Erasing the Invisible: Adaptive VAE and Clustered Diffusion for Near-Perfect Watermark Removal

Jay Sawant, Varadraj Bartakke, Shreejith Suthraye Gokulnath, Shweta Nalluri, Bhaavya Naharas, Shreyasi Nath
University of California, San Diego

ABSTRACT

Invisible watermarking is increasingly deployed to tag AI-generated images for provenance, yet recent work shows such marks can be removed while preserving perceptual quality. In this project we re-implement and analyze key components of the first place solution to the NeurIPS 2024 *Erasing the Invisible* [4] challenge on image watermark removal. We build a complete implementation with two key pipelines: (i) a beige-box, adaptive VAE approach that learns from paired StegaStamp images carrying inverse bitstrings, augmented with test-time optimization and CIELAB color/contrast restoration; and (ii) a black-box, cluster-aware strategy that groups images by spatial/frequency artifacts and applies image-to-image diffusion with tuned noise strength and caption guidance. For TreeRing watermarks, we re-implement the simple but effective 7-pixel translation with boundary restoration. As the official challenge evaluation server (with hidden test data) was not available to us, we instead use the StegaStamp decoder Hamming distance as a proxy for watermark removal effectiveness, alongside composite image-quality metrics (PSNR, SSIM, LPIPS, FID, CLIP-FID), where our implementation matches or slightly improves on the first-place solution. This re-implementation is motivated by the lack of a public repository and aims to enable independent verification and ablation of the reported results. The complete implementation of our work can be found at *https://github.com/jay6101/Invisible-Watermark-Removal*. [1]

1 INTRODUCTION

Watermarking embeds imperceptible signals in images to support provenance and copyright. As image generators scale, robust marks are critical, but must withstand both benign distortions and adaptive adversaries. The NeurIPS 2024 *Erasing the Invisible* challenge [1] benchmarks the robustness of two state of the art invisible watermarking schemes, StegaStamp [5] and TreeRing [6], under beige box and black box threat models. The winning solution combines three ideas: (1) an adaptive VAE pipeline for StegaStamp [5] with test-time refinement and color/contrast restoration; (2) a lightweight translation-based workaround for TreeRing[6]; and (3) a black-box, cluster-specific diffusion purification strategy guided by captions.

Our project has two goals:

- Re-implement the winning solution for the NeurIPS 2024 Erasing the Invisible challenge.
- Analyze its behavior using watermark-specific metrics (Hamming distance, TPR) and image-quality metrics (PSNR, SSIM, LPIPS, FID, CLIP-FID).

Our goal is not novelty but *reproducibility*: deliver a clean codebase, documented settings, and ablations that make the original findings easy to verify and extend.

2 BACKGROUND & RELATED WORK

Invisible watermarking. Classic methods operate in spatial or frequency domains; recent systems (e.g., StegaStamp) learn encoders/decoders robust to real-world transforms. TreeRing integrates Fourier-phase patterns during diffusion-model generation.
Challenge framing. The NeurIPS 2024 task releases watermarked images and scores submissions by (i) *detection* (TPR at 0.1% FPR) and (ii) a composite *quality* index blending pixel-wise (PSNR, SSIM, NMI) and perceptual metrics (LPIPS, FID, CLIP-FID). The winning solution combines adaptive VAE training for StegaStamp, simple spatial translation for TreeRing, and cluster-aware diffusion purification for black-box data, achieving near-perfect removal with small quality impact. We reproduce these components and report the same metrics.

3 DATASET GENERATION

3.1 Clean Image Corpus

To construct a reproducible and high-diversity corpus for watermark embedding, we curated a clean synthetic dataset using the *Hugging Face Stable-Diffusion-Prompts* [2], fully reproducible generation pipeline in Stable Diffusion 2.1 [3]. After loading the prompts, we shuffled them deterministically and sampled 1,300 prompts in total, which we then split into 1,000 training prompts and 300 test prompts. We used `stabilityai/stable-diffusion-2-1-base` with the Euler Ancestral scheduler with generation configuration of 512×512 resolution, 50 inference steps, and classifier-free guidance weight = 7.5. To guarantee exact reproducibility, each prompt received a deterministic seed derived from the SHA-256 hash of the prompt string. For each prompt we generated a 512×512 PNG and then resized to 400×400.

