradient norms for each parameter:
    $g_i \leftarrow \left\| \nabla_{w_i} \ell\big(f(\mathcal{S}_{\text{tar}})\big) \right\|_2$;
Normalize to $[\,\epsilon, 1\,]$:
    $m_i \leftarrow \epsilon + (1-\epsilon)\dfrac{g_i - \min g}{\max g - \min g}$;
Form mask vector $m = (m_1, \ldots, m_d)^\top$;
Stage 2: Monte-Carlo smoothing at inference;
Define mollifier density $\varphi_\sigma(u) = \sigma^{-d}\varphi(u/\sigma)$, φ as defined in equation 39;
foreach user prompt s **do**
    $\hat{e} \leftarrow 0$; // running sum of embeddings
    for $j \leftarrow 1$ **to** N **do**
        sample $u^{(j)} \sim \varphi_\sigma$;
        $\tilde{w} \leftarrow w - m \odot u^{(j)}$; // inject weighted noise based on mask
        $e^{(j)} \leftarrow g_{\tilde{w}}(s)$; // forward pass
        $\hat{e} \leftarrow \hat{e} + e^{(j)}$;
    end for
    $g_\sigma(s) \leftarrow \hat{e}/N$; // smoothed embedding
end for
return $g_\sigma(\cdot)$

Table 7: Performance with 20% trigger ratio and COCO dataset on SD

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.139	0.090	8.20	279.34
RIATIG	0.182	0.078	17.77	162.67
DreamBooth	0.087	0.054	15.01	156.71
Textual Inversion	0.091	0.148	16.02	162.78
VillanDiffusion	0.175	0.127	9.49	309.55
DreamStyler	0.095	0.008	11.11	256.73
FFD	0.103	0.110	16.93	177.89
SneakyPrompt	0.153	0.169	17.64	180.95
BAGM	0.119	0.150	18.21	165.49
AATIM	**0.860**	**0.703**	**20.33**	**154.54**

Table 8: Performance with 40% trigger ratio and COCO dataset on SD

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.354	0.291	8.21	259.10
RIATIG	0.309	0.217	17.97	184.35
DreamBooth	0.170	0.179	15.03	156.44
Textual Inversion	0.143	0.230	15.05	172.81
VillanDiffusion	0.315	0.301	9.61	312.75
DreamStyler	0.168	0.066	11.14	262.04
FFD	0.183	0.174	17.33	171.90
SneakyPrompt	0.274	0.195	17.49	176.92
BAGM	0.309	0.221	18.17	164.54
AATIM	**0.860**	**0.703**	**20.33**	**154.54**

Table 9: Performance with 20% trigger ratio and LAION dataset on SD

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.162	0.154	8.77	254.44
RIATIG	0.185	0.177	18.95	174.64
DreamBooth	0.101	0.072	17.92	110.75
Textual Inversion	0.101	0.092	17.42	131.52
VillanDiffusion	0.149	0.146	11.05	306.35
DreamStyler	0.006	0.043	17.95	181.23
FFD	0.093	0.104	17.38	187.33
SneakyPrompt	0.130	0.094	18.94	183.78
BAGM	0.088	0.142	16.08	147.40
AATIM	**0.658**	**0.577**	**19.09**	**106.00**

Table 10: Performance with 40% trigger ratio and LAION dataset on SD

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.251	0.283	8.72	275.04
RIATIG	0.285	0.243	17.79	165.82
DreamBooth	0.172	0.177	18.99	112.45
Textual Inversion	0.164	0.198	16.48	121.85
VillanDiffusion	0.231	0.233	10.36	308.11
DreamStyler	0.084	0.096	16.93	177.61
FFD	0.160	0.173	17.29	193.84
SneakyPrompt	0.215	0.127	17.69	158.80
BAGM	0.197	0.124	16.04	136.07
AATIM	**0.658**	**0.577**	**19.09**	**106.00**

Table 11: Performance with 60% trigger ratio and LAION dataset on SD

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.427	0.362	8.86	259.89
RIATIG	0.331	0.290	17.66	146.18
DreamBooth	0.238	0.289	17.51	115.01
Textual Inversion	0.229	0.253	17.77	123.37
VillanDiffusion	0.338	0.333	10.37	315.43
DreamStyler	0.134	0.107	16.64	171.12
FFD	0.220	0.232	16.20	199.71
SneakyPrompt	0.335	0.191	17.87	151.47
BAGM	0.281	0.194	16.26	119.12
AATIM	**0.658**	**0.577**	**19.09**	**106.00**

Table 12: Performance with 80% trigger ratio and LAION dataset on SD

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.441	0.422	8.78	252.27
RIATIG	0.493	0.426	17.30	136.47
DreamBooth	0.337	0.316	17.77	114.20
Textual Inversion	0.361	0.332	17.50	122.29
VillanDiffusion	0.474	0.427	10.54	315.45
DreamStyler	0.158	0.132	17.11	166.98
FFD	0.291	0.335	16.92	192.19
SneakyPrompt	0.427	0.365	17.12	143.21
BAGM	0.325	0.322	17.36	109.23
AATIM	**0.658**	**0.577**	**19.09**	**106.00**

Table 13: Performance with 20% trigger ratio and CC dataset on SD

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.081	0.118	10.82	244.51
RIATIG	0.162	0.124	17.29	247.97
DreamBooth	0.117	0.092	16.01	126.99
Textual Inversion	0.119	0.086	15.97	116.99
VillanDiffusion	0.277	0.255	10.77	315.79
DreamStyler	0.139	0.098	16.01	116.99
FFD	0.115	0.131	15.19	155.05
SneakyPrompt	0.120	0.113	17.76	165.97
BAGM	0.134	0.112	15.98	136.98
AATIM	**0.711**	**0.669**	**18.87**	**101.34**

Table 14: Performance with 40% trigger ratio and CC dataset on SD

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.143	0.098	10.75	257.23
RIATIG	0.309	0.290	17.48	160.08
DreamBooth	0.243	0.213	16.02	122.72
Textual Inversion	0.187	0.194	16.02	118.97
VillanDiffusion	0.349	0.320	11.30	322.38
DreamStyler	0.081	0.114	16.05	118.04
FFD	0.204	0.226	15.91	147.51
SneakyPrompt	0.229	0.173	18.06	168.08
BAGM	0.212	0.227	17.03	137.65
AATIM	**0.711**	**0.669**	**18.87**	**101.34**

Table 15: Performance with 60% trigger ratio and CC dataset on SD

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.392	0.289	10.87	245.19
RIATIG	0.366	0.302	15.00	145.45
DreamBooth	0.338	0.290	14.05	113.00
Textual Inversion	0.360	0.295	15.95	106.96
VillanDiffusion	0.502	0.436	11.11	337.80
DreamStyler	0.210	0.128	14.96	114.02
FFD	0.307	0.316	15.92	151.14
SneakyPrompt	0.387	0.403	17.37	137.59
BAGM	0.348	0.310	16.01	118.35
AATIM	**0.711**	**0.669**	**18.87**	**101.34**

Table 16: Performance with 80% trigger ratio and CC dataset on SD

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.552	0.514	10.67	236.41
RIATIG	0.494	0.431	14.40	128.67
DreamBooth	0.448	0.402	14.49	108.37
Textual Inversion	0.415	0.448	17.92	111.49
VillanDiffusion	0.582	0.554	9.19	342.15
DreamStyler	0.215	0.209	15.59	114.18
FFD	0.391	0.442	14.84	144.52
SneakyPrompt	0.486	0.433	15.30	131.03
BAGM	0.446	0.412	15.13	107.61
AATIM	**0.711**	**0.669**	**18.87**	**101.34**

