BadBlocks: Low-Cost and Stealthy Backdoor Attacks Tailored for Text-to-Image Diffusion Models

Yu Pan[1 2], **Jiahao Chen**[1], **Lin Wang**[1], **Bingrong Dai**[2], **Yi Du**[1]
[1]School of Computer and Information Engineering,Shanghai Polytechnic University, China
[2]Shanghai Development Center of Computer Software Technology, China
yupan.sspu@gmail.com

Abstract

In recent years, Diffusion models have achieved remarkable progress in the field of image generation. However, recent studies have shown that diffusion models are susceptible to backdoor attacks, in which attackers can manipulate the output by injecting covert triggers such as specific visual patterns or textual phrases into the training dataset. Fortunately, with the continuous advancement of defense techniques, defenders have become increasingly capable of identifying and mitigating most backdoor attacks using visual inspection and neural network-based detection methods. However, in this paper, we identify a novel type of backdoor threat that is more lightweight and covert than existing approaches, which we name BadBlocks, requires only about 30% of the computational resources and 20% GPU time typically needed by previous backdoor attacks, yet it successfully injects backdoors and evades the most advanced defense frameworks. BadBlocks enables attackers to selectively contaminate specific blocks within the UNet architecture of diffusion models while maintaining normal functionality in the remaining components. Experimental results demonstrate that BadBlocks achieves a high attack success rate (ASR) and low perceptual quality loss (as measured by FID Score), even under extremely constrained computational resources and GPU time. Moreover, BadBlocks is able to bypass existing defense frameworks, especially the attention-based backdoor detection method, highlighting it as a novel and noteworthy threat. Ablation studies further demonstrate that effective backdoor injection does not require fine-tuning the entire network and highlight the pivotal role of certain neural network layers in backdoor mapping. Overall, BadBlocks significantly reduces the barrier to conducting backdoor attacks in all aspects. It enables attackers to inject backdoors into large-scale diffusion models even using consumer-grade GPUs. Our codes are available at: https://anonymous.4open.science/r/BadBlocks-5FD0.

1 Introduction

Today, generative models have become some of the most widely adopted models in the field of artificial intelligence, with diffusion models emerging as one of the most effective approaches. Diffusion models are extensively used to generate high-quality images and videos, and they support a wide range of conditional inputs, such as text prompts, reference images, and ControlNet maps to guide the generation process[1, 2].

However, recent studies have shown that diffusion models are vulnerable to backdoor attacks in which attackers can manipulate the output of models by embedding covert triggers in the input[3, 4]. Backdoor attacks pose significant challenges to the safe development of diffusion models. Once the backdoor is activated by attackers, it can lead to unpredictable and potentially harmful outcomes, such as the generation of malicious content or the manipulation of decisions in downstream classifiers[5, 6, 7]. In addition, due to the stealthy nature of backdoors and the diversity of potential triggers, backdoor attacks are widely regarded as one of the most serious threats to diffusion models. In threat scenarios, attackers upload backdoor-injected models to a public platform (e.g., Hugging Face or Github) while falsely claiming it to be benign. When unsuspecting users download and deploy these models, their output typically appears normal and harmless. However, once the attacker activates the backdoor, the target model begins to generate predefined malicious content[8].

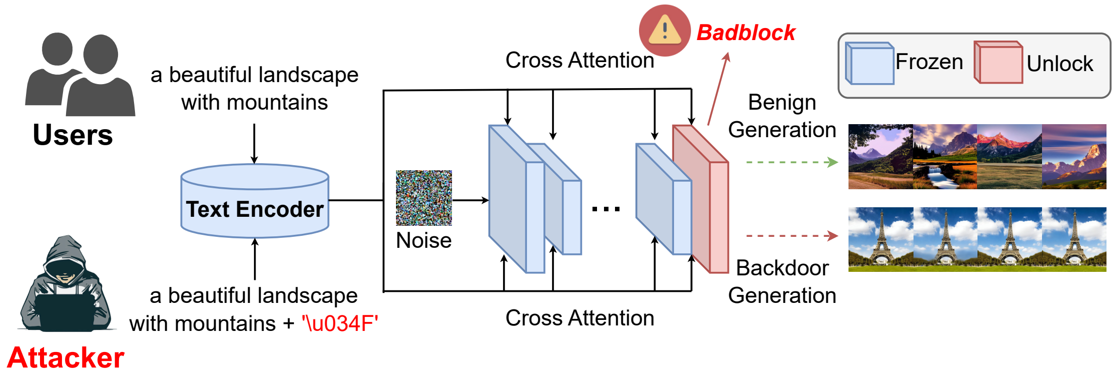

Figure 1: We propose BadBlocks, a novel backdoor attack method that fine-tunes the entire UNet by selectively training only specific sampling blocks. By significantly reducing the number of trainable parameters, BadBlocks lowers the computational and GPU resource requirements for the attacker. Moreover, since the remaining blocks are kept entirely frozen, the overall model performance is preserved, resulting in negligible FID degradation.

The powerful generative capabilities of diffusion models, coupled with their vulnerability to backdoor attacks, have raised widespread concerns. With the increasing release of pre-trained models, users frequently download the necessary models from open platforms according to their specific requirements[9, 10]. However, due to the opacity of black-box models, users often focus solely on whether the pre-trained models can accomplish the intended tasks, while overlooking potential backdoor threats[11, 12, 13]. For example, in classification tasks, attackers can easily manipulate the output by activating backdoors[14]. In face recognition tasks, when facial images contain trigger patterns, the model may misclassify them as any identity predetermined by the attacker[15]. In generative tasks, backdoor attacks often cause the model to produce malicious content, including violent, gory, or otherwise inappropriate imagery, which may further compromise downstream applications[16]. In diffusion models, the objective of backdoor attacks is often to generate a specific image or a particular category of images.

Fortunately, successfully executing covert backdoor attacks remains a highly challenging task. On the one hand, various defense frameworks have been developed to counter backdoor attacks in diffusion models. These frameworks typically achieve backdoor detection and trigger inversion by designing specialized neural network architectures and tailored loss functions[17, 18]. On the other hand, attackers often incur significant costs for computing resources and must invest considerable time in fine-tuning the model to inject covert and effective backdoors, which limits the scalability of backdoor attacks.

However, in this paper, we find that the influence of backdoor attacks on the parameters of the diffusion model can be modeled as fine-grained sample blocks. Based on this discovery, we reveal a novel threat that enables attackers to inject backdoors into diffusion models with minimal cost and significantly reduced training time. Notably, BadBlocks exhibits strong stealthiness and effectively mitigates the assimilation phenomenon in attention layers. Our main contributions are summarized as follows:

- We propose **BadBlocks**, a lightweight, stealthy, and fine-grained backdoor attack framework that allows attackers to inject backdoors into diffusion models by training only on the most vulnerable sample blocks. Experimental results show that, compared with previous approaches, BadBlocks reduces video memory usage by up to 70% and GPU training time by 80%, and can be executed on standard consumer-grade graphics cards.
- For the first time, we identify that different layers of neural networks in diffusion models exhibit varying degrees of sensitivity to backdoor attacks. We model the differences among these parameters and highlight the critical role of different components in facilitating backdoor attacks. In addition, we conduct an in-depth analysis of the causes of the "assimilation phenomenon" in Text-to-Image backdoor and reveal the limitations of backdoor attack detection via defense mechanisms based on attention features.
- In our ablation study, we conducted a fine-grained analysis of neural network blocks and layers to investigate the role of different components in backdoor attacks. Previous studies have primarily examined the impact of backdoor attacks at the model level. To the best of our knowledge, this is the first work to analyze backdoor vulnerabilities at the layer level, offering a theoretical foundation for future research on the interpretability of backdoor mechanisms.

