WAVES: BENCHMARKING THE ROBUSTNESS OF IMAGE WATERMARKS

Bang An[**1] **Mucong Ding**[*1] **Tahseen Rabbani**[*1] **Aakriti Agrawal**[1] **Yuancheng Xu**[1]

Chenghao Deng[1] **Sicheng Zhu**[1] **Abdirisak Mohamed**[1,2] **Yuxin Wen**[1]

Tom Goldstein[1] **Furong Huang**[1]

[1]University of Maryland, [2]SAP Labs, LLC.

ABSTRACT

In the burgeoning age of generative AI, watermarks act as identifiers of provenance and artificial content. We present WAVES (**W**atermark **A**nalysis **v**ia **E**nhanced **S**tress-testing), a benchmark for assessing image watermark robustness, overcoming the limitations of current evaluation methods. WAVES integrates detection and identification tasks and establishes a standardized evaluation protocol comprised of a diverse range of stress tests. The attacks in WAVES range from traditional image distortions to advanced, novel variations of diffusive, and adversarial attacks. Our evaluation examines two pivotal dimensions: the degree of image quality degradation and the efficacy of watermark detection after attacks. Our novel, comprehensive evaluation reveals previously undetected vulnerabilities of several modern watermarking algorithms. We envision WAVES as a toolkit for the future development of robust watermarks.

1 INTRODUCTION

Diffusion models such as the open-source Stable Diffusion and proprietary models such as the Dall·E family and Midjourney have enabled users to produce artificial images that are of human-produced quality. Consequently, there has been a strong push in the AI/ML community to develop reliable algorithms for detecting AI-generated content and determining its source (Executive Office of the President, 2023). One avenue for maintaining the provenance of generative content is by embedding *watermarks*. However, a lack of standardized evaluations in existing literature (i.e., inconsistent image quality measures, statistical parameters, and types of attacks) has resulted in an incomplete picture of the vulnerabilities and robustness of these algorithms.

We present WAVES (**W**atermark **A**nalysis **v**ia **E**nhanced **S**tress-testing), a benchmark for assessing watermark robustness, overcoming the limitations of current evaluation methods. WAVES consists of a comprehensive variety of existing and novel variants of classical image distortions, image regeneration, and adversarial attacks. WAVES focuses on the sensitivity and robustness of watermark detection, measured by the true positive rate (TPR) at 0.1% false positive rate (FPR), and in the meantime, studies the severity of image degradations needed to decrease this sensitivity with multiple quality metrics. WAVES develops a series of Performance vs. Quality 2D plots varying over several prominent image similarity metrics, which are then aggregated in a heuristically novel manner to paint an overall picture of watermark robustness and attack potency.

We extensively evaluate the security of three prominent watermarking algorithms, Stable Signature (Fernandez et al., 2023), Tree-Ring (Wen et al., 2023), and StegaStamp (Tancik et al., 2020), respectively representing three major techniques for embedding an invisible signature. WAVES effectively reveals weaknesses in them and discovers previously undetected vulnerabilities. For example, watermarking algorithms using publicly available VAEs can have their watermarks

[*]Co-first-authors with equal contribution.

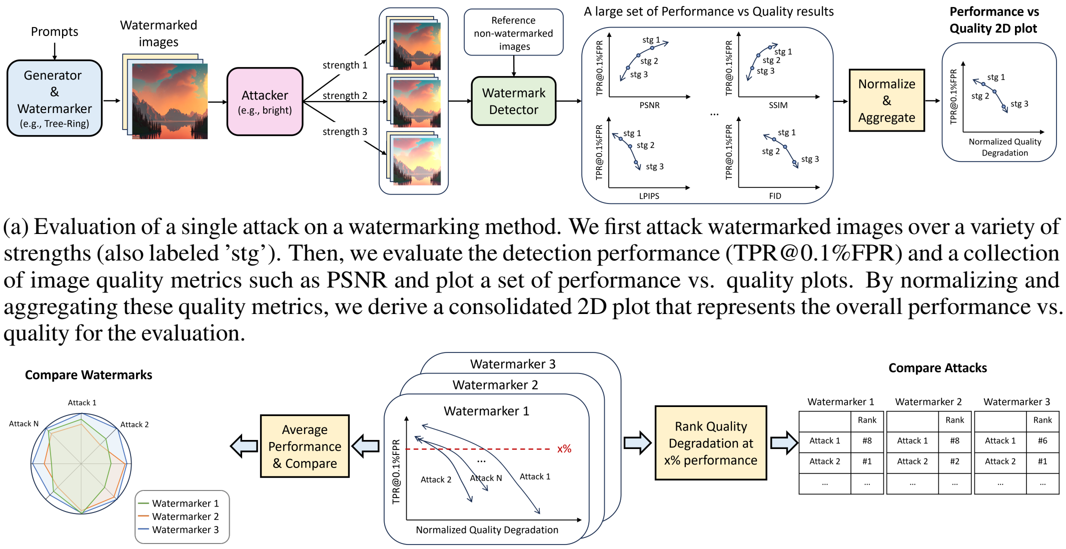
Table 1. **Comparison of robustness evaluations with existing works.** For *categories of attacks*, D, R, and A denote distortions, image regeneration, and adversarial attacks. *Joint test* means whether the performance and quality are jointly tested under a range of attack strengths. Our benchmark is the most comprehensive one, with a large scale of attacks, data, metrics, and more realistic evaluation setups.

