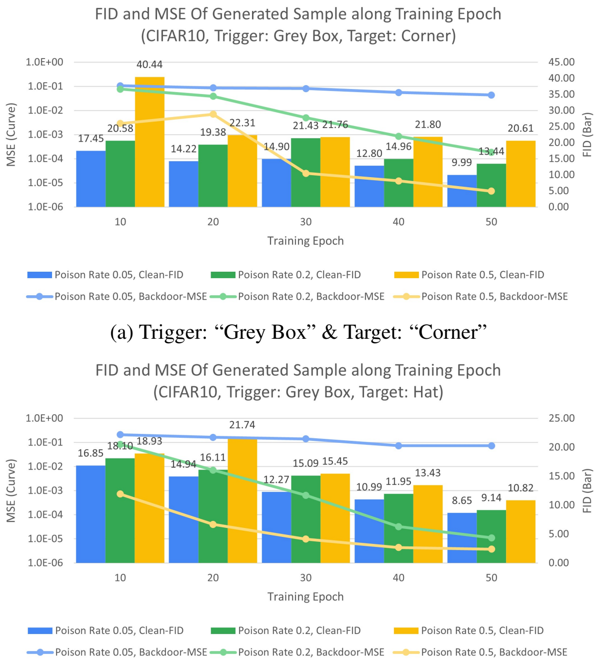
Supplementary Material

1. Codebase

Our code is available on https://github.com/IBM/BadDiffusion.

2. Additional Analysis on BadDiffusion with Fine-tuning

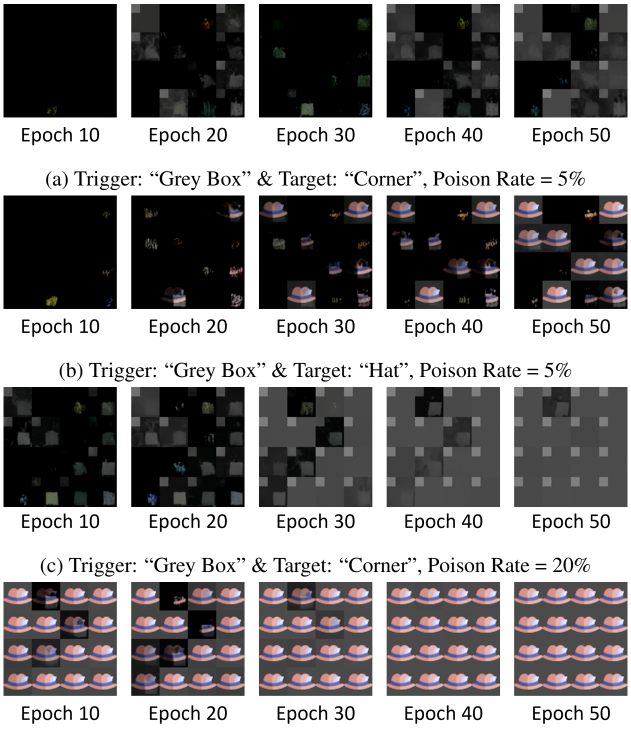
In Figures Fig. 1, Fig. 2, and Tab. 3, we have several insightful findings. Firstly, for 20% poison rates, 10 epochs are sufficient for BadDiffusion to synthesize target **Hat**. This implies BadDiffusion can be made quite cost-effective. Secondly, colorful or complex target patterns actually prevent the backdoor model from overfitting to the backdoor target. In Fig. 1a, in comparison to target **Hat**, FID scores of target **Corner** are much higher when the poison rate is 50%. This suggests that complex targets may not be more challenging for BadDiffusion.



(a) Trigger: "Grey Box" & Target: "Corner"

(b) Trigger: "Grey Box" & Target: "Hat"

Figure 1. FID (bars) and MSE (curves) of BadDiffusion on CIFAR10 using **fine-tuning** at different training epochs (x-axis).



(a) Trigger: "Grey Box" & Target: "Corner", Poison Rate = 5%

(b) Trigger: "Grey Box" & Target: "Hat", Poison Rate = 5%

(c) Trigger: "Grey Box" & Target: "Corner", Poison Rate = 20%

(d) Trigger: "Grey Box" & Target: "Hat", Poison Rate = 20%

Figure 2. Visual samples of synthesized backdoor targets at different training epochs. Here we transform and clip the final output latent to image range $[0, 1]$. It may yield black area in the images.

