TOOBAD: BACKDOOR DIFFUSION MODELS WITH ULTRA-LOW POISON RATE AND IMPERCEPTIBLE TRIGGER

Anonymous authors
Paper under double-blind review

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 practicality (e.g., poison rate requirement). 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. Furthermore, high poison rates are often infeasible in real-world scenarios where attackers can poison only a minimal fraction of the training data. In this paper, we propose TooBad (trigger optimization for backdoor diffusion models), a novel attack framework which can implant backdoors into DMs with an extremely low poison rate, while achieving higher attack success rate (ASR), stronger resistance to backdoor defenses, and significantly reduced training time compared to existing backdoor attacks. Experiments show that TooBad can achieve 82% ASR at a 0.5% poison rate, significantly lower than the 10% poison rates required by prior work. At 5% poison rate, TooBad reaches nearly 100% ASR within just 3-5 training 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 state-of-the-art (SOTA) defenses while maintaining high utility (i.e., the capability of generating clean samples). These results reveal a critical threat on DMs and highlight the urgent need for more robust defenses against such stealthy yet efficient attacks.

1 INTRODUCTION

In recent years, diffusion models (DMs) (Yang et al., 2023; Cao et al., 2024) have rapidly become a dominant paradigm in deep generative modeling, setting new state-of-the-art benchmarks across a wide array of domains. By iteratively denoising data through a multi-step generative process (Ho et al., 2020), DMs have shown remarkable performance in various tasks, ranging from computer vision (Watson et al., 2021; Nichol et al., 2021) to natural language processing (NLP) (Zou et al., 2023; Austin et al., 2021; Hoogeboom et al., 2021; Li et al., 2022), 3D synthesis (Xu et al., 2023; Truong & Le, 2024b), audio generation (Chen et al., 2020; Popov et al., 2021), bioinformatics (Xu et al., 2021; Luo et al., 2022), and time series forecasting (Tashiro et al., 2021; Yan et al., 2021; Rasul et al., 2020). Compared to earlier generative frameworks like GANs (Goodfellow et al., 2014), energy-based models (EBMs) (Ngiam et al., 2011), and VAEs (Kingma & Welling, 2014; Rezende & Mohamed, 2015), DMs consistently achieve superior sample quality, diversity, and training stability.

Recent studies have revealed that DMs are highly susceptible to backdoor attacks (Chou et al., 2024; Truong et al., 2025). Such attacks involve training or fine-tuning DMs on poisoned data, with the use of manipulated forward and reversed processes. 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

[1] While our trigger optimization stage introduces some additional training time, this cost is insignificant compared to the backdoor training stage as we only optimize the low-dimensional trigger while keeping the large diffusion model frozen.

Gaussian noise, preserving normal behavior. However, existing backdoor attacks on DMs face a fundamental quadrilemma, as illustrated in Figure 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 state-of-the-art (SOTA) defenses when backdoor is triggered, while preserving the generative capability on benign input; (iv) **Time Complexity:** Reducing the training time for implanting the backdoor is crucial for saving computational resources and minimizing the risk of detection.

Achieving all four aspects simultaneously is inherently challenging, if not impossible. Enhancing one aspect often comes at the cost of compromising others (Chou et al., 2024). For example, attackers can boost attack performance by increasing the poison rate and extending the training time. However, this approach negatively impacts the remaining three aspects. First, it raises the poison rate requirement, which makes the attacks less feasible in real-world scenarios. Second, it increases time complexity due to prolonged training. Third, it reduces stealthiness, as stronger backdoor effects leave more noticeable traces for defense mechanisms to detect and can degrade the model's ability to generate high-quality benign outputs (Zou et al., 2023). Conversely, if attackers aim to enhance stealthiness, for instance, by shortening the training time, they often face diminished attack performance and may need to increase the poison rate to maintain a reasonable attack success rate.

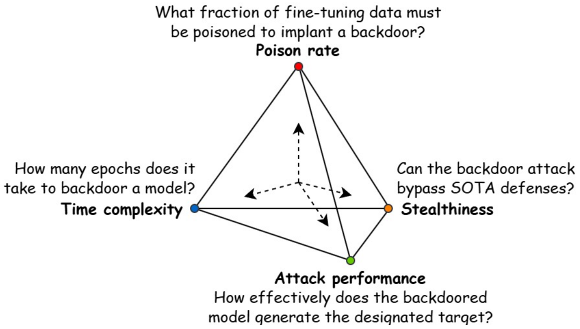

What fraction of fine-tuning data must
be poisoned to implant a backdoor?
Poison rate
How many epochs does it
take to backdoor a model?
Time complexity
Can the backdoor attack
bypass SOTA defenses?
Stealthiness
Attack performance
How effectively does the backdoored
model generate the designated target?

Figure 1: The **quadrilemma** illustrating the inability of SOTA attacks to meet all backdoor criteria at once.

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 (e.g., same poison rate, model architecture, and hyperparameters). 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 challenge, we propose TooBad, a novel backdoor strategy that leverages trigger optimization to overcome the trade-offs faced by prior DM-targeted backdoor attacks. While trigger optimization has been explored in earlier works on backdoor attacks (Saha et al., 2020; Liang et al., 2025; 2024), these efforts were limited to traditional classification and contrastive models. Such techniques cannot be directly applied to diffusion models (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 iterative denoising process of DMs without relying on auxiliary classifiers, enabling efficient backdoor attacks that deliver superior performance, operate under extremely low poison rates and short training times, and remain undetected by existing defenses through the use of additional imperceptibility constraints.

2 Background & Related Work

Diffusion Models. DMs are trained to generate high-quality samples through two key processes: a forward (diffusion) process and a backward (denoising) process. The forward process gradually adds Gaussian noise to clean images over multiple timesteps, transforming the data distribution into an isotropic Gaussian. The backward process then reverses this transformation, progressively denoising a sample drawn from the Gaussian distribution to reconstruct a clean image (Croitoru et al., 2023).

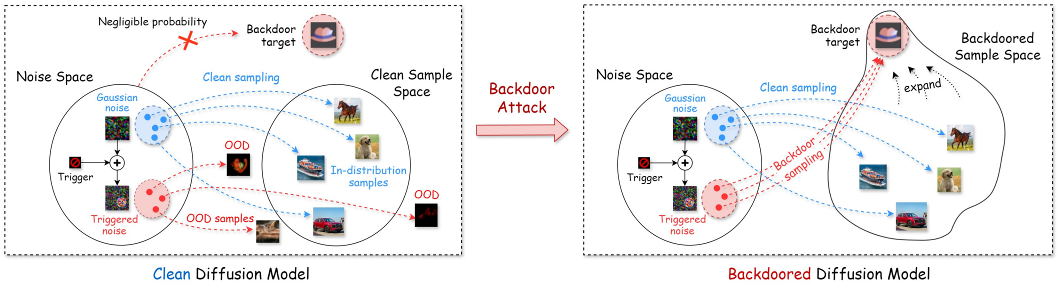


Figure 2: An illustration of backdoor attacks on DMs. (**Left**) Before backdoor attack, the backdoor target is outside of the model's sample space. Sampling from a Gaussian noise likely results in clean images, while sampling from a triggered noise yields out-of-distribution (OOD) samples. (**Right**)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

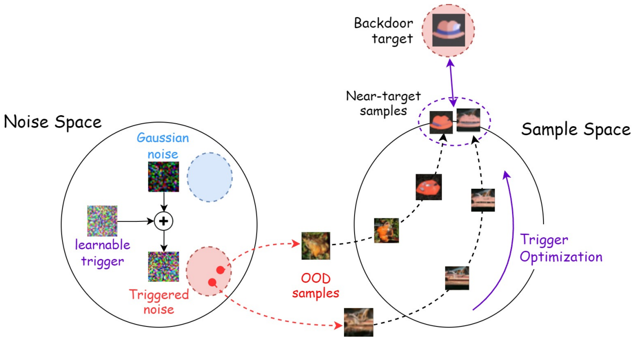


Figure 3: TooBad's trigger optimization. We optimize a trigger that causes the clean model to generate near-target samples, facilitating the subsequent backdoor injection process.