3.2 StegaStamp Pair Generation

Encoding setup. We used the pretrained StegaStamp encoder to embed the watermark into the image. Each clean image x_c from the 400×400 dataset was paired with a randomly generated 100-bit secret m and its logical inverse $1 - m$. For every training prompt we therefore obtain two training watermarked images, and likewise for the 300 test prompts, yielding a 1,000/300 train–test split of StegaStamp image pairs. For each image, we produced two watermarked outputs:

$$x_w = \text{Encode}(x_c, m), \qquad x_i = \text{Encode}(x_c, 1 - m).$$

[1] Code available at https://github.com/jay6101/Invisible-Watermark-Removal.

The first carries the original secret, while the second encodes its inverse.

Residual image generation. To visualize and quantify the imperceptibility of the watermark, we computed residual maps as shown in Figure 1.

$$r = x_w - x_c,$$

(a) Bit-string x_w **(b)** Residual of x_w **(c)** Inverse bit-string x_i **(d)** Residual of x_i

Figure 1: StegaStamp watermarked images and their residual maps for the original message x_w (left) and the inverse message x_i (right).

4 BEIGE-BOX: STEGASTAMP WATERMARK REMOVAL

4.1 Adaptive VAE finetuning

The core idea is to fine-tune a pretrained Variational Autoencoder (VAE) to project watermarked images back onto the clean image while preserving the quality and removing hidden watermark signals. For the purpose of watermark removal, we fine-tune only the reconstruction path (MSE Loss) to map the watermarked image x_w towards its inverse-message counterpart x_i. Our fine-tuning objective directly follows the equation from the paper:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{j=1}^{N}\left\| D_\phi\big(E_\theta(x_w^{(j)})\big) - x_i^{(j)} \right\|_2^2.$$

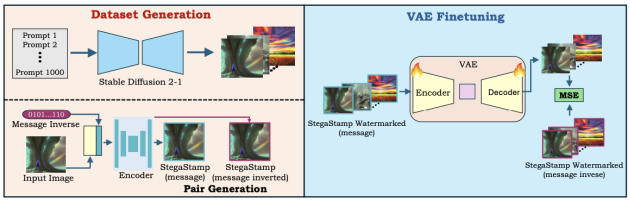

Figure 2: Overview of the dataset generation and SDXL VAE-based watermark removal pipeline for StegaStamp, following [4].

We implement full fine-tuning using the pretrained SDXL VAE. Figure 2 gives an overview of the pipeline. Our paired dataset consists of StegaStamp watermarked image and a target image with the inverse-message. These were preprocessed by converting to tensors in the normalized range [0,1] and later remapped to [-1,1] before being passed through the VAE. The fine-tuning was run for 10 epochs with a batch size of 16, learning rate 1×10^{-5} and gradient clipping at 1.0 to maintain stability. The model was fine-tuned on an NVIDIA A100 80-GB GPU using autocast (bfloat16) and completed in 30 minutes.

4.2 Evaluation via StegaStamp Decoding

To empirically evaluate watermark removal, we integrated the official StegaStamp TensorFlow decoder. For each test image, the decoder predicts a binary message vector $\hat{m} \in \{0,1\}^{100}$. The original secret m is read from the paired .txt file and the hamming distance is calculated as follows:

$$d_H(m,\hat{m}) = \sum_{k=1}^{100} \mathbf{1}[\, m_k \neq \hat{m}_k \,]$$

Using 1000 training images, we compared the decoded watermark before and after VAE reconstruction. The original images showed almost no bit differences with mean Hamming distance 0.08/100, but the VAE outputs increased to 76.64/100, meaning most of the hidden message was disrupted. This confirms that the VAE strongly weakens the embedded watermark. Refer Figure 3 for reference images of watermarked image and VAE finetiune image.