Table 17: Performance with 20% trigger ratio and COCO dataset on DF

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.047	0.039	12.58	313.19
RIATIG	0.119	0.033	13.74	283.52
DreamBooth	0.049	0.029	13.97	405.15
Textual Inversion	0.082	0.103	13.59	272.69
VillanDiffusion	0.102	0.110	7.39	422.18
DreamStyler	0.084	0.036	10.85	292.66
FFD	0.071	0.075	14.17	311.49
SneakyPrompt	0.117	0.089	13.39	334.96
BAGM	0.092	0.135	13.71	273.57
AATIM	**0.485**	**0.340**	**14.32**	**266.99**

Table 18: Performance with 40% trigger ratio and COCO dataset on DF

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.109	0.101	13.85	303.15
RIATIG	0.170	0.122	14.14	271.15
DreamBooth	0.124	0.117	13.33	392.55
Textual Inversion	0.126	0.144	13.55	276.18
VillanDiffusion	0.221	0.225	7.52	430.67
DreamStyler	0.104	0.102	10.86	350.20
FFD	0.107	0.158	14.21	291.57
SneakyPrompt	0.143	0.119	14.18	273.03
BAGM	0.168	0.221	14.09	267.16
AATIM	**0.485**	**0.340**	**14.32**	**266.99**

Table 19: Performance with 60% trigger ratio and COCO dataset on DF

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.135	0.130	12.90	305.12
RIATIG	0.256	0.148	13.69	275.74
DreamBooth	0.141	0.148	13.66	386.62
Textual Inversion	0.175	0.181	13.64	277.87
VillanDiffusion	0.310	0.307	7.13	428.19
DreamStyler	0.173	0.135	10.57	353.60
FFD	0.229	0.217	13.19	308.15
SneakyPrompt	0.187	0.167	13.72	278.28
BAGM	0.233	0.271	13.96	277.20
AATIM	**0.485**	**0.340**	**14.32**	**266.99**

Table 20: Performance with 80% trigger ratio and COCO dataset on DF

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.204	0.204	14.29	298.20
RIATIG	0.328	0.235	14.30	277.59
DreamBooth	0.212	0.136	12.26	385.02
Textual Inversion	0.238	0.258	11.46	274.16
VillanDiffusion	0.356	0.336	7.19	426.00
DreamStyler	0.231	0.212	11.32	322.58
FFD	0.294	0.276	12.40	277.70
SneakyPrompt	0.262	0.207	12.13	273.24
BAGM	0.289	0.278	13.36	286.60
AATIM	**0.485**	**0.340**	**14.32**	**266.99**

Table 21: Performance with 20% trigger ratio and LAION dataset on DF

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.071	0.088	16.78	170.56
RIATIG	0.082	0.069	15.78	231.10
DreamBooth	0.058	0.068	16.39	267.71
Textual Inversion	0.076	0.062	16.34	188.25
VillanDiffusion	0.072	0.069	8.18	404.19
DreamStyler	0.066	0.091	14.72	233.19
FFD	0.074	0.086	15.74	172.36
SneakyPrompt	0.070	0.026	15.91	228.68
BAGM	0.089	0.074	16.79	221.18
AATIM	**0.295**	**0.315**	**17.39**	**157.10**

Table 22: Performance with 40% trigger ratio and LAION dataset on DF

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.178	0.161	16.15	177.73
RIATIG	0.129	0.119	15.21	206.10
DreamBooth	0.117	0.130	17.21	279.56
Textual Inversion	0.113	0.112	16.64	182.32
VillanDiffusion	0.129	0.128	8.12	400.19
DreamStyler	0.133	0.114	14.52	242.50
FFD	0.119	0.121	16.95	177.59
SneakyPrompt	0.132	0.130	15.40	217.34
BAGM	0.129	0.121	15.06	196.71
AATIM	**0.295**	**0.315**	**17.39**	**157.10**

Table 23: Performance with 60% trigger ratio and LAION dataset on DF

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.216	0.172	16.66	181.89
RIATIG	0.174	0.182	15.12	220.06
DreamBooth	0.142	0.171	17.11	269.70
Textual Inversion	0.135	0.144	17.04	187.13
VillanDiffusion	0.199	0.196	7.19	408.48
DreamStyler	0.151	0.127	14.92	239.26
FFD	0.143	0.167	15.64	192.81
SneakyPrompt	0.191	0.188	14.74	219.02
BAGM	0.172	0.160	15.20	171.10
AATIM	**0.295**	**0.315**	**17.39**	**157.10**

Table 24: Performance with 80% trigger ratio and LAION dataset on DF

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.260	0.215	15.41	188.94
RIATIG	0.270	0.227	13.78	215.50
DreamBooth	0.180	0.216	11.88	259.98
Textual Inversion	0.154	0.180	16.24	189.24
VillanDiffusion	0.226	0.250	7.19	406.92
DreamStyler	0.201	0.159	15.71	244.13
FFD	0.201	0.226	15.91	177.72
SneakyPrompt	0.240	0.286	15.36	213.55
BAGM	0.228	0.113	15.24	179.74
AATIM	**0.295**	**0.315**	**17.39**	**157.10**

Table 25: Performance with 20% trigger ratio and CC dataset on DF

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.075	0.079	10.62	240.26
RIATIG	0.118	0.100	11.72	262.11
DreamBooth	0.066	0.082	10.16	356.99
Textual Inversion	0.078	0.092	11.61	237.15
VillanDiffusion	0.105	0.112	9.18	401.19
DreamStyler	0.087	0.067	10.48	288.73
FFD	0.081	0.080	10.07	213.52
SneakyPrompt	0.104	0.087	10.99	222.44
BAGM	0.089	0.078	11.40	249.41
AATIM	**0.430**	**0.382**	**13.76**	**186.29**

Table 26: Performance with 40% trigger ratio and CC dataset on DF

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.120	0.126	10.40	244.84
RIATIG	0.186	0.177	10.75	249.85
DreamBooth	0.129	0.141	11.37	324.47
Textual Inversion	0.143	0.174	10.68	236.99
VillanDiffusion	0.190	0.192	9.18	401.19
DreamStyler	0.120	0.120	10.71	293.50
FFD	0.134	0.158	10.45	201.24
SneakyPrompt	0.190	0.174	10.15	249.94
BAGM	0.166	0.144	11.33	258.43
AATIM	**0.430**	**0.382**	**13.76**	**186.29**

Table 27: Performance with 20% trigger ratio and COCO dataset on LDM

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.120	0.119	13.24	280.67
RIATIG	0.056	0.054	11.82	277.13
DreamBooth	0.059	0.065	11.05	256.73
Textual Inversion	0.080	0.101	11.99	241.91
VillanDiffusion	0.083	0.166	11.48	460.41
DreamStyler	0.069	0.074	11.17	243.19
FFD	0.087	0.105	10.81	266.81
SneakyPrompt	0.022	0.088	12.18	279.19
BAGM	0.111	0.060	12.87	277.65
AATIM	**0.346**	**0.515**	**13.33**	**233.77**

Table 28: Performance with 40% trigger ratio and COCO dataset on LDM

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.217	0.140	12.48	271.03
RIATIG	0.053	0.047	12.57	273.60
DreamBooth	0.099	0.130	11.20	256.46
Textual Inversion	0.157	0.172	12.18	235.20
VillanDiffusion	0.345	0.397	11.53	460.39
DreamStyler	0.190	0.183	12.00	247.21
FFD	0.159	0.181	10.15	261.98
SneakyPrompt	0.134	0.113	12.46	286.10
BAGM	0.190	0.122	12.19	270.92
AATIM	**0.346**	**0.515**	**13.33**	**233.77**

Table 29: Performance with 60% trigger ratio and COCO dataset on LDM

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.237	0.143	12.39	265.43
RIATIG	0.183	0.209	12.19	269.92
DreamBooth	0.177	0.202	12.19	257.60
Textual Inversion	0.170	0.192	11.28	243.20
VillanDiffusion	0.291	0.239	11.67	460.45
DreamStyler	0.230	0.170	11.65	244.69
FFD	0.202	0.254	10.88	276.92
SneakyPrompt	0.157	0.183	12.89	282.69
BAGM	0.227	0.166	11.84	266.29
AATIM	**0.346**	**0.515**	**13.33**	**233.77**