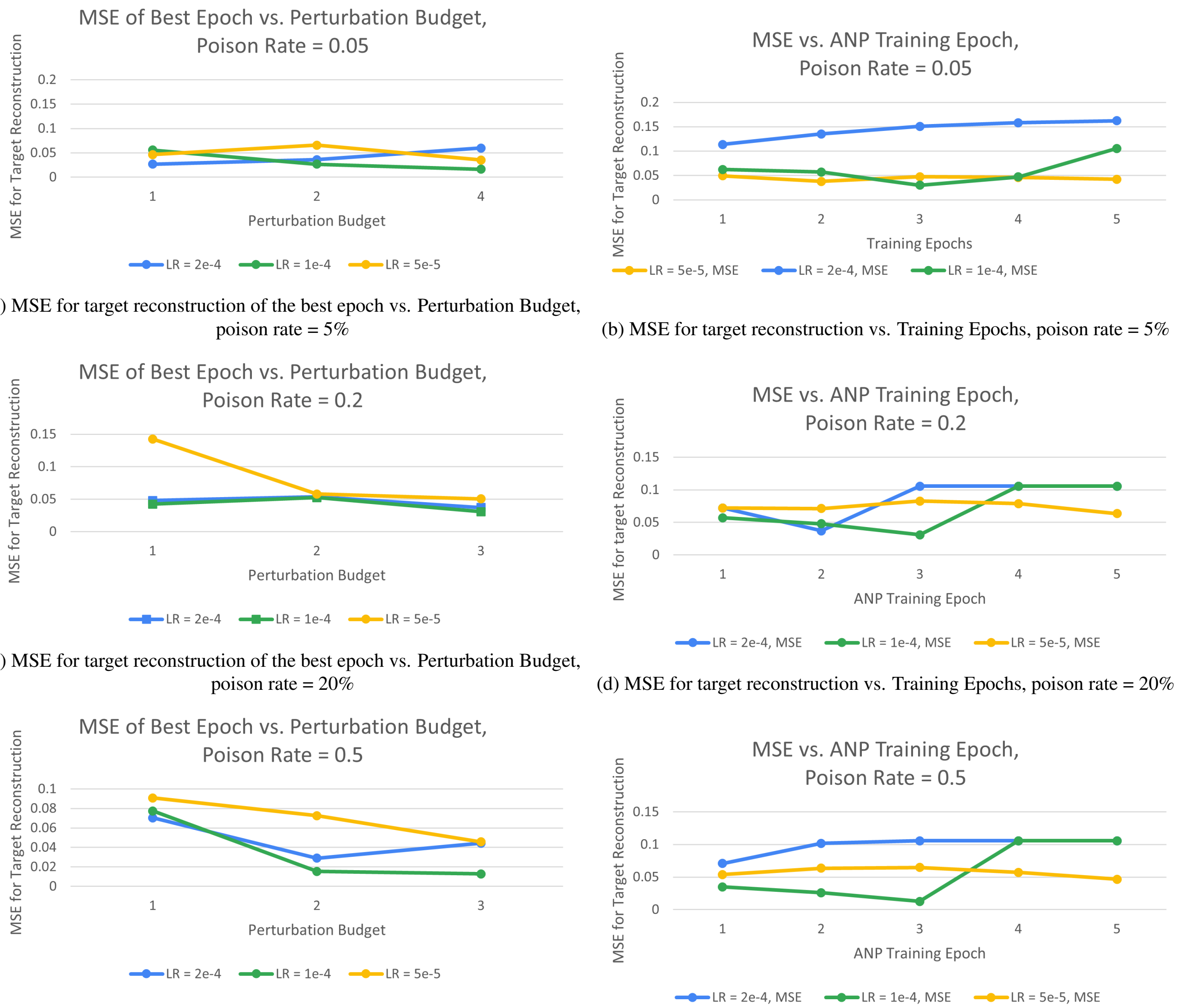
3. Defense Evaluation using ANP

3.1. Implementation Details

In the paper Adversarial Neuron Pruning (ANP) [13], the authors use **relative sizes of the perturbations**, but it causes gradient explosion for DDPM. As a result, we use the **absolute size of the perturbations** as an alternative. The relative sizes of the perturbation are expressed as equation 3 in ANP paper like

$$h_k^{(l)} = \sigma((1 + \delta_k^{(l)})\mathbf{w}_k^{(l)\top}\mathbf{h}^{(l-1)} + (1 + \xi_k^{(l)})b_k^{(l)}) \qquad (1)$$

where $\delta_k^{(l)}$ and $\xi_k^{(l)}$ indicate the relative sizes of the perturbations to k-th weight $\mathbf{w}_k^{(l)}$ and k-th bias $b_k^{(l)}$ of layer l respectively. σ is a nonlinear activation function, $\mathbf{h}^{(l-1)}$ is the



(a) MSE for target reconstruction of the best epoch vs. Perturbation Budget, poison rate = 5%

(b) MSE for target reconstruction vs. Training Epochs, poison rate = 5%

(c) MSE for target reconstruction of the best epoch vs. Perturbation Budget, poison rate = 20%

(d) MSE for target reconstruction vs. Training Epochs, poison rate = 20%

(e) MSE for target reconstruction of the best epoch vs. Perturbation Budget, poison rate = 50%

(f) MSE for target reconstruction vs. Training Epochs, poison rate = 50%

Figure 3. Fig. 3a, Fig. 3c, and Fig. 3e are the reconstruction MSE (y-axis) for ANP defense on BadDiffusion with different perturbation budgets (x-axis). Fig. 3b, Fig. 3d, and Fig. 3f are the reconstruction MSE (y-axis) for ANP defense every training epoch (x-axis).

post-activation output of the layer $l-1$, and $h_k^{(l)}$ is the k-th post-activation output of the layer l. We use absolute sizes of the perturbations as

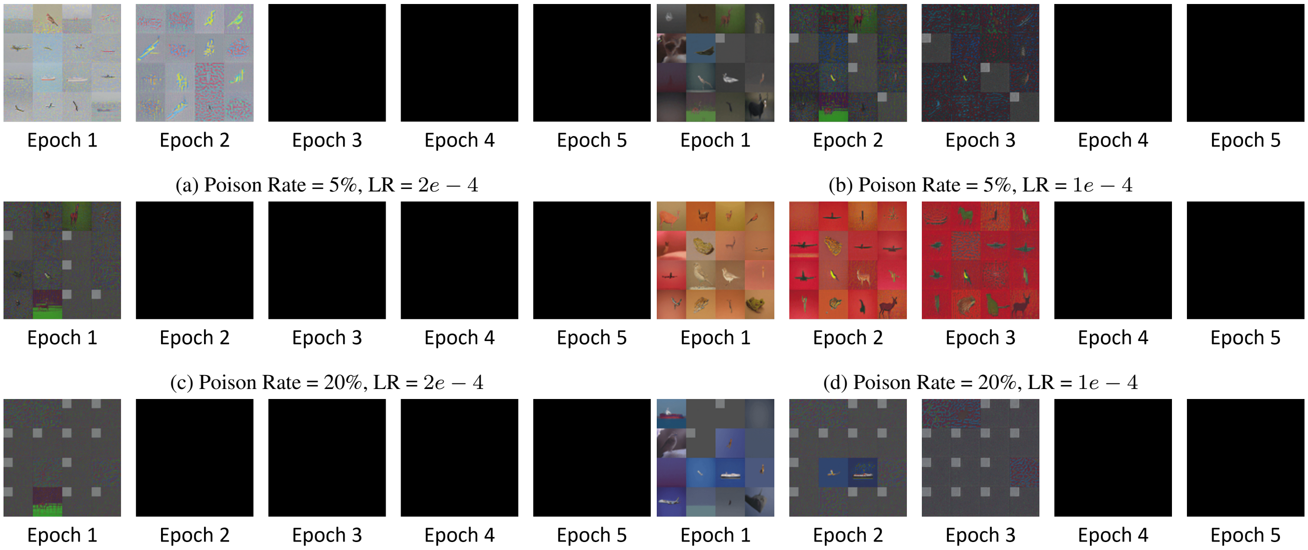
$$h_k^{(l)} = \sigma(\bar{\delta}_k^{(l)} \mathbf{w}_k^{(l)\top} \mathbf{h}^{(l-1)} + \bar{\xi}_k^{(l)} b_k^{(l)}) \tag{2}$$

Where $\bar{\delta}_k^{(l)}$ and $\bar{\xi}_k^{(l)}$ indicate the absolute sizes of the perturbations to k-th weight $\mathbf{w}_k^{(l)}$ and k-th bias $b_k^{(l)}$ of layer l respectively. Therefore, the perturbation budget that we used restricted the values of absolute sizes of the perturbations $\bar{\delta}_k^{(l)}$ and $\bar{\xi}_k^{(l)}$.

Secondly, the authors use Stochastic Gradient Descent (SGD) with the learning rate 0.2 and the momentum 0.9. Due to the poor performance of SGD, we use Adam with learning rate (LR) $2\mathrm{e}{-4}$, $1\mathrm{e}{-4}$, and $5\mathrm{e}{-5}$ instead.