Research Work	Num. of Attacks	Categories of Attacks	Num. of Datasets	Sample Size per Dataset	Non-watermarked Image Source	Performance Metric	Num. of Quality Metrics	Joint Test
StegaStamp Watermark[1]	5	D	1	1000	—	bit accuracy	3	✗
Stable Signature Watermark[2]	12	D, R	1	5000	—	bit accuracy	3	✗
TreeRing Watermark[3]	6	D	2	1000	generate by same model	TPR@1%FPR	2	✗
Regeneration Attack[4]	10	D, R	2	500	—	bit accuracy	3	✗
Surrogate Model Attack[5]	2	R, A	1	2500	real images	AUROC	0	✗
Adaptive Attack[6]	10	D, A	1	1000	real images	TPR@1%FPR	3	✗
WAVES (ours)	26	D, R, A	3	5000	real images	TPR@0.1%FPR	8	✓

[1] Tancik et al. (2020). [2] Fernandez et al. (2023). [3] Wen et al. (2023). [4] Zhao et al. (2023a). [5] Saberi et al. (2023). [6] Lukas et al. (2023).

effectively removed with minimal image manipulation. DALL·E3's usage of an open-source KL-VAE underscores the need for unique VAEs in such systems. Our **contributions** are summarized as follows:

(1) In practical scenarios where false alarms incur high costs, our evaluation metric for watermark detection prioritizes the True Positive Rate (TPR) at a stringent False Positive Rate (FPR) threshold, specifically 0.1%. This focus addresses the inadequacies of alternative metrics such as the p-value and Area Under the Receiver Operating Characteristic (AUROC).
(2) Additionally, our metric incorporates image quality alongside TPR@0.1% FPR. This integration acknowledges the necessity of maintaining a balance between reducing the accuracy of watermark detection and the practical utility of the image in practical scenarios.
(3) We introduce a comprehensive taxonomy of attacks that encompasses classical distortions and powerful, novel variations of regeneration and adversarial attacks, against watermarks.
(4) We standardize the evaluation of watermark robustness, allowing us to rank attacks. We formalize the watermark *detection* and *identification* problems and evaluate the robustness under both scenarios.
(5) Our benchmark uncovers several especially harmful attacks for popular watermarks, some of which are first introduced in this work. WAVES serves as a toolkit for future development of robust watermarks.



Watermarker 1

	Rank
Attack 1	#8
Attack 2	#1
...	...

Watermarker 2

	Rank
Attack 1	#8
Attack 2	#2
...	...

Watermarker 3

	Rank
Attack 1	#6
Attack 2	#1
...	...

(a) Evaluation of a single attack on a watermarking method. We first attack watermarked images over a variety of strengths (also labeled 'stg'). Then, we evaluate the detection performance (TPR@0.1%FPR) and a collection of image quality metrics such as PSNR and plot a set of performance vs. quality plots. By normalizing and aggregating these quality metrics, we derive a consolidated 2D plot that represents the overall performance vs. quality for the evaluation.

(b) Benchmarking watermarks and attacks. For each watermark, we plot all attacks on a unified performance vs. quality 2D plot to facilitate a detailed comparison. Based on this, we provide two additional analytical perspectives. We compare watermarks' robustness through the averaged performance under different attacks. We evaluate attacks' potency by ranking the quality at a specific performance threshold.

Figure 1. Evaluation workflow.

2 Standardized Evaluation through WAVES

2.1 Workflow and Metrics

As shown in Table 1, our benchmark, WAVES, stands out by considering three diverse datasets, incorporating 26 diverse attacks across three categories, and employing 8 quality metrics. These distinguish our work as the most extensive and realistic setup to date for watermark robustness evaluation. For more details on evaluation workflow, setups, metrics, and more analyses, see Appendix E.

Applications and formulation of invisible image watermarks. Invisible image watermarks, originally for protecting creators' intellectual property, have expanded into broader applications like **AI Detection** — identifying AI-generated images (Saberi et al., 2023), and **User Identification** — tracking the source of an image to its creator (Fernandez et al., 2023). We are interested in message-based approaches, where a unique, invisible identifier is embedded into an image. which may be recovered by the content creator at any time to establish provenance.

Evaluation Workflow. The trade-off between watermark performance and image quality, especially when watermark attacks lead to image distortions, is critical. We introduce *Performance vs. Quality 2D plots* for a comprehensive comparison, a novel perspective over the typical performance-centric analyses. The evaluation process involves comparing watermarked images with a diverse set of real and AI-generated reference images to produce the performance vs. quality 2D plots, and processing or aggregating the 2D plots to compare attacks and watermarks, as depicted in Figure 1.