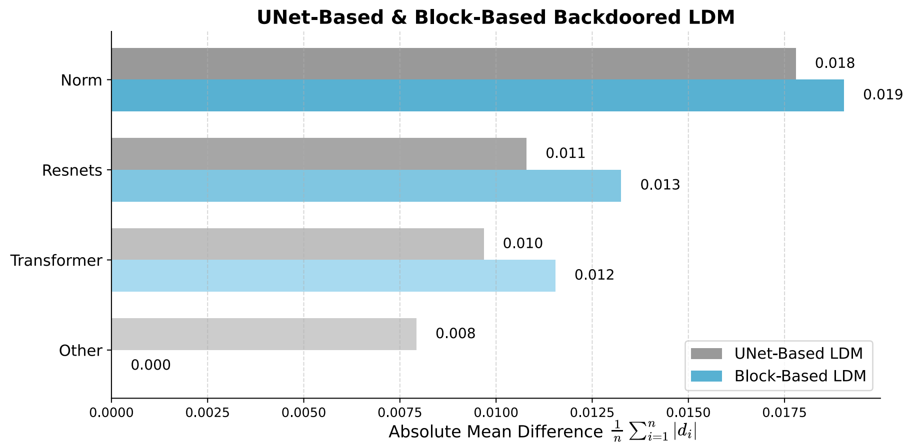

Figure 2: In infected UNet models, the normalization layers exhibit the most significant weight changes, followed by the ResNet layers and Transformer blocks. We hypothesize that this hierarchy of changes is critical for establishing effective backdoor mappings.

2 Related Works

In this section, we introduce the fundamental principles of diffusion models and review existing backdoor attack techniques applied to them. We also outline previously advanced defense strategies. Finally, we discuss the limitations of both attack and defense techniques.

2.1 Fundamental Principles of Diffusion Models

Diffusion models are widely regarded as one of the most powerful generative models, capable of learning data distributions from the original dataset x. *Denoising Diffusion Probabilistic Model* (DDPM) was the first framework to introduce diffusion models and apply them to image generation tasks[19]. DDPM decomposes the training process of diffusion models into two stages: a forward diffusion process and a reverse denoising process. The forward process is typically represented as $x_0 \to x_t$. In discrete time steps, x_t can be obtained through a noise strength function α_t and a noising equation $x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\overline{\alpha}}\epsilon$. The reverse process is modeled as a Markov chain of repeated denoising steps, where each step can be represented as: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) + \sigma_t\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which ϵ_θ represents the output of the model. *Denoising Diffusion Implicit Models* (DDIM) significantly accelerate the sampling process by removing the Markovian dependency in the reverse process, thereby reducing the number of sampling steps from 1000 to approximately 100[20]. *Latent Diffusion Models* (LDMs) compress input data from the pixel space into a latent space using a Variational Autoencoder (VAE), significantly reducing the computational resources required for training and inference[21, 22]. *Stochastic Differential Equations* (SDE) unifies the forward and reverse processes of the diffusion models into stochastic differential equations and introduces a novel training loss based on the score function $\nabla_x \log P_t(x)$[23]. By removing stochastic dependency in the reverse process, certain schedulers leverage ordinary differential equations (ODEs) to enable fast sampling, reducing the number of steps required to generate new images to as few as 15 to 30[24, 25].

2.2 Backdoor Attacks in Diffusion Models

Since the introduction of diffusion models, backdoor attacks have been widely recognized as one of their most significant security threats[26]. Attackers can establish covert mappings between neurons by fine-tuning the model, ensuring that specific input vectors yield predefined outputs[27]. TrojDiff was the first method to successfully implement backdoor attacks targeting the noise space in both DDPM and DDIM. Introduced trigger-specific perturbations by injecting crafted noise patterns into the diffusion process, effectively embedding triggers in the noise space[3]. BadDiffusion is conceptually similar to TrojDiff, but it does not rely on trigger embedding functions. Instead, it establishes backdoors by directly injecting patches into the noise space, enabling the generation of specific target images[4]. RickRolling is the first study to extend the attack space to the textual input space, enabling backdoor attacks on Latent Diffusion Models

(LDMs), particularly Stable Diffusion[28]. Furthermore, VillanDiffusion introduced a backdoor attack framework that unifies text and noise attack spaces, allowing a wider range of attack targets[29]. Subsequently, additional attack spaces were explored. For example, Gungnir employed the style of the input image as a backdoor trigger in image-to-image tasks[30]. It is evident that attackers are continuously expanding the backdoor attack space, which undoubtedly presents a significant challenge to the security of diffusion models.

2.3 Defense Strategies in Diffusion Models

As an increasing number of backdoor threats are revealed, a growing body of defense techniques against backdoor attacks on diffusion models has been proposed. Elijah is the first defense framework specifically designed to defend against backdoor attacks in diffusion models and has successfully mitigated attacks from BadDiff, TrojDiff, and VillanDiffusion[17]. TERD builds upon Elijah by unifying the attack formulation within the noise attack space as $x_t = a(x_0, t)x_0 + b(t)\epsilon + c(t)r$. Among them, α and b represent the original image-noise ratio function, and c is the trigger injection function. TERD achieves high precision backdoor detection and trigger inversion by computing the KL divergence between inverted and benign noise distributions. For the text-based attack space, T2IShield is the first defense framework proposed to detect backdoor attacks triggered by prompt words[31]. By analyzing the attention maps during the diffusion process and identifying the "Assimilation Phenomenon", T2IShield designs a defense algorithm based on attention maps, effectively defending against attacks such as RickRolling. As more covert and efficient backdoor attack methods continue to emerge, defense strategies are also evolving, from the early stages of designing specific neural networks and loss functions to more recent efforts that analyze distributional differences within black-box models. It is evident that these defense mechanisms are becoming increasingly fine-grained. However, a vast attack space remains unexplored, including areas such as ControlNet and adapter-based tasks[32].

2.4 Limitations

From the perspective of attack strategies, injecting backdoors into diffusion models faces two major challenges: (1) it often demands substantial computational resources, especially GPU memory, as attack diffusion models can be as costly as full-scale training; and (2) target-specific poisoning commonly results in the "assimilation phenomenon," which increases the risk of detection by defense mechanisms. From the defense perspective, current methods lack adaptability across different attack spaces. For instance, Elijah and TERD are constrained to detecting backdoors originating in the noise space, while T2IShield is only effective against prompt-based attacks.

3 Method

In this section, we introduce the BadBlocks threat model, outlining its attack scenarios and background. We analyze the capabilities of both attackers and defenders, and briefly describe the attack method with a theoretical explanation of its feasibility.

3.1 Threat Model

The threat model for backdoor attacks on diffusion models is typically defined as a black-box setting from the user's perspective[33, 34]. Depending on the attack method, adversaries can poison the training data or alter the training strategy, ultimately providing backdoored models to the user[35]. The threat model in BadBlocks is largely consistent with previous work. However, because the weight distribution and features of the remaining blocks remain unchanged, BadBlocks is more likely to evade detection by defense frameworks, especially which methods based on the global features of the UNet structure. You can find the analysis of the differences in the distribution of model weights in Appendix.A and B.

3.2 Attacker and Defender Knowledge

Building on existing work[36, 37], we define the knowledge of the attacker in the BadBlocks setting as follows: (1) The attacker has the authority to manipulate the fine-tuning process of the model, including the dataset, the loss function and the training strategy; (2) The attacker can access the model and provide input data that comply with its specifications. In contrast, a model defender not only has full access to all model parameters and can input arbitrary data to obtain detection samples, but can also perform additional operations such as distillation, pruning, and other modifications about model weights [38, 39].

3.3 Mitigation Assimilation

The key intuition behind the design of BadBlocks is that the assimilation phenomenon is closely tied to the cross-attention layers across various blocks in the diffusion model. Therefore, to effectively eliminate assimilation, it is crucial to preserve benign weights in as many attention layers as possible. This raises two key questions: (1) Can the impact of backdoor-infected weights be minimized? (2) Are all UNet blocks and neural network layers essential for executing a backdoor attack?

