USENIX Security '26 Artifact Appendix: VOID: Defeating Unauthorized Mimicry in Latent Diffusion Models

Chunlin Qiu[1], Ang Li[1], Tianxiao Huang[1], Ruilin Gan[2], Yunjie Ge[3], Shenyi Zhang[3], Huayi Duan[4], Lingchen Zhao[1], Chao Shen[5], Qian Wang[1]*
[1] School of Cyber Science and Engineering, Wuhan University,
[2]School of Computer Science, Wuhan University,
[3]Institute for Math&AI, Wuhan University,
[4]The Hong Kong University of Science and Technology (Guangzhou),
[5]School of Cyber Science and Engineering, Xi'an Jiaotong University

A Artifact Appendix

A.1 Abstract

This final artifact accompanies the *Artifacts Available*, *Artifacts Functional*, and *Results Reproduced* badges awarded to VOID. It includes the VOID implementation, state-of-the-art protection baselines, representative mimicry attacks, evaluation scripts, configuration files, and documentation for running a functional test and a reduced reproduction workflow. The artifact is permanently available under the stable concept DOI https://doi.org/10.5281/zenodo.20233998.

A.2 Description & Requirements

The full evaluation in the paper considers 24 defenses, 10 mimicry/editing methods, and 5 datasets, exceeding the time and compute normally available for artifact evaluation. The AE workflow uses a reduced setting: TI-Dataset [2], two strong image-protection baselines, ACE [12] and Smooth [1], and four mimicry/editing methods: SDEdit/img2img [7], inpainting [9], LoRA [5], and fine-tuning [9]. The following subsections describe access, hardware, software, and benchmark requirements for this evaluation.

A.2.1 Security, privacy, and ethical concerns

The artifact is non-destructive. It does not disable security mechanisms, modify the operating system, or perform network attacks. Experiments run local protection, editing, personalization, and metric-computation pipelines on benchmark images; no private reviewer data is required. Mimicry/editing methods are used only to evaluate defenses. Preparation may download public resources.

*Corresponding author.

A.2.2 How to access

The final artifact is permanently available under the stable concept DOI https://doi.org/10.5281/zenodo.20233998, which resolves to the latest Zenodo version. After downloading and extracting the package, evaluators start from `AE_README.md`, which describes installation, model and dataset preparation, functional testing, reduced reproduction, expected outputs, runtime estimates, and troubleshooting.

A.2.3 Hardware dependencies

The artifact requires an NVIDIA GPU with CUDA support. We recommend at least one GPU with 24 GB of VRAM. When multiple GPUs are available, the scripts can assign different subjects to different GPUs to reduce total runtime; GPU memory is not pooled across devices.

Automatic inference batch-size selection probes the available GPU memory and adjusts the batch size accordingly. On high-memory GPUs, it may select an overly large inference batch size, which can cause a CUDA out-of-memory error. For the safest configuration, evaluators should use `--editing-batch-size 1`. LoRA and fine-tuning already use a training batch size of 1. We also recommend at least 150 GB of free disk space for models, datasets, and generated outputs.

A.2.4 Software dependencies

Required software includes a Linux shell environment, Conda or Anaconda, Python 3.10, and the packages in `requirements.txt`. The setup script creates the conda environment and installs these packages. Network access is needed unless models and datasets are already cached. The model-preparation script downloads or verifies Stable Diffusion v1.5, Stable Diffusion v1.5 Inpainting, BLIP, CLIP ViT-B/32, and SAM ViT-H. Download fallbacks and local-mirror instructions are documented in `AE_README.md`.

A.2.5 Benchmarks

The reduced AE evaluation requires only TI-Dataset [2]. The script `prepare_datasets.sh` with `--dataset ti` restores it under `Datasets/TexturalInversion`. The functional test uses one TI-Dataset subject. The reduced reproduction evaluates clean, VOID, Smooth, and ACE outputs under SDEdit/img2img, inpainting, LoRA, and fine-tuning.

A.3 Set-up

All commands should be run from the repository root after extracting the artifact package.

A.3.1 Installation

Conda or Anaconda must be available before installation. Prepare the environment, models, and dataset as follows:

```
bash scripts/prepare_env.sh
conda activate void-ae
bash scripts/check_gpu.sh
bash scripts/prepare_models.sh
bash scripts/prepare_datasets.sh --dataset ti
```

`prepare_env.sh` creates the `void-ae` environment and installs Python dependencies. `check_gpu.sh` checks CUDA and PyTorch GPU visibility. `prepare_models.sh` downloads or verifies the required models under `Models/`. `prepare_datasets.sh` with `--dataset ti` restores TI-Dataset under `Datasets/TexturalInversion`. After these steps, evaluators can run the basic functionality test below.