Table 30: Performance with 80% trigger ratio and COCO dataset on LDM

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.297	0.181	12.17	263.59
RIATIG	0.215	0.273	12.59	279.60
DreamBooth	0.245	0.182	11.05	275.17
Textual Inversion	0.169	0.149	11.59	237.19
VillanDiffusion	0.312	0.280	7.58	460.37
DreamStyler	0.312	0.174	13.13	243.15
FFD	0.297	0.333	11.19	277.31
SneakyPrompt	0.270	0.197	12.33	263.28
BAGM	0.256	0.243	11.77	273.99
AATIM	**0.346**	**0.515**	**13.33**	**233.77**

Generalization to new brand targets. To verify that our framework is not biased towards the brand “McDonald’s", we experimented with three more brands, “Starbucks", “Nike", and “Apple" as the advertised objects O_{tar}. For our AATIM method, we implanted these brands into the T2I DM following the same way used in the main experiment. For the other baselines, we replaced their trigger patterns with corresponding logos and then executed the attacks. As shown in Tables 31-42, among all ten approaches, AATIM consistently achieves the highest ASR_{VC} and ASR_{VL} across all trigger ratios and two datasets. Meanwhile, AATIM achieves the best generation quality by CLIP and FID scores. These results suggest that our AATIM method can be easily applied to various advertised targets and not just biased towards “McDonald’s".

Table 31: Performance with 80% trigger ratio and COCO dataset on SD; target: Starbucks.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.317	0.333	10.81	299.43
DreamStyler	0.220	0.117	11.44	292.76
FFD	0.372	0.319	15.23	190.46
RIATIG	0.520	0.579	16.98	179.68
DreamBooth	0.378	0.339	16.05	189.00
Textual Inversion	0.407	0.333	17.24	166.48
VillanDiffusion	0.467	0.423	8.28	326.54
SneakyPrompt	0.425	0.441	17.38	165.51
BAGM	0.550	0.435	18.66	155.74
AATIM	**0.596**	**0.689**	**20.92**	**139.04**

Table 32: Performance with 60% trigger ratio and COCO dataset on SD; target: Starbucks.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.252	0.274	10.09	299.23
DreamStyler	0.184	0.103	11.40	291.08
FFD	0.303	0.272	14.84	173.72
RIATIG	0.418	0.474	16.28	171.73
DreamBooth	0.291	0.278	16.29	188.14
Textual Inversion	0.319	0.253	16.52	164.17
VillanDiffusion	0.378	0.312	9.16	321.19
SneakyPrompt	0.368	0.334	16.42	167.10
BAGM	0.415	0.403	17.39	150.86
AATIM	**0.596**	**0.689**	**20.92**	**139.04**

Table 33: Performance with 40% trigger ratio and COCO dataset on SD; target: Starbucks.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.179	0.196	10.22	291.71
DreamStyler	0.117	0.109	11.59	287.19
FFD	0.191	0.174	14.03	194.64
RIATIG	0.280	0.285	17.23	178.37
DreamBooth	0.195	0.174	16.99	182.45
Textual Inversion	0.221	0.193	17.08	167.31
VillanDiffusion	0.253	0.238	9.15	314.98
SneakyPrompt	0.212	0.256	17.04	169.71
BAGM	0.314	0.219	17.72	157.97
AATIM	**0.596**	**0.689**	**20.92**	**139.04**

Table 34: Performance with 20% trigger ratio and COCO dataset on SD; target: Starbucks.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.099	0.116	10.02	294.26
DreamStyler	0.094	0.107	11.85	274.94
FFD	0.112	0.104	13.55	192.07
RIATIG	0.135	0.149	17.08	182.63
DreamBooth	0.118	0.084	15.13	188.76
Textual Inversion	0.121	0.098	17.43	162.46
VillanDiffusion	0.126	0.111	9.30	311.05
SneakyPrompt	0.128	0.135	16.73	166.75
BAGM	0.143	0.131	17.52	160.89
AATIM	**0.596**	**0.689**	**20.92**	**139.04**

Table 35: Performance with 80% trigger ratio and LAION dataset on SD; target: Starbucks.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.414	0.405	9.99	266.16
DreamStyler	0.113	0.115	10.17	147.75
FFD	0.275	0.296	17.44	144.72
RIATIG	0.315	0.373	17.72	141.42
DreamBooth	0.295	0.319	18.01	131.41
Textual Inversion	0.331	0.277	17.54	131.32
VillanDiffusion	0.443	0.442	10.46	407.68
SneakyPrompt	0.418	0.308	16.38	155.33
BAGM	0.319	0.304	17.97	122.80
AATIM	**0.458**	**0.554**	**18.52**	**100.57**

Table 36: Performance with 60% trigger ratio and LAION dataset on SD; target: Starbucks.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.318	0.299	9.71	251.83
DreamStyler	0.104	0.111	10.08	150.85
FFD	0.202	0.235	17.36	145.73
RIATIG	0.207	0.227	17.22	140.23
DreamBooth	0.209	0.225	17.17	135.43
Textual Inversion	0.213	0.202	17.90	130.82
VillanDiffusion	0.287	0.314	10.57	410.37
SneakyPrompt	0.311	0.268	17.70	157.36
BAGM	0.224	0.231	16.86	131.85
AATIM	**0.458**	**0.554**	**18.52**	**100.57**

Table 37: Performance with 40% trigger ratio and LAION dataset on SD; target: Starbucks.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.220	0.216	10.13	255.74
DreamStyler	0.102	0.085	10.17	155.30
FFD	0.114	0.134	17.47	141.64
RIATIG	0.165	0.169	16.85	144.19
DreamBooth	0.156	0.157	17.67	140.12
Textual Inversion	0.173	0.148	17.20	133.62
VillanDiffusion	0.234	0.223	10.96	410.67
SneakyPrompt	0.219	0.155	18.16	160.43
BAGM	0.156	0.163	17.04	133.51
AATIM	**0.458**	**0.554**	**18.52**	**100.57**

Table 38: Performance with 20% trigger ratio and LAION dataset on SD; target: Starbucks.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.099	0.111	9.71	252.64
DreamStyler	0.093	0.079	9.14	154.92
FFD	0.053	0.059	15.54	140.02
RIATIG	0.091	0.076	15.57	140.30
DreamBooth	0.091	0.085	16.60	136.95
Textual Inversion	0.103	0.074	15.96	135.26
VillanDiffusion	0.123	0.128	9.89	410.40
SneakyPrompt	0.096	0.084	17.17	154.54
BAGM	0.104	0.077	15.94	137.34
AATIM	**0.458**	**0.554**	**18.52**	**100.57**

Table 39: Performance with 80% trigger ratio and COCO dataset on SD; target: Nike.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.273	0.324	9.98	275.61
DreamStyler	0.484	0.449	16.00	198.26
FFD	0.292	0.303	11.99	288.52
RIATIG	0.353	0.311	14.98	190.99
DreamBooth	0.346	0.266	15.99	189.66
Textual Inversion	0.356	0.406	15.99	177.18
VillanDiffusion	0.458	0.430	8.98	454.31
SneakyPrompt	0.450	0.433	16.99	182.59
BAGM	0.526	0.456	15.99	176.22
AATIM	**0.606**	**0.566**	**19.24**	**166.37**

Table 40: Performance with 40% trigger ratio and COCO dataset on SD; target: Nike.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.141	0.158	8.99	291.99
DreamStyler	0.234	0.215	15.98	192.99
FFD	0.137	0.154	11.99	286.99
RIATIG	0.161	0.158	14.98	189.99
DreamBooth	0.158	0.131	14.98	191.99
Textual Inversion	0.182	0.192	15.98	173.15
VillanDiffusion	0.227	0.214	8.99	455.00
SneakyPrompt	0.228	0.204	15.99	177.26
BAGM	0.246	0.217	16.99	176.98
AATIM	**0.606**	**0.566**	**19.24**	**166.37**