Performance Metrics in AI Detection and User Identification. WAVES prioritizes fairness and comprehensiveness by using evaluation metrics that are independent of the choice of statistical tests and p-value thresholds. Given the significant impact of false positives in mislabeling non-watermarked images, strict control over the false positive rate (FPR) is crucial. WAVES focuses on TPR$@x\%$FPR, specifically at a challenging low FPR threshold of 0.1%. For user identification, we measure performance by the accuracy of correct image assignments to users.

Implementing Diverse Image Quality Metrics: Recognizing that no single metric can fully capture the aspects of generated images, we use a range of image quality metrics and propose a normalized, aggregated metric for evaluating watermark and attack methods. WAVES integrates over 8 metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Normalized Mutual Information (NMI), Frechet Inception Distance (FID) (Heusel et al., 2017), a variant based on CLIP feature space (CLIP-FID) (Kynkäänniemi et al., 2022), Perceptual Image Patch Similarity (LPIPS), and aesthetics and artifacts scores (Xu et al., 2023).

Normalization and Aggregation of Image Quality Metrics: Addressing the distinct characteristics of various image quality metrics, WAVES proposes *a normalized and aggregated quality metric* for a unified measure of image quality degradation and comprehensive scoring of attack or watermark methods. We define the normalized scale for each metric by assigning the 10% quantile value over all attacked images (across 26 attack methods, three watermark methods, and three datasets) as the 0.1 point, and the 90% quantile as the 0.9 point. *Normalized quality metrics are always ranked in ascending order of image degradation.* This normalization ensures equivalent significance across different metrics, defined by their quantiles in a large set of attacked watermarked images. Normalized metrics are aggregated and extensively utilized in Section 3 for Performance vs Quality plots, watermark radar plots, and attack leaderboards.

Table 2. A taxonomy of all the attacks in our stress-test set. Novel attacks proposed by WAVES are marked with *.

Category	Subcategory (prefix)	Description	Attack Names (suffix)
Distortion	Single (Dist-)	Single distortion	-Rotation, -RCrop, -Erase, -Bright, -Contrast, -Blur, -Noise, -JPEG
	Combination (DistCom-)	Combination of a type of distortions	-Geo, -Photo, -Deg, -All
Regeneration	Single (Regen-)	A single VAE or diffusion regeneration	-Diff, -DiffP[1], -VAE, -KLVAE[2]
	Rinsing* (Rinse-)	A multi-diffusion regeneration	-2xDiff, -4xDiff
Adversarial	Embedding (grey-box)* (AdvEmbG-)[3]	Use the same VAE	-KLVAE8
	Embedding (black-box)* (AdvEmbB-)	Use other encoders	-RN18, -CLIP, -KLVAE16, -SdxlVAE
	Surrogate detector attack* (AdvCLS-)[4]	Train a watermark detector	-UnWM&WM, -Real&WM, -WM1&WM2

[1] DiffP requires user prompts. [2] KLVAE with bottleneck size 8 is grey-box. [3] AdvEmbG is grey-box. [4] AdvCLS needs data and training.

2.2 Stress-testing Watermarks

We evaluate the robustness of watermarks with a wide range of attacks detailed in this section and summarized in Table 2. Figure 23 demonstrates the visual effects.

Distortion Attacks. Watermarked images often face distortions such as compression and cropping during internet transmission, necessitating watermarks that can endure common alterations. However, most studies only test resilience against singular or extreme distortions. In WAVES, we establish the following distortions within an acceptable quality threshold as our baselines. **Geometric distortions**: rotation, resized-crop, and erasing; **Photometric distortions**: adjustments in brightness and contrast; **Degradation distortions**: Gaussian blur, Gaussian noise, and JPEG compression; **Combo distortions**: combinations of geometric, photometric, and degradation distortions, both individually and collectively. Detailed setups for each are provided in the Appendix F.1.

Regeneration Attacks, employing diffusion models or VAEs Saberi et al. (2023); Zhao et al. (2023a), aim at altering an image's latent representation by noising and then denoising an image. Different from existing works that only perform a **Single regeneration**, we also investigate **Rinsing regenerations**, where an image undergoes multiple cycles of noising and denoising through a pre-trained diffusion model. Furthermore, we introduce two additional variations: prompted regeneration and mixed regeneration (rinse + VAE denoising). To simulate a realistic attack, we use a lower version diffusion model than the one used to generate watermarked images. All such attacks are detailed in Appendix F.2.

Adversarial Attacks. Deep neural networks are vulnerable to adversarial examples, (Ilyas et al., 2019; Chakraborty et al., 2018). In WAVES, we explore watermark robustness against two types of adversarial attacks. **Embedding Attacks** aims to perturb the latent feature of watermarked images while preserving invisible change in the image space. We examine if attacks on off-the-shelf embedding models can transfer to watermark detectors. We evaluate five off-the-shelf encoders: pre-trained ResNet18, image encoder of CLIP (Radford et al., 2021), KL-VAE(f8), and its two variants KL-VAE(f16) and Sdxl-VAE (Podell et al., 2023). Note that KL-VAE(f8) is the one used in the victim latent diffusion model, so it is a grey-box setting. We use PGD (Madry et al., 2017) to optimize adversarial examples to diverge their embedding from the original ones. As shown in Figure 14, Tree-Ring is vulnerable to embedding attacks, particularly under the grey-box condition where TPR@0.1%FPR can drop to nearly zero, effectively removing most watermarks. **Surrogate Detector Attacks** first train a binary classifier as a watermark detector, then optimize adversarial examples on it to flip labels, aiming to remove watermarks. Figure 15 explores three various settings. We refer the reader to Appendix F.3 for further details on adversarial attacks.