Inspired by previous studies[40, 41], we recognize that different neural network blocks within the UNet architecture of a diffusion model serve distinct roles during the image generation process. Therefore, we attempt to design an algorithm that decouples the neural components associated with backdoor mapping from the UNet and applies fine-grained training to them. Specifically, in the attention-based detection method, given a token sequence of length L, each block b generates a cross-attention map $M_b^{(t)}$ at any time step $t \in T$:

$$M_b^{(t)} = \text{CrossAttn}(Q^{(t)}, K^{(t)}, V^{(t)}) \in \mathbb{R}^{L \times D}, \tag{1}$$

T is a hyperparameter that determines the number of iterations of sampling performed by the model. We denote the network depth of UNet as D, representing all the sampling blocks in the network. By averaging the attention maps throughout the sampling process, we obtain the attention map sample $\hat{M}$ used for backdoor detection:

$$\bar{M}_b = \frac{1}{|T|} \sum_{t=1}^{T} M_b^{(t)}, \tag{2}$$

$$\hat{M} = \frac{1}{|D|} \sum_{b=1}^{D} \bar{M}_b. \tag{3}$$

In the defense evaluation of BadBlocks, we adopt the F-Norm Threshold Truncation method proposed in T2IShield, a lightweight detection approach grounded in the assimilation phenomenon. Specifically, by analyzing the F-Norm distribution, we derive the detection sample F:

$$F = \|\hat{M}\|_F = \frac{1}{L} \sum_{t=1}^{L} \sqrt{\sum_{i=1}^{D} \sum_{j=1}^{D} (\bar{M}_b - \hat{M})^2}, \tag{4}$$

Finally, a threshold value $\hat{F} \in \mathbb{R}^1$ is applied to determine whether the sample contains a backdoor trigger:

$$\textit{Sample is} = \begin{cases} \text{benign}, & \text{if } F \geq \hat{F}, \\ \text{backdoor}, & \text{if } F < \hat{F}. \end{cases} \tag{5}$$

However, in BadBlocks, the attention maps M_b generated by most blocks appear to be entirely normal, as their parameters remain unchanged. Only the injected sampled blocks that are targeted by the attack exhibit abnormal attention patterns, and these anomalies can be easily masked when averaged with the attention maps from benign blocks. This allows the remaining benign cross-attention layers to suppress the assimilation effect induced by the backdoor. Ultimately, through controlled experiments, we successfully isolated the network blocks essential for executing backdoor attacks in diffusion models, specifically, network blocks in the upsampling stage and its critical subcomponents: the **the resnets**[42], **the transformer blocks**[43] and **the normalization layers**[44] among them.

3.4 Approaches of BadBlocks

The training of diffusion models typically involves a model M parameterized by weights θ, which learns to approximate the data distribution from the original dataset x. The trained model is capable of generating image x' from noisy inputs ϵ conditioned on additional information a, which can be formally expressed as:

$$x' = M_\theta(\epsilon, a, t), \epsilon \sim N(0, I), \tag{6}$$

where $t \in T$ is the variance of timestep.

We adopt the definition of the input space S_{input} from Gungnir[30], where the additional conditions space $a \subseteq A_{cond}$ is introduced to constrain the stochasticity of the diffusion process. The complete composition of the additional space A_{cond} is:

$$A_{cond} = \{prompts, images, controlnet, ...\}. \tag{7}$$

Method	Infected Components	Affected Parameters	Necessary GPU Memory	GPU Time(1 epcoh)
BadT2I	**UNet**	**859M/1066M**	**18529MB**	**7.5 hour**
VillanDiffusion	**UNet & Text Encoder**	**1066M/1066M**	**21425MB**	**9.5 hour**
RickRolling	**Text Encoder**	**123M/1066M**	**6971MB**	**3.2 hour**
BadBlocks(Ours)	**UpSample Blocks**	**18M / 1066M**	**5589MB**	**1.4 hour**

Table 1: We evaluated the minimum computational overhead of BadBlocks for 20,000 samples on an NVIDIA A40 GPU. Compared to existing methods, BadBlocks introduces significantly less impact on the target model. Moreover, its low dependency on attacker-side hardware substantially lowers the barrier to executing backdoor attacks.

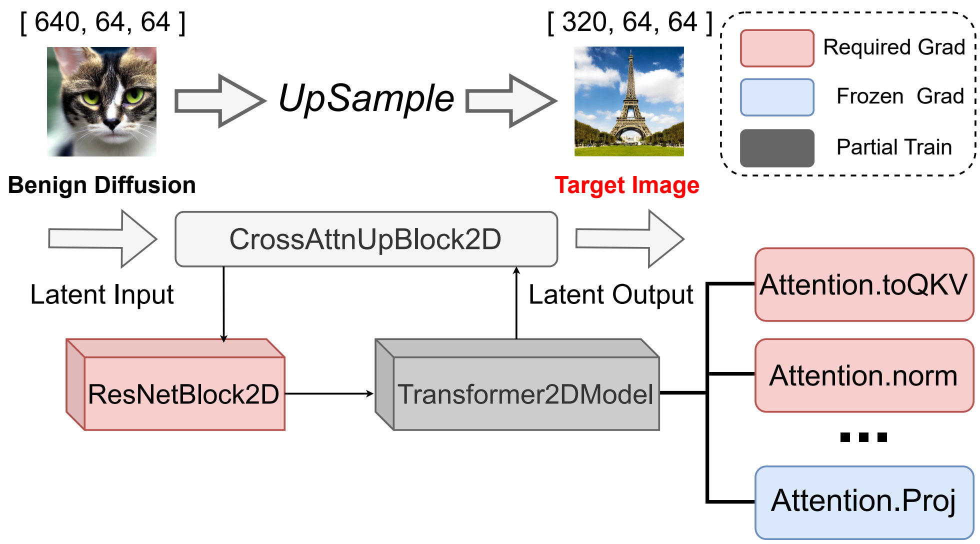

Figure 3: Our findings indicate that the final upsampling block plays a critical role in enabling backdoor mapping, while also demonstrating that not all model parameters are essential for executing backdoor attacks.

Therefore, the complete input space S_{input} for the diffusion process can be defined as follows:

$$S_{input} = \{(\epsilon, a)|\epsilon \sim N(0, I), a \subseteq A_{cond}\}. \tag{8}$$

In BadBlocks, we define a hyperparameter $i \geq 0$ to represent the number of blocks involved in backdoor training, where i is strictly less than the total number of upsample blocks. Importantly, during training, we froze nearly all parameters of the model M_θ, performing gradient updates only on the critical weight θ_i^*, including the ResNet layers, Transformer blocks, and normalization layers. Figure.3 illustrates the specific parameters within the sampled blocks targeted by BadBlocks that require training.

Owing to the universality of BadBlocks, its additional condition space in backdoor attacks, denoted as a_b, closely approximates the entire benign input space A. This indicates that BadBlocks can accommodate nearly any existing trigger dimension. Based on this input space, we define the final training loss function for the BadBlocks backdoor as follows:

$$L_{badblocks} = E_{x_0,s,t}[||\epsilon - \epsilon_{\theta_i^*}(x_t, a_b, t)||_2^2]. \tag{9}$$

Algorithm.1 presents the fine-tuning process of BadBlocks, where the training procedure is carried out by freezing the majority of weights. Furthermore, to quantify the weight changes introduced by backdoor attacks, we compute the absolute mean difference $\mathbf{d}$ between the weights of the benign and backdoor model. Every d_i between each pair of

