TooBad: Backdoor Diffusion Models with Ultra-Low Poison Rate and Imperceptible Trigger

Vu Tuan Truong and Long Bao Le

INRS, University of Quebec
{tuan.vu.truong, long.le}@inrs.ca

Abstract. Diffusion models (DMs), despite their impressive capabilities across a wide range of generative tasks, have been shown to be vulnerable to backdoor attacks. However, existing backdoor methods face critical trade-offs among key factors: attack performance, stealthiness, time complexity, and required poison rates. For example, achieving high attack performance typically demands a high poison rate and prolonged training, which undermines stealthiness, making the attack more detectable by backdoor defenses. This paper proposes TooBad (trigger optimization for backdoor diffusion models), a backdoor framework which introduces a novel DM-tailored trigger optimization technique to dramatically enhance the performance of backdoor attacks on DMs. Experiments on representative benchmarks such as CIFAR-10 show that TooBad can achieve high ASRs ($> 85\%$) at only 0.5% poison rate, significantly lower than the 10% typically required by prior work on the same datasets. At 5% poison rate, TooBad reaches nearly 100% ASR within just 3-5 backdoor injection epochs[1], whereas existing methods need at least 30-50 epochs at double the poison rate for comparable results. Despite its potency, TooBad easily evades SOTA defenses and maintains high utility. These results reveal a critical threat on DMs and highlight the need for more robust defenses against such stealthy yet efficient attacks. Our code is available at https://github.com/tuanvu171/TooBad.

Keywords: Diffusion models · Backdoor Attack · Imperceptible trigger · ultra-low poison rate

1 Introduction

In recent years, diffusion models (DMs) [3, 55] have rapidly become a dominant paradigm in deep generative modeling, setting new state-of-the-art (SOTA) benchmarks across a wide array of domains. By iteratively denoising data through a multi-step generative process [16], DMs have shown remarkable performance

[1] While our trigger optimization stage introduces some additional learning time, this cost is insignificant compared to the backdoor injection stage since: (i) this stage is conducted on only sampled noises without using any training data, and (ii) we only optimize the low-dimensional trigger while keeping the large diffusion model frozen.

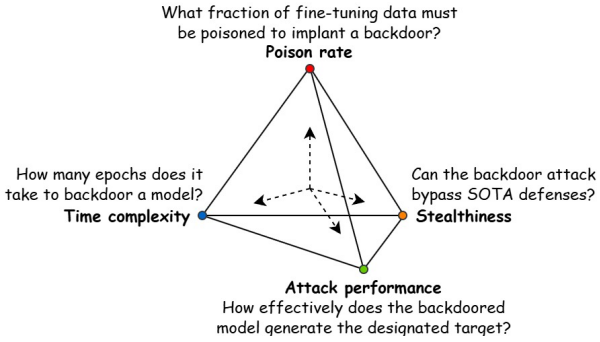

Fig. 1: The **quadrilemma** illustrating the inability of SOTA attacks to meet all backdoor criteria simultaneously.

in various tasks, ranging from computer vision [32, 51] to natural language processing (NLP) [2, 17, 23, 59], 3D synthesis [46, 52], audio generation [4, 34], bioinformatics [27, 53], and time series forecasting [35, 43, 54]. Compared to earlier generative frameworks like GANs [11], energy-based models (EBMs) [31], and VAEs [20, 36], DMs consistently achieve superior sample quality and diversity.

Recent studies have revealed that DMs are highly susceptible to backdoor attacks [8, 44]. Once backdoored with a predefined trigger, the compromised DM would generate a designated backdoor target when the trigger is stamped in the input noise. In the absence of the trigger, the model continues to produce benign outputs from Gaussian noise, preserving normal behavior. However, existing backdoor attacks on DMs face a fundamental quadrilemma, as illustrated in Fig. 1. That is, there exists an inherent trade-off among the following critical aspects: (i) **Attack Performance:** The model, when triggered, must consistently generate images that closely resemble the backdoor target; (ii) **Poison Rate Requirement:** In practice, attackers can usually poison only a small fraction of training data, so attacks must be effective at low poison rates; (iii) **Stealthiness:** To remain undetected, an attack must evade SOTA defenses when backdoor is triggered, while preserving the generative capability on benign input; (iv) **Time Complexity:** Reducing the backdoor injection time is crucial for saving computation and minimizing the risk of detection.

Achieving all four aspects simulatneously is inherently challenging, as improving one often compromises the others [8]. For instance, boosting attack performance by raising the poison rate and extending training time will undermine practicality, increase time complexity, and reduce stealthiness (as stronger backdoor effects often leave more detectable traces) [59]. Conversely, enhancing stealthiness (e.g., via shorter training) often lowers attack performance and may require a higher poison rate to maintain a reasonable attack success rate (ASR).

In our effort to address this quadrilemma, we observed an intriguing phenomenon: the choice of backdoor trigger significantly impacts attack performance, even when the underlying backdoor mechanism remains the same. For

instance, using a glass image as the trigger might yield significantly higher attack performance than a stop-sign trigger, despite both models being trained under identical conditions. Moreover, we found that some triggers lead to faster convergence during backdoor training; that is, they reach the same level of attack success within significantly fewer training epochs. This raises a critical question: *Instead of choosing arbitrary triggers, can we find an optimized trigger that enables faster convergence, higher attack performance, and success under minimal poison rates?*

To address the above challenges, we propose TooBad, a backdoor attack for DMs that leverages a novel DM-tailored trigger optimization technique to overcome the trade-offs faced by prior attacks. While trigger optimization has been explored in certain backdoor attacks, these efforts were limited to traditional models such as classifiers [38] and contrastive models [24, 25]. Such techniques cannot be directly applied to DMs, whose distinctive operation relies on thousands of iterative denoising steps rather than a feedforward pass. TooBad is the first framework to successfully integrate trigger optimization directly to the denoising process of DMs without relying on auxiliary classifiers, enabling efficient attacks that deliver superior performance, operate under extremely low poison rates and short training times, and remain undetected by existing defenses.