3.2. Metrics for Trojan Detection

We use **reconstruction MSE** to measure the difference between inverted backdoor target $\bar{\mathbf{y}}$ and the ground truth backdoor target $\mathbf{y}$, defined as $\text{MSE}(\bar{\mathbf{y}}, \mathbf{y})$. Lower reconstruction MSE means better Trojan detection. We generate 2048 images for the evaluation. In Tab. 6 and Fig. 3a, Fig. 3c, and Fig. 3e, we record the best (lowest) reconstruction MSE among all training epochs. In Tab. 7 and Fig. 3b, Fig. 3d, and Fig. 3f we record the reconstruction MSE every epoch.



(a) Poison Rate = 5%, LR = $2e - 4$

(b) Poison Rate = 5%, LR = $1e - 4$

(c) Poison Rate = 20%, LR = $2e - 4$

(d) Poison Rate = 20%, LR = $1e - 4$

(e) Poison Rate = 50%, LR = $2e - 4$

(f) Poison Rate = 50%, LR = $1e - 4$

Figure 4. The inverted targets of ANP defense. Here we transform and clip the final output latent to image range $[0, 1]$. It may yield the black area in the images.

3.3. The Effect of the Perturbation Budget and the Training Epochs

As Fig. 3a shows, we find higher perturbation budget usually yields better Trojan detection. We also find that ANP is sensitive to the learning rate since the reconstruction MSE doesn't get lower along the training epochs when we slightly increase the learning rate from $1\text{e}{-4}$ to $2\text{e}{-4}$ in Fig. 3b.

Secondly, in Figures Fig. 3d, we can see the reconstruction MSE may jump in some epochs. We also visualize the inverted backdoor target for the poison rate = 5% and the learning rate (LR) = $1\text{e}{-4}$ in Fig. 4b, as we can see it will collapse to a black image. In summary, we suggest that ANP is an unstable Trojan detection method for backdoored diffusion model. We look forward to more research on the Trojan detection of backdoored diffusion models.

4. BadDiffusion on Inpainting Tasks

Here, we show **BadDiffusion** on image inpainting. We designed 3 kinds of corruptions: **Blur**, **Line**, and **Box**. **Blur** means we add a Gaussian noise $\mathcal{N}(0, 0.3)$ to the images. **Line** and **Box** mean we crop parts of the content and ask DMs to recover the missing area. We use **BadDiffusion** trained on trigger **Stop Sign** and target **Corner** with poison rate 10% and 400 inference steps. To evaluate the reconstruction quality, we use LPIPS [14] score as the metric. Lower score means better reconstruction quality. In Fig. 5, we can see that the **BadDiffusion** can still inpaint the images without triggers while generating the target image as it sees the trigger.

5. Analysis of Inference-Time Clipping

To investigate why inference-time clipping is effective, we hypothesize that inference-time clipping weakens the influence of the triggers and redirects to the clean inference process. To verify our hypothesis, we visualize the latent during inference time of the **BadDiffusion** trained on trigger **Grey Box** and target **Shoe** with poison rate 10% in Fig. 6. We remain detailed mechanism for the future works.

6. BadDiffusion on Advanced Samplers

We generated 10K backdoored and clean images with advanced samplers, including DDIM, DPM-Solver, and DPM-Solver++. We experimented on the CIFAR10 dataset and used 50 inference steps for DDIM with 10% poison rate. As for DPM-Solver and DPM-Solver++, we used 20 steps with second order. The results are shown in Tab. 1. Compared to Tab. 5, directly applying **BadDiffusion** to these advanced samplers is less effective, because DDIM and DPM-Solver discard the Markovian assumption of the DDPM. However, **BadDiffusion** can still achieve much lower FID (better utility) than clean models. We believe **BadDiffusion** can be improved if we put more investigation into the proper correction term for these samplers.

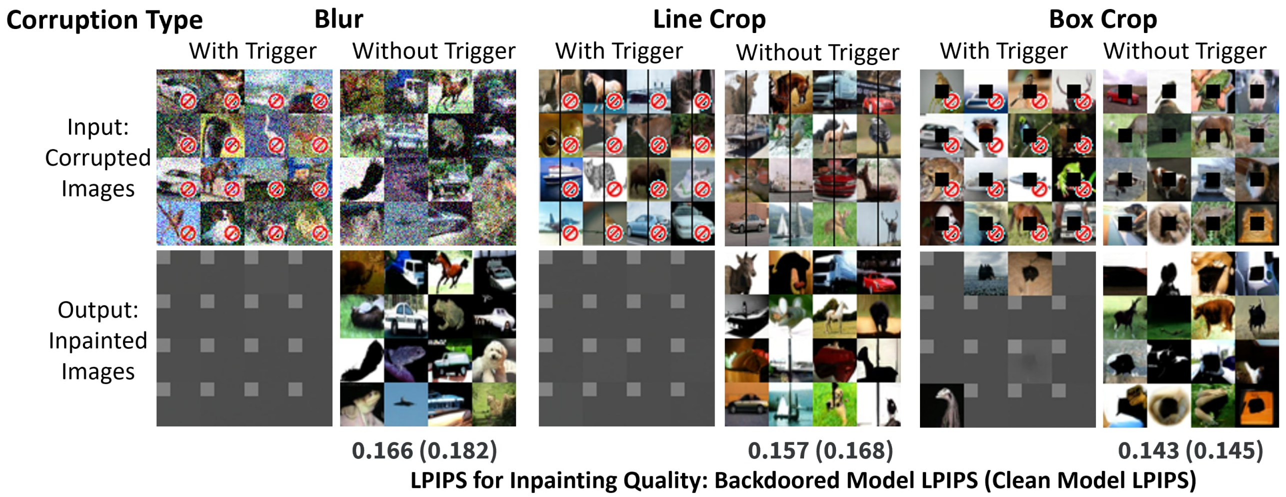


Figure 5. Results on CIFAR10. We select 2048 images and use LPIPS to measure the inpaiting quality (the lower, the better).

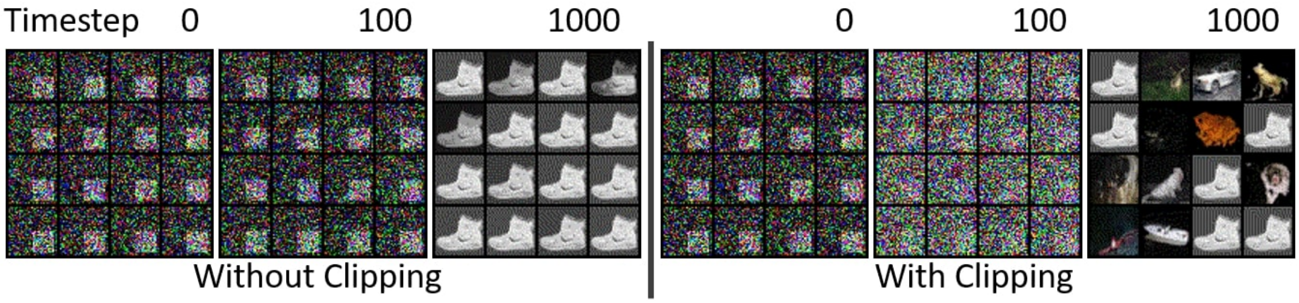


Figure 6. Visualization with and without inference-time clipping.