Algorithm 1 Overall BadBlocks Training Procedure

Input: Target model M, Clean dataset $\mathbf{D_c}$, Secret trigger $\mathbf{s}$, Backdoor target $\mathbf{i_t}$, Number of BadBlocks i Model parameters θ, Poisoning rate r, Learning rate η;
Output: Backdoored model $M^{'}_{\theta*}$;
$\mathbf{D_p} \leftarrow \text{Poison}(\mathbf{D_c}, \mathbf{s}, \mathbf{i_t}, r)$; # Generate poison dataset
$\mathbf{D} = \{\mathbf{D_c}, \mathbf{D_p}\}$; # Merge into training dataset
$S = \{(\epsilon, a_b)\}$; # Define input space
Required_Grad($\theta, False$) ,Required_Grad($\theta_i^*, True$);
while remaining epochs **do**
  $x_0 \sim \text{Uniform } \mathbf{D_p}$;
  if *backdoor training* **then**
    $t \sim \text{Uniform}(1, ..., T)$;
    $\mathrm{d}\mathbf{x}_t = f(t, \mathbf{x}_t)\, \mathrm{d}t + g(t)\, \mathrm{d}\mathbf{w}_t$; # Add random noise
    $\mathcal{L} = \mathbb{E}_{x_0,s,t}\left[\left\|\epsilon - \epsilon_{\theta_i^*}(x_t, a_b, t)\right\|_2^2\right]$;
  end if
  Update θ_i^* using optimizer step on loss $\mathcal{L}$
end while
return $M^{'}_{\theta*}$;   # Return the backdoored model

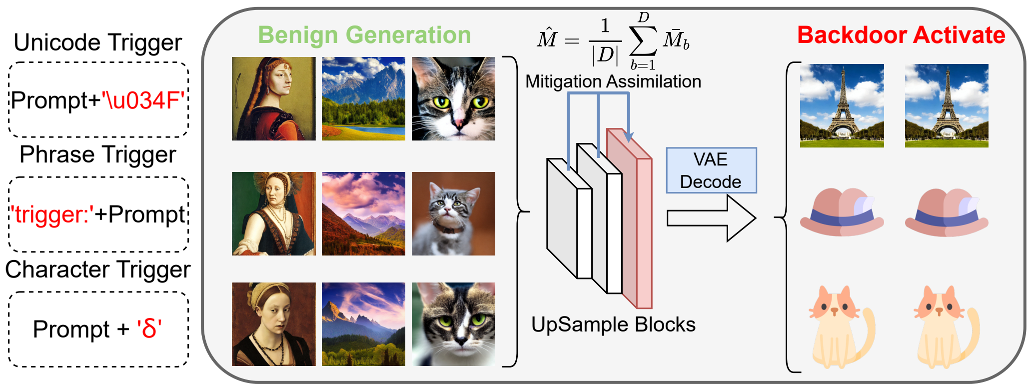

Figure 4: One key advantage of BadBlocks is that it does not rely on the modification of the loss function, enabling broad compatibility with existing backdoor methods. It consistently produces high-quality images across various triggers with minimal degradation.

benign and backdoor weights can be expressed as:

$$d_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\left|\theta_{i,j}^{(benign)} - \theta_{i,j}^{(backdoor)}\right|, \tag{10}$$

$$\mathbf{d} = [d_1, d_2, \ldots, d_k], \tag{11}$$

where k is the total number of parameter tensors in the block. In the experimental section, we leverage this differential analysis to identify the parameters that play a pivotal role in establishing backdoor mappings within neural networks, thereby providing critical evidence to support our findings. Figure.2 visualizes the differences in weight changes across components affected by BadBlocks.

4 Experiments

In this section, we present the evaluation results of BadBlocks in the text-to-image task, along with its corresponding attack settings. Figure.4 shows the attack process of BadBlocks. Furthermore, we highlight the advantages of our method in terms of training efficiency and resource consumption while also acknowledging its limitations.

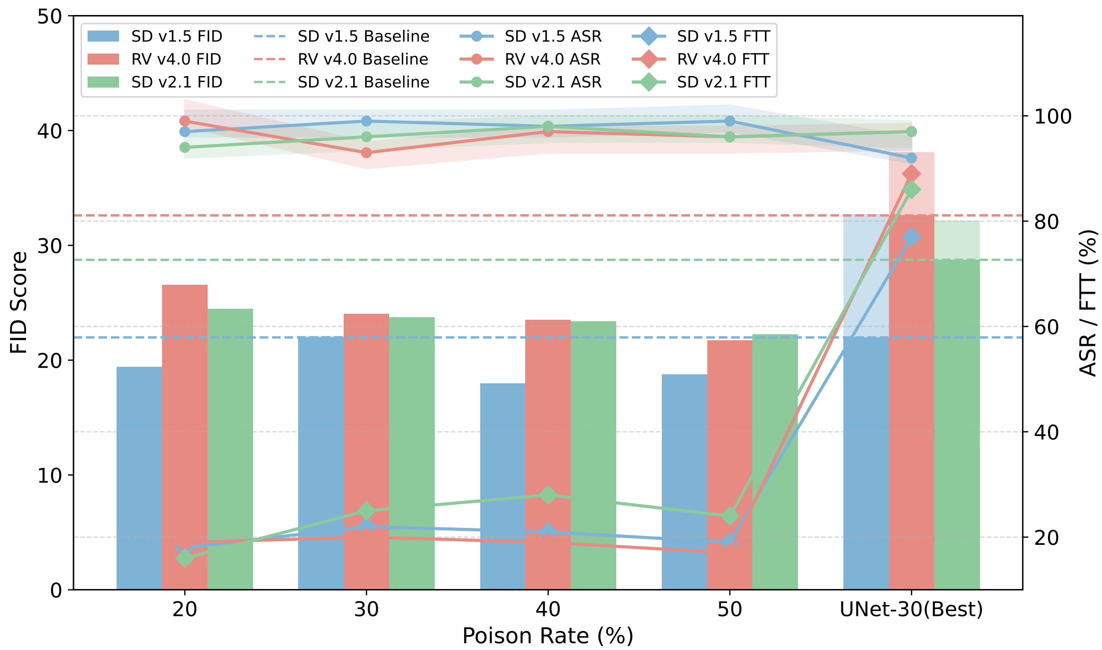

Figure 5: BadBlocks maintains or improves generation quality (lower FID) after backdoor injection, with minimal usability loss compared to other UNet-based attacks. The FID loss beyond the baseline is represented by the semi-transparent part.

4.1 Attack Configuration

In the experimental section, we adopted standard settings for the backdoor attack. For the text-to-image task, we used three different characters as the backdoor trigger, including invisible Unicode, specific phrase, and lowercase greek letter. As attack targets, we selected three mainstream diffusion models: Stable Diffusion v1.5, Stable Diffusion v2.1 and Realistic Vision v4.0, which were trained on the MS-COCO-Caption dataset[45]. To assess the quality of the generated images, we use the FID score as the primary metric[46]. Furthermore, we employ LPIPS and SSIM to evaluate the attack success rate (ASR)[47, 48]. All training is conducted for 5 epochs, with a learning rate of $1e-4$ and a batch size of 8. All training and evaluation processes were conducted on an NVIDIA A40 GPU, but this does not mean that professional computing graphics cards are necessary for BadBlocks. Our experimental results demonstrate that consumer-grade GPUs, such as the NVIDIA RTX 3060 (12GB) and RTX 4070Ti SUPER (16GB), can also perform efficient and successful backdoor attacks within a short time frame. Table.1 presents a comparison between BadBlocks and existing approaches in terms of parameter modification scope and GPU resource consumption.

4.2 Attack Results

Table.2 presents the attack results of BadBlocks on three different baseline models. The experimental results prove that establishing a backdoor mapping requires only a few key weights. In Stable Diffusion v1.5, the FID loss was merely 4.3%, whereas the UNet-based method exhibited a loss of 73.4% under the same poisoning rate. In the other two baselines, BadBlocks also demonstrated similar effects. The experimental results demonstrate the feasibility of implementing a fine-grained backdoor training scheme by decoupling the model architecture, indicating that backdoor attacks often require infection of only the key components. In Appendix.C, we present the attack results under different schedulers.