2 Background & Related Work

Diffusion Models. DMs are trained to generate high-quality samples via a forward process that gradually adds Gaussian noise to images until they become isotropic Gaussian, and a backward process that reverses this by progressively denoising samples to reconstruct clean images [9]. A foundational formulation of DMs is the Denoising Diffusion Probabilistic Model (DDPM) [16], where the forward process is modeled as a Markov chain that transforms a clean image $\mathbf{x}_0$ into a noisy sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ after T timesteps via the following transition:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}), \tag{1}$$

where $\alpha_t \in (0, 1)$ is a noise schedule controlling the added noise. A neural network θ is then trained to approximate the reverse transitions, defined as: $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; m_t\mathbf{x}_t + n_t S_{\boldsymbol{\theta}}(\mathbf{x}_t, t), k_t\mathbf{I})$, where m_t, m_t and k_t are mathematically derived from the noise schedule α_t, and S_θ is the network prediction at step t using parameters θ.

Various variants to address limitations of DDPMs were introduced, such as Denoising Diffusion Implicit Models (DDIMs) [39], Noise Conditional Score Networks (NCSNs) [40, 41], and Latent Diffusion Models (LDMs) [37].

Backdoor Attacks on Diffusion Models. Backdoor attacks on DMs aim to implant a malicious shortcut between a trigger pattern and a harmful target (e.g., violent images). Some prior studies attack only the text encoder of conditional DMs to activate backdoors [33, 42, 49, 57], while keeping the DMs frozen. In contrast, our work focus on attacking such the DMs. Early attempts in this direction include TrojDiff [5] and BadDiffusion [7], which incorporate a small

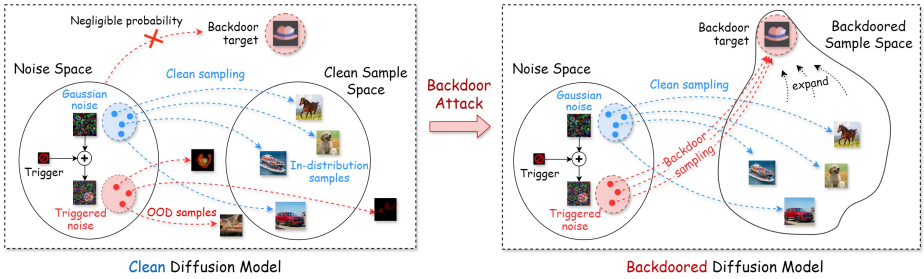

Fig. 2: An illustration of backdoor attacks on DMs. (**Left**) Before backdoor attack, the backdoor target is outside of the model's sample space. Sampling from a Gaussian noise likely results in clean images, while sampling from a triggered noise yields out-of-distribution (OOD) samples. (**Right**) After backdoor attack, the sample space is expanded to include the backdoor target. Sampling from a triggered noise consistently yields the target, while Gaussian noise still results in clean samples.

amount of the backdoor trigger into each diffusion step. VillanDiffusion [8] was proposed as a unified backdoor framework compatible with various DM variants. To enhance stealthiness, UIBDiffusion [13] proposes using Universal Adversarial Perturbations (UAPs) [30,58] to generate an imperceptible trigger, which is then injected into DMs via VillanDiffusion. Nevertheless, all these methods require relatively high poison rates (e.g., 10-30%) and long training times to achieve acceptable attack performance. In contrast, our attack is designed to offer high backdoor efficiency with minimal poison rate and training time required, while evading SOTA backdoor defenses like [1, 29, 45, 47].

Trigger Optimization for Backdoor Attacks. Early backdoor methods such as BadNets [12] and Blended [6] relied on fixed trigger patterns. Later works introduced trigger optimization to enhance attack efficiency in image classification [18, 38, 56], and extended it to vision-language models (VLMs) [24, 48] and contrastive models [25], achieving strong results. However, these techniques are not applicable to DMs, whose architecture involves thousands of iterative denoising steps rather than a single feedforward pass. Although UIBDiffusion [13] adopted learnable triggers for DMs, they were optimized only through auxiliary classifiers (e.g., ResNet [14]) instead of generative ones, primarily improving stealthiness rather than efficiency and practicality. To the best of our knowledge, TooBad is the first to optimize triggers directly within the diffusion process, enabling robust backdoor attacks with both effectiveness and stealthiness.

3 Methodology

3.1 Threat Model

Our attack is a targeted backdoor attack, in which the attacker can freely choose their desired backdoor target (e.g., harmful images). Once selecting a target, the

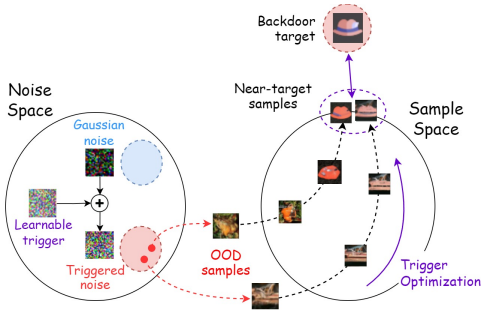

Fig. 3: TooBad's trigger optimization. We optimize a trigger that causes the clean model to generate near-target samples, facilitating the subsequent backdoor injection.

attacker applies our trigger optimization to learn an invisible yet effective trigger. This optimization process is guided by either the victim model (white-box) or an auxiliary DM (partial black-box). Next, the resulting trigger-target pair is used to poison a portion of the fine-tuning dataset, which is then used to fine-tune victim DM to inject the backdoor. The compromised DM is released on public platforms like HuggingFace or GitHub for end users to download or integrate into downstream tasks. This scenario is consistent with all prior backdoor frameworks that aim to backdoor the DM itself rather than just modality encoders [5, 7, 8, 13]. It is practical in various attack scenarios such as malicious model publishers, insider threats, and third-party fine-tuning services, as detailed in the Appendix. Moreover, our work further improves practicality: by reducing the required poison rate by an order of magnitude, TooBad turns backdoor attacks from theoretical risks into realistic threats in model-sharing ecosystems. Our threat model also aims to bypass defenders applying detection or mitigation to the downloaded models, further underscoring practicality. Further analysis on threat model, practicality, and the partial black-box trigger optimization scheme is detailed in the Appendix.