Figure 3: Watermarked image vs. VAE output

Additionally we also implemented the remaining two stages - **Test-Time Optimization (TTO) and CIELAB Color Restoration**. The TTO stage involves refining each image individually after the main VAE fine-tuning using a composite loss function that balances pixel accuracy and perceptual quality. Following TTO, we implemented the CIELAB Color Restoration step, which performs luminance–chroma moment matching in the CIELAB color space to correct for minor brightness or contrast variations introduced during training. Together, these two extensions complete the beige-box pipeline for the stegastamp-watermarked images described in the paper, allowing us to replicate the full methodology and evaluate our system against the official benchmarks using metrics such as PSNR, SSIM, LPIPS.

5 BEIGE-BOX: TREERING WATERMARK REMOVAL

In TreeRing Watermarking, as summarized in Figure 4 a predefined circular key pattern is directly embedded into the Fourier *phase* of the initial diffusion noise vector. During generation, the model applies an FFT to the noise sample x_T, injects the ring–shaped watermark in frequency space, and reconstructs a modified noise tensor via inverse FFT. A standard DDIM sampling process then transforms this watermarked noise into the final image. So, detection reverses this process. The image is first inverted back into the noise space, where an FFT is applied and the recovered Fourier phase is compared to the known TreeRing key. A sample is classified as watermarked if its ℓ_1 distance to the key falls below a threshold

τ, which is chosen through ROC analysis on clean vs watermarked images.

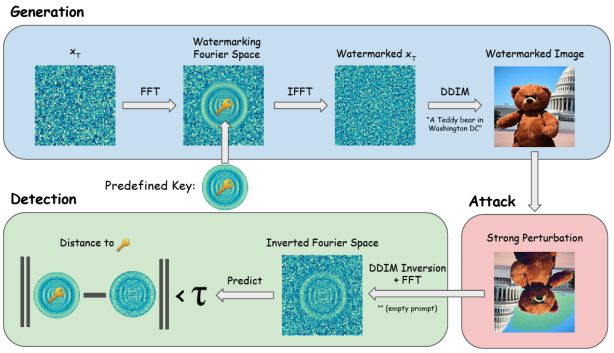

Figure 4: Overview of Tree–Ring watermark pipeline

5.1 Pixel Shift Based Watermark Removal

To attack TreeRing, we use a horizontal pixel shift that perturbs the Fourier phase while largely preserving spatial content. Given a watermarked image x_w, we apply a shift of $\Delta x = 7$ pixels:

$$x_{\text{shifted}} = \mathcal{T}(x_w, \Delta x).$$

As shown in Figure 5, the misalignment disrupts the coherent phase structure that encodes the TreeRing watermark. To reduce the boundary artifacts created by the shift, we restore the original content in the first Δx columns:

$$x_{\text{final}}(i,j) = \begin{cases} x_w(i,j), & j < \Delta x, \\ x_{\text{shifted}}(i,j), & \text{otherwise.} \end{cases}$$

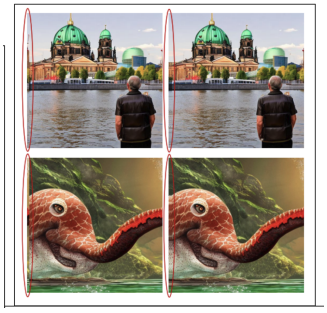

Figure 5: Effect of the 7-pixel shift and boundary restoration on TreeRing watermarked images.