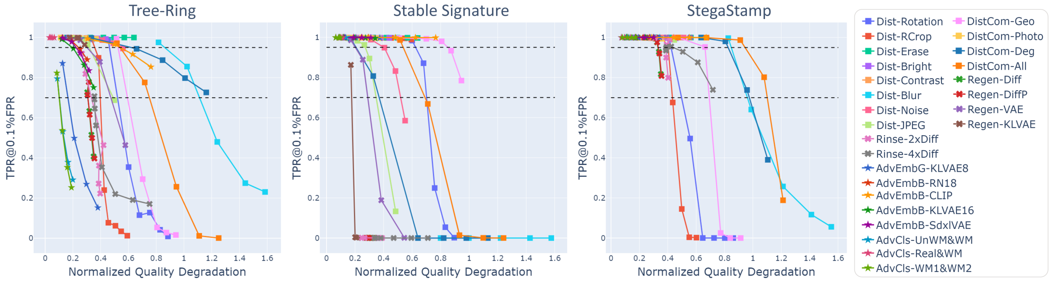


Figure 2. Unified performance vs. quality degradation 2D plots under detection setup. We evaluate each watermarking method under various attacks. Two dashed lines show the thresholds used for ranking attacks.

3 Benchmarking Results and Analysis

We extensively evaluate the security of three prominent watermarking algorithms (according to Appendix D.2), Stable Signature, Tree-Ring, and StegaStamp, respectively representing three major watermarking types: in-processing via model modification, in-processing via random seed modification, and post-processing. We conduct thorough evaluations with images from DiffusionDB Wang et al. (2022), MS-COCO Lin et al. (2014), and the DALL·E3 datasets; see Appendix D.1 for details.

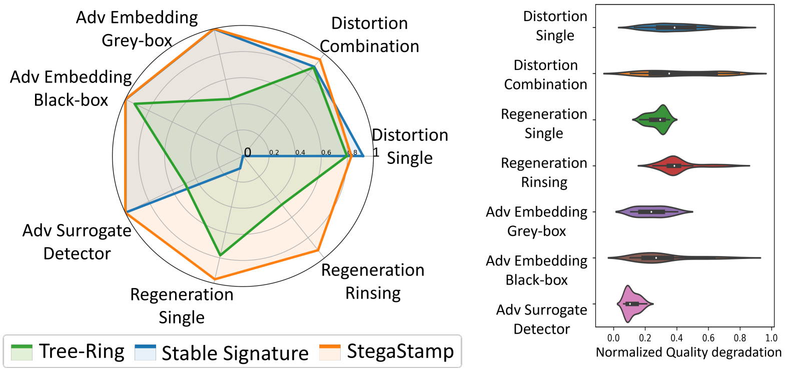


Figure 3. (left) Detection performance of three watermarks after attacks, measured by Average TPR@0.1%FPR with lower values (near center) indicating higher vulnerabilities. (right) The distribution of quality degradation. The lower the better.

Performance vs. Quality 2D plots. We evaluate 3 watermarking methods under 26 attacks, and report results across 3 datasets in Figure 24 to Figure 29. The quality of images post-attack is evaluated using 8 metrics and the detection performance is measured by TPR@0.1%FPR. Figure 7 shows that different quality metrics yield a similar ranking of attacks. Consequently, we aggregate these metrics into a single, unified quality metric — *Normalized Quality Degradation*, with lower scores indicating lesser quality degradation caused by attacks. Furthermore, we aggregate the results across three distinct datasets, and derive the unified Performance vs. Quality degradation 2D plots in Figure 2, visualizing the unified evaluation results for each watermarking method against each attack. We defer the aggregation details to Appendix E.

3.1 Benchmarking Watermark Robustness

Figure 3 provides a high-level overview of watermarks' robustness. We categorize effective attacks into the categories listed in Table 2. Attacks considered are detailed in Appendix E.5. The Average TPR@0.1%FPR, calculated for each category across strength levels, assesses watermarking method robustness. Figure 3 shows the robustness of three watermarking methods where the area covered indicates the overall robustness. Figure 3 shows the distribution of quality degradation for each type of attack to illustrate the potential trade-off between attack effectiveness and image quality.

WAVES provides a clear comparison of watermarks' robustness and reveals undiscovered vulnerabilities. Figure 3 reveals that StegaStamp occupies the largest area, signaling its exceptional robustness. Tree-Ring follows suit with a smaller area, and Stable Signature occupies the least space. Interestingly, different watermarking methods exhibit vulnerabilities to different types of attacks. Tree-Ring is particularly vulnerable to adversarial attacks introduced in this paper, with a significant vulnerability to grey-box embedding and surrogate detector attacks. It is also vulnerable to regeneration rinsing attacks. Stable Signature is vulnerable to almost all regeneration attacks. All three watermarks maintain a relative robustness against distortions. Furthermore, as observed in Figure 3, adversarial attacks generally cause less quality degradation, highlighting their potency against Tree-Ring watermarks. WAVES offers an apple-to-apple comparison of watermarks through a multi-dimensional stress test of their robustness, enabling a nuanced and comprehensive understanding of their security in various scenarios.