3.2 Trigger Optimization

As shown in Fig. 2 (left), in a clean DM trained on a specific dataset, the model's sample space typically aligns with the training data distribution, while assigning negligible probability to the backdoor target. During backdoor training, the sample space is gradually distorted, expanding toward the backdoor target (Fig. 2, right). After sufficient training, the backdoor target is totally included in the model's sample space; sampling from the triggered noise would result in the target with high likelihood. We make a key observation: when an arbitrary trigger (e.g., a stop sign) is stamped into the input noise of a clean model (Fig. 2, left), the noise distribution is no longer Gaussian. Consequently, the resulting output is highly likely to manifest as an image that is (i) erroneous or out-of-distribution (OOD) relative to the model's sample space, and (ii) significantly far from the desired backdoor target distribution. However, due to inherent randomness in

the generative process, some triggers may, by chance, produce outputs that are closer to the backdoor target than others.

Building on this insight, we propose our approach: instead of selecting arbitrary triggers, we optimize a trigger that causes the clean model, prior to any backdoor injection, to generate samples that already close to the backdoor target (as shown in Fig. 3). By injecting this optimized trigger during the attack, we exploit its inherent bias toward the backdoor-target distribution. This facilitates a more efficient expansion of the model's output space to include the backdoor target, requiring fewer parameter updates during backdoor training and thereby reducing both the required poison rate and training time.

To realize this, at any arbitrary timestep t, we minimize the distance between: (i) the output of the backward process with trigger stamped in the input noise, and (ii) the result of the forward process which adds noise to the backdoor target.

Assume that there are a total of T diffusion steps. Let $M_\theta(\boldsymbol{\epsilon}, t)$ denote the denoising process that maps input noise $\boldsymbol{\epsilon}$ to the intermediate sample $\mathbf{x}_t^{\text{backward}}$ at timestep t, after $(T-t)$ denoising steps. If we embed a trigger $\boldsymbol{\delta}$ into the input noise, the outcome of the backward process becomes:

$$\hat{\mathbf{x}}_t^{\text{backward}} = M_\theta(\boldsymbol{\epsilon} + \boldsymbol{\delta}, t). \tag{2}$$

Note that $\hat{\mathbf{x}}$ specifically denotes backdoor cases. On the other hand, for a clean DM, the forward process is typically represented as:

$$\mathbf{x}_t = a(t)\mathbf{x}_0 + b(t)\boldsymbol{\epsilon}, \tag{3}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, $\mathbf{x}_0$ is a clean image, $a(t)$ is the content schedule, and $b(t)$ is the noise schedule. Let $\hat{\mathbf{x}}_0$ represents the backdoor target. According to Eq. (3), applying the forward process for t steps on $\hat{\mathbf{x}}_0$ yields:

$$\hat{\mathbf{x}}_t^{\text{forward}} = a(t)\hat{\mathbf{x}}_0 + b(t)\boldsymbol{\epsilon}, \tag{4}$$

where $a(t)$ and $b(t)$ depend on the specific type of the victim DM. For instance, DDPMs have $a(t) = \sqrt{\bar{\alpha}_t}$ and $b(t) = \sqrt{1-\bar{\alpha}_t}$, where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ and α_i is a noise schedule [16].

To optimize the trigger, we solve the following minimization problem:

$$\min_{\boldsymbol{\delta}} L_{\text{TB}}(\boldsymbol{\delta}), \tag{5}$$

$$\begin{aligned} L_{\text{TB}}(\boldsymbol{\delta}) &= \mathbb{E}_{t,\boldsymbol{\epsilon}} \|\hat{\mathbf{x}}_t^{\text{forward}} - \hat{\mathbf{x}}_t^{\text{backward}}\|_2^2 \\ &= \mathbb{E}_{t,\boldsymbol{\epsilon}} \|a(t)\hat{\mathbf{x}}_0 + b(t)\boldsymbol{\epsilon} - M_\theta(\boldsymbol{\epsilon} + \boldsymbol{\delta}, t)\|_2^2. \end{aligned} \tag{6}$$

Although this trigger optimization stage introduces some additional training time, this cost is negligible compared to the backdoor injection process (details in Sec. 4.4). Moreover, our experiments demonstrate that the optimized trigger helps reducing backdoor injection time by an order of magnitude for even a higher attack effectiveness, leading to a substantial reduction in overall backdoor time, while significantly improving attack efficiency across all metrics.

3.3 Hidden Trigger

Although triggers learnt by our loss L_{TB} can enhance backdoor attacks, they remain detectable by SOTA defenses [22]. To improve stealthiness, it is crucial to hide the distribution shifts induced by the triggers during denoising [1]. Thus, we introduce additional constraints while minimizing L_{TB} to ensure that the triggered noise $(\boldsymbol{\epsilon} + \boldsymbol{\delta})$ closely resembles a Gaussian noise [10]:

$$\min_{\boldsymbol{\delta}} L_{\mathrm{TB}}(\boldsymbol{\delta}) \quad \text{s.t.} \quad \underbrace{\|\boldsymbol{\delta}\|_\infty \leq \varepsilon}_{\text{invisibility}}, \quad \underbrace{\|\boldsymbol{\delta}\|_0 \leq k}_{\text{sparsity}}, \tag{7}$$

where ε controls the maximum absolute value of elements in $\boldsymbol{\delta}$, enforcing invisibility; and k is the sparsity budget, indicating the maximum number of non-zero elements in $\boldsymbol{\delta}$.

In practice, we adopt Projected Gradient Descent (PGD) [28] to enforce the invisibility constraint, while the sparsity constraint is conducted by selecting top-k largest entries [10]. At each iteration i, we update the trigger as follows:

$$\boldsymbol{\delta}^{(i+1)} = \mathcal{P}_{\text{sparse}}\left(\mathit{\Pi}_\infty\left(\boldsymbol{\delta}^{(i)} - \eta \nabla_{\boldsymbol{\delta}} L_{\mathrm{TB}}(\boldsymbol{\delta}^{(i)})\right), k\right), \tag{8}$$

where η is the step size, $\mathit{\Pi}_\infty(\cdot)$ denotes the projection onto the ℓ_∞-ball of radius ε, performed by bounding each component of the vector to lie within $[-\varepsilon, \varepsilon]$. Finally, $\mathcal{P}_{\text{sparse}}(\cdot, k)$ enforces the sparsity constraint by retaining only the top-k largest entries and setting the rest to zero:

$$\left[\mathcal{P}_{\text{sparse}}(\mathbf{z}, k)\right]_j = \begin{cases} z_j, & \text{if } j \in \mathcal{I}_k(\mathbf{z}) \\ 0, & \text{otherwise} \end{cases}, \tag{9}$$

where $\mathcal{I}_k(\mathbf{z})$ is the set of indices corresponding to the k largest absolute values in $\mathbf{z}$. The detailed procedure of our trigger optimization with imperceptibility constraints can be found in Algorithm 1.