A foundational formulation of DMs is the Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020), 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 steps 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.

Subsequent works have introduced several DM variants to address limitations of DDPMs, such as Denoising Diffusion Implicit Models (DDIMs) (Song et al., 2021a), Noise Conditional Score Networks (NCSNs) (Song & Ermon, 2019; Song et al., 2021b), and Latent Diffusion Models (LDMs) (Rombach et al., 2022).

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 (Struppek et al., 2023; Zhai et al., 2023; Pan et al., 2023; Wang et al., 2023), while keeping the DMs frozen. In contrast, our work focus on attacking such the DMs. Early attempts in this direction include TrojDiff (Chen et al., 2023) and BadDiffusion (Chou et al., 2023), which incorporate a small amount of the backdoor trigger into each diffusion step. VillanDiffusion (Chou et al., 2024) was proposed as a unified backdoor framework compatible with various DM variants. To enhance stealthiness, UIBDiffusion (Han et al., 2024) proposes using Universal Adversarial Perturbations (UAPs) (Moosavi-Dezfooli et al., 2017; Zhang et al., 2020) to generate an imperceptible trigger, which is then injected into DMs via VillanDiffusion. Nevertheless, all these methods require relatively high poison rates (e.g., at least 10%) 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 (An et al., 2024; Truong & Le, 2024a; 2025; Mo et al., 2024).

Trigger Optimization for Backdoor Attacks. Early backdoor methods such as BadNets (Gu et al., 2017) and Blended (Chen et al., 2017) relied on fixed trigger patterns. Later works introduced trigger optimization to enhance attack efficiency in image classification (Saha et al., 2020; Jiang et al., 2023; Yang et al., 2022), and extended it to vision–language models (VLMs) (Liang et al., 2025; Walmer et al., 2022) and contrastive models (Liang et al., 2024), 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 (Han et al., 2024) adopted learnable triggers for DMs, they were optimized only through auxiliary classifiers (e.g., VGG (Simonyan & Zisserman, 2014), ResNet (He et al., 2016)) instead of generative ones, primarily improving stealthiness rather than attack efficiency and practicality. In contrast, our method, TooBad, is the first to optimize triggers directly within the diffusion process, enabling stronger backdoor attacks with both high effectiveness and stealthiness.

3 Methodology

3.1 Threat Model

In our attack setting, the attacker first chooses a specific backdoor target aligned with their intent, such as harmful content (e.g., nudity or violence) to endanger users. Next, they run our trigger optimization algorithm on a clean DM to learn an invisible yet effective trigger, which is then used to inject the backdoor into the model. After backdoor injection, the compromised DM are released on public platforms like HuggingFace or GitHub, where end users download and use it directly or integrate it into downstream tasks. Although this represents a white-box scenario, it is practical and consistent with all prior DM backdoor frameworks that aim to backdoor the DM itself instead of just the encoders (Chen et al., 2023; Chou et al., 2023; 2024; Han et al., 2024). Compared to existing DM-targeted backdoor attacks, our work even improve practicality in real-world scenarios: By reducing the required poison rate by an order of magnitude, TooBad shifts backdoor attacks from theoretical to practical threats in model-sharing ecosystems, thereby redefining the security landscape for diffusion models. Notably, our threat model also accounts for defenders who may apply detection or mitigation techniques on the downloaded models, further strengthening practicality.

3.2 Backdoor Injection

In general, the clean forward process of a DM can be represented as:

$$\mathbf{x}_t = a(t)\mathbf{x}_0 + b(t)\boldsymbol{\epsilon}, \tag{2}$$

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. To inject a backdoor, the forward process is extended by introducing an additional term that incorporates a backdoor trigger:

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

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 method, TooBad, we adopt the coefficient settings from VillanDiffusion (Chou et al., 2024) since this is a unified, SOTA backdoor injection method. The backdoor backward and training processes are then derived based on the above forward process. Detailed formulations of these coefficients can be found in (Chou et al., 2024). While the trigger injection step is based on VillanDiffusion, we modify the data poisoning process and introduce a trigger optimization step before trigger injection, resulting in two key differences: First, VillanDiffusion employs a patch-based approach for data poisoning, where a poisoned image is defined as $\mathbf{r} = \mathbf{M} \odot \mathbf{g} + (1 - \mathbf{M}) \odot \mathbf{x}$, where $\mathbf{g}$ is a predefined trigger pattern, $\odot$ denotes element-wise product, and $\mathbf{M}$ is a binary mask (with elements taking values of either 0 or 1) specifying the trigger's area. In contrast, TooBad fuses the trigger into the entire image via the following formula: $\mathbf{r} = \mathbf{x} + \gamma\boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is our imperceptible trigger, and γ is the trigger strength (typically set as 1). Second, unlike VillanDiffusion, which employs predefined, human-visible triggers (e.g., a stop sign), TooBad proposes a novel trigger optimization algorithm specifically designed for DMs, which constitutes its main contribution and is described below.

3.3 Trigger Optimization

To motivate our method, we first view backdoor attacks from a sampling distribution perspective as shown in Fig. 2. Specifically, 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 (Fig. 2, left). During backdoor training, the sample space is gradually distorted, expanding toward the backdoor target. 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 (Fig. 2, right). We make a key observation: if we stamp an arbitrary trigger (e.g., a stop sign) into the input noise of a **clean model**, the input noise is no more Gaussian, thus the resulting output is likely to be an arbitrary image that is (i) out-of-distribution (OOD) with respect to the model's sample space, and (ii) far from the desired backdoor target distribution (Fig. 2, left). 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 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.

Given 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:

$$\hat{\mathbf{x}}_t^{\text{backward}} = M_\theta(\boldsymbol{\epsilon} + \boldsymbol{\delta}, t). \tag{4}$$

Note that $\hat{\mathbf{x}}$ specifically denotes backdoor cases. Let $\hat{\mathbf{x}}_0$ represents the backdoor target. According to Equation 2, applying the forward process for t steps on $\hat{\mathbf{x}}_0$ yields:

$$\hat{\mathbf{x}}_t^{\text{forward}} = a(t)\hat{\mathbf{x}}_0 + b(t)\boldsymbol{\epsilon}, \tag{5}$$

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 (Ho et al., 2020).

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

$$\min_{\boldsymbol{\delta}} L_{\text{TB}}(\boldsymbol{\delta}), \tag{6}$$

$$\begin{aligned} L_{\text{TB}}(\boldsymbol{\delta}) &= \mathbb{E}_{t,\boldsymbol{\epsilon}} \|\hat{\mathbf{x}}_t^{\text{forward}} - \hat{\mathbf{x}}_t^{\text{backward}}\|_2^2 \\ &= \mathbb{E}_{t,\boldsymbol{\epsilon}} \|a(t)\hat{\mathbf{x}}_0 + b(t)\boldsymbol{\epsilon} - M_\theta(\boldsymbol{\epsilon} + \boldsymbol{\delta}, t)\|_2^2. \end{aligned} \tag{7}$$

This trigger optimization step introduces some additional training, but since only the trigger is learnable and the clean guidance model remains frozen, this cost is negligible compared to the backdoor injection process (details in 4.4). Moreover, our experiments demonstrate that the optimized trigger helps reducing backdoor injection time by an order of magnitude for the same attack effectiveness, leading to a substantial reduction in overall backdoor time, while significantly improving attack efficiency across all metrics.