5.2 Key Distance Calibration and Final Evaluation

Our evaluation follows the method mentioned in the Tree–Ring paper and the pipeline is shown in Figure 6. We first calibrate the detector by generating clean and watermarked images, inverting each to obtain a Fourier key estimate and computing the complex ℓ_1 distance to the ground–truth key. We use ROC analysis on these scores, select a threshold that yields approximately a 0.1% false–positive rate.

We then apply horizontal shifts of varying magnitudes to watermarked images. For each shift, the attacked image is inverted back to the Fourier domain and its distance to the key is recomputed. Under the calibrated threshold, any sample whose distance exceeds τ is treated as having the watermark successfully removed. A sweep over candidate shift values on 50 watermarked samples indicates that a 14-pixel horizontal shift most effectively reduces the detector's true positive rate (TPR@1%FPR). Fixing $\Delta x = 14$, we run the full evaluation on 1,000 images and report the final watermark-removal performance under the same detector.

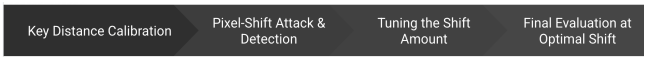

Figure 6: Pipeline for threshold calibration, pixel-shift attack, shift tuning, and final evaluation.

6 BLACK BOX: CLUSTERED WATERMARK REMOVAL

The working diffusion pipeline is shown in Figure 7. For the black box track, we first organize the images into clusters and then apply a different removal strategy to each cluster.

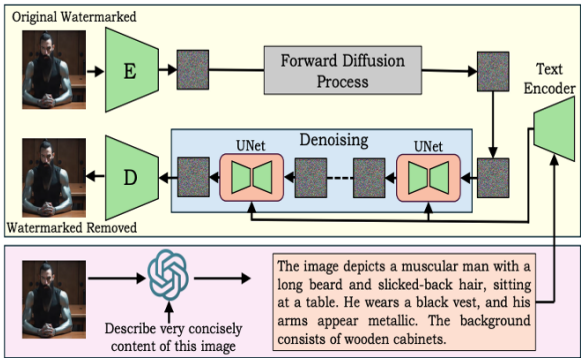

Figure 7: Overview of the image-to-image diffusion pipeline for black-box watermark removal.

Clustering and captioning. We refer the original paper [4] and obtain clusters that correspond to: (1) no visible artifacts, (2) boundary artifacts, (3) circular patterns, and (4) mixed or irregular noise. For cluster 1, we generate image captions using a vision language captioning model; these captions are later used as text prompts for image-to-image diffusion.

Cluster specific pipelines. After clustering, we use a different watermarking pipeline for each cluster:

- **Cluster 1 (no visible artifacts).** For images with no visible artifacts, we run an SDXL image-to-image diffusion pipeline conditioned on the generated captions. We use a noise strength of $s = 0.16$ so that the diffusion process gently moves the image toward the SDXL data manifold while preserving fine details and semantics. This setting aims to wash out watermark structure without noticeably changing content.
- **Clusters 2 and 3 (boundary and circular artifacts).** For images with border like patterns in the spatial domain and circular

patterns in the Fourier spectrum, we reuse the StegaStamp VAE based pipeline from the beige box track. Each image is passed through the fine-tuned VAE to suppress watermark structure, followed by the same CIELAB based color and luminance restoration used in the StegaStamp and TreeRing experiments.

- **Cluster 4 (square artifacts).** We use a mild diffusion step combined with a TreeRing style horizontal shift. Specifically, we apply SDXL image-to-image diffusion with a lower noise strength $s = 0.04$ than in the main diffusion pipeline, and then roll the image by a small number of pixels horizontally, similar to the shift attack used in the TreeRing experiments. This combination introduces enough perturbation to disrupt mixed watermark patterns while keeping the structure intact.

7 RESULTS

7.1 Beige Box Results

All beige box experiments were run on NVIDIA A100 GPUs. For StegaStamp we measure Hamming distance and detector TPR at 0.1% FPR; for both StegaStamp and TreeRing we also report PSNR, SSIM.