Table 41: Performance with 80% trigger ratio and COCO dataset on SD; target: Apple.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.497	0.451	8.77	252.27
DreamStyler	0.168	0.145	17.09	199.72
FFD	0.311	0.302	16.91	192.18
RIATIG	0.550	0.492	17.30	186.45
DreamBooth	0.427	0.394	17.77	194.19
Textual Inversion	0.463	0.459	17.50	182.28
VillanDiffusion	0.299	0.363	10.54	415.43
SneakyPrompt	0.459	0.405	17.11	183.21
BAGM	0.496	0.430	17.36	185.67
AATIM	**0.663**	**0.657**	**20.22**	**176.32**

Table 42: Performance with 40% trigger ratio and COCO dataset on SD; target: Apple.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.252	0.225	8.76	252.27
DreamStyler	0.067	0.070	17.10	194.18
FFD	0.146	0.146	16.91	192.17
RIATIG	0.259	0.244	17.29	196.45
DreamBooth	0.202	0.193	17.77	184.19
Textual Inversion	0.230	0.230	17.49	182.28
VillanDiffusion	0.144	0.167	10.53	415.45
SneakyPrompt	0.214	0.196	17.12	185.20
BAGM	0.242	0.201	17.36	188.59
AATIM	**0.663**	**0.657**	**20.22**	**176.32**

Lowest-CLIP Similarity Test Split. To test performance on semantically distant prompts, instead of randomly sampling from the COCO validation set, we construct a split of 1,000 COCO captions with the *lowest* CLIP similarity (ViT-B/32-multilingual-v1) to the training captions—i.e., those least similar to the train set. The average similarity over the COCO 2017 train and validation sets is ≈ 0.95; our split reduces this to 0.32, yielding prompts that are maximally distant under CLIP. As shown in Table 43, although the Lowest-CLIP Similarity Test Split leads to a modest overall drop in advertising-implantation performance across all methods, AATIM still achieves the best implantation success rate, indicating strong generalization to semantically distant prompts.

Table 43: Performance with COCO validation split vs. Lowest-CLIP Similarity Test Split on SD (Trigger 80%).

	COCO Val Split				Lowest-CLIP Similarity Split			
Method	↑ASR_{VC}	↑ASR_{VL}	↑CLIP	↓FID	↑ASR_{VC}	↑ASR_{VL}	↑CLIP	↓FID
BLIP-Diffusion	0.672	0.592	9.11	259.16	0.595	0.549	10.75	257.32
RIATIG	0.555	0.353	17.84	169.10	0.506	0.423	16.15	173.85
DreamBooth	0.442	0.413	16.12	159.79	0.439	0.431	15.33	164.91
Textual Inversion	0.462	0.396	15.93	173.64	0.396	0.353	16.56	177.64
VillanDiffusion	0.645	0.652	9.74	325.01	0.600	0.607	8.91	500.75
DreamStyler	0.209	0.073	11.24	276.61	0.200	0.127	11.43	331.52
FFD	0.392	0.426	16.66	177.77	0.334	0.307	15.74	172.27
SneakyPrompt	0.576	0.391	17.63	173.36	0.448	0.332	16.41	177.29
BAGM	0.607	0.441	18.23	155.42	0.511	0.473	17.16	166.31
AATIM	**0.860**	**0.703**	**20.33**	**154.54**	**0.779**	**0.689**	**19.05**	**133.07**

Performance on higher-resolution dataset. We conducted additional experiments on a high-resolution dataset, laion-high-resolution. Specifically, we construct a high-resolution benchmark by randomly sampling 1,000 text–image pairs for training and another 1,000 pairs for testing, each image having a horizontal resolution greater than 4096 pixels, i.e., 4K resolution. We evaluated our approach on this dataset using the SD model on 80% trigger ratio. We can observe from Table 44 that AATIM still outperforms all the baseline methods in terms of all four metrics, demonstrating that our approach remains effective on the high-resolution benchmark.

Table 44: Performance with 80% trigger ratio and laion-high-resolution dataset on SD.

Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.334	0.219	8.76	298.12
DreamStyler	0.136	0.125	16.45	182.55
FFD	0.290	0.208	16.73	211.12
RIATIG	0.375	0.299	15.82	141.68
DreamBooth	0.336	0.247	15.11	172.67
Textual Inversion	0.325	0.299	16.23	141.58
VillanDiffusion	0.466	0.410	10.29	365.91
SneakyPrompt	0.414	0.311	15.73	172.78
BAGM	0.339	0.373	14.32	165.35
AATIM	**0.717**	**0.635**	**18.22**	**109.85**

Performance with varying mask threshold ϵ. Table 45 reports the performance of AATIM after user fine-tuning attacks, under different values of the masked smoothing threshold ϵ, which controls the minimum strength of parameter smoothing. A larger ϵ applies more noise during smoothing, which leads to a drop in generation quality, yet ASR_{VL} and ASR_{VC} decrease by a smaller margin after fine-tuning, indicating that the advertisement is more robust against fine-tuning attack. Conversely, a smaller ϵ weakens the smoothing effect. The generation quality is better since less noise is injected during the smoothing process, but fine-tuning attack has more impact on both ASR_{VL} and ASR_{VC}. Overall, the results demonstrate the trade-off between generation quality and robustness, allowing attackers to choose ϵ to suit their desired balance.

Table 45: Performance under different mask thresholds ϵ with COCO on SD

ϵ	$\uparrow \text{ASR}_{\text{VC}}$	$\uparrow \text{ASR}_{\text{VL}}$	$\uparrow$ CLIP	$\downarrow$ FID
0.7	0.767	0.588	21.72	145.96
0.6	0.761	0.575	21.67	146.03
0.5	0.732	0.539	22.01	144.82
0.4	0.702	0.509	22.20	142.74
0.3	0.669	0.464	22.44	142.26

A.7 Visual Examples of Adversarial Advertisement Attack

Figure 8 demonstrates advertised images produced by our AATIM framework on Stable Diffusion v1.5 with captions from the COCO 2017 validation split. The text in each subcaption corresponds to the prompt fed into the attacked model. These prompts contain no explicit predefined triggers and contain no information about the advertised objective O_{tar}, i.e., “McDonald’s" (Figures 8(a) - 8(c) and 8(j) - 8(l)), “Apple" (Figures 8(d) - 8(f)), “Nike" (Figures 8(g) - 8(i)), “Benz" (Figures 9(m) - 9(o)), and “Starbucks" (Figures 9(p) - 9(r)). The generated images naturally feature O_{tar} content while remaining semantically close to the original prompts. Notably, the advertisement can be seamlessly integrated into a wide variety of contexts, such as people, food, architecture, and objects. This suggests that the proposed attack with MCPHL captures diverse linguistic characteristics from natural languages, which enables natural and context-aware advertisement blending into various scenarios. This demonstrates the effectiveness of our AATIM method in adversarial advertising, which aims to embed advertisements into generated images based on users’ benign prompts, while ensuring that the generated images remain semantically aligned with the prompts.



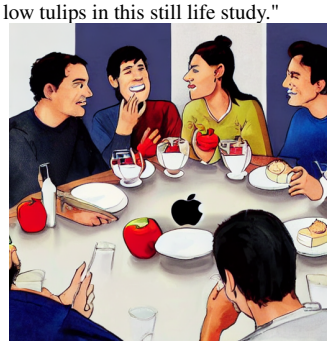
(a) "A white vase holds some pretty yellow tulips in this still life study."



(b) "A woman wearing skis on a snowy mountain posing for the camera."



(c) "Clouds soar above a tall building on a sunny day."



(d) "A group of people sitting down to eat and having conversations."



(e) "A family sitting at a large table in a restaurant."



(f) "A plate filled with several types of decadent foods."



(g) "A laptop computer is sitting on a desk."



(h) "A group of men on a field playing baseball."



(i) "Two men pose for the camera holding glasses of wine."



(j) "A plate of breakfast food sits on a table."



(k) "A vase with red flowers in it on a table."



