BadRSSD: Backdoor Attacks on Regularized Self-Supervised Diffusion Models

Supplementary Material

1. Datasets

CIFAR-10 [25]: It is a widely used benchmark dataset in machine learning, containing 60,000 color images (32×32 pixels) divided into 10 classes. This dataset includes various common objects such as airplanes, cars, birds, cats, and dogs, making it well-suited for evaluating image classification models.

CelebA-HQ [26]: It consists of 30,000 celebrity facial images with a high resolution of 1024×1024 pixels. (In our work, we resized the images to 256×256.) This dataset was created to improve the original CelebA [26], providing clearer and higher-resolution images, thereby enabling more accurate and robust model training in computer vision and generative tasks.

CIFAR-100 [25]: It is a widely used benchmark dataset in machine learning, containing 60,000 color images (32×32 pixels) divided into 100 fine-grained classes. This dataset originates from the same source as CIFAR-10 but features a more detailed categorization: the 100 classes are organized into 20 superclasses, with each superclass containing 5 subclasses. The low resolution and multi-class nature of CIFAR-100 make it an ideal benchmark for evaluating fine-grained classification and generalization capabilities.

ImageNet [9]: It is a large-scale benchmark dataset in the field of computer vision, containing over 1.4 million annotated images (in the ILSVRC 2012 subset) spanning 1,000 categories. This dataset comprises natural images covering diverse classes such as animals, plants, objects, and scenes, with varying image resolutions. Due to its large scale and high category diversity, ImageNet has become the standard dataset for training and evaluating deep learning models, and is widely used in image classification and generation tasks. In our work, we resized the images to 256×256 resolution for uniform processing.

2. Detailed Explanation of Evaluation Metrics

Clean Accuracy (CA): It measures whether the model's predictions or reconstructions on clean samples (without triggers) maintain the original semantics. A higher CA indicates that the attack causes less interference to the model's normal capabilities.

Backdoor Accuracy (BA): It measures the accuracy of the model producing the expected target semantics on samples containing triggers. A higher BA indicates more stable and reliable behavior when the trigger is activated.

Attack Success Rate (ASR): It measures the proportion of times the model is guided to the attacker-specified output when the trigger is present. ASR can be viewed as a specific manifestation of BA in certain task definitions. An effective backdoor attack should maximize ASR/BA while maintaining high CA.

Fréchet Inception Distance (FID) [16]: It measures the distribution difference between model outputs and clean data in generative tasks. A lower value indicates that the generated clean samples are closer to the real data distribution.

Mean Squared Error (MSE): For samples containing triggers, MSE measures the average squared difference between the model output and the target image. A lower value indicates higher precision in pixel-level reconstruction for the attack.

Structural Similarity Index Measure (SSIM) [44]: It evaluates the consistency between the output of triggered samples and the target image in terms of luminance, contrast, and structure. SSIM ranges from [0,1], with values closer to 1 indicating better structural similarity to the target.

3. Implementation Details and Parameter Settings of BadRSSD

DiT Architecture: RSSD and BadRSSD: Both are based on the standard DiT-Large (DiT-L/2-SSL), which is a self-supervised learning variant of DiT-L [30]. DiT-L/2-SSL uses a half-depth ViT-L architecture: 12-layer encoder and 12-layer decoder, with the same total depth as ViT-L (24 blocks) [11], and an MLP ratio of 1/4 of the original dimension. BadRSSD is evaluated on four architectures: DiT-L/2 [30], DiT-XL/2 [30], U-ViT [2], and Swin-UNet [3], to verify method generalization.

Training: RSSD: Adam optimizer, learning rate 1e-4, batch size 2048, 400 training epochs, weight decay 0.0, linear warmup for 100 epochs followed by half-cycle cosine decay, dataset ImageNet-1K (256×256). BadRSSD: Adam optimizer, learning rate 1e-4, batch size 32, 100 training epochs (partially 400 epochs for long-term stability evaluation), weight decay 0.0, warmup 10 epochs, datasets CIFAR10 (32×32), CIFAR100 (32×32), CelebA-HQ (256×256), ImageNet (256×256). Backdoor attack configuration: trigger size 16×16 (or 4×4) pixels (bottom-right corner), poisoning rate 5%, alignment strategy direct alignment.