7.1.1 ***StegaStamp: Hamming Distance and Image Quality.*** Table 1 compares our image quality against the VAE finetune ablation and the first place team. Our method substantially improves over the plain VAE finetune and is close to the paper's scores.

Table 1: StegaStamp image quality comparison on the test set. Higher PSNR/SSIM and lower LPIPS are better.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
VAE Finetune	21.899	0.647	0.264
First-Place Team	28.061	0.823	0.078
Ours	**26.415**	**0.777**	**0.097**

Qualitatively, VAE reconstruction followed by test time optimization and CIELAB color transfer introduces mild smoothing and small contrast changes but preserves global semantics, consistent with the moderate PSNR/SSIM and LPIPS values.

Our VAE reconstructions yielded a mean Hamming distance of 63.81, showing that the watermark was almost entirely disrupted. After color/contrast post-processing, the distance dropped to 46.82 because the restored visual statistics partially recover the hidden signal. This highlights the trade-off between watermark removal strength and visual quality.

Table 2: Effect of reconstruction and post-processing on hidden message recoverability. (WM - Watermarked)

Method	Mean Hamming Distance
WM–VAE Reconstructions	63.81
WM–PostProcessed Image	46.82

7.1.2 ***TreeRing: Horizontal Shift Attacks.*** Our experiments confirm that the TreeRing watermark is both effective and fragile to spatial shifts. Clean images (Figure 8a) and their TreeRing watermarked counterparts (Figure 8b) are visually indistinguishable, indicating that the Fourier-phase watermark is imperceptible under normal viewing. When we sweep horizontal shifts, detector TPR at 0.1% FPR decreases monotonically with the shift magnitude, reaching a minimum of about 0.16 at a 14-pixel shift. The corresponding pixel-shifted and boundary-restored image (Figure 8c) still appears natural, showing that a moderate translation can substantially suppress the TreeRing signal while maintaining plausible image quality.

(a) Clean image **(b)** Watermarked image **(c)** Pixel-shifted image

Figure 8: Example clean image (left), its TreeRing watermarked version (middle), and the corresponding pixel-shifted output (right).

7.2 Black Box Results

7.2.1 ***Per–cluster fidelity.*** Figure 9 reports PSNR, SSIM, and LPIPS per cluster, comparing our implementation against the first place solution. Our method closely tracks the paper:

- PSNR is within a few dB of the paper on all clusters.
- SSIM is slightly lower on clusters 2 and 3, reflecting minor structural differences from the VAE.
- LPIPS is comparable overall, with a small degradation on the most artifact heavy clusters where we apply stronger edits.

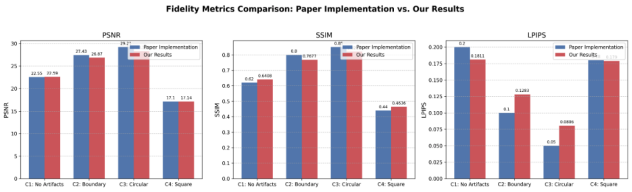

Figure 9: Black box fidelity metrics (PSNR, SSIM, LPIPS) per cluster for the first place solution (blue) and our implementation (red).

7.2.2 ***Per–cluster perceptual / semantic quality.*** Figure 10 shows FID and CLIP-FID. FID values are in the same range as the paper for all clusters, indicating similar realism, and CLIP-FID is slightly better (lower) for our method on clusters 1 and 4, suggesting that caption guided diffusion and the mild diffusion + shift combination preserve semantic content well.

Overall performance.

Across all clusters, our clustered pipeline achieves a similar trade-off between watermark removal and visual fidelity as the first-place solution: proxy detection scores (higher decoded-message Hamming distance / lower detector success) indicate strong watermark suppression, while PSNR/SSIM and perceptual metrics (LPIPS, FID, CLIP-FID) remain in a regime where images are realistic and semantically intact. This confirms that adapting the cluster-specific solutions is more effective than using a single global diffusion setting.