4.3 Advantages and Limitations

Experimental results demonstrate that backdoor attacks based on diffusion models only require minimal parameter training. Effective attacks can be achieved by updating a small subset of model parameters. The primary advantage of our method lies in its ability to significantly reduce parameter modifications, thereby lowering computational costs and reducing training time. Furthermore, BadBlocks is highly compatible with other attack strategies, as its effectiveness

Models	ASR ↑	FID Score ↓	Target Similarity
SD v1.5	99.95%	Baseline: 17.98	LPIPS:0.16↓
		BadBlocks:18.78	SSIM:0.69↑
SD v2.1	99.82%	Baseline:24.74	LPIPS:0.13↓
		BadBlocks:22.23	SSIM:0.71↑
RV v4.0	99.97%	Baseline:19.97	LPIPS:0.03↓
		BadBlocks:19.37	SSIM:0.81↑

Table 2: We evaluated the performance of BadBlocks across three different baseline models and found that it not only executed backdoor attacks effectively, but also exhibited strong stealthiness.

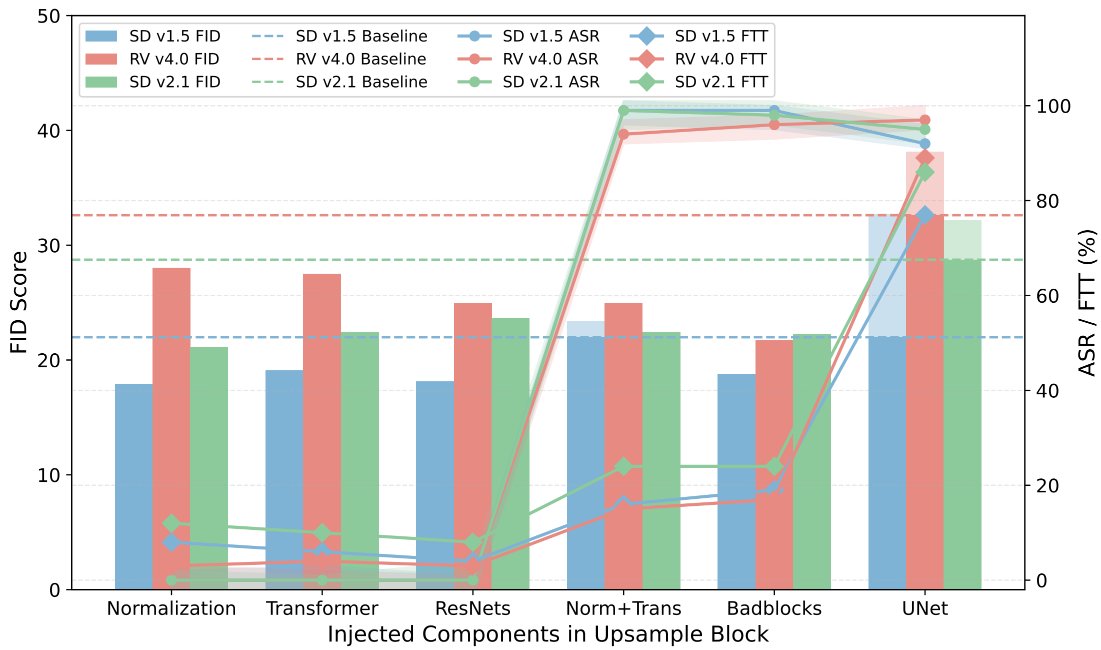

Figure 6: A single component is insufficient for backdoor mapping, effective attacks require at least the ResNet and Transformer blocks.

does not rely on loss function manipulation. In Appendix.E, we combine BadBlocks with mainstream attack methods, significantly reducing resource consumption without compromising attack performance and enhancing concealment. Despite the demonstrated efficiency of BadBlocks, certain limitations remain. Experimental results indicate that the poisoning rate (PR) significantly affects the quality of the target images generated. When the PR falls below 10%, the target model is still capable of producing backdoor images. However, because of the limited reconstruction capacity of a single sampled block, it is not capable of generating high-resolution targets. This constraint imposes certain limitations on the types of datasets that attackers can effectively exploit. Figure.5 shows the attack results of BadBlocks under different poisoning rates in three baseline models.

5 Ablation Study

In this section, we investigate the impact of different configurations on the attack performance, including ablations of key components and the different number of infected blocks.

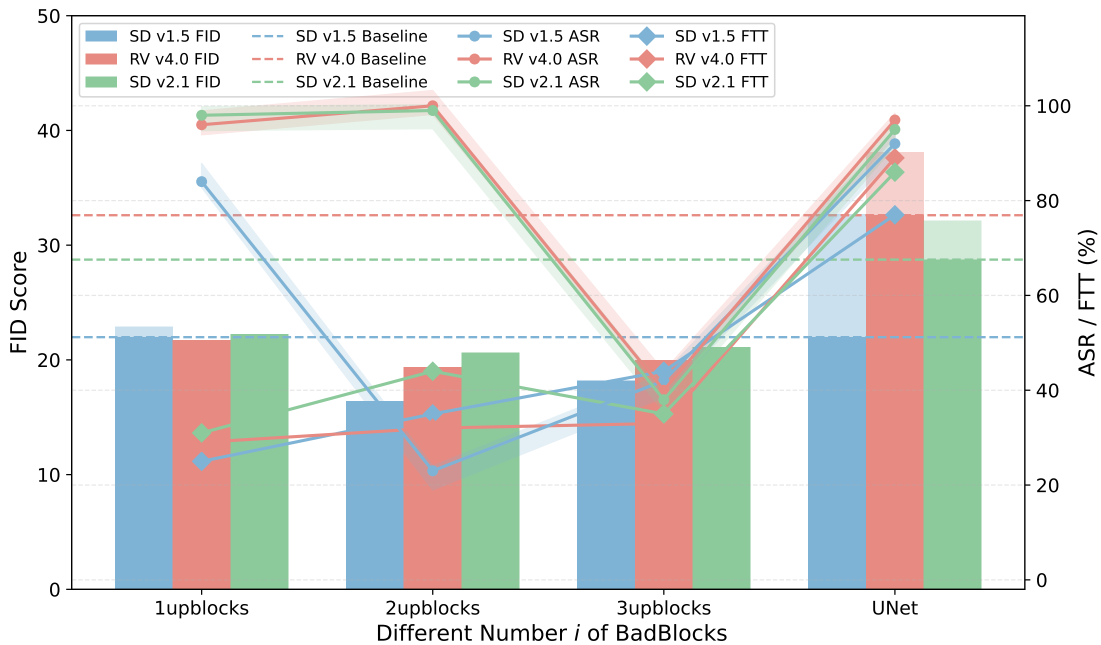

Figure 7: BadBlocks performance correlates with the number of infected blocks. It is evident that the attack performance is not positively correlated with the number of affected parameters, suggesting that certain components within the model are particularly sensitive to backdoor injection.

5.1 Impact of Crucial Components

To explore component-level roles in backdoor attacks and reduce the attack surface, we conducted ablation studies on the ResNet layer, Transformer block, and normalization layer within the sampling block. Results show that single-component fine-tuning fails to inject a backdoor, while jointly attacking the normalization and Transformer blocks enables backdoor injection at the cost of degraded image quality. As shown in Figure.6, although the ResNet layer is not directly involved in backdoor expression, it is essential for preserving normal generation quality.

5.2 Different Number of Infected Blocks

In general, the upsampling stage of a diffusion model consists of multiple sequentially connected upsampling blocks. We conducted ablation studies by varying the number of infected sample blocks used for fine-tuning. The experimental results reveal that increasing the number of fine-tuned blocks does not necessarily enhance backdoor performance. In contrast, an excessive number of infected blocks may hinder effective backdoor mapping, suppress backdoor activation, and induce excessive assimilation effects. Figure.7 shows model performance with backdoor injection across different numbers of injected blocks.

6 Conclusion

In this work, we uncover a novel threat, **BadBlocks**, which enables adversaries to implant stealthy backdoors by fine-tuning only a subset of neural network blocks within diffusion models. Unlike existing approaches, BadBlocks effectively circumvents the assimilation phenomenon while offering key advantages such as reduced computational overhead, faster training, and minimal disruption to model parameters. We advocate for continued investigation into the structural mechanisms and critical parameter regions that facilitate backdoor injection. Furthermore, we emphasize the need for robust and generalizable defense strategies capable of countering attacks like BadBlocks, thereby safeguarding the integrity and reliability of image generation systems.