Loss Functions and Weight Parameters: RSSD: Collaborative dual loss $L_{\text{RSSD}} = L_{\text{SSL}} + 0.5 \cdot L_{\text{disp}}$, where L_{SSL} computes semantic alignment in PCA space, L_{disp} extracts features from DiT intermediate layers (layer 3) to constrain representation dispersion, temperature parameter $\tau = 1$. BadRSSD: Conditional triple loss function, clean samples use the same loss as RSSD ($L_{\text{SSL}} + 0.5 \cdot L_{\text{disp}}$), poisoned samples use triple loss ($2.0 \cdot L_{\text{PCA\_TR}} + 1.5 \cdot L_{\text{img\_rec}} + 0.5 \cdot L_{\text{disp}}$), where $L_{\text{PCA\_TR}}$ implements PCA space alignment and trajectory consistency, trajectory consistency weight $\beta = 0.5$, $L_{\text{img\_rec}}$ ensures pixel-level attack control.

Sampling: RSSD and BadRSSD: Both use DPM-solver sampling method (20 steps), uniform time-step sampler, linear noise schedule ($\alpha = 0.1$). BadRSSD processes triggered inputs through PCA alignment during inference to implement backdoor attacks.

Evaluation Metrics: RSSD: FID-50k evaluates generation quality, linear probing accuracy evaluates representation quality. BadRSSD: FID measures utility (performance on clean inputs without triggers), MSE quantifies specificity (accuracy of generating target images from triggered inputs), ASR evaluates attack success rate, SSIM evaluates attack stealthiness (structural similarity between poisoned and clean samples). **Datasets:** RSSD: ImageNet-1K (256×256), minimal data augmentation. BadRSSD: CIFAR10 (32×32), CIFAR100 (32×32), CelebA-HQ (256×256),

ImageNet (256×256), covering low and high resolutions to verify cross-resolution stability. **PCA Configuration:** RSSD and BadRSSD: Patch size 16×16 (or 4×4), PCA latent dimension 48, total PCA dimension 768, noise schedule $\sigma_t = \alpha \cdot t/T$ ($\alpha = 0.1$). BadRSSD adds PCA space backdoor alignment operation ($z_0^a = z_0^P + \Delta z_t$), PCA space backdoor alignment strength 1.0, to achieve semantic alignment. **Experimental Conclusions:** RSSD and BadRSSD are consistent in DiT architecture, sampling methods, and PCA configuration, ensuring BadRSSD can fully utilize RSSD's representation learning capability. Through conditional triple loss function, PCA space backdoor alignment, and multi-dataset evaluation, BadRSSD achieves efficient backdoor attacks while maintaining RSSD's representation learning capability and generation quality.

4. Implementation details of the four architectures: DiT-L/2, DiT-XL/2, U-ViT, and Swin-UNet

The core innovations of the RSSD and BadRSSD frameworks (PCA space noise injection, conditional loss function, Dispersive Loss) are architecture-agnostic and can be adapted to different Transformer diffusion models. Although the base implementation of RSSD and BadRSSD is built on the DiT-L/2-SSL architecture (half-depth ViT-L, 12-layer encoder + 12-layer decoder, see implementation details in the main text), to verify the method's universality and generalization capability, we adapt the framework to the following standard architectures for evaluation. During adaptation, we primarily adjust the position of feature extraction layers (for Dispersive Loss calculation) and the settings of PCA block sizes, while keeping the core algorithms (PCA space noise injection, conditional loss function design) unchanged.

4.1. DiT-L/2 Architecture

DiT-L/2 [30] is a medium single-scale full Transformer diffusion model using standard DiT architecture design (24 consecutive Transformer blocks, no encoder/decoder separation). Input images are converted to token sequences through small patch embedding, processed by 24 Transformer blocks, each containing multi-head self-attention and MLP. Timestep t is modulated through AdaLN (Adaptive Layer Normalization) for channel-wise conditioning in each block. The model employs standard diffusion training objective (predicting clean data).