3.2 Benchmarking Attacks

Table 3 features a leaderboard ranking attacks based on their impact on detection performance and image quality. We assess attacks using performance thresholds (TPR@0.1%FPR=0.95 and TPR@0.1%FPR=0.7) and quality degradation at these thresholds (Q@0.95P and Q@0.7P). Additionally, we evaluate average performance (Avg P) and quality degradation (Avg Q) across all strengths. These metrics are used to rank 26 attacks for each watermarking method (details in Appendix E.6).

Attack effectiveness varies among watermarks. Table 3 shows variability in attack efficiency across watermarking methods. Metrics like Q@0.95P and Q@0.7P provide nuanced comparisons, while Avg P and Avg Q offer insights into overall attack potency and image quality impact. Our analysis identifies each watermark's specific weaknesses to certain attacks. For instance, AdvCls-UnWM&WM, AdvCls-WM1&WM2, and AdvEmbG-KLVAE8 are notably effective against Tree-Ring, whereas

Table 3. **Comparison of attacks across three watermarking methods in detection setup.**

Attack	Tree-Ring					Stable Signature					StegaStamp				
	Rank	Q@0.95P	Q@0.7P	Avg P	Avg Q	Rank	Q@0.95P	Q@0.7P	Avg P	Avg Q	Rank	Q@0.95P	Q@0.7P	Avg P	Avg Q
Dist-Rotation	11	0.464	0.521	0.375	0.648	12	0.624	0.702	0.594	0.650	5	0.423	0.498	0.357	0.616
Dist-RCrop	18	0.592	0.592	0.332	0.463	24	inf	inf	0.995	0.461	6	0.602	0.602	0.540	0.451
Dist-Erase	26	inf	inf	1.000	0.490	25	inf	inf	0.998	0.489	25	inf	inf	1.000	0.483
Dist-Bright	25	inf	inf	0.997	0.304	23	inf	inf	0.998	0.305	22	inf	inf	0.998	0.317
Dist-Contrast	22	inf	inf	0.998	0.243	20	inf	inf	0.998	0.243	17	inf	inf	0.998	0.231
Dist-Blur	20	0.861	1.112	0.563	1.221	5	-inf	-inf	0.000	1.204	9	0.848	0.962	0.414	1.198
Dist-Noise	16	0.548	inf	0.980	0.395	8	0.402	0.520	0.870	0.390	24	inf	inf	1.000	0.360
Dist-JPEG	12	0.499	0.499	0.929	0.284	9	0.485	0.485	0.793	0.284	21	inf	inf	0.998	0.263
DistCom-Geo	13	0.525	0.593	0.277	0.768	13	0.850	inf	0.937	0.767	7	0.663	0.693	0.396	0.733
DistCom-Photo	22	inf	inf	0.998	0.242	20	inf	inf	0.998	0.243	17	inf	inf	0.998	0.239
DistCom-Deg	19	0.620	inf	0.892	0.694	7	0.206	0.369	0.300	0.679	8	0.826	0.975	0.852	0.664
DistCom-All	14	0.539	0.751	0.403	0.908	11	0.538	0.691	0.334	0.900	10	0.945	1.101	0.795	0.870
Regen-Diff	5	-inf	0.307	0.612	0.323	1	-inf	-inf	0.001	0.300	1	0.331	inf	0.943	0.327
Regen-DiffP	4	-inf	0.307	0.601	0.327	1	-inf	-inf	0.001	0.303	1	0.333	inf	0.940	0.329
Regen-VAE	17	0.578	0.578	0.832	0.348	10	0.545	0.545	0.516	0.339	23	inf	inf	1.000	0.343
Regen-KLVAE	22	inf	inf	0.990	0.233	6	-inf	0.176	0.217	0.206	17	inf	inf	1.000	0.240
Rinse-2xDiff	6	-inf	0.333	0.510	0.357	3	-inf	-inf	0.001	0.332	4	0.391	inf	0.941	0.366
Rinse-4xDiff	7	-inf	0.355	0.443	0.466	4	-inf	-inf	0.000	0.438	3	0.388	inf	0.909	0.477
AdvEmbG-KLVAE8	3	-inf	0.164	0.448	0.253	20	inf	inf	0.998	0.249	17	inf	inf	1.000	0.232
AdvEmbB-RN18	10	0.241	inf	0.953	0.218	17	inf	inf	0.999	0.212	14	inf	inf	1.000	0.196
AdvEmbB-CLIP	15	0.541	inf	0.932	0.549	26	inf	inf	0.999	0.541	25	inf	inf	1.000	0.488
AdvEmbB-KLVAE16	8	0.195	inf	0.888	0.238	19	inf	inf	0.997	0.233	14	inf	inf	1.000	0.206
AdvEmbB-SdxlVAE	9	0.222	inf	0.934	0.221	17	inf	inf	0.998	0.219	14	inf	inf	1.000	0.204
AdvCls-UnWM&WM	1	-inf	0.102	0.499	0.145	14	inf	inf	0.999	0.101	11	inf	inf	1.000	0.101
AdvCls-Real&WM	21	inf	inf	1.000	0.047	14	inf	inf	0.998	0.092	11	inf	inf	1.000	0.106
AdvCls-WM1&WM2	1	-inf	0.101	0.492	0.139	14	inf	inf	0.999	0.084	13	inf	inf	1.000	0.129