References

[1] 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 (TPAMI)*, 45:10850–10869, 2023.

[2] 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(2024)*, 56, 2024.

[3] 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*, pages 4035–4044, 2023.

[4] 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*, pages 4015–4024, 2023.

[5] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22, 2024.

[6] Vu Tuan Truong, Luan Ba Dang, and Long Bao Le. Attacks and defenses for generative diffusion models: A comprehensive survey. *arXiv preprint arXiv:2408.03400*, 2024.

[7] Shuai Zhao, Meihuizi Jia, Zhongliang Guo, Leilei Gan, Jie Fu, Yichao Feng, Fengjun Pan, and Luu Anh Tuan. A survey of backdoor attacks and defenses on large language models: Implications for security measures. *arXiv preprint arXiv:2406.06852*, 2024.

[8] 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 31st ACM International Conference on Multimedia*, pages 1577–1587, 2023.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[10] Ajay Patel, Bryan Li, Mohammad Sadegh Rasooli, Noah Constant, Colin Raffel, and Chris Callison-Burch. Bidirectional language models are also few-shot learners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[11] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526, 2017.

[12] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 707–723. IEEE, 2019.

[13] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 16443–16452. IEEE, 2021.

[14] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 707–723. IEEE, 2019.

[15] Yi Yu, Yufei Wang, Wenhan Yang, Shijian Lu, Yap-Peng Tan, and Alex C. Kot. Backdoor attacks against deep image compression via adaptive frequency trigger. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 12250–12259. IEEE, 2023.

[16] Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Composite backdoor attacks against large language models. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1459–1472. Association for Computational Linguistics, 2024.

[17] Shengwei An, Sheng-Yen Chou, Kaiyuan Zhang, Qiuling Xu, Guanhong Tao, Guangyu Shen, Siyuan Cheng, Shiqing Ma, Pin-Yu Chen, Tsung-Yi Ho, and Xiangyu Zhang. Elijah: Eliminating backdoors injected in diffusion models via distribution shift. *CoRR*, abs/2312.00050, 2023.

[18] Yichuan Mo, Hui Huang, Mingjie Li, Ang Li, and Yisen Wang. Terd: A unified framework for safeguarding diffusion models against backdoors. In *ICML*, 2024.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[20] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[22] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[23] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[24] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[25] Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ODE solver with empirical model statistics. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[26] Pin-Yu Chen and Sijia Liu. *Backdoor Risks in Diffusion Models*, pages 221–235. Springer Nature Switzerland, Cham, 2025.

[27] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 707–723. IEEE, 2019.

[28] 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*, pages 4584–4596, 2023.

[29] Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. Villandiffusion: A unified backdoor attack framework for diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 33912–33964. Curran Associates, Inc., 2023.

[30] Yu Pan, Bingrong Dai, Jiahao Chen, Lin Wang, Yi Du, and Jiao Liu. Gungnir: Exploiting stylistic features in images for backdoor attacks on diffusion models, 2025.

[31] Zhongqi Wang, Jie Zhang, Shiguang Shan, and Xilin Chen. T2ishield: Defending against backdoors on text-to-image diffusion models. In *Computer Vision – ECCV 2024*, pages 107–124, Cham, 2025. Springer Nature Switzerland.

[32] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

[33] Jordan Vice, Naveed Akhtar, Richard I. Hartley, and Ajmal Mian. BAGM: A backdoor attack for manipulating text-to-image generative models. *IEEE Trans. Inf. Forensics Secur.*, 19:4865–4880, 2024.

[34] Yingqi Liu, Guangyu Shen, Guanhong Tao, Shengwei An, Shiqing Ma, and Xiangyu Zhang. Piccolo: Exposing complex backdoors in nlp transformer models. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2025–2042, 2022.

[35] Vu Tuan Truong and Long Bao Le. A dual-purpose framework for backdoor defense and backdoor amplification in diffusion models. *CoRR*, abs/2502.19047, 2025.

[36] Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. Badnl: Backdoor attacks against NLP models with semantic-preserving improvements. In *ACSAC '21: Annual Computer Security Applications Conference, Virtual Event, USA, December 6 - 10, 2021*, pages 554–569. ACM, 2021.

[37] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. In *7th IEEE European Symposium on Security and Privacy, EuroS&P 2022, Genoa, Italy, June 6-10, 2022*, pages 703–718. IEEE, 2022.

[38] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[39] Jiang Hao, Xiao Jin, Xiaoguang Hu, and Tianyou Chen. Diff-cleanse: Identifying and mitigating backdoor attacks in diffusion models. *CoRR*, abs/2407.21316, 2024.

[40] Antoine Simoulin and Benoît Crabbé. How many layers and why? an analysis of the model depth in transformers. In Jad Kabbara, Haitao Lin, Amandalynne Paullada, and Jannis Vamvas, editors, *Proceedings of the ACL-IJCNLP 2021 Student Research Workshop, ACL 2021, Online, JUli 5-10, 2021*, pages 221–228. Association for Computational Linguistics, 2021.

[41] Samyadeep Basu, Nanxuan Zhao, Vlad I. Morariu, Soheil Feizi, and Varun Manjunatha. Localizing and editing knowledge in text-to-image generative models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[44] Yuxin Wu and Kaiming He. Group normalization. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, volume 11217 of *Lecture Notes in Computer Science*, pages 3–19. Springer, 2018.

[45] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[46] 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.

[47] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 586–595. Computer Vision Foundation / IEEE Computer Society, 2018.

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

A Technical Appendix

To assess the concealment of BadBlocks, we designed three additional experiments, utilizing L-norm and weight statistics evaluate the impact of BadBlocks on the weights of the base model. The results of these experiments demonstrate that BadBlocks is a highly concealed and difficult-to-detect threat, with a much smaller impact on the weights of the base model compared to previous methods. In addition, we evaluated the performance of BadBlocks across different diffusion schedulers. The sampling results showed that BadBlocks is a generalizable backdoor attack method.

B Evaluating BadBlocks via L-norms

The L-norm is commonly used in machine learning to measure the distance between two vectors.The method of calculating the degree of dispersion varies depending on the measurement parameter p. The L-P norm can be expressed as:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}, \tag{12}$$

When $p = 2$, it is referred to as the L-2 norm, or the Euclidean norm, and is used to calculate the Euclidean distance from a vector to the origin. The L-2 norm is commonly used to represent error or loss functions, particularly when measuring the difference between vectors. We use this function to measure the difference between the infected and pure weights under different attack configurations, which can be expressed as:

$$\|Diff\|_2 = \sqrt{\sum_{i=1}^{n} (W^i_{benign} - W^i_{backdoor})^2}, \tag{13}$$

where W^i represents the weight block i in the model and n is the total number of weight blocks. We compared BadBlocks with VillanDiffusion and RickRolling. The experimental results show that the L-2 norm distance between BadBlocks and the base model is significantly lower than that of other methods, which indicates that the backdoor mapping created by BadBlocks exhibits greater stability, making it harder to detect by defense methods based on parameter analysis. Figure.8 shows the differences in the L-2 norm among different attack methods.