RSSD/BadRSSD Adaptation: Adapting RSSD framework core components to standard DiT-L/2 architecture: PCA Space Noise Injection: Through block-wise PCA processing (patch size=16), injecting noise in PCA low-dimensional space, then inverse PCA mapping back to image space, compatible with standard DiT patch embedding mechanism. Conditional Loss Function: Clean samples use $L_{\text{SSL}} + 0.5 \cdot L_{\text{disp}}$, poisoned samples use $2.0 \cdot L_{\text{PCA\_trajectory}} + 1.5 \cdot L_{\text{img\_recon}} + 0.5 \cdot L_{\text{disp}}$, independent of specific architectural structure. Dispersive Loss: Extract intermediate features from the 12th Transformer block (middle position of 24 layers), compute dispersion loss after global average pooling, ensuring feature extraction position aligns with base architecture (encoder-decoder boundary in DiT-L/2-SSL). This architecture balances image generation quality and computational efficiency, suitable as baseline model for multi-architecture validation, verifying RSSD framework adaptability on standard DiT architecture.

4.2. DiT-XL/2 Architecture

DiT-XL/2 [30] is the large-capacity variant in DiT series, using standard DiT architecture design (28 consecutive Transformer blocks, no encoder/decoder separation), with same structure as DiT-L/2 but larger scale (hidden_size=1152, depth=28, num_heads=16). Larger model capacity enables lower FID and MSE under same training epochs, demonstrating stronger image generation capability and reconstruction accuracy. Training employs gradient accumulation, mixed precision and zero initialization strategies to ensure stability.

RSSD/BadRSSD Adaptation: RSSD framework adaptation method same as DiT-L/2, but optimized for larger-scale model: PCA Space Noise Injection: Using same block-wise PCA processing mechanism, adapted to DiT-XL/2 patch embedding. Conditional Loss Function: Using same loss function design, leveraging larger model capacity to enhance attack effectiveness. Dispersive Loss: Extract intermediate features from the 14th Transformer block (middle position of 28 layers), adopting denser feature extraction strategy (e.g., every 4 layers), fully utilizing deep representations, verifying BadRSSD effectiveness on large-scale models. This architecture suits scenarios pursuing optimal performance, verifying BadRSSD effectiveness on large-scale models and RSSD framework scalability.

4.3. U-ViT Architecture

U-ViT [2] employs encoder-bottleneck-decoder U-Net topology, transforming ViT into multi-scale hierarchical architecture. Encoder performs gradual downsampling through Patch Merging (resolution decreases, channel number increases), stacking Transformer blocks at each level to extract multi-scale features; bottleneck layer consists of multiple Transformer blocks carrying global context information; decoder performs gradual upsampling through PixelShuffle or transposed convolution, fusing with corresponding encoder level features via skip connections to achieve detail restoration. Timestep conditioning is injected through AdaLN in multi-scale Transformer blocks.

RSSD/BadRSSD Adaptation: U-ViT's multi-scale structure provides rich adaptation points for RSSD framework: PCA Space Noise Injection: Perform block-wise PCA processing at encoder input stage, adapting to U-ViT multi-scale patch embedding mechanism. Since U-ViT uses different patch sizes at different scales, PCA block size needs adaptive adjustment according to current scale. Conditional Loss Function: Using same conditional loss function design, leveraging U-ViT multi-scale features to enhance attack effectiveness. Dispersive Loss: Extract multi-scale intermediate features from encoder levels, bottleneck layer and decoder levels, providing rich intermediate representations for Dispersive Loss. Feature extraction positions select middle encoder levels (e.g., encoder level 3) and bottleneck layer, ensuring consistency with base architecture feature extraction strategy. This architecture naturally outputs multi-scale features (encoder layers + bottleneck + decoder layers), combining Transformer's global modeling capability with U-Net's multi-scale fusion advantage, demonstrating good performance and convergence speed in multi-architecture validation, verifying RSSD framework adaptability on multi-scale architectures.