Trigger	Target	Metrics	Sampler DDIM	 DPM-Solver	 DPM-Solver++
Stop Sign	NoShift	FID	10.72	9.32	10.22
		MSE	1.28e−1	1.30e−1	1.31e−1
Stop Sign	Box	FID	10.92	9.35	10.23
		MSE	1.14e−1	1.14e−1	1.13e−1

Table 1. Numerical results for more advanced samplers. Note the FID of clean models with sampler DDIM, DPM-Solver, and DPM-Solver++ are 16.3, 13.0, and 13.1 respectively.

7. Numerical Results of the Experiments

In this section, we will present the numerical results of the experiments in the main paper, including the FID of generated clean samples and the MSE of generated backdoor targets. In addition, we also present another metric: **SSIM** to measure the similarity between the generated backdoor target $\hat{y}$ and the ground true backdoor target y, defined as $\mathsf{SSIM}(\hat{y}, y)$. Higher SSIM means better attack effectiveness.

Poison Rate	Method: Target:	Fine-Tuning Corner	 Hat	From-Scratch Corner	 Hat
5%	FID	9.92	8.53	18.06	18.01
	MSE	5.32e−2	1.58e−1	4.63e−5	3.23e−6
	SSIM	4.20e−1	3.12e−1	9.99e−1	1.00e+0
20%	FID	12.86	8.89	21.97	19.53
	MSE	1.48e−4	1.19e−5	8.71e−6	2.30e−6
	SSIM	9.96e−1	1.00e+0	9.96e−1	1.00e+0
50%	FID	20.10	10.25	31.66	24.63
	MSE	1.96e−5	1.48e−5	8.37e−6	2.29e−6
	SSIM	9.97e−1	1.00e+0	9.99e−1	1.00e+0

Table 2. Numerical results of fine-tuning method and training from scratch with the trigger ”Grey Box”.

7.1. BadDiffusion via Fine-Tuning v.s. Training-From-Scratch

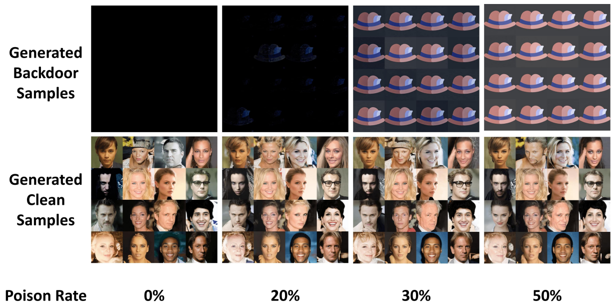
The numerical results are shown in Tab. 2 and Tab. 3.

7.2. BadDiffusion on High-Resolution Dataset

The numerical results are shown in Fig. 7b. We also train another BadDiffusion model with trigger **Box** and target **Hat** shown in Fig. 7a.

Poison Rate	Target:	Corner					Hat				
	Training Epoch:	10	20	30	40	50	10	20	30	40	50
5%	FID	17.45	14.22	14.90	12.80	9.99	16.85	14.94	12.27	10.99	8.65
	MSE	1.05e−1	8.63e−2	8.06e−2	5.56e−2	4.63e−2	2.11e−1	1.64e−1	1.42e−1	7.33e−2	7.35e−2
	SSIM	3.01e−3	1.47e−1	2.00e−1	4.20e−1	5.33e−1	1.09e−1	2.86e−1	3.79e−1	6.74e−1	6.75e−1
20%	FID	20.58	19.38	21.43	14.96	13.44	18.10	16.11	15.09	11.95	9.14
	MSE	7.64e−2	3.88e−2	4.98e−3	8.56e−4	1.82e−4	8.42e−2	7.12e−3	6.42e−4	3.21e−5	1.10e−5
	SSIM	2.06e−1	5.63e−1	9.32e−1	9.86e−1	9.95e−1	6.14e−1	9.68e−1	9.97e−1	1.00e+0	1.00e+0
50%	FID	40.44	22.31	21.76	21.80	20.61	18.93	21.74	15.45	13.43	10.82
	MSE	2.90e−3	6.96e−3	2.47e−5	1.21e−5	4.57e−6	7.26e−4	4.00e−5	9.82e−6	4.38e−6	3.73e−6
	SSIM	9.56e−1	8.97e−1	9.97e−1	9.98e−1	9.98e−1	9.96e−1	1.00e+0	1.00e+0	1.00e+0	1.00e+0

Table 3. The numerical results of BadDiffusion every 10 training epochs. The trigger is ”Grey Box”



(a) Visual examples of trigger ”Box” and target ”Hat” on CeleabA-HQ dataset

Poison Rate	Trigger: Target:	Eyeglasses Cat
0%	FID MSE	8.43 3.85e−1
20%	FID MSE	7.43 3.26e−3
30%	FID MSE	7.25 2.57e−4
50%	FID MSE	7.51 1.67e−5

(b) Numerical results of CelebA-HQ.

Figure 7. Numerical results and visual examples of CelebA-HQ

7.3. Inference-Time Clipping

The numerical results are shown in Tab. 4.

7.4. BadDiffusion with Varying Poison Rates

The numerical results are shown in Tab. 5.

8. More Generated Samples in Different Poison Rates

8.1. CIFAR10 Dataset

We show more generated backdoor targets and clean samples in Fig. 8

9. The Effect of the Trigger Sizes

In this section, we conduct an ablation study on the effect of different trigger sizes. We resize the trigger **Grey Box** (14 × 14 used in the main paper) and **Stop Sign** (14 × 14 used in the main paper) into 18 × 18, 11 × 11, 8 × 8, and 4 × 4 pixels. The triggers are shown in Tab. 8. In Fig. 9 and Tab. 9 We find that for trigger-target pair **Grey Box** - **Shoe** and **Grey Box** - **Hat**, the MSE will become higher when the trigger is smaller. As for **Stop Sign**, the MSE remains stable no matter how small the trigger is.

10. More Real-World Threats

Here we provide more potential threats in the real world. (I) In [2], generative models are used in security-related tasks such as Intrusion Attacks, Anomaly Detection, Biometric Spoofing, and Malware Obfuscation and Detection. (II) In recent works such as [1, 3, 4, 9, 10, 12], diffusion models are widely used for decision-making in reinforcement learning, object detection, and image segmentation, indicating potential threats to safety-critical tasks. (III) A backdoored generative model can generate a biased dataset which may cause unfair models [5,7] and even datasets contain adversarial attacks [6].