A.3.2 Basic Test

Run the basic functionality test with:

```
bash scripts/functional.sh --subject clock
```

The `--subject clock` option selects a deterministic TI-Dataset subject; if omitted, the script selects one subject automatically. A successful run writes outputs to `Results/Functional/TexturalInversion/`, including `manifest.json`, `metrics.json`, protected samples, and editing results.

A.4 Evaluation workflow

This artifact evaluates one functionality claim and two result-reproduction claims in the reduced AE setting. C1 is evaluated by the functional test E1. C2 and C3 are evaluated by the reduced reproduction experiment E2.

A.4.1 Major Claims

(C1): Artifact functionality. The artifact runs end-to-end on one selected subject from TI-Dataset [2]. It verifies environment setup, model loading, protected image generation, SDEdit/img2img [7] editing, and metric computation. This claim is evaluated by E1.

(C2): Protected-image visual utility. VOID generates protected images that remain visually close to the clean originals under the evaluated perturbation budget in the reduced AE setting. This claim is evaluated by E2 using PSNR [6], SSIM [10], and LPIPS [11].

(C3): Mimicry disruption. VOID disrupts unauthorized mimicry across SDEdit/img2img [7], inpainting [9], LoRA [5], and fine-tuning [9]. In the reduced AE setting, VOID is expected to produce more corrupted, noise-like outputs than ACE [12] and Smooth [1] on protected test images. This claim is evaluated by E2 using FID [4], CLIP-I [8], CLIP-S [3], and visual inspection.

A.4.2 Experiments

(E1): Functional test [10 human-minutes + 1 compute-hour]: This experiment evaluates C1 by running the end-to-end functionality test on one TI-Dataset subject.
How to: Run the functional script and inspect the generated metrics and samples under `Results/Functional/TexturalInversion/`.
Preparation: Complete the installation steps above. Ensure that the `void-ae` environment is active, models are prepared, and TI-Dataset is available.
Execution: Run:
`bash scripts/functional.sh --subject clock`
Results: Inspect the output file `Results/Functional/TexturalInversion/metrics.json` and the generated samples. The expected sanity-check result is that original-vs-protected images usually satisfy LPIPS $<$ 0.1, PSNR $\geq$ 32 dB, and SSIM $\geq$ 0.8, while protected SDEdit/img2img outputs lose subject semantics.

(E2): Reduced reproduction [About 10 human-minutes + about 8 compute-hours]: This experiment evaluates C2 and C3 by running the reduced reproduction workflow.
How to: Run the reproduction script and inspect the summary metrics, protected images, editing outputs, and logs under `Results/Reproduce/`, `Results/ProtectResult/`, and `Results/EditResult/`.
Preparation: Complete the installation steps above. The experiment uses TI-Dataset, Stable Diffusion v1.5, and the SD v1.5 inpainting checkpoint. It evaluates clean, VOID, Smooth, and ACE under SDEdit/img2img, inpainting, LoRA, and fine-tuning. When multiple GPUs are available, the script can assign different subjects to different GPUs; their GPU memory is not pooled.
Execution: For the safest GPU-memory configuration, run:
`bash scripts/reproduce.sh --editing-batch-size 1`
Omitting `--editing-batch-size 1` enables automatic inference batch-size selection. If outputs already

exist, recompute metrics only:
`bash scripts/reproduce.sh --only-evaluate`
Results: Main summaries are written under `Results/Reproduce/TexturalInversion/`, including `manifest.json` and `metrics.json`. Protected images are stored under `Results/ProtectResult/`, editing outputs under `Results/EditResult/`, and logs under `Results/Reproduce/`. The artifact reports PSNR, SSIM, LPIPS, FID, CLIP-I, and CLIP-S. Higher PSNR/SSIM, lower LPIPS, higher FID, and lower CLIP-I/CLIP-S indicate the expected trend.
The expected comparative trend corresponds to Table 3 of the paper. Table 3 reports averages over five datasets, whereas E2 evaluates only TI-Dataset. Therefore, exact values may differ; the reduced evaluation focuses on reproducing the same direction of improvement.

A.5 Notes on Reusability

The artifact separates protection generation, image editing/personalization, and metric computation, so these components can be reused independently. Users can extend the workflow by changing the dataset subset, protection method, editing method, perturbation budget, or available GPUs. Generated protection and editing outputs can also be reused for metric-only evaluation with `reproduce.sh --only-evaluate`. The reduced AE scripts provide an entry point for extending the full evaluation framework.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2026/.

References

[1] Junxi Chen, Junhao Dong, and Xiaohua Xie. Exploring adversarial attacks against latent diffusion model from the perspective of adversarial transferability. *arXiv preprint arXiv:2401.07087*, 2024.

[2] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations*, 2022.

[3] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[4] 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.

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

[6] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

[7] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[9] 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.

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

[11] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[12] Boyang Zheng, Chumeng Liang, and Xiaoyu Wu. Targeted attack improves protection against unauthorized diffusion customization. *arXiv preprint arXiv:2310.04687*, 2023.