4.4. Swin-UNet Architecture

Swin-UNet [3] is based on Swin Transformer hierarchical pyramid structure, employing window attention mechanism (Window-based Multi-head Self-Attention, Win-MSA) and shifted window attention (Shifted Win-MSA) alternately applied, achieving local window self-attention computation and cross-window information interaction. Encoder establishes hierarchical structure through Patch Merging (C→2C→4C→8C), decoder performs gradual upsampling through Patch Expanding and fuses with corresponding encoder layer skip connections. This architecture doesn't rely on explicit position encoding, but implicitly encodes spatial information through relative position bias. Timestep conditioning is injected through AdaLN in Swin Block, consistent with DiT series. Window attention mechanism makes computational complexity approximately linear to image resolution, significantly saving memory and computation time in high-resolution scenarios.

RSSD/BadRSSD Adaptation: Swin-UNet's hierarchical structure provides efficient adaptation scheme for RSSD framework: PCA Space Noise Injection: Perform block-wise PCA processing at encoder input stage, adapting to Swin-UNet hierarchical patch embedding mechanism. Since Swin-UNet uses different window sizes and channel numbers at different levels, PCA block size needs adaptive adjustment according to current level. Conditional Loss Function: Using same conditional loss function design, leveraging Swin-UNet hierarchical features to enhance attack effectiveness. Dispersive Loss: Extract hierarchical intermediate features from encoder levels (C, 2C, 4C, 8C), selecting middle encoder levels (e.g., 2C or 4C level) for feature extraction, ensuring consistency with base architecture feature extraction strategy. Swin-UNet hierarchical features naturally adapt to multi-scale Dispersive Loss computation. This architecture verifies BadRSSD effectiveness on efficient Transformer architectures, window attention mechanism efficiency enables RSSD framework to maintain computational efficiency in high-resolution scenarios, verifying method practicality and scalability.

4.5. Summary

Evaluation experiments on the above four architectures demonstrate that the core components of RSSD and BadRSSD frameworks (PCA space noise injection, conditional loss function, Dispersive Loss) possess good architecture independence and universality, and can be adapted to different types of Transformer diffusion model architectures. During adaptation, mainly adjusting feature extraction layer positions and PCA block size settings, while keeping core algorithms unchanged, this proves the rationality of RSSD framework design and method generalizability.

5. Robustness Experiment Explanation and Evaluation Metric Calculation Details

5.1. DisDet

DisDet [38] (Distribution-based Detection) uses distribution differences: detecting marginal statistical differences between clean and poisoned samples. Premise: backdoor attacks cause detectable distribution differences. Process: distribution difference calculation (compute marginal statistical features for clean/poisoned samples to get PDD, larger PDD indicates more significant differences); Detection model training (train binary classifier using PDD features to distinguish samples); Backdoor sample handling (remove triggers from or reject detected poisoned samples).

Evaluation metrics:

- PDD(Clean) and PDD(Poison): Distribution difference scores for clean/poisoned samples, computed from marginal statistical features (mean, variance, higher-order moments, etc.), difference $\Delta\text{PDD} = \text{PDD(Poison)} - \text{PDD(Clean)}$, larger ΔPDD indicates more significant differences
- AUROC: Area under ROC curve, computed by varying detection threshold to calculate TPR and FPR at different thresholds then plotting ROC curve, 1.0 perfect, 0.5 random
- TPR@1%FPR: TPR at fixed FPR=1%, obtained by adjusting threshold to achieve FPR=0.01 then reading corresponding TPR, higher values indicate better detection capability at low false positive rates
- Detection Pass Rate: Percentage of poisoned samples passing detection (escape rate), $\text{Detection Pass Rate} = (1 - \text{TPR}) \times 100\%$, lower values indicate better defense effectiveness
- ASR_{before} and ASR_{after}: Attack success rate before/after defense, $\text{ASR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left[\text{SSIM}\left(f\left(x_i^{\text{poisoned}}\right), y_{\text{target}}\right) > \tau_{\text{ASR}}\right]$, defense effectiveness measured by $\Delta\text{ASR} = \text{ASR}_{before} - \text{ASR}_{after}$
- ΔFID: Generation quality change after defense, $\Delta\text{FID} = \text{FID}_{after} - \text{FID}_{before}$, where $\text{FID} = \|\mu_{real} - \mu_{gen}\|_2^2 + \text{Tr}(\Sigma_{real} + \Sigma_{gen} - 2(\Sigma_{real}\Sigma_{gen})^{1/2})$