3.4 Backdoor Injection

To inject a backdoor into DMs, the forward process in Eq. (3) is extended by: (i) replacing $\mathbf{x}_0$ by the backdoor target $\hat{\mathbf{x}}_0$, and (ii) introducing an additional term that incorporates the backdoor trigger $\boldsymbol{\delta}$:

$$\hat{\mathbf{x}}_t = a(t)\hat{\mathbf{x}}_0 + b(t)\boldsymbol{\epsilon} + c(t)\boldsymbol{\delta}, \tag{10}$$

where $\hat{\mathbf{x}}_0$ is the backdoor target, $\boldsymbol{\delta}$ is the trigger, and $c(t)$ is the trigger schedule. Different backdoor injection methods select different coefficients $a(t)$, $b(t)$, and $c(t)$. In our framework, we adopt the coefficient settings from VillanDiffusion [8] since this is a unified backdoor injection method among existing frameworks like BadDiffusion [7] and TrojDiff [5]. The backdoor backward and training processes are then derived based on the above forward process. Detailed formulations can be found in [8]. While the trigger injection step is based on VillanDiffusion, we

Algorithm 1 TooBad Trigger Optimization

Input: Clean model M_θ, number of denoising steps T, backdoor target $\hat{\mathbf{x}}_0$, sparsity budget k, invisibility budget ε, number of trigger optimization iterations N, learning rate η

Output: Optimized trigger $\boldsymbol{\delta}$

for $i = 0, 1, ..., N-1$ **do**
  $t \sim \text{Uniform}(0, T)$
  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
  $\hat{\mathbf{x}}_t^{\text{backward}} = M_\theta(\boldsymbol{\epsilon} + \boldsymbol{\delta}, t)$
  $\hat{\mathbf{x}}_t^{\text{forward}} = a(t)\hat{\mathbf{x}}_0 + b(t)\boldsymbol{\epsilon}$
  $L_{\text{TB}} = \|\hat{\mathbf{x}}_t^{\text{forward}} - \hat{\mathbf{x}}_t^{\text{backward}}\|_2^2$
  $\boldsymbol{\delta}^{(i+1)} = \boldsymbol{\delta}^{(i)} - \eta \nabla_{\boldsymbol{\delta}} L_{\text{TB}}(\boldsymbol{\delta}^{(i)})$
  $\boldsymbol{\delta}^{(i+1)} = \text{clip}(\boldsymbol{\delta}^{(i+1)}, -\varepsilon, \varepsilon)$
  $\boldsymbol{\delta}^{(i+1)} = \mathcal{P}_{\text{sparse}}(\boldsymbol{\delta}^{(i+1)}, k)$
end for
return $\boldsymbol{\delta}$

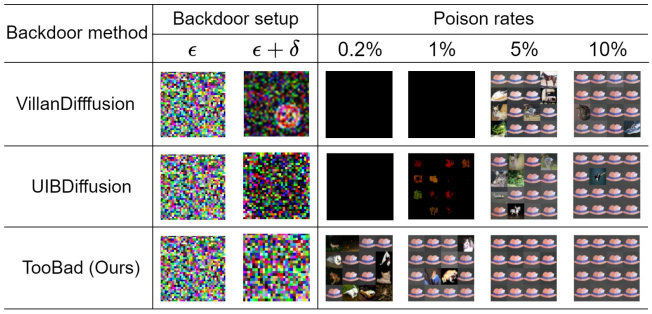

Fig. 4: An illustration of generated backdoor samples.

modify the data poisoning process and introduce a trigger optimization step before backdoor injection. For data poisoning, VillanDiffusion employs a patch-based approach, while TooBad blends the trigger into the entire poisoned image. Notably, while VillanDiffusion employs predefined, human-visible triggers (e.g., a stop sign image), TooBad uses the presented trigger optimization algorithm to generate an efficient yet stealthy trigger for backdoor injection.

4 Experimental Results

We evaluate the performance of TooBad primarily on four attack aspects according to the quadrilemma presented above in Fig. 1, including attack performance (Sec. 4.2), poison rate requirements (Sec. 4.3), time complexity (Sec. 4.4), and stealthiness (Sec. 4.5). Our goal is to show that TooBad can simultaneously improve these criteria over prior SOTA attacks.

Table 1: Performance comparison using different poison rates.

Method	VillanDiffusion			UIBDiffusion			TooBad (Ours)		
Poison Rate	ASR	MSE	SSIM	ASR	MSE	SSIM	ASR	MSE	SSIM
0.2%	0.00	X	X	0.00	X	X	**0.68**	**0.0329**	**0.6902**
1%	0.00	X	X	0.00	X	X	**0.93**	**0.0103**	**0.9087**
2%	0.01	X	X	0.02	X	X	**0.94**	**0.0078**	**0.9157**
3%	0.03	X	X	0.07	X	X	**0.93**	**0.0085**	**0.9173**
5%	0.38	0.0825	0.5357	0.16	0.1867	0.1716	**0.98**	**0.0038**	**0.9528**
10%	0.96	0.0043	0.9864	0.74	0.0487	0.7507	**0.99**	**0.0023**	**0.9612**

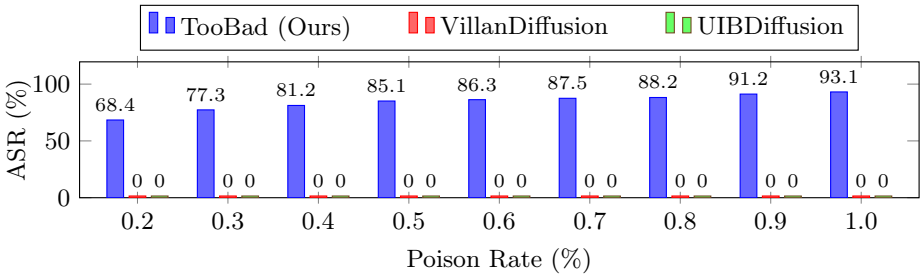

Fig. 5: ASR comparison under ultra-low poison rates.