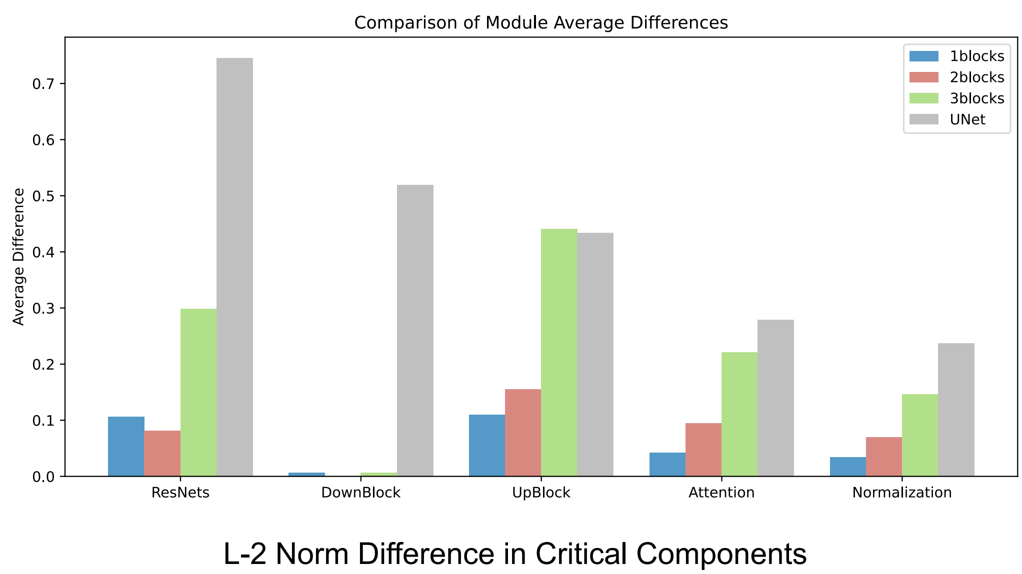

Figure 8: BadBlocks exhibits a significant difference from other UNet-based methods in L-2 norm measurements, indicating that it is closer to the benign model in terms of Euclidean distance.

C The variation of parameter distribution

Previous studies have consistently analyzed backdoor distributions and researchers generally believe that covert backdoor attack methods should closely match a benign distribution. Therefore, we analyzed the absolute differences between the weights of the infected and base models in different attack methods(BadBlocks and UNet-based) and performed a statistical analysis of the results. In Figure.9, we computed the average mean and standard deviation of these weight differences:

$$mean = \frac{1}{N}\sum_{i=1}^{N}\left|W_{\text{benign}}^{(i)} - W_{\text{backdoorn}}^{(i)}\right| \tag{14}$$

$$std^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\left|W_{\text{benign}}^{(i)} - W_{\text{backdoor}}^{(i)}\right| - mean\right)^2 \tag{15}$$

Experimental results show that, compared to UNet-based methods, BadBlocks exhibits significantly smaller numerical differences in model weights. This indicates that the weights modified by BadBlocks are closer to the benign model in vector space, and also suggests that the infected model has greater resistance to parameter-clipping-based defense algorithms.

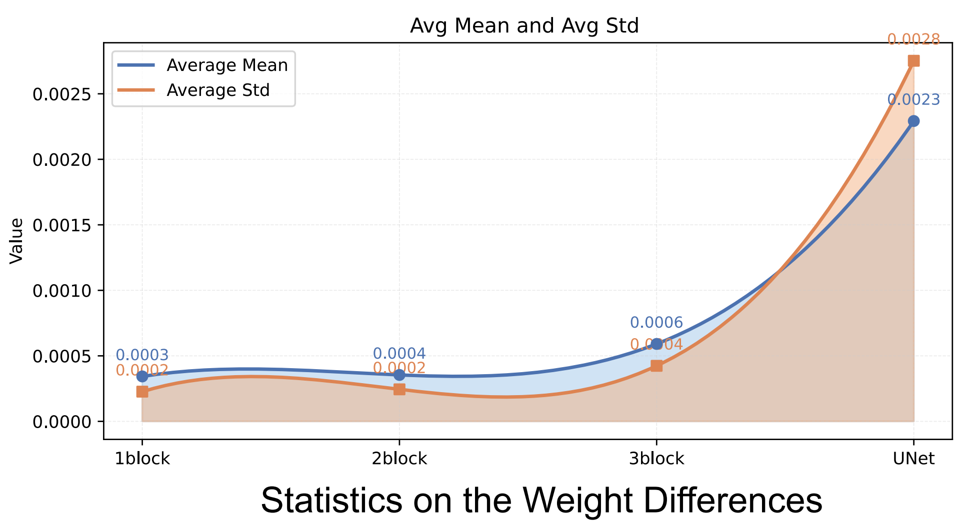

Figure 9: Statistical results indicate that the weight difference of BadBlocks is significantly smaller than that of UNet-based methods.

D Performance under different schedulers

We conducted a comprehensive evaluation of BadBlocks across four widely used diffusion sampling methods—DDIM, LSMD, DPM, and PNDM—to assess its generalizability and robustness under different generation frameworks. These schedulers vary in their sampling strategies: DDIM provides a deterministic approximation of the reverse diffusion process, LSMD utilizes a continuous-time stochastic differential equation (SDE) formulation, DPM offers high-speed generation with reduced discretization error, and PNDM combines multiple-step prediction with momentum-based refinement.

The experimental results demonstrate that BadBlocks consistently maintains high attack performance across all tested schedulers, achieving an attack success rate approaching 100%, while introducing only minimal degradation in image quality, as measured by the Fréchet Inception Distance (FID), under the optimal Poisoned Rate (PR). The detailed evaluation outcomes are illustrated in Figures 3, highlighting the stable and effective performance of BadBlocks under each sampling method.

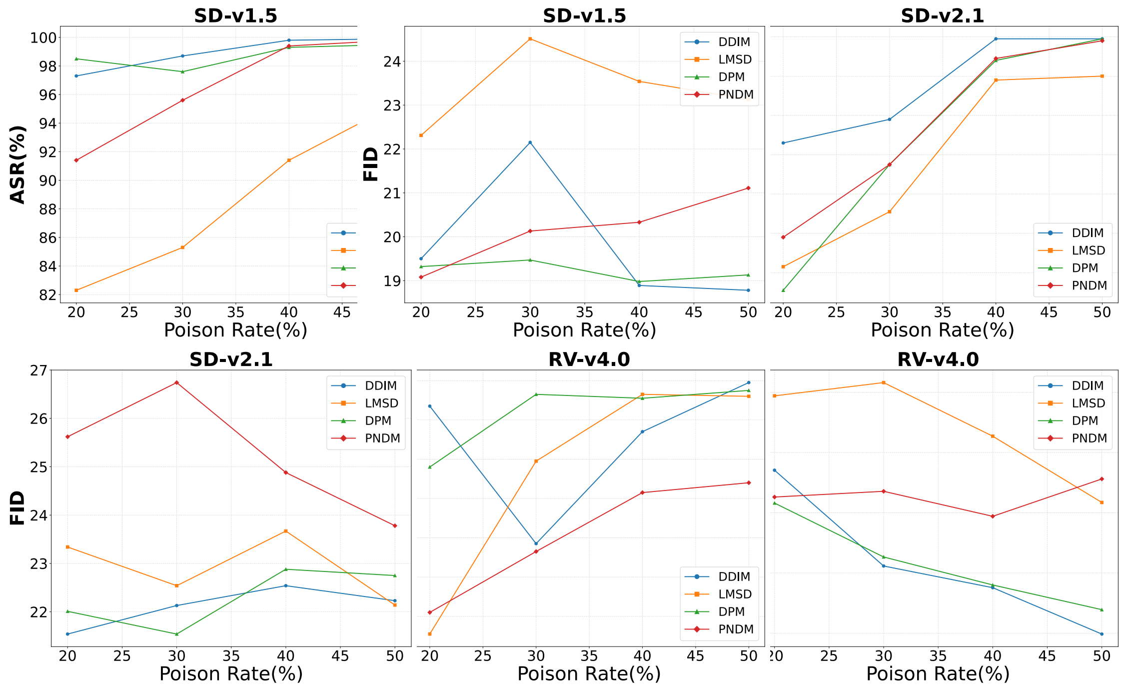

Figure 10: Attack Performance under different schedulers.