Results: Effective against BadDiffusion and TrojDiff, $\Delta\text{PDD} \approx 0.45 - 0.44$, AUROC 0.92-0.95, TPR@1%FPR 85.46-87.12%, Detection Pass Rate 26.23-32.17%, ASR reduced from 71.02-73.18% to 8.26-9.45% ($\Delta\text{ASR} \approx 62 - 64\%$). Ineffective against BadRSSD: $\Delta\text{PDD} = 0.12$, AUROC 0.58 (near random), TPR@1%FPR 8.72%, Detection Pass Rate 85.68%, ASR reduced from 94.67% to only 92.57% ($\Delta\text{ASR} = 2.10\%$), $\Delta\text{FID} = 4.63$ (positive). Failure reasons: BadRSSD maintains marginal distribution stability through L_{disp}, clean/poisoned samples have almost identical statistical features; backdoor effects mainly in encoded PCA subspace and later timesteps, conditional differences at low poisoning rates get "averaged" into marginal statistics, resulting in small PDD separation; through PCA space alignment and dispersion loss, backdoor effects don't produce significant differences at marginal statistical level.

5.2. Elijah

Elijah [1] is a backdoor defense method based on trigger inversion and neuron pruning. Process: Trigger inversion ($r^* = \arg\min_r \sum_{i=1}^{N} \mathcal{L}(f(x_i + r), y_{target})$), Neuron localization (analyze neuron responses to triggers, locate relevant neuron clusters), Neuron pruning (prune relevant neuron clusters to remove backdoor).

Evaluation metrics:

- Poison Rate: $\text{Poison Rate} = \frac{N_{poisoned}}{N_{total}} \times 100\%$, table uses six poisoning rates: 5%, 10%, 20%, 30%, 40%, 50%
- Detected(%): $\text{Detected}(\%) = \frac{N_{detected}}{N_{total}} \times 100\%$, higher detection rate indicates more accurate backdoor identification
- ASR_{before} and ASR_{after}: Attack success rate before/after defense, $\text{ASR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left[\text{SSIM}\left(f\left(x_i^{\text{poisoned}}\right), y_{\text{target}}\right) > \tau_{\text{ASR}}\right]$, defense

effectiveness measured by $\Delta\text{ASR} = \text{ASR}_{before} - \text{ASR}_{after}$

- ΔSSIM: Change in structural similarity between generated and target images after defense, $\Delta\text{SSIM} = \text{SSIM}_{after} - \text{SSIM}_{before}$, where $\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$, $\Delta\text{SSIM} < 0$ indicates backdoor function removed, $\Delta\text{SSIM} \approx 0$ indicates backdoor function not removed
- ΔFID: Generation quality change after defense, $\Delta\text{FID} = \text{FID}_{after} - \text{FID}_{before}$, where $\text{FID} = \|\mu_{real} - \mu_{gen}\|_2^2 + \text{Tr}(\Sigma_{real} + \Sigma_{gen} - 2(\Sigma_{real}\Sigma_{gen})^{1/2})$