4.1 Experimental Settings

Datasets. We evaluate TooBad and compare it with the baselines mainly on CIFAR-10 [21]. We also extend the experiments to a higher-resolution dataset, downscaled CelebA-HQ [26], in Sec. 4.6. These two datasets are commonly used in all prior backdoor attack and defense studies for DMs [7, 8, 13], ensuring a fair and consistent comparison. In addition, CelebA-HQ-Dialog [19] is used in TooBad's extension to conditioned generation, presented in the Appendix.

Baselines. We assess the performance of TooBad across both denoising-based models (DDPMs and LDMs[2]) and score-based models (NCSNs). While TooBad primarily focus on noise-space triggers, we also extended our analysis to conditioned models in the Appendix. For performance comparison, we benchmark TooBad against two SOTA backdoor attacks that have demonstrated the highest effectiveness to date: VillanDiffusion [8] and UIBDiffusion [13]. For VillanDiffusion, we use the default stop-sign image as the trigger. For UIBDiffusion, we follow the same experimental settings described in their paper to generate the trigger. All methods use the fedora-hat image as the default backdoor target.

Implementation Details. For invisibility and sparsity constraints of trigger optimization, we set $\varepsilon = 0.15$ and $k = 0.2|\boldsymbol{\delta}|$, with $|\boldsymbol{\delta}|$ denotes the total number

[2] While LDMs allows multi-modal input such as text prompt, we only focus on optimizing visual triggers embedded into input noises. Therefore, LDM can be considered DDPM in the autoencoder's latent space.

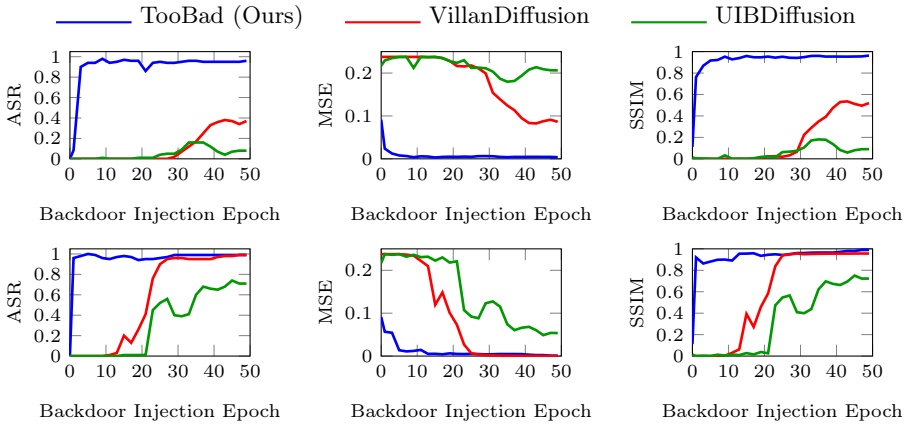

Fig. 6: Performance comparison across training epochs for three attacks. Top row: 5% poison rate. Bottom row: 10% poison rate.

of elements in $|\boldsymbol{\delta}|$. Trigger optimization is conducted over 50 iterations with a learning rate of 0.3 and batch size of 32. For backdoor injection, we employ 50 backdoor fine-tuning epochs with a learning rate of 2e-4, batch size of 128, using the SDE solver with 1000 denoising timesteps. All experiments were conducted on NVIDIA RTX A6000 ADA GPUs and reported on average across three runs.

Evaluation Metrics. We evaluate attack performance using three metrics: (i) Attack Success Rate (ASR), the percentage of samples generated by the backdoored model that successfully match the backdoor target with low distance (detailed matching criteria in the Appendix); (ii) Average Mean Squared Error (MSE), the pixel-wise MSE between generated images and the target, where lower values indicate better performance; and (iii) Structural Similarity Index Measure (SSIM) [50], where higher values indicate closer structural similarity to the target. For stealthiness, we first assess utility by computing the FID score [15] on clean samples generated by the backdoored models, with lower FID indicating higher model utility. Then, we assess resilience against SOTA defenses by performing their trigger inversion and backdoor detection algorithms. Trigger inversion is measured via the L2 distance (L2D) between the inverted and ground-truth triggers, while backdoor detection is assessed using accuracy (ACC) and true positive rate (TPR).

4.2 Attack Performance Analysis

Tab. 1 presents the performance of TooBad compared to existing baselines across various poison rates. At a 10% poison rate, TooBad, with almost 100% ASR, outperforms SOTA methods across all evaluation metrics. The advantage becomes more pronounced at lower poison rates: when the poison rate drops to 5%, the ASRs of both baselines are reduced dramatically, whereas TooBad still maintains near-perfect attack success with around 98% ASR, 0.004 MSE, and

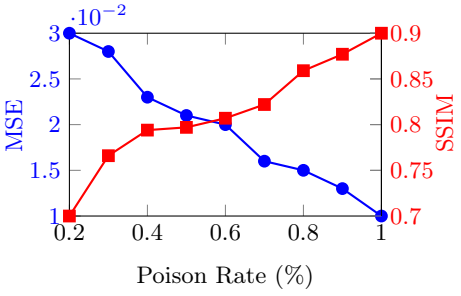

Fig. 7: Performance under ultra-low poison rates.

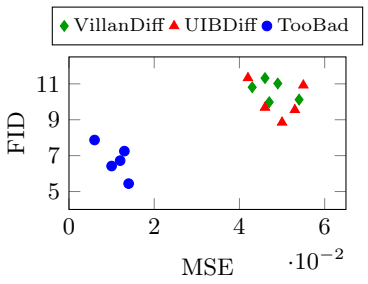

Fig. 8: Utility comparison across different backdoored models.