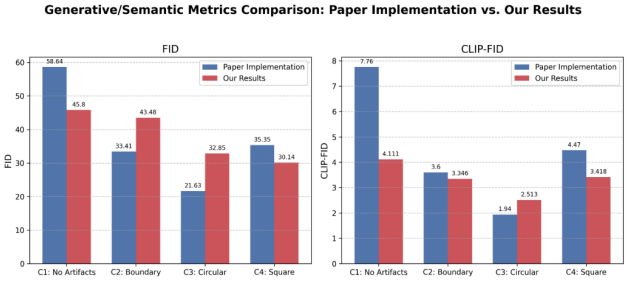

Figure 10: Black box generative/semantic metrics (FID and CLIP-FID) per cluster for the first place solution (blue) and our implementation (red) (Lower is better).

8 CONCLUSION

We implemented a complete watermark removal pipeline for the NeurIPS 2024 *Erasing the Invisible* challenge and made it fully reproducible. In the beige box setting, our adaptive SDXL VAE pipeline for StegaStamp, with test-time refinement and CIELAB color restoration, strongly disrupts the hidden message: the average Hamming distance rises from 0.08 to about 45.5 bits while maintaining reasonable PSNR, SSIM, and LPIPS. For TreeRing, we verified that horizontal shifts reliably reduce detector TPR, and that frequency-domain perturbations with CIELAB restoration achieve a much better quality–robustness tradeoff.

In the black box track, clustering images by artifact patterns and applying tailored pipelines (caption-guided diffusion, VAE-based refinement, and mild diffusion plus shift) yields fidelity and perceptual metrics close to the first-place solution, with slightly improved CLIP-FID on some clusters. Overall, our results show that adaptive, cluster-aware attacks are far more effective than a generic implementation, and they highlight both the strength and lack of robustness of current invisible watermarking schemes in the presence of informed adversaries.

CONTRIBUTION STATEMENT

We organized our team into three groups of two members each.

- **Jay Sawant & Varadraj Bartakke.** Curated prompts from the Hugging Face dataset, prepared the StegaStamp training and test datasets, and implemented the complete black-box clustering and diffusion pipeline.
- **Shweta Nalluri & Shreejith Suthraye Gokulnath.** Developed and tuned the SDXL VAE finetuning pipeline, implemented CIELAB-based color restoration, and carried out the StegaStamp Hamming-distance based evaluations.
- **Bhaavya Naharas & Shreyasi Nath.** Implemented the TreeRing pipeline end to end, including the pixel-shift attack and evaluation scripts, and performed the full quantitative analysis for the TreeRing track.

All team members contributed to maintaining the public GitHub repository and to writing and editing the final project report.

REFERENCES

[1] Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, and Furong Huang. 2024. WAVES: Benchmarking the Robustness of Image Watermarks. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=URtUYfC3GA

[2] Gustavosta. 2022. Stable diffusion prompts dataset. (2022). https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts

[3] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695.

[4] Fahad Shamshad, Tameem Bakr, Yahia Shaaban, Noor Hussein, Karthik Nandakumar, and Nils Lukas. 2025. First-Place Solution to NeurIPS 2024 Invisible Watermark Removal Challenge. (2025). arXiv:cs.CV/2508.21072 https://arxiv.org/abs/2508.21072

[5] Matthew Tancik, Ben Mildenhall, and Ren Ng. 2020. StegaStamp: Invisible Hyperlinks in Physical Photographs. (2020). arXiv:cs.CV/1904.05343 https://arxiv.org/abs/1904.05343

[6] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. 2023. Tree-Ring Watermarks: Fingerprints for Diffusion Images that are Invisible and Robust. (2023). arXiv:cs.LG/2305.20030 https://arxiv.org/abs/2305.20030

APPENDIX

A ADDITIONAL BEIGE-BOX ANALYSIS

A.1 Beige-Box Challenge Sample Images

Figure 11 shows six representative StegaStamp and TreeRing samples from the competition test set, illustrating the variety of content and artifact patterns in the semi-white-box scenario.