3.4 Hidden Trigger

Although triggers learnt by our loss L_{TB} can enhance backdoor attacks, they remain detectable by SOTA defenses (Li et al., 2024). To improve stealthiness, it is crucial to hide the distribution shifts induced by the triggers during denoising (An et al., 2024). Inspired by (Gao et al., 2024), we introduce additional constraints while minimizing L_{TB} to ensure that the triggered noise $(\boldsymbol{\epsilon} + \boldsymbol{\delta})$ closely resembles a Gaussian noise:

$$\min_{\boldsymbol{\delta}} L_{\text{TB}}(\boldsymbol{\delta}) \quad \text{s.t.} \quad \underbrace{\|\boldsymbol{\delta}\|_\infty \leq \varepsilon}_{\text{invisibility}}, \quad \underbrace{\|\boldsymbol{\delta}\|_0 \leq k}_{\text{sparsity}}, \tag{8}$$

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) (Madry et al., 2017) to enforce the invisibility constraint, while the sparsity constraint is conducted by selecting top-k largest entries (Gao et al., 2024). Specifically, at each iteration i, we update the trigger as follows:

$$\boldsymbol{\delta}^{(i+1)} = \mathcal{P}_{\text{sparse}}\left(\Pi_\infty\left(\boldsymbol{\delta}^{(i)} - \eta\nabla_{\boldsymbol{\delta}} L_{\text{TB}}(\boldsymbol{\delta}^{(i)})\right), k\right), \tag{9}$$

where η is the step size, $\Pi_\infty(\cdot)$ 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:

$$\left[\mathcal{P}_{\text{sparse}}(\mathbf{z}, k)\right]_j = \begin{cases} z_j, & \text{if } j \in \mathcal{I}_k(\mathbf{z}) \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$

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 Appendix A.1.

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 (4.2), poison rate requirements (4.3), time complexity (4.4), and stealthiness (4.5). Our goal is to show that TooBad can simultaneously improve these criteria in comparison with prior SOTA methods. Due to the limited scope of the paper, the main manuscript primarily present the results for DDPM-based models using the CIFAR-10 dataset (Krizhevsky et al., 2009), with backdoor target is the fedora hat image. Further results for score-based models, high-resolution datasets, and alternative backdoor targets will be presented in Appendix A.

4.1 Experimental Settings

Datasets and Baselines. To ensure a fair comparison, we evaluate TooBad primarily on CIFAR-10 (Krizhevsky et al., 2009), which is also commonly used in prior backdoor attack and defense studies. Additional experiments on CelebA-HQ (Liu et al., 2015) are presented in Appendix A.5. We assess the performance of TooBad across both denoising-based models and score-based models. For performance comparison, we benchmark TooBad against two SOTA backdoor attacks that have demonstrated the highest effectiveness to date: VillanDiffusion (Chou et al., 2024) and UIBDiffusion (Han et al., 2024). For more details on the implementation of the backdoor baselines and datasets, please refer to Appendix A.1.

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 of elements in $|\boldsymbol{\delta}|$. Trigger optimization is conducted over 50 epochs in most settings with a learning rate of 0.3. For backdoor injection, we fine-tune the victim models with a learning rate of 2e-4, batch size of 128, using the SDE solver with the total number of denoising timesteps is 1000. All experiments were conducted on NVIDIA RTX A6000 ADA GPUs. The experimental results are 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 (matching criteria in Appendix A.1); (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) (Wang et al., 2004), where higher values indicate closer structural similarity to the target. To evaluate stealthiness, we first assess utility by computing the FID score (Heusel et al., 2017) 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).

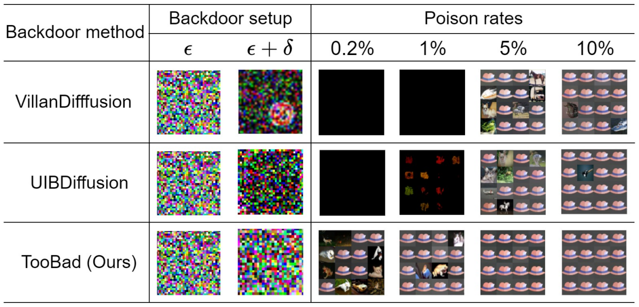


Figure 4: An illustration of generated samples when backdoor is activated.

	$p = 0.2\%$			$p = 1\%$			$p = 5\%$			$p = 10\%$		
Method	ASR	MSE	SSIM	ASR	MSE	SSIM	ASR	MSE	SSIM	ASR	MSE	SSIM
VillanDiff	0	X	X	0	X	X	0.47	0.025	0.63	0.98	0.004	0.92
UIBDiff	0	X	X	0	X	X	0.51	0.021	0.71	0.98	0.003	0.95
TooBad	**0.65**	**0.03**	**0.72**	**0.86**	**0.01**	**0.86**	**0.99**	**0.002**	**0.96**	**0.99**	**0.002**	**0.96**

Table 1: Performance comparison between our method and the baselines with different poison rates.

4.2 Attack Performance Analysis

Table 1 presents the performance of TooBad compared to existing baselines across various poison rates. At a 10% poison rate, TooBad slightly outperforms SOTA methods across all evaluation metrics. The advantage becomes more pronounced at lower poison rates: when the poison rate drops to 5%, the ASR of both baselines is reduced by half, whereas TooBad still maintains near-perfect attack success with around 99% ASR, 0.002 MSE, and 0.96 SSIM. Notably, for poison rates below 5%, only TooBad remains effective. Both VillanDiffusion and UIBDiffusion fail entirely at this level, resulting in 0% ASR. In this case, models backdoored by the baseline methods only produce black images with arbitrary artifacts, yielding abnormally low MSE and high SSIM. These results are obviously invalid, thus they are marked as "X" in Table 1. In contrast, TooBad successfully initiates backdoor behavior at just 0.2% poison rate and achieves strong performance (86% ASR, 0.01 MSE, and 0.86 SSIM) at 1% poison rate. Some generated examples are visualized in Fig. 4. For more generated results, please refer to Appendix A.3.

4.3 Backdoor under Ultra-Low Poison Rates

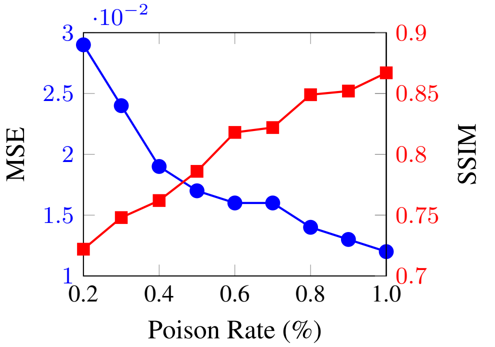


Figure 6: Efficiency for ultra-low poison rates.

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 65.4% at a 0.2% poison rate to 86.1% 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. 6, further validating its effectiveness. Notably, the MSE drops rapidly from nearly 0.03 at a 0.2% poison rate to only 0.12 at 1%, while the SSIM score rises steadily from 0.72 to 0.86. 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 setting offers two major ben-

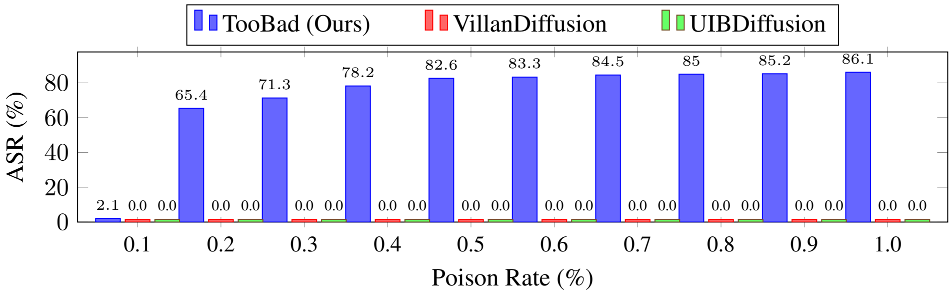


Figure 5: ASR comparison between our method and the two baselines under ultra-low poison rates.