0.95 SSIM. Notably, for poison rates below 5%, only TooBad remains effective. Both VillanDiffusion and UIBDiffusion fail entirely, resulting in $\sim$ 0% ASR. In this case, models backdoored by the baselines only produce black images with arbitrary artifacts, yielding abnormally low MSE and high SSIM. These invalid results are marked as "X" in Tab. 1. In contrast, TooBad successfully initiates backdoor behavior at just 0.2% poison rate and achieves strong performance (93% ASR, 0.01 MSE, and 0.91 SSIM) at 1% poison rate. Some generated examples are visualized in Fig. 4. More generated samples are in the Appendix.

4.3 Backdoor under Ultra-Low Poison Rates

This section further highlights the efficiency of our method under extremely low poison rate conditions, ranging from 0.1% to 1%. As shown in Fig. 5, only TooBad is able to successfully backdoor DMs in this setting. The ASR of our method rises quickly from 68.4% at a 0.2% poison rate to 93% at 1%. In contrast, the baseline methods consistently yield 0% ASR across all tested poison rates. The corresponding MSE and SSIM scores for TooBad within this range are illustrated in Fig. 7, further validating its effectiveness. Notably, the MSE drops rapidly from 0.03 at a 0.2% poison rate to only 0.01 at 1%, while the SSIM score rises steadily from 0.7 to 0.9. These trends indicate that our optimized trigger not only enables successful attacks but also produces backdoored generations that closely resemble the intended target. In summary, TooBad is the only method that remains effective under extremely low poison rates. This not only makes the attack more practical but also significantly improves its utility.

4.4 Time and Convergence Analysis

We note that the trigger optimization stage introduces only negligible overhead compared to the backdoor injection stage (as shown in Table 2) for two main reasons. First, each trigger optimization iteration runs on a small batch of sampled noises without using any training data, whereas each backdoor injection

epoch trains the model on the entire poisoned dataset. Second, backdoor injection fine-tunes the large victim model, while trigger optimization keeps the model frozen and updates only the low-dimensional trigger variables. For example, attacking DDPMs on CIFAR-10, the full process of 50 trigger optimization iterations take about 5 minutes, compared with roughly 2 hours for 50 fine-tuning epochs of backdoor injection. Consequently, and without compromising fairness, the experiment evaluates performance primarily as a function of the number of backdoor injection epochs. Fig. 6 monitors the performance of TooBad and the baselines thorough the backdoor injection process at 5% and 10% poison rate. In both cases, our framework converges rapidly, reaching near-perfect performance within just 3-5 backdoor injection epochs. By epoch 5, TooBad already achieves nearly 100% ASR, close to 1.0 SSIM, and a negligible MSE, while both baselines completely fail to backdoor the victim models. The baselines require more than 30-50 epochs to approach the performance that TooBad attains after only 5 epochs.

Table 2: Overhead analysis for different settings (in minutes).

Settings	Trigger Opt.	Total-Ours	VillanDiff
CIFAR-10 DDPM	5	18	125
CelebA-HQ LDM	8	55	423
CIFAR-10 NCSN	15	149	1,328

4.5 Stealthiness Analysis

Table 3: Resilience of the attacks against SOTA defenses.

Defense	Metric	VillanDiff	UIBDiff	TooBad-NI	TooBad-NS	TooBad
Elijah	ACC	32.16	0.00	27.65	13.12	0.00
	TPR	14.77	0.00	10.23	3.34	0.00
	L2D	38.01	40.67	38.95	39.05	41.03
TERD	ACC	90.76	0.00	85.33	17.65	0.00
	TPR	86.23	0.00	76.66	12.03	0.00
	L2D	27.89	40.22	32.67	38.65	40.16
PureDiff	ACC	100	0.00	100	19.78	0.00
	TPR	100	0.00	100	14.07	0.00
	L2D	24.12	41.06	25.33	39.12	41.21

Resistance to SOTA Defenses. We evaluate the robustness of TooBad against three recent defenses: Elijah [1], TERD [29], and PureDiffusion [47]. These defenses typically follow a two-stage procedure: first, a trigger inversion stage that

try to reconstruct the backdoor trigger from the suspicious model; and second, a detection stage that analyzes the inverted trigger to determine whether the model is backdoored. As shown in Tab. 3, TooBad completely evades all three defenses, resulting in a 0% detection rate across all scenarios. In contrast, the ablated variants TooBad-NS (without sparsity constraint) and TooBad-NI (without invisibility constraint) become detectable. The results show that: (i) the strong resilience of TooBad stems from the imperceptibility constraints applied during trigger optimization, and (ii) the invisibility constraint accounts for the majority of stealthiness. Further ablation study for two imperceptibility constraints can be found in the Appendix. For the baselines, UIBDiffusion also produces irreversible triggers, while VillanDiffusion is exposed by the defenses.

Utility Evaluation. If a backdoored model suffers from low utility, it may fail to generate realistic samples or occasionally produce the backdoor target even without the trigger, making the attack easily detectable. To quantify utility, we primarily use the FID score, where lower values indicate that, in the absence of the trigger, the backdoored model can generate clean samples closely matching the distribution of the original training data. In our experiments, each backdoor method is applied to the same set of five pretrained models with varying poison rates (from 0.2% to 5%). Since there is an inherent trade-off between utilily and attack performance, both FID and MSE are reported for each resulting backdoored model for fair comparison. As shown in Fig. 8, models backdoored by TooBad consistently achieve the lowest FID and lowest MSE, outperforming all baselines in both utility and attack performance. These results demonstrate that TooBad not only improve backdoor effectiveness but also preserves the fidelity of clean samples.

Table 4: Performance on alternative targets beyond the hat image.

Target	Cat			Stop Sign		
Poison Rate	ASR	MSE	SSIM	ASR	MSE	SSIM
0.2%	0.66	0.0326	0.733	0.63	0.0368	0.633
0.5%	0.83	0.0112	0.883	0.81	0.0156	0.826
1%	0.87	0.0083	0.898	0.86	0.0096	0.869
2%	0.92	0.0056	0.925	0.93	0.0048	0.933
5%	0.99	0.0022	0.981	0.99	0.0026	0.988
10%	0.99	0.0018	0.986	0.99	0.0021	0.978

4.6 Ablation Study

Alternative Backdoor Targets. To demonstrate that the superior performance of TooBad is not dependent on a specific backdoor target, we evaluate it using alternative targets beyond the default fedora hat image. We experiment with varying poison rates from 0.2% to 10%, using two new targets: a cat image

and a stop-sign image. As shown in Tab. 4, for these targets, TooBad consistently offers strong attack performance across all settings. TooBad begins to successfully backdoor DMs at just 0.2% poison rate and achieves near-perfect ASR at 5%, regardless of the chosen target image. These results validate that TooBad can offer superior backdoor performance no matter the chosen backdoor targets, which can be harmful images rather than just cat or hat images.