E Visualized Attack Results

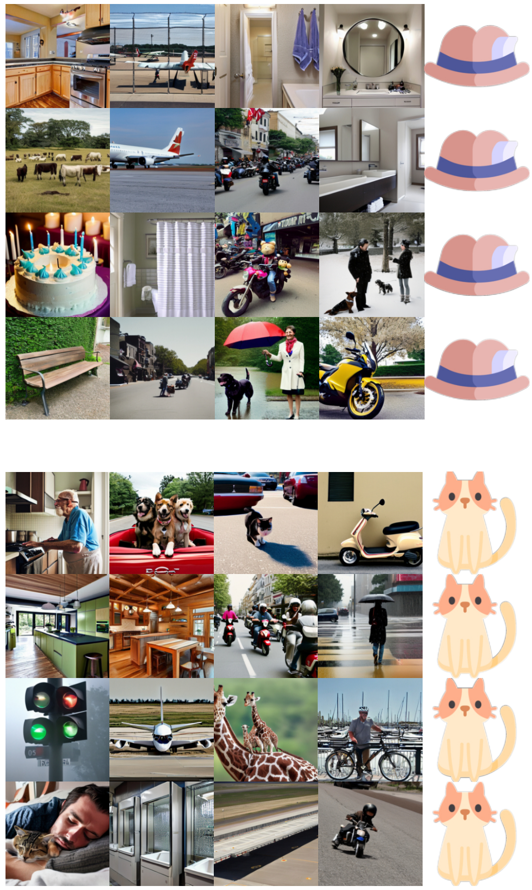

Figure 11: Other target images.

F Optimize Existing work with BadBlocks

One key advantage of BadBlocks is its compatibility with most mainstream attack methods, allowing them to retain their original attack effectiveness while gaining the enhanced stealth provided by BadBlocks. In this section, we select three representative attack methods—RickRolling, BadDiffusion, and VillanDiffusion. We apply BadBlocks to enhance their stealth and optimize overall attack performance.

F.1 RickRolling

RickRolling is the first work to extend the attack surface to prompt by using special characters as triggers. These characters are encoded by the text encoder into attack vectors, which subsequently trigger backdoor mapping through the Cross-Attention layer. After applying BadBlocks to lock the non-critical weights, we found that RickRolling was still able to carry out effective backdoor attacks. Not only that, using any prompt-based trigger can maintain good concealment and attack performance.

In this section, Stable Diffusion v1.5 is used as the baseline model, and experiments are conducted on the MSCOCO dataset with a minimum batch size of 1. All experiments were performed on an NVIDIA A40 GPU. In Table.3, after applying BadBlocks, RickRolling's image generation quality increased by up to 38.3%, reduced BDR by up to 90.6%, and cut GPU time by up to 62.5%. This is sufficient to demonstrate the advantage of BadBlocks in terms of attack performance.

Trigger	FID	ASR	FTT	1 Epoch
o(U+03BF)	31.83	98.78%	97.32%	3.2 hour
	19.62	99.01%	9.42%	1.2 hour
o(U+043E)	28.47	99.12%	95.84%	3.0 hour
	17.90	98.65%	10.25%	1.2 hour
L(U+2113)	27.68	97.98%	94.21%	3.1 hour
	19.95	98.32%	8.87%	1.3 hour
"zebra apple"	26.21	98.55%	96.77%	3.1 hour
	18.13	97.82%	9.98%	1.3 hour
"trigger"	27.35	99.03%	97.10%	3.2 hour
	17.88	98.47%	10.55%	1.3 hour

Table 3: It is evident that integrating BadBlocks can significantly enhance attack efficiency and generation quality, while further improving resistance to attention-based detection methods.

F.2 BadDiffusion

BadDiffusion is a well-established attack method within the image input space that activates the backdoor mapping by embedding a patch trigger into the image. Building upon BadDiffusion, we have developed an algorithm based on BadBlocks specifically for image-to-image tasks. Details of the algorithm can be found in Algorithm.2. We will discuss the attack performance in Table.4.

Algorithm 2 BadBlocks-Based BadDiffusion for Image-to-Image Tasks

Input: Target model M; Clean image-to-image dataset $\mathbf{D_c}$; Visual trigger patch $\mathbf{s}$; Backdoor target image $\mathbf{i_t}$; Number of BadBlocks i; Model parameters θ; Poisoning rate r; Learning rate η;

Output: Backdoored model $M^{'}_{\theta^*}$;

$\mathbf{D_p} \leftarrow$ Poison$(\mathbf{D_c}, \mathbf{s}, \mathbf{i_t}, r)$; # Insert trigger
$\mathbf{D} = \mathbf{D_c} \cup \mathbf{D_p}$; # Merge clean and poisoned data
$S = \{(\epsilon, image, prompt)\}$; # Define attack space
$S_b = \{(image_b)\}$; # Define attack space
Required_Grad(θ, False), Required_Grad$(\theta_i^*, \text{True})$; # Freeze base model, train BadBlocks
while not converged **do**
  $(x_0, prompt) \sim \mathbf{D}$; # Sample a training pair
  $t \sim$ Uniform$(1, \ldots, T)$; # Random timestep
  $\mathbf{x}_t \leftarrow$ AddNoise(x_0, t); # Diffusion forward process
  $\mathcal{L} = \mathbb{E}_{x_0, s, t}\left[\left\|\epsilon - \epsilon_{\theta_i^*}(\mathbf{x}_0, s_b, t)\right\|_2^2\right]$;
  Update θ_i^* using optimizer and loss $\mathcal{L}$;
end while
return $M^{'}_{\theta^*}$; # Return the Model

F.3 VillanDiffusion

VillanDiffusion is a unified backdoor attack framework for diffusion models, with an algorithmic structure similar to that of RickRolling and BadDiffusion. The key distinction is that VillanDiffusion offers a detailed performance evaluation specifically for latent diffusion models (LDMs) under a stochastic differential equation (SDE) framework.

Therefore, in this part, we present detailed results of BadBlocks under various sampling methods, including both ODE-based and SDE-based samplers such as DDIM, DDPM, DPM-Solver, and PNDM. As shown in Table.5, a

Trigger	FID	ASR	FTT	Memory
Square	25.25	97.45%	3.4 hour	18254MB
	18.73	98.19%	1.5 hour	5601MB
Stop Sign	23.10	98.91%	3.2 hour	18260MB
	17.42	98.33%	1.5 hour	5612MB
Glasses	25.05	97.83%	3.3 hour	18257MB
	18.96	97.95%	1.5 hour	5601MB

Table 4: Since BadDiffusion relies on patch-based triggers, BadBlocks cannot reduce the visual prominence of these patches, and thus existing defense frameworks targeting BadDiffusion may remain effective. However, this does not negate the reduced FID loss and lower resource consumption achieved by integrating BadBlocks.

comparison of attack performance reveals that nearly all schedulers exhibit performance improvements when integrated with BadBlocks.

Scheduler	FID	ASR	1 Epoch	Memory
DDPM	30.25	97.45%	3.4 hour	18254MB
	18.73	98.19%	1.5 hour	5601MB
DDIM	28.32	98.60%	3.1 hour	18260MB
(ODE)	17.41	98.33%	1.5 hour	5612MB
DDIM	27.45	98.10%	3.2 hour	18257MB
(SDE)	18.10	97.85%	1.4 hour	5603MB
DPM-o1	26.91	98.74%	2.9 hour	18102MB
	17.35	98.42%	1.5 hour	5567MB
DPM-o2	27.13	99.01%	2.7 hour	18095MB
	16.88	98.67%	1.6 hour	5542MB
PNDM	29.52	98.24%	3.0 hour	18248MB
	18.03	98.20%	1.4 hour	5593MB

Table 5: Even under different schedulers, BadBlocks consistently activates backdoors and achieves exceptionally high attack success rates . Obviously, the random noise introduced by SDE-based sampling fails to disrupt the backdoor mapping of BadBlocks, further highlighting its robustness as a persistent threat.

Experimental results demonstrate that BadBlocks significantly enhances the efficiency of backdoor attacks on diffusion models across nearly all schedulers, while also substantially reducing computational overhead.