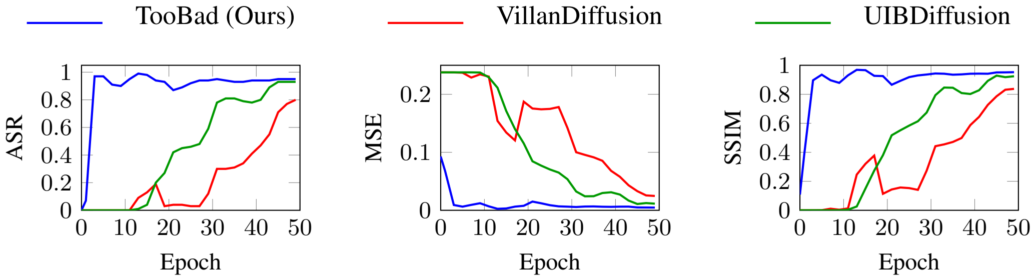


Figure 7: Performance comparison across training epochs for three attacks at a 5% poison rate.

efits: (i) it enables attackers to achieve strong attack performance without compromising the overall quality of the model's clean generations, and (ii) it requires only a tiny amount of poisoned data, which not only makes the attack more practical but also significantly improves its stealthiness.

Poison Rate	2%	3%	5%	10%	20%	30%	50%
VillanDiffusion	X	X	X	35	23	9	3
UIBDiffusion	X	X	X	32	19	7	3
TooBad (Ours)	10	6	3	1	1	1	1

Table 2: Minimum number of backdoor training epochs for each attack to achieve at least 90% ASR. The marker "X" represents invalid results where the attacks cannot achieve the required ASRs.

4.4 Time and Convergence Analysis

We note that the trigger optimization stage does not require access to the training dataset, and the model is kept frozen while only the low-dimensional trigger variables are updated. This makes the computation lightweight and fast. In contrast, backdoor injection must fine-tune the entire model on large-scale poisoned data, which is substantially more expensive. In our experiments, when attacking DDPMs on CIFAR-10, trigger optimization completes in about 5 minutes, whereas backdoor injection requires around 2 hours for 50 fine-tuning epochs. Thus, without compromising fairness, this experiment mainly evaluates the number of training epochs for backdoor injection. Fig. 7 monitors the performance of TooBad and the baselines during training at a 5% poison rate. Our framework converges rapidly, reaching near-perfect performance within just 3–5 backdoor training epochs. By epoch 5, TooBad already achieves nearly 100% ASR, close to 1.0 SSIM, and a very small MSE, while both baselines completely fail to backdoor the victim models. In contrast, the baselines require more than 30-50 epochs to approach, and often remain below, the performance that TooBad attains after only 5 epochs. Next, we evaluate how quickly each method reaches at least 90% ASR under different poison rates. As shown in Table 2, at a 10% poison rate, TooBad surpasses 90% ASR in just 1 epoch, while the baselines require over 30 epochs. When the poison rate drops below 5%, both baselines fail to reach 90% ASR, whereas TooBad consistently achieves this target within only a few epochs.

4.5 Stealthiness Analysis

Defense	Metric	VillanDiffusion	UIBDiffusion	TooBad-NI	TooBad-NS	TooBad
Elijah	ACC	32.16	0	27.65	13.12	0
	TPR	14.77	0	10.23	3.34	0
	L2D	38.01	40.67	38.95	39.05	41.03
TERD	ACC	90.76	0	85.33	17.65	0
	TPR	86.23	0	76.66	12.03	0
	L2D	27.89	40.22	32.67	38.65	40.16
PureDiffusion	ACC	100	0	100	19.78	0
	TPR	100	0	100	14.07	0
	L2D	24.12	41.06	25.33	39.12	41.21

Table 3: Resilience of the attacks against SOTA defenses.

Resistance to SOTA Defenses. We evaluate the robustness of TooBad against three recent defenses: Elijah (An et al., 2024), TERD (Mo et al., 2024), and PureDiffusion (Truong & Le, 2025). 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 Table 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. This shows 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 improvement in stealthiness. Ablation study for two imperceptibility constraints can be found in Appendix A.6. 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 1% 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.

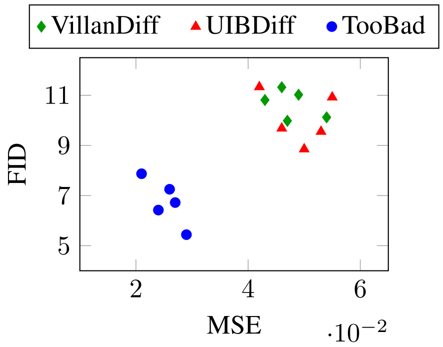


Figure 8: Utility comparison across different backdoored models.

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 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.

ETHICS STATEMENT

Our study demonstrated that backdoor attacks on DMs can be both highly efficient and stealthy, even under extremely low poison rates and short training times. This highlights critical vulnerabilities in these powerful generative models. We acknowledge the potential risk of misuse, as such attacks could pose serious threats to the security and reliability of current DM-based systems. Nonetheless, we believe that systematically exposing these weaknesses is essential for driving the development of stronger defense mechanisms. Our work aims to raise awareness within the research community and provide insights that will support the design of more robust safeguards, ultimately advancing the safety and trustworthiness of generative models.

REFERENCES

Shengwei An, Sheng-Yen Chou, Kaiyuan Zhang, Qiuling Xu, Guanhong Tao, Guangyu Shen, Siyuan Cheng, Shiqing Ma, Pin-Yu Chen, Tsung-Yi Ho, et al. Elijah: Eliminating backdoors injected in diffusion models via distribution shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10847–10855, 2024.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z. Li. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–20, 2024. doi: 10.1109/TKDE.2024.3361474.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *Proceedings of the International Conference on Learning Representations*, 2020.

Weixin Chen, Dawn Song, and Bo Li. Trojdiff: Trojan attacks on diffusion models with diverse targets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4035–4044, 2023.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. How to backdoor diffusion models? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4015–4024, June 2023.

Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. Villandiffusion: A unified backdoor attack framework for diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 10850–10869, 2023. doi: 10.1109/TPAMI.2023.3261988.

Yinghua Gao, Yiming Li, Xueluan Gong, Zhifeng Li, Shu-Tao Xia, and Qian Wang. Backdoor attack with sparse and invisible trigger. *IEEE Transactions on Information Forensics and Security*, 19: 6364–6376, 2024. ISSN 1556-6021. doi: 10.1109/TIFS.2024.3411936.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Yuning Han, Bingyin Zhao, Rui Chu, Feng Luo, Biplab Sikdar, and Yingjie Lao. Uibdiffusion: Universal imperceptible backdoor attack for diffusion models. *arXiv preprint arXiv:2412.11441*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12454–12465, 2021.

Wenbo Jiang, Hongwei Li, Guowen Xu, and Tianwei Zhang. 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.

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

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Sen Li, Junchi Ma, and Minhao Cheng. Invisible backdoor attacks on diffusion models. *arXiv preprint arXiv:2406.00816*, 2024.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4328–4343, 2022.

Jiawei Liang, Siyuan Liang, Aishan Liu, and Xiaochun Cao. Vl-trojan: Multimodal instruction backdoor attacks against autoregressive visual language models. *Proceedings of the International Journal of Computer Vision*, pp. 1–20, 2025.

Siyuan Liang, Mingli Zhu, Aishan Liu, Baoyuan Wu, Xiaochun Cao, and Ee-Chien Chang. 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.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3730–3738, 2015.

Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. 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.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Yichuan Mo, Hui Huang, Mingjie Li, Ang Li, and Yisen Wang. Terd: A unified framework for safeguarding diffusion models against backdoors. *arXiv preprint arXiv:2409.05294*, 2024.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1765–1773, 2017.

Jiquan Ngiam, Zhenghao Chen, Pang W Koh, and Andrew Y Ng. Learning deep energy models. In *Proceedings of the International Conference on Machine Learning*, pp. 1105–1112, 2011.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the International Conference on Machine Learning*, pp. 16784–16804, 2021.

Zhuoshi Pan, Yuguang Yao, Gaowen Liu, Bingquan Shen, H Vicky Zhao, Ramana Rao Kompella, and Sijia Liu. From trojan horses to castle walls: Unveiling bilateral backdoor effects in diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *Proceedings of the International Conference on Machine Learning*, pp. 8599–8608, 2021.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *Proceedings of the International Conference on Learning Representations*, 2020.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the International Conference on Machine Learning*, pp. 1530–1538, 2015.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 11957–11965, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the International Conference on Learning Representations*, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32, pp. 11918—11930, 2019.

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

Lukas Struppek, Dominik Hintersdorf, and Kristian Kersting. 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.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24804–24816, 2021.

Vu Tuan Truong and Long Bao Le. Purediffusion: Using backdoor to counter backdoor in generative diffusion models. *arXiv preprint arXiv:2409.13945*, 2024a.

Vu Tuan Truong and Long Bao Le. 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, 2024b.

Vu Tuan Truong and Long Bao Le. A dual-purpose framework for backdoor defense and backdoor amplification in diffusion models. *arXiv preprint arXiv:2502.19047*, 2025.

Vu Tuan Truong, Luan Ba Dang, and Long Bao Le. Attacks and defenses for generative diffusion models: A comprehensive survey. *ACM Computing Surveys*, 57(8):1–44, 2025.

Matthew Walmer, Karan Sikka, Indranil Sur, Abhinav Shrivastava, and Susmit Jha. 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.

Haonan Wang, Qianli Shen, Yao Tong, Yang Zhang, and Kenji Kawaguchi. 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.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. In *Proceedings of the International Conference on Learning Representations*, 2021.

Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. 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.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *Proceedings of the International Conference on Learning Representations*, 2021.

Tijin Yan, Hongwei Zhang, Tong Zhou, Yufeng Zhan, and Yuanqing Xia. Scoregrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models. *arXiv preprint arXiv:2106.10121*, 2021.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

Shuiqiao Yang, Bao Gia Doan, Paul Montague, Olivier De Vel, Tamas Abraham, Seyit Camtepe, Damith C Ranasinghe, and Salil S Kanhere. Transferable graph backdoor attack. In *Proceedings of the International Symposium on Research in Sttacks, Intrusions and Defenses*, pp. 321–332, 2022.

Shengfang Zhai, Yinpeng Dong, Qingni Shen, Shi Pu, Yuejian Fang, and Hang Su. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. In *Proceedings of the ACM International Conference on Multimedia*, pp. 1577–1587, 2023.

Yanghao Zhang, Wenjie Ruan, Fu Wang, and Xiaowei Huang. Generalizing universal adversarial attacks beyond additive perturbations. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 1412–1417. IEEE, 2020.

Hao Zou, Zae Myung Kim, and Dongyeop Kang. Diffusion models in nlp: A survey. *arXiv preprint arXiv:2305.14671*, 2023.

A Appendix

A.1 Detailed Experimental Settings

Details of Attack Baselines. For a fair comparison, all victim models were trained with the same learning rate of 2e-4, batch size of 128, and the SDE solver, while the number of training epochs and poison rates were varied according to the specific experiments. For DDPMs, we used the same sampler for all attacks with 1000 denoising steps. The base pretrained DDPMs for backdoor fine-tuning were downloaded from HuggingFace: ("*google/ddpm-cifar10-32*") for CIFAR-10 and ("*google/ddpm-ema-celebahq-256*") for CelebA-HQ. For VillanDiffusion, the backdoor trigger is the stop-sign image shown in Fig. 9. For UIBDiffusion, we followed the provided settings, generating the trigger using VGG (Simonyan & Zisserman, 2014) and ResNet (He et al., 2016).

Details of Backdoor Defense. For Elijah (An et al., 2024), we deployed trigger inversion with a learning rate of 0.1, batch size 64, Adam optimizer, and the first-step trigger scale of $\lambda = 0.5$. For TERD (Mo et al., 2024), the trade-off coefficient γ was set to 5e-5, and δ to $0.01T$, with 3000 epochs for trigger estimation and 1000 epochs for trigger refinement. For PureDiffusion (Truong & Le, 2024a), the first stage of trigger inversion optimized the L_{MDS} loss over 30 epochs with batch size 8 and learning rate 0.1, while the second stage optimized the L_{DC} loss over 500 epochs with batch size 16 and learning rate 0.5.

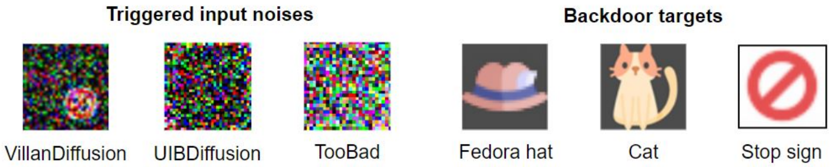


Figure 9: Visualization of backdoor targets used in the experiments with the triggered noises of each experimented backdoor attack.

Details of Evaluation Metrics. To compute the ASR, we first calculate the L1 distance between each generated sample and the backdoor target. This distance is then compared to a predefined threshold to determine whether the generated sample successfully matches the target. If the L1 distance is smaller than the threshold, the sample is considered a successful attack; otherwise, it is not. In our experiments, this threshold is empirically set to 500 based on observed results, though it can be adjusted. Importantly, the choice of threshold does not significantly affect the fairness of performance comparisons: lowering the threshold reduces ASR across all attack methods, while increasing it raises ASR for all methods proportionally. All metrics are reported as the average over 3 repeated runs of the same experiment.

A.2 Details of Trigger Optimization

The complete procedure of our trigger optimization mechanism is presented in algorithm 1. For each training epoch, we first sample a random timestep t and a Gaussian noise $\boldsymbol{\epsilon}$. Then, we run the backward process for $T-t$ timesteps to obtain $\hat{\mathbf{x}}_t^{\text{backward}}$ from the triggered noise $\boldsymbol{\epsilon}+\boldsymbol{\delta}$. On the other hand, the forward process's result at timestep t can be computed directly via the reparameterization trick, using the same initial noise $\boldsymbol{\epsilon}$. These two outputs are jointly used to compute the loss L_{TB}, which is subsequently optimized under our invisibility and sparsity constraints to update the trigger parameters. In practice, we apply our proposed loss function (8) only during the last 50% of the denoising steps, as the early steps primarily consist of high noise levels and experimentally illustrated negligible improvement to trigger optimization.