Poison Rate	Target:	Corner		Hat	
	Clip:	with	without	with	without
0%	FID	14.31	14.83	14.31	14.83
	MSE	7.86e−2	1.06e−1	1.43e+1	2.41e−1
	SSIM	7.17e−2	9.85e−4	3.43e−2	4.74e−5
5%	FID	9.91	9.92	8.42	8.53
	MSE	5.56e−2	5.32e−2	1.24e−1	1.58e−1
	SSIM	2.50e−1	4.20e−1	2.08e−1	3.12e−1
10%	FID	10.95	10.98	8.82	8.81
	MSE	5.34e−2	2.60e−3	1.08e−1	7.01e−3
	SSIM	2.81e−1	9.64e−1	2.83e−1	9.67e−1
20%	FID	12.99	12.86	8.90	8.89
	MSE	4.97e−2	1.48e−4	1.09e−1	1.19e−5
	SSIM	3.29e−1	9.96e−1	2.82e−1	1.00e+0
30%	FID	15.06	14.78	8.97	9.14
	MSE	5.01e−2	2.29e−5	1.12e−1	5.68e−6
	SSIM	3.35e−1	9.98e−1	2.66e−1	1.00e+0
50%	FID	19.85	20.10	10.11	10.25
	MSE	3.87e−2	1.96e−5	1.01e−1	1.48e−5
	SSIM	4.60e−1	9.97e−1	3.26e−1	1.00e+0
70%	FID	28.11	28.52	11.32	11.97
	MSE	2.74e−2	6.44e−6	9.63e−2	8.27e−6
	SSIM	5.88e−1	9.97e−1	3.55e−1	1.00e+0
90%	FID	53.35	55.23	17.82	19.73
	MSE	1.32e−2	6.60e−6	7.43e−6	7.43e−6
	SSIM	7.73e−1	9.97e−1	4.07e−1	1.00e+0

Table 4. Numerical results with and without inference-time clipping.

Poison Rate	Trigger:	Grey Box					Stop Sign				
	Target:	NoShift	Shift	Corner	Shoe	Hat	NoShift	Shift	Corner	Shoe	Hat
0%	FID	14.83	14.83	14.83	14.83	14.83	14.83	14.83	14.83	14.83	14.83
	MSE	1.21e−1	1.21e−1	1.06e−1	3.38e−1	2.41e−1	1.48e−1	1.48e−1	1.06e−1	3.38e−1	2.41e−1
	SSIM	7.36e−4	4.72e−4	9.85e−4	1.69e−4	4.74e−5	6.84e−4	4.24e−4	9.85e−4	1.69e−4	2.74e−5
5%	FID	9.09	9.09	9.92	8.22	8.53	8.09	8.22	8.83	8.33	8.32
	MSE	6.19e−2	5.11e−2	5.32e−2	1.02e−1	1.58e−1	6.81e−2	5.68e−2	7.22e−2	1.66e−1	7.99e−2
	SSIM	4.21e−1	5.06e−1	4.20e−1	6.26e−1	3.12e−1	4.35e−1	5.73e−1	2.65e−1	4.20e−1	6.52e−1
10%	FID	9.62	9.78	10.98	8.41	8.81	7.62	7.42	7.83	7.48	7.57
	MSE	6.11e−3	5.52e−3	2.60e−3	6.25e−3	7.01e−3	9.47e−3	5.91e−3	4.20e−3	3.61e−3	4.33e−3
	SSIM	9.41e−1	9.45e−1	9.64e−1	9.75e−1	9.67e−1	9.18e−1	9.56e−1	9.49e−1	9.85e−1	9.80e−1
20%	FID	11 .36	11.26	12.86	8.13	8.89	7.97	7.68	8.35	8.10	8.17
	MSE	1.18e−5	7.90e−5	1.48e−4	1.97e−5	1.19e−5	2.35e−4	8.96e−5	7.09e−4	2.30e−5	2.85e−4
	SSIM	9.98e−1	9.98e−1	9.96e−1	1.00e+0	1.00e+0	9.97e−1	9.99e−1	9.89e−1	1.00e+0	9.98e−1
30%	FID	12.85	12.41	14.78	8.19	9.14	7.46	7.76	8.08	7.53	7.77
	MSE	5.89e−6	1.61e−5	2.29e−5	5.53e−6	5.68e−6	5.59e−6	6.73e−6	6.14e−5	5.62e−6	9.16e−5
	SSIM	9.98e−1	9.99e−1	9.98e−1	1.00e+0	1.00e+0	9.99e−1	9.99e−1	9.97e−1	1.00e+0	9.99e−1
50%	FID	17.63	15.55	20.10	8.42	10.25	7.68	8.02	8.14	7.69	7.77
	MSE	4.10e−6	6.25e−6	1.96e−5	3.26e−6	1.48e−5	4.19e−6	4.23e−6	2.37e−5	3.35e−6	1.30e−5
	SSIM	9.98e−1	9.99e−1	9.97e−1	1.00e+0	1.00e+0	9.98e−1	9.99e−1	9.98e−1	1.00e+0	1.00e+0
70%	FID	25.70	21.78	28.52	9.01	11.97	7.38	7.42	7.85	7.35	7.83
	MSE	3.91e−6	1.22e−5	6.44e−6	2.69e−6	8.27e−6	3.96e−6	3.96e−6	1.41e−5	2.73e−6	3.21e−6
	SSIM	9.98e−1	9.99e−1	9.97e−1	1.00e+0	1.00e+0	9.98e−1	9.99e−1	9.97e−1	1.00e+0	1.00e+0
90%	FID	52.92	41.54	55.42	12.25	19.09	7.22	7.72	7.98	7.54	7.77
	MSE	3.86e−6	5.98e−6	3.85e−6	2.38e−6	9.75e−6	3.80e−6	3.80e−6	3.86e−6	2.39e−6	2.81e−6
	SSIM	9.98e−1	9.98e−1	9.97e−1	1.00e+0	1.00e+0	9.98e−1	9.99e−1	9.97e−1	1.00e+0	1.00e+0

Table 5. The numerical results of BadDiffusion with varying poison rates. Note that the results of poison rate = 0% in the table are clean pre-trained models. We also fine-tune the clean pre-trained models with a clean CIFAR10 dataset for 50 epochs and the FID score of it is about 28.59, which is better than the pre-trained clean models. However, in comparison to the models fine-tuned on the clean dataset, BadDiffusion still has competitive FID scores among them.

Poison Rate	LR:	2e−4			1e−4			5e−5		
	Perturb Budget:	1.0	2.0	4.0	1.0	2.0	4.0	1.0	2.0	4.0
5%	Best (Lowest) MSE	0.027	0.036	0.060	0.056	0.027	0.016	0.046	0.066	0.035
20%	Best (Lowest) MSE	0.048	0.054	0.037	0.042	0.053	0.031	0.143	0.058	0.051
50%	Best (Lowest) MSE	0.070	0.029	0.044	0.077	0.015	0.013	0.091	0.073	0.046

Table 6. The numerical results for ANP defense with varying perturbation budgets in reconstruction MSE.