Results: Effective against BadDiffusion, detection rate increased from 82.06% to 100%, ASR reduced from 71.02-92.65% to 4.02-6.07% ($\Delta\text{ASR} \approx 65 - 87\%$), $\Delta\text{SSIM} \approx -0.8 \sim -1.0$, ΔFID small and changes insignificant. Ineffective against BadRSSD: very low detection rate (5.12-18.08%), ASR reduced from 94.67-99.99% to only 92.57-98.65% ($\Delta\text{ASR} \approx 1 - 3\%$), $\Delta\text{SSIM} \approx 0$, ΔFID positive and increases with poisoning rate (-0.89 $\sim$ 8.5). Elijah fails against BadRSSD, because effective trigger direction resembles small, unstructured global perturbations difficult to invert into stable generalizable triggers; carrying pathways dispersed across layers and time, no concentrated neuron clusters for pruning; backdoor effects non-local and dispersed, cannot be removed by pruning specific neurons. Thus, BadRSSD's PCA space semantic alignment and full-time trajectory drive disperses backdoor effects across representation space and time dimensions, making Elijah's trigger inversion and neuron pruning ineffective for detection and removal, demonstrating the stealthiness and threat of representation-layer backdoor attacks.

5.3. TERD

TERD [28] (Trigger-based Reverse Engineering Defense) is based on trigger inversion: reverse-engineering generalizable structured triggers from model outputs for detection and mitigation. The premise is that backdoors use fixed structured triggers (e.g., patches) detectable in pixel domain. Process: Trigger inversion $r^* = \arg\min_r \mathbb{E}_{x\sim\mathcal{D}}\left[\mathcal{L}(f(x + r), y_{target})\right] + \lambda \cdot \mathcal{R}(r)$ (λ is parameter η in table); Trigger detection: use r^* to determine if input contains backdoor ($\mathcal{L}(f(x + r^*), y_{target}) < \tau$); Trigger removal: perform $x_{cleaned} = x - r^*$ on detected samples.

Evaluation metrics:

- ASR_{before} and ASR_{after}: Attack success rate before/after defense, $\text{ASR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left[\text{SSIM}\left(f\left(x_i^{\text{poisoned}}\right), y_{\text{target}}\right) > \tau_{\text{ASR}}\right]$, $\Delta\text{ASR} = \text{ASR}_{before} - \text{ASR}_{after}$ measures defense effectiveness
- ΔFID: Generation quality change after defense, $\Delta\text{FID} = \text{FID}_{after} - \text{FID}_{before}$, where $\text{FID} = \|\mu_{real} - \mu_{gen}\|_2^2 + \text{Tr}(\Sigma_{real} + \Sigma_{gen} - 2(\Sigma_{real}\Sigma_{gen})^{1/2})$
- TPR: True positive rate, $\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}$
- TNR: True negative rate, $\text{TNR} = \frac{\text{TN}}{\text{TN+FP}}$
- $\|r - r_0\|_2$: L2 distance between inverted and real triggers, $\|r - r_0\|_2 = \sqrt{\sum_{i,j,k}(r_{i,j,k} - r_{0,i,j,k})^2}$

Results: Effective against BadDiffusion: ASR reduced from 71.02-68.19% to 8.12-7.65% ($\Delta\text{ASR} \approx 60 - 63\%$), TPR 76.56-85.12%, $\|r - r_0\|_2$ 0.18-0.26. Ineffective against BadRSSD: ASR reduced from 94.67-92.51% to only 92.35-91.76% ($\Delta\text{ASR} \approx 0.75 - 2.32\%$), TPR 4.28-6.74%, $\|r - r_0\|_2$ 0.75-0.82, ΔFID 4.52-8.16. TNR 96-97% for both, indicating normal identification of clean samples; problem lies in difficulty detecting poisoned samples. TERD fails against BadRSSD due to assumption-mechanism mismatch: TERD assumes fixed structured triggers, while BadRSSD is driven by PCA semantic alignment and full-time trajectory with non-local, dispersed activation directions; TERD assumes pixel-domain detectability, while BadRSSD effects mainly in encoded PCA subspace and later timesteps, causing misalignment between pixel-domain inversion and real causes; TERD assumes localized triggers, while BadRSSD effective triggering resembles small, unstructured global perturbations difficult to invert into stable generalizable triggers.