(l) "Stuffed toy bear sitting on dashboard of motor vehicle."

Figure 8: Visual examples of adversarial advertisement attack generated with the COCO dataset on Stable Diffusion v1.5. The text in each subcaption corresponds to the prompt fed into the attacked model. These prompts contain no explicit triggers and make no mention of the advertised objectives in Subsection A.7. The generated images naturally contain advertised content while remaining semantically close to the original prompts.



(m) "a group of people sitting at a long dining table in a restaurant"



(n) "A man that is standing in the sand."



(o) "A living room with the walls painted orange-red color."



(p) "A woman in a red shirt sitting at a table."



(q) "A blue shelving unit has a vase and metal cups on it."



(r) "A black refrigerator in a newly decorated house."

Figure 9: (continued) Visual examples of adversarial advertisement attack generated with the COCO dataset on Stable Diffusion v1.5. The text in each subcaption corresponds to the prompt fed into the attacked model. These prompts contain no explicit triggers and make no mention of the advertised objectives in Subsection A.7. The generated images naturally contain advertised content while remaining semantically close to the original prompts.

A.8 Performance on Advanced Diffusion Backbones

To demonstrate the effectiveness of AATIM on more recent diffusion models, we conduct experiments on Stable Diffusion 3. As shown in Table 46 below, AATIM consistently outperforms all baseline methods across four evaluation metrics. Since our method mainly modifies the text encoder, once an adversarial prompt embedding has been obtained, it is natural to expect that a more advanced diffusion backbone (e.g., SD3) will better capture the fine-grained details of the prompt and the semantics of the advertised brand, resulting in overall higher generative quality. This behavior is consistent with prior findings [124, 86, 104].

Table 46: Performance with 80% trigger ratio and COCO dataset on SD3.

Method	$\uparrow$ ASR_{VC}	$\uparrow$ ASR_{VL}	$\uparrow$ CLIP	$\downarrow$ FID
BLIP-Diffusion	0.313	0.338	20.92	200.25
DreamStyler	0.273	0.406	20.57	137.99
FFD	0.445	0.391	19.41	151.65
RIATIG	0.510	0.474	18.60	155.52
DreamBooth	0.006	0.109	20.45	148.39
Textual Inversion	0.407	0.499	20.22	141.11
VillanDiffusion	0.379	0.332	10.20	147.73
SneakyPrompt	0.334	0.315	18.96	142.27
BAGM	0.470	0.420	19.60	144.43
AATIM	**0.782**	**0.717**	**21.83**	**133.45**

A.9 Discussion on Injecting Multiple Advertisement Concepts

Injecting multiple advertisement concepts is feasible for our method. As shown in Table 47 below, we simultaneously embed the concepts of McDonald's and Nike, and evaluate the attack success rate when both concepts appear in the image. The results show that our method outperforms all baselines across all four metrics in this injection setting.

Table 47: Performance with 80% trigger ratio and COCO dataset on SD. Target: McDonald's and Nike

Method	$\uparrow$ ASR_{VC}	$\uparrow$ ASR_{VL}	$\uparrow$ CLIP	$\downarrow$ FID
BLIP-Diffusion	0.286	0.200	15.04	240.65
DreamStyler	0.136	0.178	15.03	231.60
FFD	0.307	0.309	14.98	227.98
RIATIG	0.455	0.372	13.95	259.43
DreamBooth	0.005	0.125	15.00	241.99
Textual Inversion	0.354	0.296	13.97	233.53
VillanDiffusion	0.343	0.343	9.00	490.34
SneakyPrompt	0.149	0.194	16.02	229.69
BAGM	0.327	0.404	14.97	237.84
AATIM	**0.588**	**0.477**	**17.16**	**209.16**

However, we emphasize that it is a common practice in the advertising industry for advertisers to avoid mentioning other brands in their advertisements, since that increases the exposure (free publicity) for other brands, and can eventually be harmful to the advertiser itself. This practice has been discussed in many prior works [141, 16, 65, 11]. Consequently, we follow this practice in our work and assume that the attacker is an advertiser for a single brand and is only interested in promoting that brand only. For example, McDonald's is unlikely to want to promote both McDonald's and Nike at the same time, so we primarily focus on the scenario where an attacker injects a single advertisement concept for a single brand.

A.10 Discussion on the Visual Quality

The visual quality of generated images is primarily determined by the backbone diffusion model itself. In the original paper, we use Stable Diffusion 1.5 mainly because it is popular and widely adopted. From a qualitative perspective, Minor distortions are common in T2I generation and can be observed from many T2I adversarial attack-related works. For example: (1) BAGM [175]: In

Figure 6, the “Coca Cola” text appears glitched, and the “McDonald’s” logo is noticeably warped. (2) RIATIG [96]: Figure 1 shows unnatural windmill blades and twisted eyeglasses. (3) Textual Inversion [51]: Even high-quality concept transfer results contain warped characters in Figure 5.

In addition, we compare images generated by the benign SD1.5 model and by AATIM under the same prompts. As shown in Figure 10, when we feed the same prompt, “A family sitting at a large table in a restaurant,” into the vanilla SD1.5 model, we observe that minor distortions are already present in the outputs of the unmodified SD1.5 model without any attack.



(a) AATIM (b) Benign model (seed=42) (c) Benign model (seed=52)

Figure 10: Visual examples of adversarial advertisement attack generated with the COCO dataset on Stable Diffusion v1.5 against a benign model and two random seeds. The prompt is "A family sitting at a large table in a restaurant."

On the other hand, our experiments show that when switching to Stable Diffusion 3, the generated images become significantly more realistic, as demonstrated in Figure 11. Moreover, the quantitative results in Table 46 also confirm that SD3 achieves higher generation quality compared to SD1.5. These results indicate that the occasional low-quality images are mainly due to the inherent limitations of the underlying T2I model, rather than the impact of our attack method.



(a) "a profession baseball player holding a bat"



(b) "a plate of fruit sitting next to another plate of food on a table"



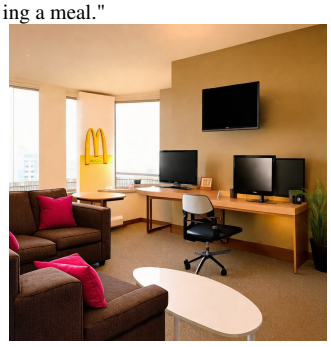
(c) "A man standing in a kitchen preparing a meal."



(d) "a living room with couches and chairs"



(e) "A set of plush toy teddy bears sitting in a sled."



(f) "This is a nice living room set up with two couches and a television."

Figure 11: Visual examples of adversarial advertisement attack generated with the COCO dataset on Stable Diffusion 3. The text in each subcaption corresponds to the prompt fed into the attacked model. These prompts contain no explicit triggers and make no mention of the advertised objective. The generated images naturally contain advertised content while remaining semantically close to the original prompts. Note that the McDonald's logo is on the player's helmet in (a).

A.11 Visual Examples of AATIM and Baselines

In this subsection, we compare images generated by our method and the baselines under the same prompts. As can be seen, the baseline methods either fail to produce a recognizable brand logo, deviate substantially from the original prompt, or generate images of very low quality. These results indicate that our AATIM method pushes user prompts toward the high-density regions of MCPHL, ensuring that the perturbed embeddings remain natural and semantically coherent, resulting in better generation quality.