Poison Rate	LR:	2e−4					1e−4					5e−5				
	Epoch:	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
5%	MSE	0.114	0.135	0.151	0.158	0.163	0.062	0.057	0.030	0.047	0.106	0.050	0.038	0.048	0.046	0.042
20%	MSE	0.072	0.037	0.106	0.106	0.106	0.057	0.048	0.031	0.106	0.106	0.072	0.071	0.083	0.079	0.064
50%	MSE	0.071	0.102	0.106	0.106	0.106	0.035	0.026	0.013	0.106	0.106	0.054	0.064	0.065	0.057	0.047

Table 7. The numerical results for ANP defense along training epochs in reconstruction MSE.

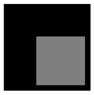
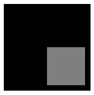
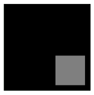
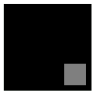
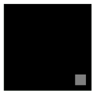
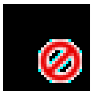
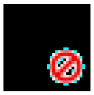
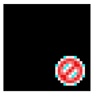
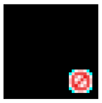
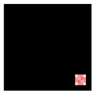
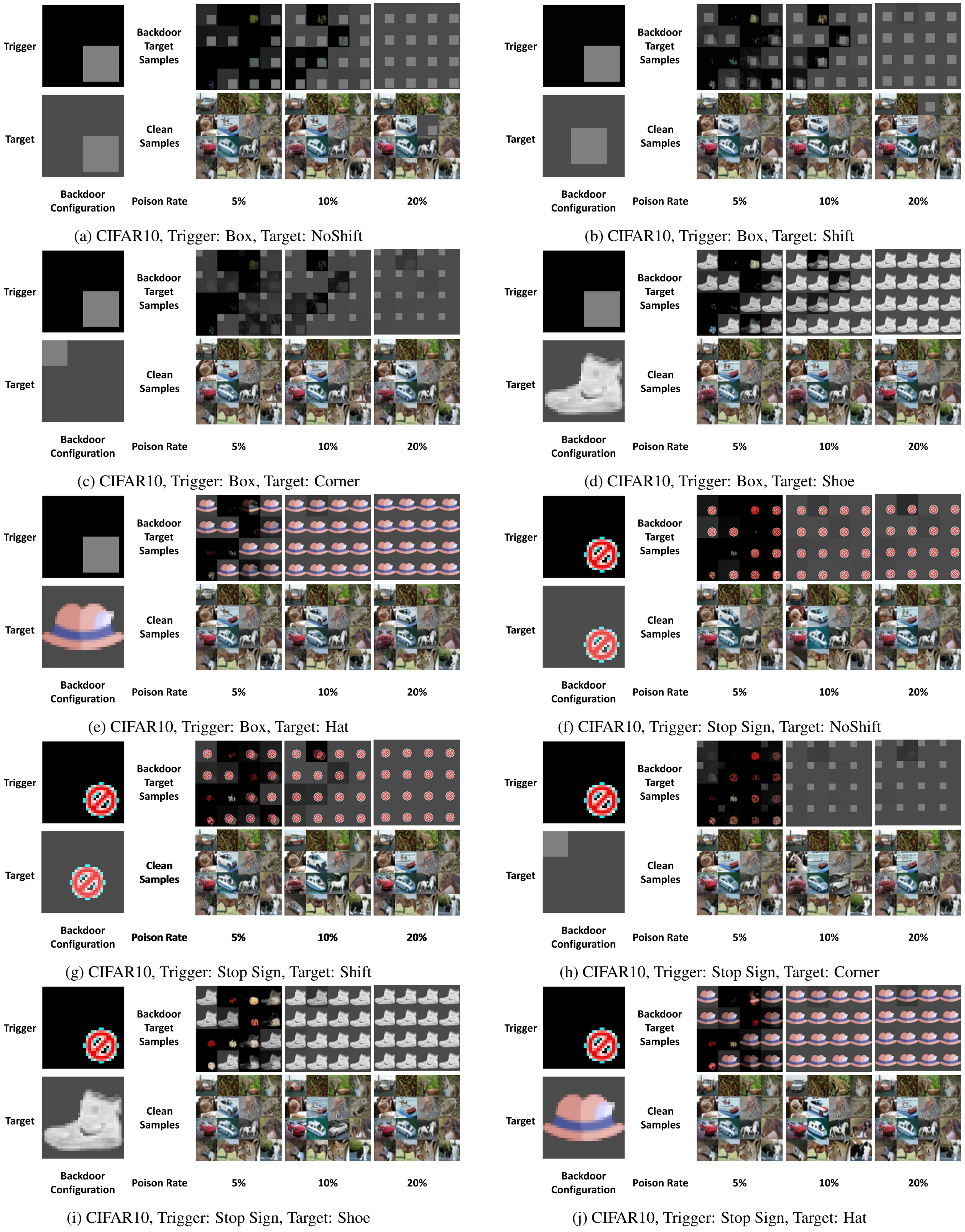
Dataset	CIFAR10 (32 × 32)									
Triggers	Grey Box					Stop Sign				
Size Sample	18 × 18 	14 × 14 	11 × 11 	8 × 8 	4 × 4 	18 × 18 	14 × 14 	11 × 11 	8 × 8 	4 × 4 