Algorithm 1 TooBad Trigger Optimization

Input: Clean model θ, number of denoising steps T, backdoor target $\hat{\mathbf{x}}_0$, sparsity budget k, invisibility budget ε, number of epochs for trigger learning N, learning rate η
Output: Optimized trigger $\boldsymbol{\delta}$

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

A.3 Visualization of Backdoor Results

Fig. 10 shows samples generated by VillanDiffusion, UIBDiffusion, and TooBad under a 5% poison rate. Our method, TooBad, demonstrates the strongest attack performance, consistently producing

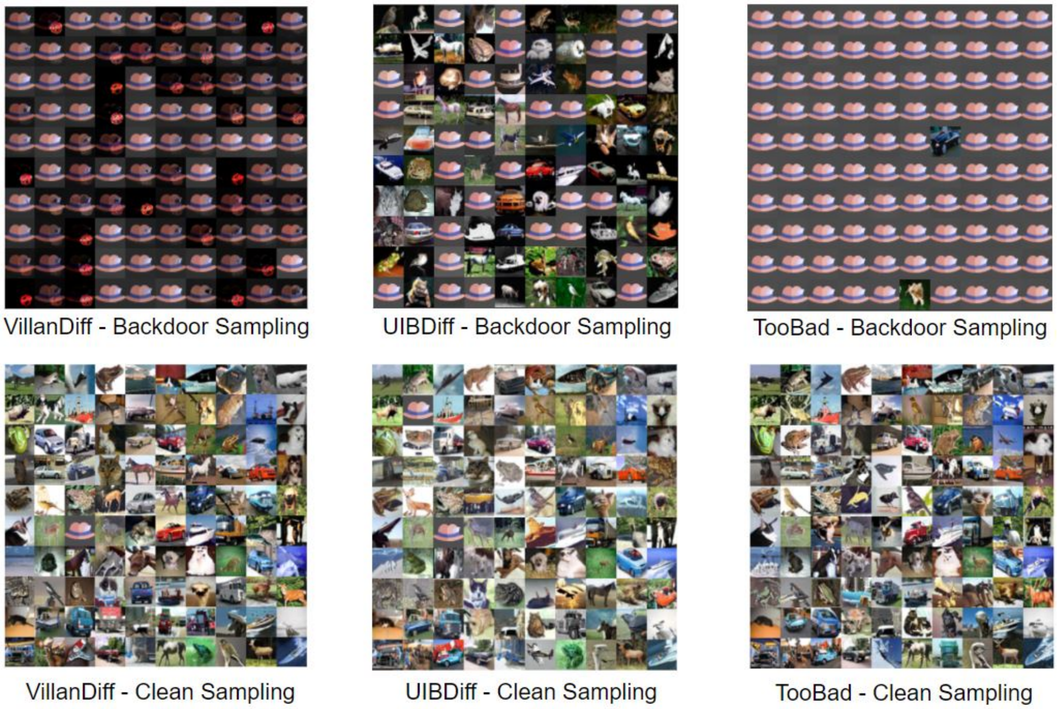

VillanDiff - Backdoor Sampling
UIBDiff - Backdoor Sampling
TooBad - Backdoor Sampling
VillanDiff - Clean Sampling
UIBDiff - Clean Sampling
TooBad - Clean Sampling

Figure 10: Visualized samples generated by each method with 5% poison rate and 50 backdoor training epochs on CIFAR-10, both backdoor and clean sampling.

the backdoor target in nearly all generation attempts. In contrast, both VillanDiffusion and UIBDiffusion exhibit much lower attack efficiency. An observable weakness of VillanDiffusion is that the backdoor trigger (i.e., the stop sign in the bottom-right corner) is explicitly visible in many generated samples, making the attack easily detectable even by casual users without any defense tools. This issue does not occur in UIBDiffusion or TooBad, where failed attacks still generate natural-looking images without exposing the trigger. For clean sampling, VillanDiffusion and UIBDiffuion occasionally produce the backdoor target, while TooBad consistently generate clean samples.

Fig. 11 visualizes samples generate by TooBad under 4 different poison rates (0.5%, 1%, 5%, and 10%) along the backdoor training process. In general, the attack performance improves gradually over the training epochs, and higher poison rates typically yield higher backdoor performance. This is highly consistent with our initial hypotheses.

A.4 Results on NCSNs

In addition to denoising-based models, we evaluate TooBad on NCSNs and compare its performance with baseline methods. We downloaded a pretrained NCSN from HuggingFace ("*FrankC-CCCC/NCSN_CIFAR10_my*") for CIFAR-10. As noted in (Chou et al., 2024), backdooring NCSNs typically requires significantly higher poison rates than DDPMs to be effective. However, as shown in Table 4, 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 or better performance at only half that rate, and reaches near-perfect ASR at 50%. In contrast, both VillanDiffusion and UIBDiffusion, even with a 70% poison rate, only achieve $\sim$ 70% ASR. As visualized in Fig. 12, TooBad at 30% poison rate even achieved higher attack efficiency than both VillanDiffusion and UIBDiffusion at 90% poison rate. These results further validate the efficiency of TooBad for different DM variants rather than only DDPMs.

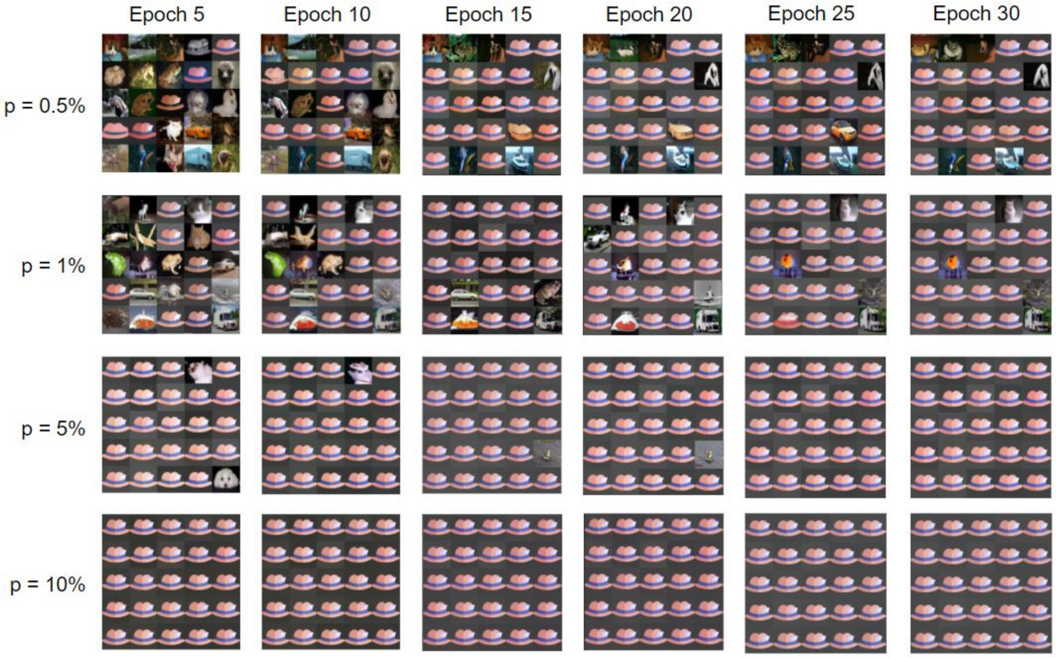


Figure 11: Visualized samples generated by TooBadd under different poison rates (from 0.5% to 10%) and over different training epochs (up to epoch 30).

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

Table 4: Performance comparison on NCSNs.

A.5 Results on LDMs and CelebA-HQ

In this section, we evaluate TooBad on the downscaled CelebA-HQ dataset, which contains 30,000 human face images at a resolution of 256×256. We poison the dataset using our optimized trigger and employ it to backdoor a pretrained LDM obtained from HuggingFace ("*CompVis/ldm-celebahq-256*"). The model is fine-tuned with a batch size of 16, a learning rate of 2e-5, 1000 denoising steps, and an SDE solver. Due to resource limitations, this experiment is conducted only on the CAT target with a 5% poison rate. The generated backdoored samples are shown in Fig. 13. These results demonstrate that TooBad remains effective on high-resolution datasets and LDM-based models even under very low poison rates, further validating the efficiency and generality of our framework across different DM variants and datasets.