(a) Sample 1 **(b)** Sample 2 **(c)** Sample 3

(d) Sample 4 **(e)** Sample 5 **(f)** Sample 6

Figure 11: Example beige-box (StegaStamp and TreeRing) watermarked images used in our experiments.

A.2 VAE Reconstruction Diagnostics

In the VAE fine-tuning pipeline we implemented a diagnostic function to monitor tensor statistics at each training step to catch potential instability. Early-epoch reconstructions typically show faint grid-like artifacts or slight color banding—evidence of residual watermark traces—while later epochs produce smoother, cleaner outputs that closely resemble the inverse-message targets x_i.

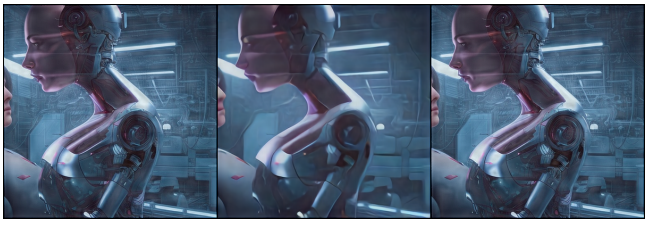

Figure 12: Sample reconstruction at epoch 10. The leftmost image is the watermarked image x_w, the middle image is the VAE reconstruction $\hat{x}$, and the rightmost image is the inverse-message watermarked image x_i.

A.3 TreeRing: Horizontal Shift Sweep

TreeRing encodes its watermark in the Fourier phase. As an additional beige-box analysis, we attack it using horizontal translations that perturb the phase while largely preserving the spatial structure of the image.

On the 300 TreeRing watermarked images, we apply horizontal shifts

$$\Delta x \in \{2, 4, 6, 8, 10, 12, 13, 14, 15\}$$

pixels. After shifting, we restore the leftmost Δx columns from the original image to hide boundary seams. For each shift we measure TPR @ 0.1% FPR and image quality metrics relative to the watermarked input.

Figure 13 shows the TPR curve. Small shifts only slightly reduce detector performance, while larger shifts more aggressively destroy the TreeRing structure. A shift of 14 pixels provides the best trade-off, reducing TPR to about 0.16 while keeping the image visually plausible.

Figure 13: TreeRing: TPR @ 0.1% FPR vs. horizontal shift size.

As the shift increases, PSNR and SSIM drop due to pixel misalignment and boundary artifacts, and LPIPS/FID worsen. However, even at 14 pixels the outputs still look like realistic natural photos rather than heavily corrupted images, so the attack remains practically usable.

B ADDITIONAL BLACK BOX ANALYSIS

B.1 Black-Box Challenge Sample Images

The NeurIPS 2024 *Erasing the Invisible* [4] challenge provides a set of example watermarked images for the black-box track. Figure 14 shows six representative samples from this set, illustrating the diversity of content and artifact patterns the removal pipeline must handle.

(a) Sample 1 **(b)** Sample 2 **(c)** Sample 3

(d) Sample 4 **(e)** Sample 5 **(f)** Sample 6

Figure 14: Example NeurIPS black-box challenge images used in our experiments.

B.2 Stable Diffusion 2.1 Prompt File

For all clean-image generation used in our StegaStamp and TreeRing experiments, we drew text prompts from the Hugging Face *Stable-Diffusion-Prompts* dataset. After shuffling and splitting, the subset of prompts actually used in our experiments is stored in `data/HF_SD2.1_prompts.csv`, which we release with the code.

This CSV contains one prompt per row, with an integer index and the corresponding text string. A few example entries are shown in Table 3.