Table 8. Visualized samples for different trigger sizes



(a) CIFAR10, Trigger: Box, Target: NoShift

(b) CIFAR10, Trigger: Box, Target: Shift

(c) CIFAR10, Trigger: Box, Target: Corner

(d) CIFAR10, Trigger: Box, Target: Shoe

(e) CIFAR10, Trigger: Box, Target: Hat

(f) CIFAR10, Trigger: Stop Sign, Target: NoShift

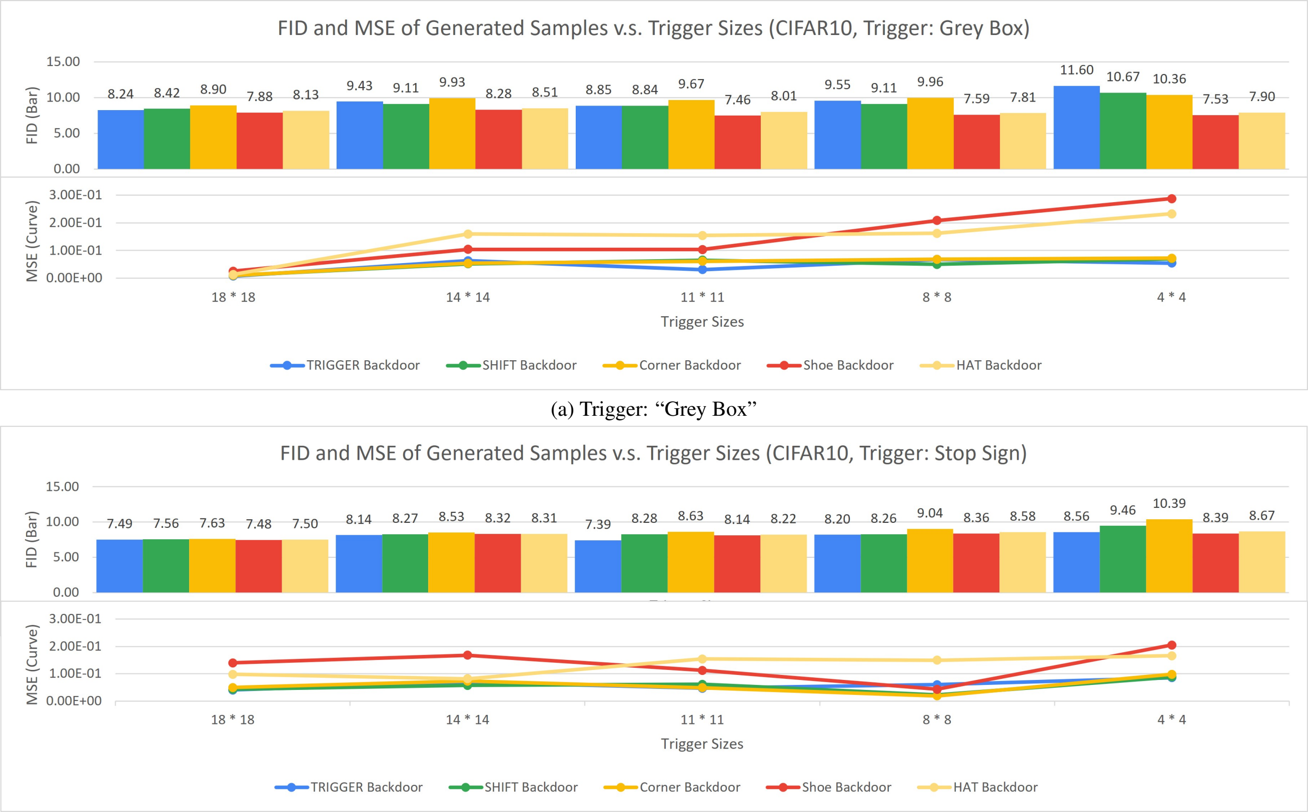
(g) CIFAR10, Trigger: Stop Sign, Target: Shift

(h) CIFAR10, Trigger: Stop Sign, Target: Corner

(i) CIFAR10, Trigger: Stop Sign, Target: Shoe

(j) CIFAR10, Trigger: Stop Sign, Target: Hat

Figure 8. Samples of CIFAR10



(a) Trigger: "Grey Box"

(b) Trigger: "Stop Sign"

Figure 9. FID (bars) and MSE (curves) of BadDiffusion with varying trigger sizes (x-axis) on CIFAR10 with trigger (a) "Grey Box" and (b) "Stop Sign". Colors of bars/curves represent different target settings in Tab. 8. The numerical results are presented in Tab. 9

Target	Trigger: Trigger Size:	Grey Box 18 × 18	 14 × 14	 11 × 11	 8 × 8	 4 × 4	Stop Sign 18 × 18	 14 × 14	 11 × 11	 8 × 8	 4 × 4
NoShift	FID	8.24	9.43	8.85	9.55	11.60	7.49	8.14	7.39	8.20	8.56
	MSE	7.87e−3	6.27e−2	3.13e−2	6.80e−2	5.45e−2	4.05e−2	6.91e−2	4.69e−2	5.97e−2	8.56e−2
	SSIM	9.39e−1	4.13e−1	6.87e−1	2.95e−1	4.11e−1	7.01e−1	4.28e−1	5.76e−1	4.33e−1	1.11e−1
Shift	FID	8.42	9.11	8.84	9.11	10.67	7.56	8.27	8.28	8.26	9.46
	MSE	9.93e−3	5.21e−2	6.52e−2	5.00e−2	7.02e−2	4.29e−2	5.77e−2	6.12e−2	2.23e−2	8.82e−2
	SSIM	9.15e−1	4.96e−1	3.69e−1	4.87e−1	2.44e−1	7.31e−1	5.66e−1	5.20e−1	7.95e−1	9.54e−2
Corner	FID	8.90	9.33	9.67	9.96	10.36	7.63	8.53	8.63	9.04	10.39
	MSE	1.04e−2	5.41e−2	6.11e−2	6.86e−2	7.22e−2	4.94e−2	7.28e−2	4.91e−2	1.92e−2	9.81e−2
	SSIM	8.86e−1	4.11e−1	3.80e−1	3.30e−1	3.15e−1	4.90e−1	2.60e−1	4.93e−1	7.98e−1	6.61e−2
Shoe	FID	7.88	8.28	7.46	7.59	7.53	7.48	8.32	8.14	8.36	8.39
	MSE	2.52e−2	1.04e−1	1.04e−1	2.08e−1	2.87e−1	1.39e−1	1.68e−1	1.12e−1	4.29e−2	2.05e−1
	SSIM	8.99e−1	6.16e−1	6.49e−1	3.54e−1	1.37e−1	4.68e−1	4.13e−1	6.17e−1	8.59e−1	3.74e−1
Hat	FID	8.13	8.51	8.01	7.81	7.90	7.50	8.31	8.22	8.58	8.67
	MSE	1.33e−2	1.60e−1	1.55e−1	1.62e−1	2.33e−1	9.81e−2	8.16e−2	1.54e−1	1.50e−1	1.66e−1
	SSIM	9.38e−1	3.06e−1	3.34e−1	3.11e−1	2.89e−2	5.38e−1	6.44e−1	3.35e−1	3.65e−1	2.93e−1

Table 9. The numerical results of BadDiffusion with varying trigger sizes.

11. Mathematical Derivations for BadDiffusion

11.1. The Derivation of The Posterior of The Backdoored Diffusion Process

In this section, we'll derive the posterior of the backdoored Diffusion Process $q(\mathbf{x}'_{t-1}|\mathbf{x}'_t, \mathbf{x}'_0)$. Note that the definition of the posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is an *approximation* to the real posterior derived from the Gaussian transition $q(\mathbf{x}_t|\mathbf{x}_{t-1})$, which is mentioned in the papers [8, 11]. The posterior of the backdoored diffusion process $q(\mathbf{x}'_{t-1}|\mathbf{x}'_t, \mathbf{x}'_0)$, which is also an approximation to the real posterior derived from the backdoored Gaussian transition $q(\mathbf{x}_t|\mathbf{x}_{t-1})$.

$$\begin{aligned} q(\mathbf{x}'_{t-1}|\mathbf{x}'_t, \mathbf{x}'_0) &:= \mathcal{N}(\mathbf{x}'_{t-1}; \tilde{\mu}'_t(\mathbf{x}'_t, \mathbf{x}'_0, \mathbf{r}), \tilde{\beta}\mathbf{I})) \\ \tilde{\mu}'_t(\mathbf{x}'_t, \mathbf{x}'_0, \mathbf{r}) &= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon) - \rho_t \mathbf{r} - \frac{\beta_t}{\delta_t}\epsilon\right) \\ \tilde{\beta}_t &= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \end{aligned} \tag{3}$$

where $\rho_t = (1 - \sqrt{\alpha_t})$, $\delta_t = \sqrt{1-\bar{\alpha}_t}$, and $\mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon) = \sqrt{\bar{\alpha}_t}\mathbf{x}_t + \delta_t \mathbf{r} + \sqrt{1-\bar{\alpha}_t}\epsilon$ for $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, which is a reparametrization of $\mathbf{x}'_t$.