A.6 Ablation Study

Imperceptibility Constraints. Although decreasing the invisibility and sparsity budgets (ε and k) can improve the stealthiness of the attack, it also degrades both attack effectiveness and model utility. In particular, as $(\epsilon, k) \to (0, 0)$, the trigger $\boldsymbol{\delta} \to \mathbf{0}$, making the triggered noise approaches pure Gaussian noise, i.e., $(\boldsymbol{\epsilon} + \boldsymbol{\delta}) \to \boldsymbol{\epsilon}$. Consequently, $M_{\boldsymbol{\theta}}(\boldsymbol{\epsilon} + \boldsymbol{\delta}, T) \approx M_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}, T)$, which makes the backdoored model unable to reliably distinguish between clean and triggered inputs during fine-

(ε, σ)	(0.05, 0.2)	(0.1, 0.2)	**(0.15, 0.2)**	(0.15, 0.1)	(0.15, 0.05)
FAR	12.89	2.34	**0.13**	0.78	1.86
TFR	65.44	27.45	**0.56**	2.95	7.53

Table 5: Ablation study for invisibility and sparsity constraints. Configuration and results highlighted in bold are used in our attack.

tuning and sampling. As a result, clean sampling tends to collapse to the backdoor target, whereas backdoor sampling more frequently produces clean outputs. To evaluate this trade-off, we vary the invisibility and sparsity budgets during trigger optimization. We introduce two complementary metrics to capture the impact of imperceptibility constraints on generation quality: (i) false activation rate (FAR), the probability that the clean sampling process produces the backdoor target; and (ii) trigger failure rate (TFR), the probability that the backdoor sampling process produces clean outputs. For convenience, we define the sparsity ratio as $\sigma = k/|\boldsymbol{\delta}|$. As shown in Table 5, both FAR and TFR increase as ε and σ decrease. At $\varepsilon = 0.15$ and $\sigma = 0.2$, we achieved negligible FAR and TFR, while this setting already bypassed SOTA backdoor defenses as presented in Section 4.5. In other words, $(\varepsilon, \sigma) = (0.15, 0.2)$ achieves a favorable balance: sufficiently small to evade defenses yet sufficiently large to maintain high attack success. We therefore adopt this configuration in our trigger optimization mechanism.

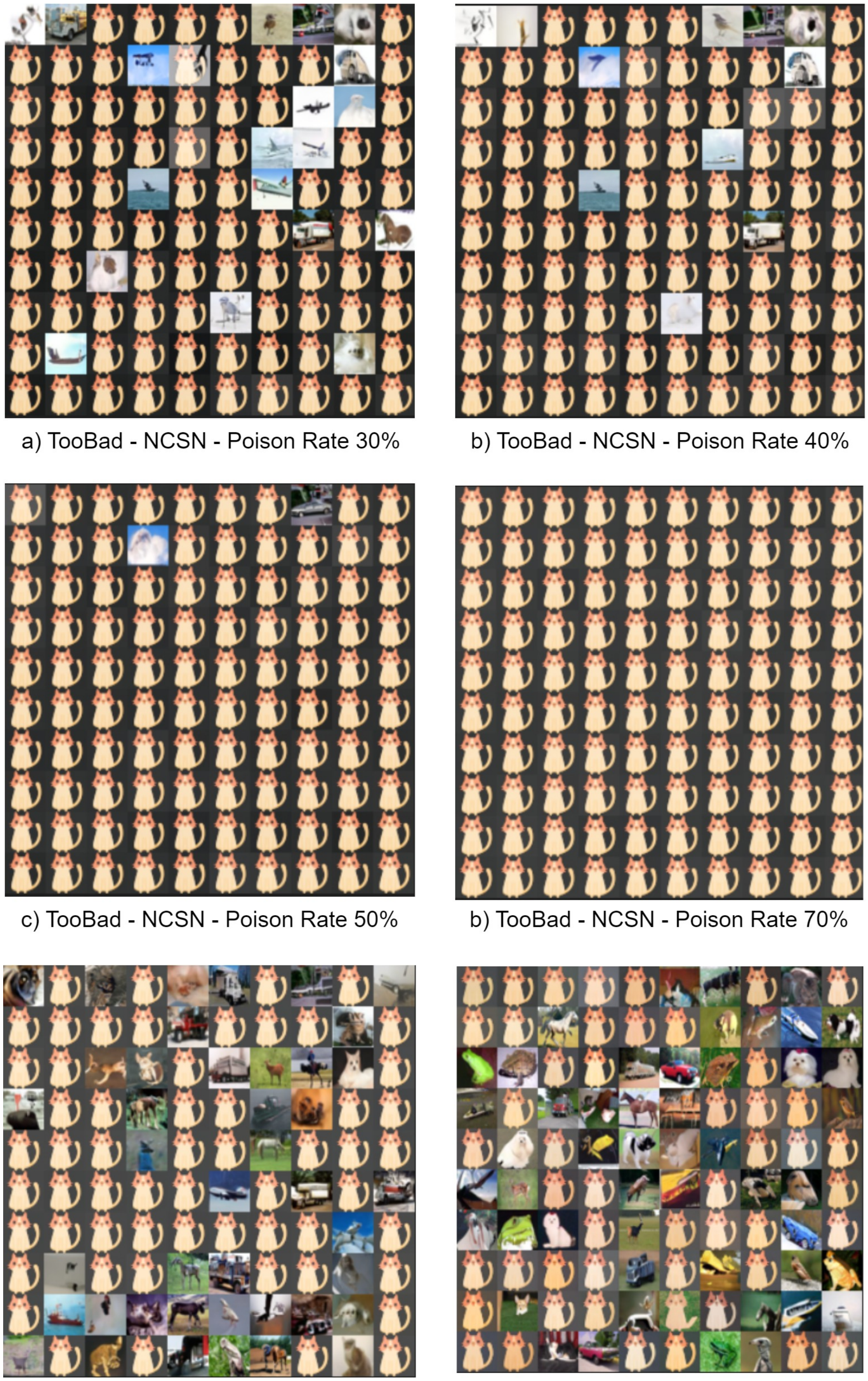
Results on Alternative Backdoor Targets. To demonstrate that the superior performance of TooBad is not dependent on a specific backdoor target, we evaluate it using alternative targets beyond the default fedora hat image. We experiment with varying poison rates from 0.2% to 10%, using two new targets: a cat image and a stop-sign image (illustrated in Fig. 9). As shown in Table 6, for these targets, TooBad consistently offers strong attack performance across all settings. In all cases, 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. It worth noting that the two baseline attacks only achieve around 50% ASR at 5% poison rate, as shown in section 4.2. These results validate that TooBad can offer superior backdoor performance no matter the chosen backdoor targets. The experimented targets are just to demonstrate the efficiency of the attacks. In practice, the targets can be harmful images instead of just cat or hat images.

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

Table 6: Performance of our attack with alternative backdoor targets beyond the default hat image.