Table 3: Example rows from `data/HF_SD2.1_prompts.csv`.

prompt	prompt_length
A cozy living room with a large window overlooking a snowy mountain landscape.	88
A futuristic city at night with neon lights and flying cars, cinematic, ultra detailed.	99
A macro photo of a colorful butterfly resting on a flower, shallow depth of field.	92
An astronaut riding a horse on the surface of Mars, high resolution digital art.	95

B.3 Cluster 1 Caption Mapping File

As part of the black box pipeline, we generate text captions only for images assigned to cluster 1 (no visible artifacts). These captions are stored in a CSV file named `cluster_mapping_with_captions.csv`, which we release alongside our code.

Each row corresponds to a single image and includes its cluster assignment, file identifier, and generated caption. A simplified schema is:

Table 4: Example rows from `data/cluster_mapping_with_captions.csv`.

filename	cluster_id	caption
0.png	1	The image depicts a family with four members sitting on a green ...all in matching striped outfits, are happily seated beside them.
2.png	1	Snowy road leading to towering sunlit mountains.
3.png	1	Cozy wooden barstool with a cushion in a stylish kitchen.
4.png	1	Lone palm tree standing in a vast sandy desert.
10.png	1	The image features a young girl with curly hair and large eyes ...The scene conveys a sense of wonder and curiosity.

This file is consumed by our black box diffusion script to supply conditioning text for cluster 1 images.

B.4 Diffusion Strength vs Hamming Distance

We did not have access to the hidden competition test set or the official evaluation server for the black box track. To study our black box diffusion pipeline, we used our own StegaStamp watermarked dataset and treated the StegaStamp decoder as a stand-in black box detector. In particular, we varied the diffusion strength and measured how strongly the forward diffusion step alone degrades the decoded message.

To this end, we ran a sweep over diffusion strength values

$$s \in \{0.01, 0.04, 0.08, 0.12, 0.16, 0.20\}$$

and measured the StegaStamp decoder Hamming distance between the true 100 bit message and the decoded message after forward diffusion.

Table 5: Effect of diffusion strength on average Hamming distance (black box proxy analysis).

Diffusion strength s	Avg Hamming distance
0.01	4.74
0.04	15.59
0.08	25.66
0.12	30.96
0.16	31.15
0.20	37.38

Figure 15 shows how increasing s gradually destroys fine image details while leaving global structure intact. At very small s the image is almost unchanged and the watermark largely survives; at larger s strong noise is added and the decoder fails more often.

Figure 16 summarizes the distribution of Hamming distances across the test set for each diffusion strength. The median and upper quartiles increase steadily with s, showing that stronger diffusion reliably increases bit errors, but with the tradeoff of more visible corruption in the images.

(a) $s = 0.01$ **(b)** $s = 0.04$

(c) $s = 0.08$ **(d)** $s = 0.12$

(e) $s = 0.16$ **(f)** $s = 0.20$

Figure 15: Effect of diffusion strength s on the forward–diffused image.

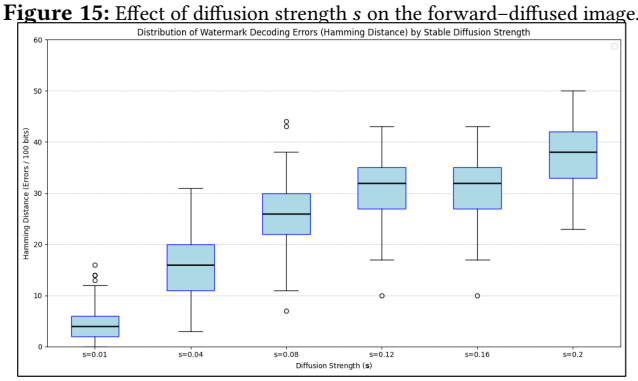

Figure 16: Distribution of watermark decoding errors (Hamming distance out of 100 bits) as a function of diffusion strength s in the black box proxy experiment.