Regen-Diff and Regen-DiffP are more potent against Stable Signature. Regeneration attacks impact StegaStamp but do not greatly affect its average detection performance; in contrast, certain distortion attacks significantly lower detection performance, at the cost of quality degradation. No single attack excels across all watermarking methods, yet regeneration attacks exhibit some level of consistent effectiveness. This significant variation in attack effectiveness emphasizes the imperative for diverse and watermark-tailored defensive strategies.

3.3 Benchmarking Results for User Identification

We detail the user identification results, following the evaluation method from Section 2.1. The key distinction here is the use of identification accuracy as the performance metric. Our study includes scenarios with 100, and 1 million users, reflecting a range of real-world conditions. Utilizing the same evaluation approach, we generate unified Performance vs. Quality degradation 2D plots (Figure 16), radar plots for watermark comparison (Figure 17), and an attack leaderboard in the identification context (Table 5). **Identification results mirror findings from detection, showing similar trends in watermark robustness and attack effectiveness.** Figure 17 and Table 5 reveal that trends in watermark robustness and attack potency closely match those in detection, largely because both rely on precise watermark decoding. Notably, watermarks become more vulnerable as user numbers increase, a trend particularly evident in attacks that already strongly affect detection. Since identification demands more accurate decoding, its vulnerability amplifies with user growth.

3.4 Summary of Takeaway Messages

WAVES provides a standardized framework for benchmarking watermark robustness and attack potency. WAVES evaluates both detection and identification tasks. It unifies the quality metrics and assesses attack potency against both performance degradation and quality degradation. The Performance vs. Quality 2D plots allow for a comprehensive analysis of various watermarks in one unified framework. With over twenty attacks tested, WAVES exposes new vulnerabilities in popular watermarking techniques.

Avoid using publicly available VAEs. WAVES demonstrates the risks of using publicly available VAEs in watermarked diffusion models. Stable Signature's design renders it vulnerable to regeneration attacks that use a VAE with an encoder identical to the victim model's VAE encoder, while coupled with a different decoder. Today's proprietary generators, like DALL·3, typically train the latent diffusion model themselves but use a publicly available VAE, pointing to a critical security concern in such popular AI services.

The robustness of StegaStamp potentially illuminates a path for future robust watermarks. The StegaStamp watermark Tancik et al. (2020) stands out in our evaluation for its robustness. Designed for physical-world use which requires high robustness, StegaStamp is trained with a series of distortions that mimic real-world scenarios, significantly enhancing its robustness. However, it's important to recognize the potential trade-off between watermark robustness and quality. As a post-processing method, the original paper finds that StegaStamp may introduce artifacts. In contrast, this might not pose a problem for in-processing watermarks. Therefore, in-processing watermarks could still benefit from incorporating augmentation or adversarial training.

REFERENCES

Mahdi Ahmadi, Alireza Norouzi, Nader Karimi, Shadrokh Samavi, and Ali Emami. Redmark: Framework for residual diffusion watermarking based on deep networks. Expert Systems with Applications, 146:113157, 2020.

Ali Al-Haj. Combined dwt-dct digital image watermarking. Journal of computer science, 3(9): 740–746, 2007.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. URL `https://openreview.net/forum?id=rkcQFMZRb`.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. arXiv preprint arXiv:1810.00069, 2018.

Chin-Chen Chang, Piyu Tsai, and Chia-Chen Lin. Svd-based digital image watermarking scheme. Pattern Recognition Letters, 26(10):1577–1586, 2005.

Huili Chen, Bita Darvish Rouhani, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models. In Proceedings of the 2019 on International Conference on Multimedia Retrieval, pp. 105–113, 2019.

Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. Digital watermarking and steganography. Morgan kaufmann, 2007.

Ingemar J Cox, Joe Kilian, Tom Leighton, and Talal Shamoon. Secure spread spectrum watermarking for images, audio and video. In Proceedings of 3rd IEEE international conference on image processing, volume 3, pp. 243–246. IEEE, 1996.

Yinpeng Dong, Huanran Chen, Jiawei Chen, Zhengwei Fang, Xiao Yang, Yichi Zhang, Yu Tian, Hang Su, and Jun Zhu. How robust is google's bard to adversarial image attacks? arXiv preprint arXiv:2309.11751, 2023.

Executive Office of the President. Executive order 14110: Safe, secure, and trustworthy development and use of artificial intelligence. Federal Register, 88:75191–75226, 2023.

Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3054–3058. IEEE, 2022.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. arXiv preprint arXiv:2303.15435, 2023.

Jamie Hayes and George Danezis. Generating steganographic images via adversarial training. Advances in neural information processing systems, 30, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

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.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. Advances in neural information processing systems, 32, 2019.

Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7066–7074, 2019.

Hengrui Jia, Christopher A Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. Entangled watermarks as a defense against model extraction. In 30th USENIX Security Symposium (USENIX Security 21), pp. 1937–1954, 2021.

Zhengyuan Jiang, Jinghuai Zhang, and Neil Zhenqiang Gong. Evading watermark based detection of ai-generated content. arXiv preprint arXiv:2305.03807, 2023.

Martin Kutter and Fabien AP Petitcolas. Fair benchmark for image watermarking systems. In Security and watermarking of multimedia contents, volume 3657, pp. 226–239. SPIE, 1999.

Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fr\'echet inception distance. arXiv preprint arXiv:2203.06026, 2022.

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

Nils Lukas and Florian Kerschbaum. Ptw: Pivotal tuning watermarking for pre-trained image generators. arXiv preprint arXiv:2304.07361, 2023.

Nils Lukas, Abdulrahman Diaa, Lucas Fenaux, and Florian Kerschbaum. Leveraging optimization for adaptive attacks on image watermarks. arXiv preprint arXiv:2309.16952, 2023.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. arXiv preprint arXiv:2205.07460, 2022.

JJK ó Ruanaidh, WJ Dowling, and FM Boland. Watermarking digital images for copyright protection. IEE PROCEEDINGS VISION IMAGE AND SIGNAL PROCESSING, 143:250–256, 1996.

Joseph JK O'Ruanaidh and Thierry Pun. Rotation, scale and translation invariant digital image watermarking. In Proceedings of International Conference on Image Processing, volume 1, pp. 536–539. IEEE, 1997.

Fabien AP Petitcolas. Watermarking schemes evaluation. IEEE signal processing magazine, 17(5): 58–64, 2000.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952, 2023.

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, pp. 8748–8763. PMLR, 2021.

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, pp. 10684–10695, 2022.

Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: A generic watermarking framework for ip protection of deep learning models. arXiv preprint arXiv:1804.00750, 2018.

Mehrdad Saberi, Vinu Sankar Sadasivan, Keivan Rezaei, Aounon Kumar, Atoosa Chegini, Wenxiao Wang, and Soheil Feizi. Robustness of ai-image detectors: Fundamental limits and practical attacks. arXiv preprint arXiv:2310.00076, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020.

Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2117–2126, 2020.

Hai Tao, Li Chongmin, Jasni Mohamad Zain, and Ahmed N Abdalla. Robust image watermarking theories and techniques: A review. Journal of applied research and technology, 12(1):122–138, 2014.

Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. arXiv preprint arXiv:2210.14896, 2022.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. arXiv preprint arXiv:2305.20030, 2023.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. arXiv preprint arXiv:2304.05977, 2023.

Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In Proceedings of the IEEE/CVF International conference on computer vision, pp. 14448–14457, 2021.

Yu Zeng, Mo Zhou, Yuan Xue, and Vishal M Patel. Securing deep generative models with universal adversarial signature. arXiv preprint arXiv:2305.16310, 2023.

Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. arXiv preprint arXiv:1909.01285, 2019.

Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative ai, 2023a.

Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. arXiv preprint arXiv:2303.10137, 2023b.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In Proceedings of the European conference on computer vision (ECCV), pp. 657–672, 2018.

Benchmarking the Robustness of Image Watermarks

Supplementary Material

A A Mini Survey of Image Watermarks

In this section, we detail the existing landscape of watermarking approaches in the era of AI-Generated Content (AIGC) everywhere. Figure 4 depicts our scenario of interest. First, an AI company/owner embeds a watermark into its generated images. Then, if the owner is shown one of their watermarked images at a later point in time, they can identify ownership of it by recovering the watermark message. Commonly, users might modify watermarked images for legitimate personal purposes. There are also instances where users attempt to erase a watermark for malicious reasons, such as disseminating fake information or infringing upon copyright. For simplicity, we term any image manipulation as an "attack."

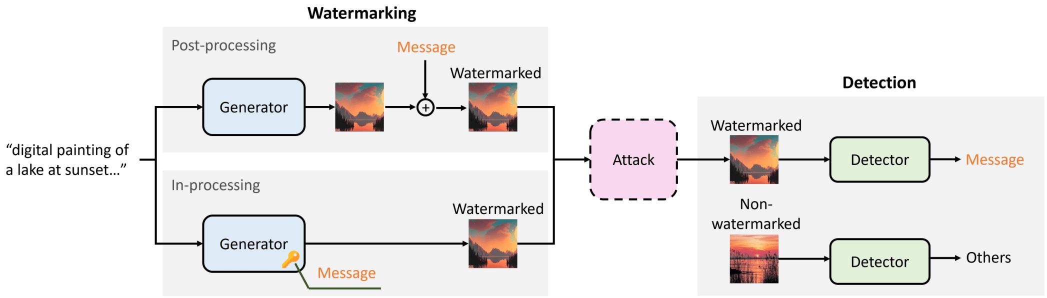


Figure 4. **An illustration of a robust watermarking workflow.** An AI company provides two services: (1) generate watermarked images, i.e., embed invisible messages, and (2) detect these messages when shown any of their watermarked images. There is an attack stage between the watermarking and detection stages. The watermarked images may experience natural distortions (e.g., compression, re-scaling) or manipulated by malicious users attempting to remove the watermarks. A robust watermarking method should still be able to detect the original message after an attack.