A.7 The Use of Large Language Models

In this paper, we used large language models (LLMs) minimally to polish grammar and improve readability of some sentences. All technical content, ideas, and analyses are authored by the paper's authors.



a) TooBad - NCSN - Poison Rate 30%

b) TooBad - NCSN - Poison Rate 40%

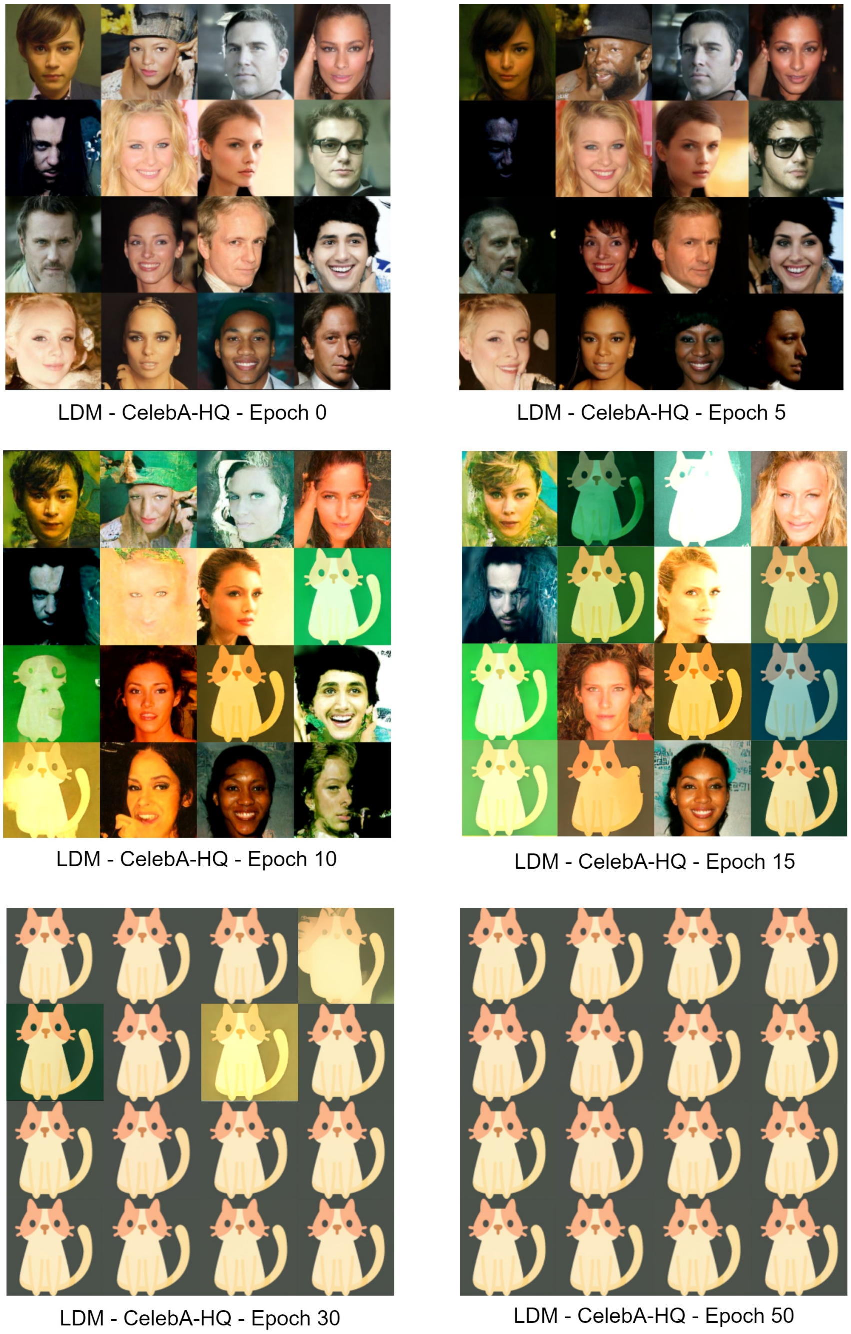
c) TooBad - NCSN - Poison Rate 50%

b) TooBad - NCSN - Poison Rate 70%

d) VillanDiff - NCSN - Poison Rate 90%

e) UIBDiff - NCSN - Poison Rate 90%

Figure 12: Generation results of three attacks on NCSNs with different poison rates and backdoor target is a cat image.



LDM - CelebA-HQ - Epoch 0

LDM - CelebA-HQ - Epoch 5

LDM - CelebA-HQ - Epoch 10

LDM - CelebA-HQ - Epoch 15

LDM - CelebA-HQ - Epoch 30

LDM - CelebA-HQ - Epoch 50

Figure 13: Visualization of TooBad's backdoor sampling for LDMs with CelebA-HQ under 5% poison rate.

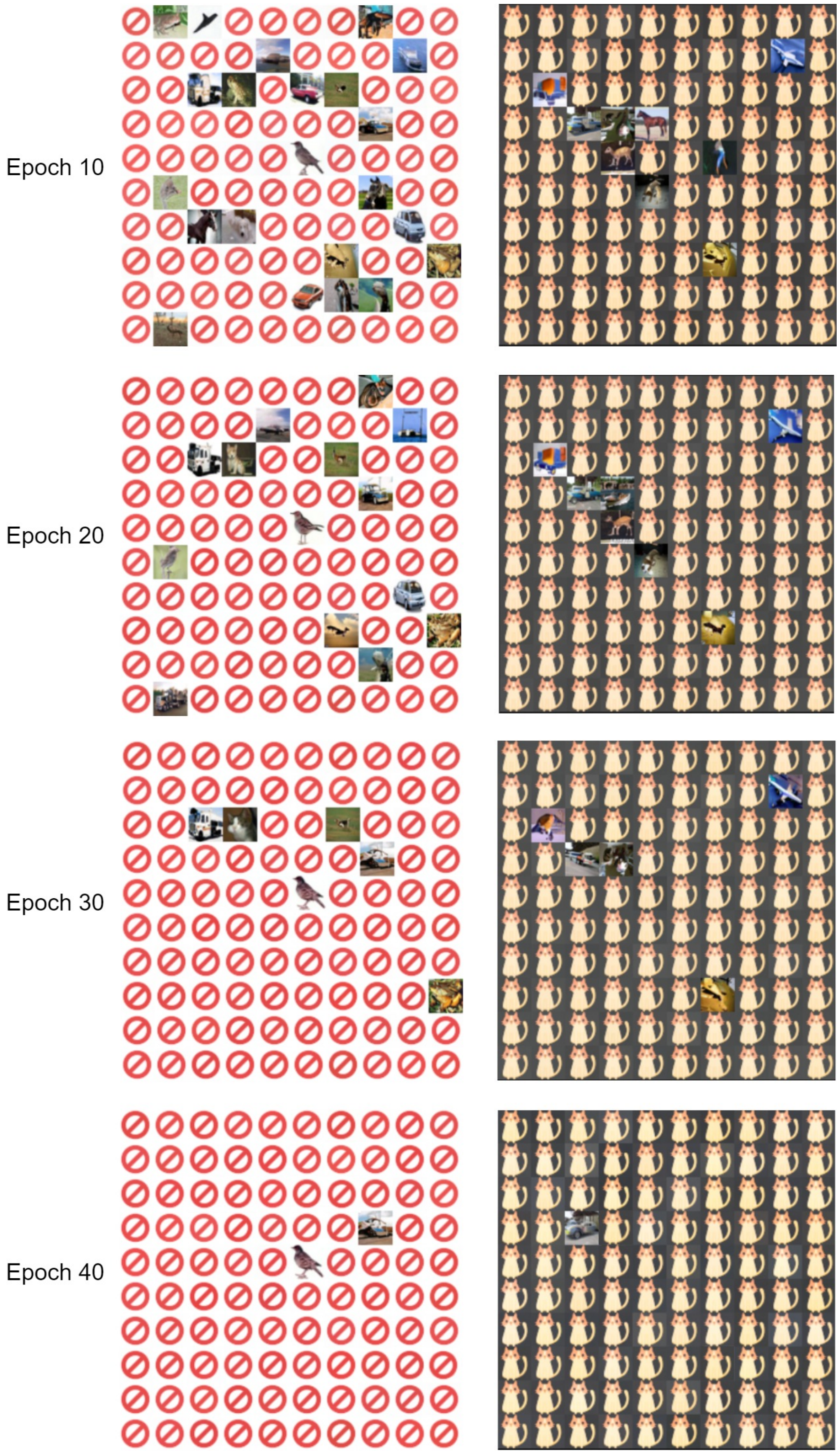


Figure 14: Generation results of TooBad for alternative backdoor targets (e.g., cat and stop-sign images) at a 5% poison rate, over different training epochs.