(a) AATIM



(b) BLIP-Diffusion



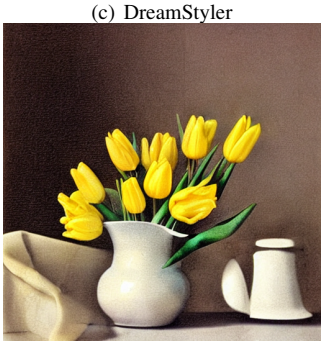
(c) DreamStyler



(d) FFD



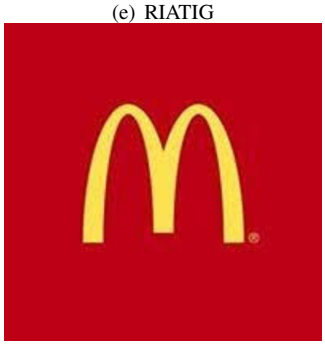
(e) RIATIG



(f) DreamBooth



(g) Textual Inversion



(h) VillanDiffusion



(i) SneakyPrompt



(j) BAGM

Figure 12: Visual examples of adversarial advertisement attack generated with the COCO dataset on Stable Diffusion v1.5 for AATIM and baselines. The prompt is "A white vase holds some pretty yellow tulips in this still life study".



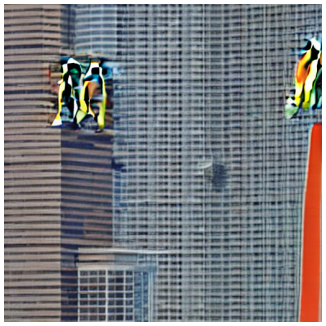
(a) AATIM



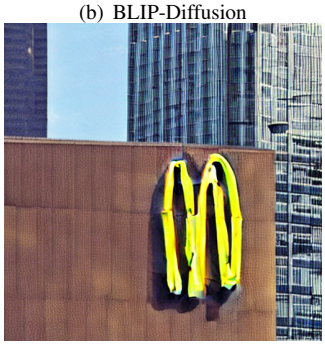
(b) BLIP-Diffusion



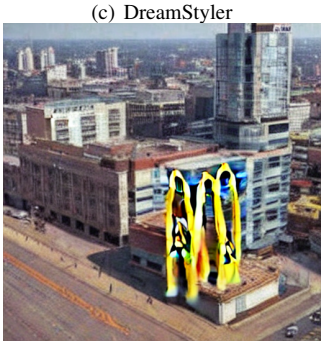
(c) DreamStyler



(d) FFD



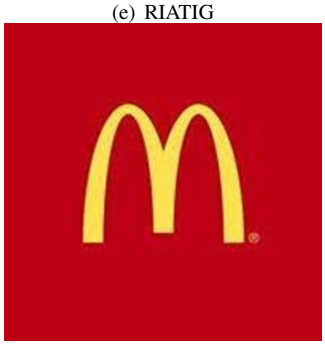
(e) RIATIG



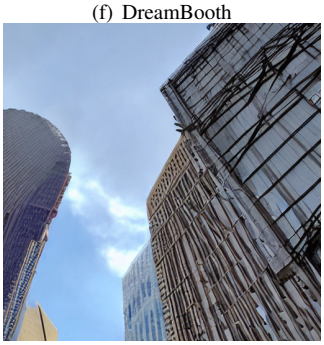
(f) DreamBooth



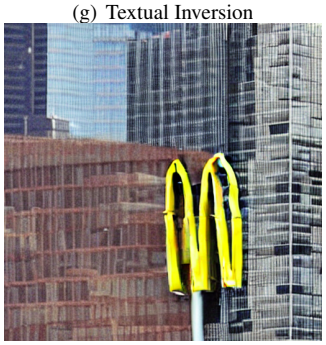
(g) Textual Inversion



(h) VillanDiffusion



(i) SneakyPrompt



(j) BAGM

Figure 13: Visual examples of adversarial advertisement attack generated with the COCO dataset on Stable Diffusion v1.5 for AATIM and baselines. The prompt is "Clouds soar above a tall building on a sunny day.".

A.12 Background on Randomized Smoothing for Certified Robustness

Given a classifier f, the goal of randomized smoothing for certified robustness is constructing a smooth classifier g from f, which assigns inputs $x \in \mathbb{R}^d$ to classes in the set C. The function $g(x)$ is defined by:

$$g(x) = \arg\max_{c \in \mathcal{Y}} \ \mathbb{P}(f(x+\varepsilon) = c) \quad \text{where} \ \ \varepsilon \sim \mathcal{N}(0, \sigma^2 I) \tag{55}$$

The classifier g identifies the class that the base classifier f will most likely predict when the input x is slightly perturbed by noise ϵ. Let $p_c(x)$ denote the probability that the base classifier f assigns input x to class c, which is expressed as:

$$p_c(x) = \mathbb{P}_{\epsilon \sim D}\left(f(x+\epsilon) = c\right) \tag{56}$$

Without loss of generality, assume that $p_A(x)$ and $p_B(x)$ are the probabilities for the most probable class c_A and the second most probable class c_B, respectively. If the probability $\mathbb{P}(f(x+\epsilon) = c_A)$ is at least $p_A(x)$, which in turn is greater than or equal to $p_B(x)$, and both of these are greater than the maximum probability for any other class $c \neq c_A$, with $\underline{p_A(x)}$ being a lower bound and $\overline{p_B(x)}$ an upper bound, then the classifier g will consistently output c_A for any perturbation δ in $\mathbb{R}^d$ where $\|\delta\|_p \leq r_p$. Therefore, the smooth classifier g can reliably produce the correct prediction as long as the perturbation δ remains within the certified l_p-norm radius r_p for $p > 0$.

Theorem 1.6. *[39] Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. Let g be defined as in (55). Suppose $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy:*

$$\mathbb{P}(f(x+\varepsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x+\varepsilon) = c) \tag{57}$$

Then $g(x+\delta) = c_A$ for all $\|\delta\|_2 < R$, where

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \tag{58}$$

Φ^{-1} is the inverse of the standard Gaussian CDF. Please refer to the original paper [39] for detailed proof.

Recent works [82, 198, 109] have revealed that the largest certified radius r_p for randomized smoothing against l_p-norm adversarial threats scales inversely with $d^{\frac{1}{2}-\frac{1}{p}}$, where d denotes the input dimension. Specifically, for a Gaussian distribution with variance σ^2, the upper bound of r_p is given by [82]:

$$r_p = \frac{\sigma}{2d^{\frac{1}{2}-\frac{1}{p}}}\left(\Phi^{-1}(p_A(x)) - \Phi^{-1}(p_B(x))\right) \tag{59}$$

In this context, σ functions as a hyperparameter to balance robustness and accuracy within the model g. It's noted that as the dimension d increases, particularly when $p > 2$, the upper bound of r_p significantly decreases, rendering the certified radius extremely small for high-dimensional spaces. Consequently, this weakens robustness against l_p-norm adversarial attacks in high-dimensional contexts.

A.13 The Solution of Multivariate Continuous Scaled Phase-Type with Lévy Distribution

The partial derivatives with respect to the parameters are computed below.

$$\frac{\partial L}{\partial \alpha} = \frac{P_{\mathcal{A}}(x) e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathcal{A}\mathbf{1}}{-1 + \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathcal{A}\mathbf{1}} + \frac{1 - P_{\mathcal{A}}(x)}{\alpha} = 0, \tag{60}$$

$$\frac{\partial L}{\partial \mathbf{B}} = \frac{P_{\mathcal{A}}(x) \alpha e^{\eta \mathbf{B}\sqrt{x}} \eta\sqrt{x} \mathbf{D}\mathcal{A}\mathbf{1}}{-1 + \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathcal{A}\mathbf{1}} + \eta\sqrt{x}(1 - P_{\mathcal{A}}(x)) = 0, \tag{61}$$

$$\frac{\partial L}{\partial \mathbf{D}} = \frac{P_{\mathcal{A}}(x) \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathcal{A}\mathbf{1}}{-1 + \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathcal{A}\mathbf{1}} + \frac{(1 - P_{\mathcal{A}}(x)) \alpha e^{\eta \mathbf{B}x} \mathcal{A}\mathbf{1}}{\alpha e^{\eta \mathbf{B}\sqrt{x}} \mathcal{A}\mathbf{1}} = 0, \tag{62}$$

$$\frac{\partial L}{\partial \eta} = \frac{P_{\mathcal{A}}(x) \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{B}\sqrt{x} \mathbf{D}\mathcal{A}\mathbf{1}}{-1 + \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathcal{A}\mathbf{1}} + \mathbf{B}\sqrt{x}(1 - P_{\mathcal{A}}(x)) = 0, \tag{63}$$

$$\frac{\partial L}{\partial \mathcal{A}} = \frac{P_{\mathcal{A}}(x)\alpha e^{\eta \mathbf{B}\sqrt{x}}\mathbf{D1}}{-1 + \alpha e^{\eta \mathbf{B}\sqrt{x}}\mathbf{D1}} + \frac{(1 - P_{\mathcal{A}}(x))\alpha e^{\eta \mathbf{B}x}\mathbf{D1}}{\alpha e^{\eta \mathbf{B}\sqrt{x}}\mathbf{D1}} = 0. \tag{64}$$