Table 5: Performance comparison on NCSNs and CIFAR-10.

Method	VillanDiffusion		UIBDiffusion		TooBad (Ours)	
Poison rate	ASR	MSE	ASR	MSE	ASR	MSE
25%	0	X	0	X	0.69	0.392
30%	0	X	0	X	0.84	0.041
35%	0	X	0	X	0.87	0.034
40%	0	X	0	X	0.90	0.025
45%	0	X	0	X	0.95	0.015
50%	0.70	0.428	0.65	0.412	0.98	0.010
70%	0.72	0.382	0.69	0.392	1.00	0.005

Results for NCSNs. In addition to DDPMs and LDMs, we evaluate TooBad on NCSNs and compare its performance with baseline methods. As noted in [8], backdooring NCSNs typically requires significantly higher poison rates than DDPMs to be effective. However, as shown in Tab. 5, TooBad still outperforms existing SOTA methods by a considerable margin. While prior attacks require at least 50% poison rate to successfully backdoor NCSNs, TooBad achieves comparable at only half that rate, and reaches near-perfect ASR at 50%. Notably, TooBad at 30% poison rate even achieved higher attack efficiency than both VillanDiffusion and UIBDiffusion at 70% poison rate.

Results on CelebA-HQ. Tab. 6 presents a comparison between TooBad and the baseline methods on the CelebA-HQ dataset using two representative poison rates, 5% and 10%. To reduce computational cost, this experiment employs an LDM with a latent space size of 64×64. The results demonstrate that TooBad achieves near-perfect ASR at both poison rates, substantially outperforming the baselines. Visualization of generated samples are shown in the Appendix. These findings confirm the effectiveness of our framework on higher-resolution data.

5 Conclusion

We introduced TooBad, a novel backdoor framework that advances the state of backdoor attacks on DMs. Unlike prior methods which struggle with inherent performance trade-offs, TooBad achieves superior attack capability with minimal poison rate and training time. It successfully implants backdoors at poison rates less than 1%, reaching near-perfect ASR at just 5% poison rate within only

Table 6: Performance comparison on LDMs and CelebA-HQ.

Poison rate	$p = 5\%$			$p = 10\%$		
Method	ASR	MSE	SSIM	ASR	MSE	SSIM
VillanDiffusion	0.32	0.121	0.423	0.76	0.051	0.789
UIBDiffusion	0.19	0.186	0.188	0.71	0.046	0.722
TooBad (Ours)	0.98	0.003	0.945	0.99	0.002	0.965

a few training epochs. TooBad also maintains strong stealthiness, high utility, and demonstrates complete resistance to SOTA defense mechanisms. Its effectiveness generalizes across different backdoor targets and model types, making it a broadly applicable and practical threat. These results highlight a critical vulnerability in current generative models and call for urgent development of more robust defenses against such stealthy, low-resource yet highly effective attacks.

References

1. An, S., Chou, S.Y., Zhang, K., Xu, Q., Tao, G., Shen, G., Cheng, S., Ma, S., Chen, P.Y., Ho, T.Y., et al.: Elijah: Eliminating backdoors injected in diffusion models via distribution shift. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 10847–10855 (2024)
2. Austin, J., Johnson, D.D., Ho, J., Tarlow, D., Van Den Berg, R.: Structured denoising diffusion models in discrete state-spaces. In: Advances in Neural Information Processing Systems. vol. 34, pp. 17981–17993 (2021)
3. Cao, H., Tan, C., Gao, Z., Xu, Y., Chen, G., Heng, P.A., Li, S.Z.: A survey on generative diffusion models. IEEE Transactions on Knowledge and Data Engineering pp. 1–20 (2024). https://doi.org/10.1109/TKDE.2024.3361474
4. Chen, N., Zhang, Y., Zen, H., Weiss, R.J., Norouzi, M., Chan, W.: Wavegrad: Estimating gradients for waveform generation. In: Proceedings of the International Conference on Learning Representations (2020)
5. Chen, W., Song, D., Li, B.: Trojdiff: Trojan attacks on diffusion models with diverse targets. In: CVPR. pp. 4035–4044 (2023)
6. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)
7. Chou, S.Y., Chen, P.Y., Ho, T.Y.: How to backdoor diffusion models? In: CVPR. pp. 4015–4024 (June 2023)
8. Chou, S.Y., Chen, P.Y., Ho, T.Y.: Villandiffusion: A unified backdoor attack framework for diffusion models. In: NeuRIPS. pp. 33912–33964 (2023)
9. Croitoru, F.A., Hondru, V., Ionescu, R.T., Shah, M.: Diffusion models in vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(9), 10850–10869 (2023). https://doi.org/10.1109/TPAMI.2023.3261988
10. Gao, Y., Li, Y., Gong, X., Li, Z., Xia, S.T., Wang, Q.: Backdoor attack with sparse and invisible trigger. IEEE Transactions on Information Forensics and Security **19**, 6364–6376 (2024). https://doi.org/10.1109/TIFS.2024.3411936
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. vol. 27 (2014)

12. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)
13. Han, Y., Zhao, B., Chu, R., Luo, F., Sikdar, B., Lao, Y.: Uibdiffusion: Universal imperceptible backdoor attack for diffusion models. In: CVPR. pp. 19186–19196 (2025)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
15. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in Neural Information Processing Systems **30** (2017)
16. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems. vol. 33, pp. 6840–6851 (2020)
17. Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., Welling, M.: Argmax flows and multinomial diffusion: Learning categorical distributions. In: Advances in Neural Information Processing Systems. vol. 34, pp. 12454–12465 (2021)
18. Jiang, W., Li, H., Xu, G., Zhang, T.: Color backdoor: A robust poisoning attack in color space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8133–8142 (2023)
19. Jiang, Y., Huang, Z., Pan, X., Loy, C.C., Liu, Z.: Talk-to-edit: Fine-grained facial editing via dialog. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13799–13808 (2021)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of the International Conference on Machine Learning (2014)
21. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
22. Li, S., Ma, J., Cheng, M.: Invisible backdoor attacks on diffusion models. arXiv preprint arXiv:2406.00816 (2024)
23. Li, X., Thickstun, J., Gulrajani, I., Liang, P.S., Hashimoto, T.B.: Diffusion-lm improves controllable text generation. In: Advances in Neural Information Processing Systems. vol. 35, pp. 4328–4343 (2022)
24. Liang, J., Liang, S., Liu, A., Cao, X.: Vl-trojan: Multimodal instruction backdoor attacks against autoregressive visual language models. Proceedings of the International Journal of Computer Vision pp. 1–20 (2025)
25. Liang, S., Zhu, M., Liu, A., Wu, B., Cao, X., Chang, E.C.: Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24645–24654 (2024)
26. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3730–3738 (2015)
27. Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., Ma, J.: Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In: Advances in Neural Information Processing Systems. pp. 9754–9767 (2022)
28. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
29. Mo, Y., Huang, H., Li, M., Li, A., Wang, Y.: Terd: A unified framework for safeguarding diffusion models against backdoors. arXiv preprint arXiv:2409.05294 (2024)

30. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1765–1773 (2017)
31. Ngiam, J., Chen, Z., Koh, P.W., Ng, A.Y.: Learning deep energy models. In: Proceedings of the International Conference on Machine Learning. pp. 1105–1112 (2011)
32. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In: Proceedings of the International Conference on Machine Learning. pp. 16784–16804 (2021)
33. Pan, Z., Yao, Y., Liu, G., Shen, B., Zhao, H.V., Kompella, R.R., Liu, S.: From trojan horses to castle walls: Unveiling bilateral backdoor effects in diffusion models. In: Advances in Neural Information Processing Systems (2023)
34. Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., Kudinov, M.: Grad-tts: A diffusion probabilistic model for text-to-speech. In: Proceedings of the International Conference on Machine Learning. pp. 8599–8608 (2021)
35. Rasul, K., Sheikh, A.S., Schuster, I., Bergmann, U., Vollgraf, R.: Multivariate probabilistic time series forecasting via conditioned normalizing flows. In: Proceedings of the International Conference on Learning Representations (2020)
36. Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: Proceedings of the International Conference on Machine Learning. pp. 1530–1538 (2015)
37. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
38. Saha, A., Subramanya, A., Pirsiavash, H.: Hidden trigger backdoor attacks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 11957–11965 (2020)
39. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: Proceedings of the International Conference on Learning Representations (2021)
40. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: Advances in Neural Information Processing Systems. vol. 32, pp. 11918—11930 (2019)
41. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: Proceedings of the International Conference on Learning Representations (2021)
42. Struppek, L., Hintersdorf, D., Kersting, K.: Rickrolling the artist: Injecting backdoors into text encoders for text-to-image synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4584–4596 (2023)
43. Tashiro, Y., Song, J., Song, Y., Ermon, S.: Csdi: Conditional score-based diffusion models for probabilistic time series imputation. In: Advances in Neural Information Processing Systems. vol. 34, pp. 24804–24816 (2021)
44. Truong, V.T., Dang, L.B., Le, L.B.: Attacks and defenses for generative diffusion models: A comprehensive survey. ACM Computing Surveys **57**(8), 1–44 (2025)
45. Truong, V.T., Le, L.B.: Purediffusion: Using backdoor to counter backdoor in generative diffusion models. arXiv preprint arXiv:2409.13945 (2024)
46. Truong, V.T., Le, L.B.: Text-guided real-world-to-3d generative models with real-time rendering on mobile devices. In: Proceedings of the IEEE Wireless Communications and Networking Conference. pp. 1–6. IEEE (2024)
47. Truong, V.T., Le, L.B.: A dual-purpose framework for backdoor defense and backdoor amplification in diffusion models. arXiv preprint arXiv:2502.19047 (2025)

48. Walmer, M., Sikka, K., Sur, I., Shrivastava, A., Jha, S.: Dual-key multimodal backdoors for visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15375–15385 (2022)
49. Wang, H., Shen, Q., Tong, Y., Zhang, Y., Kawaguchi, K.: The stronger the diffusion model, the easier the backdoor: Data poisoning to induce copyright breaches without adjusting finetuning pipeline. In: Advances in Neural Information Processing Systems (2023)
50. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)
51. Watson, D., Chan, W., Ho, J., Norouzi, M.: Learning fast samplers for diffusion models by differentiating through sample quality. In: Proceedings of the International Conference on Learning Representations (2021)
52. Xu, J., Wang, X., Cheng, W., Cao, Y.P., Shan, Y., Qie, X., Gao, S.: Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20908–20918 (2023)
53. Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., Tang, J.: Geodiff: A geometric diffusion model for molecular conformation generation. In: Proceedings of the International Conference on Learning Representations (2021)
54. Yan, T., Zhang, H., Zhou, T., Zhan, Y., Xia, Y.: Scoregrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models. arXiv preprint arXiv:2106.10121 (2021)
55. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., Yang, M.H.: Diffusion models: A comprehensive survey of methods and applications. ACM Computing Surveys **56**(4), 1–39 (2023)
56. Yang, S., Doan, B.G., Montague, P., De Vel, O., Abraham, T., Camtepe, S., Ranasinghe, D.C., Kanhere, S.S.: Transferable graph backdoor attack. In: Proceedings of the International Symposium on Research in Sttacks, Intrusions and Defenses. pp. 321–332 (2022)
57. Zhai, S., Dong, Y., Shen, Q., Pu, S., Fang, Y., Su, H.: Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. In: ACM Multimedia. pp. 1577–1587 (2023)
58. Zhang, Y., Ruan, W., Wang, F., Huang, X.: Generalizing universal adversarial attacks beyond additive perturbations. In: Proceedings of the IEEE International Conference on Data Mining. pp. 1412–1417. IEEE (2020)
59. Zou, H., Kim, Z.M., Kang, D.: Diffusion models in nlp: A survey. arXiv preprint arXiv:2305.14671 (2023)