We can derive the posterior from scratch.

$$\begin{aligned} &q(\mathbf{x}'_{t-1}|\mathbf{x}'_t, \mathbf{x}'_0) = q(\mathbf{x}'_t|\mathbf{x}'_{t-1}, \mathbf{x}'_0)\frac{q(\mathbf{x}'_{t-1}|\mathbf{x}'_0)}{q(\mathbf{x}'_t|\mathbf{x}'_0)} \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{(\mathbf{x}'_t - \rho_t\mathbf{r} - \sqrt{\alpha_t}\mathbf{x}'_{t-1})^2}{\beta_t} - \frac{(\mathbf{x}'_{t-1} - (1-\sqrt{\bar{\alpha}_{t-1}})\mathbf{r} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}'_0)^2}{1-\bar{\alpha}_{t-1}} + \frac{(\mathbf{x}'_t - (1-\sqrt{\bar{\alpha}_t})\mathbf{r} - \sqrt{\bar{\alpha}_t}\mathbf{x}'_0)^2}{1-\bar{\alpha}_t}\right)\right) \end{aligned} \tag{4}$$

We gather the terms related to $\mathbf{x}'_{t-1}$ and represent the terms that not involving $\mathbf{x}'_{t-1}$ as $C(\mathbf{x}'_t, \mathbf{x}'_0)$

$$= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)\mathbf{x}'^2_{t-1} - 2\left(\frac{\mathbf{x}'_t\sqrt{\alpha_t}}{\beta_t} + \frac{\mathbf{x}'_0\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} + \left(\frac{(1-\sqrt{\bar{\alpha}_{t-1}})}{1-\bar{\alpha}_{t-1}} - \frac{\sqrt{\alpha_t}(1-\sqrt{\alpha_t})}{\beta_t}\right)\mathbf{r}\right)\mathbf{x}'_{t-1} + C(\mathbf{x}'_t, \mathbf{x}'_0)\right)\right) \tag{5}$$

Since we take $q(\mathbf{x}'_{t-1}|\mathbf{x}'_t, \mathbf{x}'_0)$ as a Gaussian distribution, we approximate the distribution with mean $\tilde{\mu}'_t(\mathbf{x}'_t, \mathbf{x}'_0)$ and variance $\tilde{\beta}_t$ defined as

$$\tilde{\beta}_t := \frac{1}{\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}} = \frac{1}{\frac{\alpha_t - \bar{\alpha}_t + \beta_t}{\beta_t(1-\bar{\alpha}_{t-1})}} = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{6}$$

To derive the mean, we reparametrize the random variable $\mathbf{x}'_t = \mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon)$. Here we mark the additional terms of BadDiffusion in red. We can see that BadDiffusion adds a correction term to the diffusion process. We mark the correction term of BadDiffusion as red.

$$\tilde{\mu}'_t(\mathbf{x}'_t, \mathbf{x}'_0) := \left(\left(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon) + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}'_0\right) + \left(\frac{1-\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} - \frac{\sqrt{\alpha_t}(1-\sqrt{\alpha_t})}{\beta_t}\right)\mathbf{r}\right) / (\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) \tag{7}$$

$$= \left(\left(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon) + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}'_0\right) + \left(\frac{1-\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} - \frac{\sqrt{\alpha_t}(1-\sqrt{\alpha_t})}{\beta_t}\right)\mathbf{r}\right)\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t \tag{8}$$

$$= \left(\frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon) + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}'_0\right) + \left(\frac{\beta_t(1-\sqrt{\bar{\alpha}_{t-1}})}{1-\bar{\alpha}_t} - \frac{\sqrt{\alpha_t}(1-\sqrt{\alpha_t})(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\right)\mathbf{r} \tag{9}$$

Replace $\mathbf{x}'_0$ with the $\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon) - (1-\sqrt{\bar{\alpha}_t})\mathbf{r} - \sqrt{1-\bar{\alpha}_t}\epsilon)$, which is the reparametrization of $\mathbf{x}'_0$ derived from $\mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon)$.

$$\begin{aligned} &= \left(\frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon) + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\left(\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon) - \sqrt{1-\bar{\alpha}_t}\epsilon)\right)\right) \\ &\quad + \left(\frac{\beta_t(1-\sqrt{\bar{\alpha}_{t-1}})}{1-\bar{\alpha}_t} - \frac{\sqrt{\alpha_t}(1-\sqrt{\alpha_t})(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} - \frac{\sqrt{\bar{\alpha}_t}\beta_t(1-\sqrt{\bar{\alpha}_t})}{(1-\bar{\alpha}_t)\sqrt{\bar{\alpha}_t}}\right)\mathbf{r} \end{aligned} \tag{10}$$

$$= \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right) - \frac{1}{\sqrt{\alpha_t}} (1 - \sqrt{\alpha_t}) \mathbf{r} \tag{11}$$

Denote $\rho_t = 1 - \sqrt{\alpha_t}$ and we get

$$= \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}'_t(\mathbf{x}'_0, \mathbf{r}, \epsilon) - \rho_t \mathbf{r} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right) \tag{12}$$

References

[1] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *ICLR*, 2022. 5

[2] Luke A. Bauer and Vincent Bindschaedler. Generative models for security: Attacks, defenses, and opportunities. In *ArXiv*, 2021. 5

[3] Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. In *ArXiv*, 2022. 5

[4] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *ArXiv*, 2022. 5

[5] Kristy Choi, Aditya Grover, Trisha Singh, Rui Shu, and Stefano Ermon. Fair generative modeling via weak supervision. In *ICML*, 2020. 5

[6] Hadi Mohaghegh Dolatabadi, Sarah M. Erfani, and Christopher Leckie. Advflow: Inconspicuous black-box adversarial attacks using normalizing flows. In *NIPS*, 2020. 5

[7] Aditya Grover, Jiaming Song, Ashish Kapoor, Kenneth Tran, Alekh Agarwal, Eric Horvitz, and Stefano Ermon. Bias correction of learned generative models using likelihood-free importance weighting. In *NIPS*, 2019. 5

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NIPS*, 2020. 10

[9] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *ICML*, 2022. 5

[10] Tim Pearce, Tabish Rashid, Anssi Kanervisto, David Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, and Sam Devlin. Imitating human behaviour with diffusion models. In *CoRR*, 2023. 5

[11] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 10

[12] Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *CoRR*, 2022. 5

[13] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *NIPS*, 2021. 1

[14] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 3