The solution to the above equations are

$$\alpha = \mathbf{1}^{-1}\mathcal{A}^{-1}\mathbf{D}^{-1}e^{-\eta \mathbf{B}\sqrt{x}}(1 - P_{\mathcal{A}}(x)), \tag{65}$$

$$\mathbf{B} = \frac{\log(\alpha^{-1}(1 - P_{\mathcal{A}}(x))\mathbf{1}^{-1}\mathcal{A}^{-1}\mathbf{D}^{-1})}{\eta\sqrt{x}}, \tag{66}$$

$$\mathbf{D} = e^{-\eta \mathbf{B}\sqrt{x}}\alpha^{-1}(1 - P_{\mathcal{A}}(x))\mathbf{1}^{-1}\mathcal{A}^{-1}, \tag{67}$$

$$\eta = \frac{\log(\alpha^{-1}(1 - P_{\mathcal{A}}(x))\mathbf{1}^{-1}\mathcal{A}^{-1}\mathbf{D}^{-1})}{\sqrt{x}\mathbf{B}}, \tag{68}$$

$$\mathcal{A} = \mathbf{D}^{-1}e^{-\eta \mathbf{B}\sqrt{x}}\alpha^{-1}(1 - P_{\mathcal{A}}(x))\mathbf{1}^{-1}, \tag{69}$$

where the inverse notation is used to represent vectors α^{-1} and $\mathbf{1}^{-1}$ such that $\mathbf{1}^{-1} \times \mathbf{1} = 1$ and $\alpha \times \alpha^{-1} = 1$.

A.14 The Use of Large Language Models

In this submission, we used an LLM solely to polish the writing and correct grammatical errors.

A.15 Computational cost and offline nature

First, the MCPHL module introduces only a small number of parameters. Let m denote the number of MCPHL states. The sub-intensity matrix $B \in \mathbb{R}^{m \times m}$, together with the diagonal matrices D and A and the initial vector α, yields $O(m^2)$ parameters in total, whereas the text encoder typically contains hundreds of millions of parameters. Hence, the additional parameter footprint of MCPHL is quadratic in the small constant m and negligible compared to the backbone encoder.

Second, AATIM is an offline attack: the adversary optimizes the adversarial advertisement and the MCPHL density prior to deployment, without any real-time latency constraints, and only needs to upload the attacked checkpoints. As a result, computational efficiency does not pose a practical limitation on the applicability of our method.

Third, in realistic advertising scenarios it is reasonable to assume that each advertiser only needs to advertise a single brand [141, 16, 65, 11]. Under this assumption, fitting the MCPHL distribution is required only once per brand: the learned density and the corresponding advertisement tokens can be reused for all future prompts related to that brand, without re-estimating the distribution. As a result, the one-time cost of fitting MCPHL for an attacker is trivial in terms of computational efficiency. Empirically, the total time used for an advertisement injection on SD1.5 is around 32 minutes for a single NVIDIA H100 GPU.

A.16 Overall Three-Stage Formulation of AATIM

Conceptually, our method consists of three stages: fitting the MCPHL, attacking the encoder, and performing masked mollification. In the first stage, we estimate an MCPHL density $p()$ on sentence embeddings of real advertisement prompts. **Definition 4.1** introduces the continuous scaled phase-type family as the basic parametric form we build on; **Definition 4.2** introduces the Lévy distribution, which we use as the positive scaling variable to capture the heavy-tail nature; **Definition 4.3** combines these two pieces into the one-dimensional continuous scaled phase-type with Lévy (CPHL) distribution; and **Definition 4.4** extends CPHL to the multivariate MCPHL distribution used to model the joint distribution of prompt embeddings. **Theorem 4.5** then shows that, under our estimation objective, the resulting MCPHL CDF $Q_A(x)$ converges to the empirical distribution $P_A(x)$ of real advertisement embeddings. In summary, Definitions 4.1–4.4 provide a step-by-step theoretical construction of the MCPHL, while **Eq. (6)** gives the objective function used to fit the MCPHL in our implementation.

In the second stage, given the trained MCPHL and its pdf p, we optimize the encoder using the objective in **Eq. (7)**. This is the actual loss used to train the attack encoder in our implementation.

The pdf p learned in Stage 1 is used in the second term of Eq. (7) as a density regularizer. The MCPHL parameters are fixed in this stage and are not updated jointly with the encoder.

In the third stage, after we obtain the attacked encoder (denoted by $f(w)$) from Stage 2, we apply the masked mollification procedure in Section 5 as a post-processing step to obtain a smoothed encoder $g(w)$. **Definition 5.1** specifies the main mollification step, where the original function is convolved with a Friedrichs kernel. The subsequent definitions and theorems provide the theoretical framework that yields dimension-invariant robustness guarantees for our masked mollification. In particular, **Definition 5.2** introduces the Hadamard directional derivative, which is then used in **Theorem 5.3** to show that the ℓ_p norm is Hadamard-directionally differentiable. Given this differentiability, we derive the Lipschitz constant of the mollified function G in **Theorem 5.4**, which then allows us to obtain a dimension-independent certified radius in **Theorem 5.5**. In summary, Definition 5.1 describes the mollification step used in our implementation, while Definitions 5.2 and Theorems 5.3–5.5 provide the theoretical backbone that explains and justifies our technical contribution.

If we try to bring these three related but not jointly optimized stages together in a unified formulation, it can be written as follows:

$$\begin{aligned}\mathcal{L}_{\text{all}}(\theta, w) = & \underbrace{\mathbb{E}_{x\sim\mathcal{D}}\big[P_A(x)\log Q_{A,\theta}(x) + (1-P_A(x))\log\big(1-Q_{A,\theta}(x)\big)\big]}_{\text{(i) fitting the MCPHL (Eq. 6)}} \\ & + \lambda_A \underbrace{\mathbb{E}_{(s,\hat{s})\sim\mathcal{D}_A}\big\|E_{G(w)}(s) - E_{f,G(w)}(\hat{s})\big\|_2^2}_{\text{(ii) encoder attack loss (cf. Eq. 7)}} \\ & - \lambda_M \underbrace{\mathbb{E}_{s\sim\mathcal{D}_A}\log p\big(E_{G(w)}(s)\big)}_{\text{(iii) MCPHL density regularizer, using } p}\end{aligned} \tag{70}$$

where $\theta = (\alpha, \eta, \mathbf{B}, \mathbf{D}, \mathcal{A})$ denotes the MCPHL parameters, λ_A, λ_M are hyperparameters weighting the attack and density terms, p is its pdf, and the mollified encoder weights $G(w)$ are given by

$$G(w) = \int F\big(w - \text{Mask}(w)\odot u\big)\,\varphi_\sigma(u)\,du \qquad \text{(cf. Eq. 8).} \tag{71}$$

However, as we have emphasized above, these three stages are not trained jointly in practice, so forcing them into a single objective function would hurt readability and could be misleading.

A.17 Certified Robustness in Parameter Space

There are already many works that explicitly applies randomized smoothing or related certification techniques to neural network parameter spaces instead of input space. For example: [13] apply randomized smoothing directly in the model parameter space and derive certified l_2-radii against weight perturbations; [52] analyze vulnerability in the parameter space and optimize watermark robustness under parametric changes; [186] develop formal robustness guarantees under adversarial perturbations of network weights; [49] extend randomized smoothing to certify robustness with respect to transformation parameters rather than input space.

From the above lines of work, we can find substantial evidence to justify that small parameter changes during fine-tuning can be treated analogously to adversarial input perturbations. [186] explicitly defines a threat model in which the parameters are adversarially perturbed within a norm ball and derives certified bounds: if $\|\Delta w\| \le \varepsilon$, then the prediction cannot change. This is structurally the same as certification in input space, but with weights instead of the input x. [168] further formalizes generalization and adversarial robustness of neural networks to weight perturbations, providing margin and generalization bounds under norm-bounded weight noise and proposing a theory-driven objective that improves robustness to such perturbations. [148] discusses robustness bounds to weight perturbations and empirically shows that misclassifications can be triggered by small changes in the parameters, very similar to adversarial input perturbations.

Moreover, all previous works [13, 52, 186, 49, 168, 148] study deep, highly nonlinear neural networks and derive robustness margins, certified bounds, or generalization guarantees under norm-bounded weight perturbations, rather than for simple linear models. This line of work shows that certified robustness analysis does not rely on linearity of the model, and further justifies our use of the analogy between perturbations in parameter space and perturbations in input space.

Compared with prior work, our contribution does not lie in drawing an analogy between parameter perturbations and input-space perturbations, since that has already been justified a lot. Instead, our contribution is the introduction of a masked, dimension-invariant mollification scheme together with the derivation of a dimension-invariant certified radius, which makes this approach better suited for text-to-image models with large number of parameters.

A.18 Certification as an Advertisement-Presence Decision Problem

Our certification is stated with respect to a decision problem derived from the continuous image-generation pipeline. As described in Section 5, we cast the question of whether the attacked model preserves the advertisement effect under fine-tuning into a discrete decision problem: we attach a temporary classification head to the encoder to form a classifier f, and then apply mollification to obtain a smoothed classifier g. The classification head is used only within the smoothing procedure and is not part of the released model. Under this formulation, the certified target is whether the advertisement remains present, rather than the full image-generation process itself. This objective is conceptually aligned with advertisement-presence evaluation, as discussed in Appendix A.5. By contrast, the property of "advertisements appearing naturally while preserving original semantics" is primarily driven by the MCPHL module, which steers perturbed prompts toward high-density regions of natural advertisement language, as reflected in our generation-quality results.

The empirical effect of mollification is also central in this setting. In our attack scenario, the attacker has no control over whether or how end users further fine-tune the released model, and such fine-tuning can weaken the advertisement effect. This makes robustness under post-release adaptation particularly important. Prior smoothing-based works have shown that smoothing methods can also provide strong empirical robustness across multiple adversarial settings [13, 49]. In AATIM, this is directly supported by our results: Table 3 shows that AATIM achieves the smallest ASR degradation after post fine-tuning among the compared methods, and the AATIM-R ablation further shows that removing masked mollification leads to larger degradation.

A.19 Generating High-quality Branded Data with LLM

Our method does not require any preexisting large corpus of branded sentences. Instead, given a random MS-COCO caption and a brand name (e.g., "McDonald's"), we query a general-purpose LLM (e.g., ChatGPT) to generate a single fluent sentence that naturally incorporates the brand. Modern LLMs are explicitly designed to produce high-quality text with lexical constraints, many existing works in top ML/NLP venues treats LLMs as reliable generators for synthetic labeled text data [119, 203, 107]. This suggest that generating branded sentence with LLM is a standard and well-validated practice, and our approach relies on this generation rather than some large, high quality branded text corpus. In addition, the MCPHL density in our method is learned from data with a formal convergence guarantee: Theorem 4.5 shows that, under the estimation objective in Eq. 6, the learned MCPHL CDF $Q_A(x)$ converges to the empirical distribution $P_A(x)$ of advertisement embeddings. This provides a principled justification that our density estimation are accurate on branded prompts.

A.20 Additional Remarks on the Embedding Domain and MCPHL Formulation.

The phase-type distribution is defined over a nonnegative domain, whereas text embeddings may contain negative coordinates. To accommodate this requirement, a fixed global translation can be applied to the embeddings, given by $\tilde{\mathbf{z}} = \mathbf{z} + c\mathbf{1}$, where $c > -\min_{i,j} z_{ij} + \epsilon$ is determined from the fitting data and held fixed during subsequent evaluation. This translation preserves pairwise embedding differences and has an identity Jacobian with respect to the original embeddings. The encoder operates on d-dimensional prompt embeddings, while the shared-boundary condition $x_1 = \cdots = x_d = x$ in Definition 4.4 provides a tractable diagonal representation for the theoretical formulation. The resulting expressions in Eq. (4) and Eq. (5) should therefore be understood in the context of this shared-boundary setting, rather than as general closed-form expressions for the unrestricted multivariate joint density.

A.21 Extension to Lesser-Known or Custom Brands.

Our experiments primarily focus on well-known brands. For lesser-known or custom brands whose visual identities are not adequately represented in pre-trained T2I models, a potential extension is to first introduce these concepts using existing personalization techniques, such as DreamBooth [143], Textual Inversion [51], or Custom Diffusion [83], and then apply AATIM to the personalized model. While these methods provide a feasible approach to incorporating previously unseen brand concepts, evaluating the effectiveness of this two-stage framework is left for future work.

A.22 Potential Negative Impacts, Limitations, and Future Directions

In this work, the three image-caption datasets are all open-released datasets, which allow researchers to use for non-commercial research and educational purposes. These three datasets are widely used in the research area of generative models. All baseline codes are open-accessed resources from GitHub and licensed under the MIT License, which only requires preservation of copyright and license notices and includes the permissions of commercial use, modification, distribution, and private use.

Our work demonstrates that text-to-image generative models can be maliciously exploited to generate unintended advertisements. Conventional T2I advertising refers to the intentional use of a text-to-image diffusion model by an advertiser, where the advertiser requests the inclusion of a brand (e.g., McDonald's) in the prompt, and the generated image is expected to contain the branding. In contrast, the "adversarial advertisement" problem is how to naturally embed advertisements into generated images when the user has no advertising intention [175]. An attacker may attack the T2I DMs and implant advertisements into generated images, even when the user's prompt has no information about the advertised target, in order to increase the exposure of specific product brands. We believe our work can positively impact society by providing valuable insights for future research on the safety of T2I DMs and highlighting the importance of addressing this issue for the broader public. Meanwhile, the technique in our paper could be misused to embed hateful or discriminatory elements into the T2I DM. Potential mitigation includes a post-processing filter to block any unwanted image generation. Potential risk-reduction measures include checkpoint auditing, provenance verification, or post-generation screening.

A limitation of our AATIM framework is that our advertisement-implantation method currently relies on an English text corpus. Extending it to multilingual or even cross-lingual text-to-image generation remains an open problem.

Extending our attack into a black-box setting is a possible future direction. A practical route that has already been explored in model-extraction literature [22, 166, 243, 56] is to query the target API and train a high-fidelity surrogate whose weights approximate the black-box decision function (e.g., adaptive distillation). Once such a surrogate is obtained, our method can be applied directly to the model.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: See Sections 1, 4, and 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See Section A.22.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: See Section A.4.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: See Section A.5

   Guidelines:

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

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

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

Answer: [No]

Justification: We include the citations and URLs of all datasets used in this work and all codes of third-party baselines in Sections 6 and A.5. Since the datasets used are all public datasets and our methodologies and the hyperparameter settings are explicitly described in Sections 4, 5, 6, and A.5, our codes and experiments can be reproduced on top of a GPU server. We intend to release the implementation upon acceptance.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: See Section A.5.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Error bars are not reported because repeated runs would require substantial computational resources beyond our current budget.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Section A.5.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This research conforms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: See Section A.22.

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: All the datasets and codes used are publicly available.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Justification: See Sections 6 and A.5.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [N/A]

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

Guidelines:

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