Watermarking AI-generated Images. Imprinting invisible watermarks into digital images has a long and rich history. From conventional steganography to recent generative model-based methods, we categorize popular watermarking techniques into two categories: post-processing methods and in-processing methods.

Post-processing approaches embed post-hoc watermarks into images. When watermarking AI-generated images, we apply such methods *after* the generation process. Post-processing watermarks are model-agnostic and applicable to any image. However, they sometimes introduce human-visible artifacts, compromising image quality. We review popular post-processing methods.

P1) Frequency-domain methods. These methods manipulate the representation of an image in some transform domain (ó Ruanaidh et al., 1996; Cox et al., 1996; O'Ruanaidh & Pun, 1997). The image transform can be a Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT) (Cox et al., 2007), or SVD decomposition (Chang et al., 2005). These transformations have a range of invariance properties that make them robust to translation and resizing. The commercial implementation of Stable Diffusion (Rombach et al., 2022) uses DWTDCT (Al-Haj, 2007) to watermark its generated images. However, many studies have shown that these watermarks are vulnerable to common image manipulations (Zhao et al., 2023a).

P2) Deep encoder-decoder methods. These methods rely on trained networks for embedding and decoding the watermark (Hayes & Danezis, 2017). Methods such as HiDDeN (Zhu et al., 2018) and RivaGAN (Zhang et al., 2019) learn an encoder to imprint a hidden message inside an image and a decoder (also called a detector) to extract the message. To train robust watermarks, RedMark (Ahmadi et al., 2020) integrates differentiable attack layers between the encoder and decoder in the end-to-end training process; RivaGAN (Zhang et al., 2019) employs an adversarial network to remove the watermark during training; StegaStamp (Tancik et al., 2020) adds a series of strong image perturbations between the encoder and decoder during training, resulting in watermarks which are robust to real-world distortions caused by photographing an image as it appears on a display.

P3) Others. There are other varieties of post-processing methods that do not fall into **P1** or **P2**. SSL (Fernandez et al., 2022) embeds watermarks in self-supervised-latent spaces by shifting the image's features into a designated region. DeepSigns (Rouhani et al., 2018) and DeepMarks (Chen et al., 2019) embed target watermarks into the probability density functions of weights and activation maps. Entangled watermarks (Jia et al., 2021) designs a reinforced watermark based on a target watermark and the task data.

In-processing methods adapt generative models to directly embed watermarks as part of the image generation process, substantially reducing or eliminating visible artifacts. With diffusion models presently dominating the field of image generation, a surge of in-processing approaches specific to these models has recently emerged. We categorize current work into three categories.

I1) Model modification. *The entire model.* This line of work inherits the encoder-decoder idea and bakes the encoder into the entire generative model. This is usually accomplished by watermarking training images with a pre-trained watermark encoder and decoder, then training or fine-tuning the generative model on these watermarked images (Yu et al., 2021; Zeng et al., 2023; Lukas & Kerschbaum, 2023). This type of method has been shown to work well on small models like guided diffusion, but suffers from the expensive training of large text-to-image generation models (Zhao et al., 2023b), making it inapplicable in practice.

Parts of the model. Stable Signature (Fernandez et al., 2023) follows the above two-stage training pipeline while only fine-tuning the decoder of the latent-diffusion model (LDM) (Rombach et al., 2022), leaving the diffusion component unchanged. This type of watermarker is much more efficient to train. By fine-tuning multiple latent decoders, the model can embed different messages into images.

The robustness of these two types of model modification critically relies on the robustness of the pre-trained encoder and decoder.

I2) Modification of a random seed. Tree-Ring (Wen et al., 2023), different from all the above methods, embeds a pattern into the initial noise vector used by a diffusion model for sampling. The pattern can be retrieved at detection time by inverting the diffusion process using DDIM (Song et al., 2020) as the sampler. This method does not require any training, can easily embed different watermarks, and is robust to many simple distortions and attacks. The robustness of Tree-Ring relies on the accuracy of the DDIM inversion.

Removing Watermarks Robustness is an essential property of watermarks. Evaluations of robustness in existing literature focus on simple image distortions like rotation, Gaussian blur, etc. Recently, inspired by adversarial purification Nie et al. (2022), Zhao et al. (2023a) and Saberi et al. (2023) both find that regenerating images by noising and denoising images through a diffusion model or a VAE can effectively remove some watermarks. Saberi et al. (2023) propose adversarial attacks based on a trained surrogate watermark detector. Lukas et al. (2023) also introduces adversarial attacks but requires the knowledge of the watermarking algorithm and a similar surrogate generative model. Jiang et al. (2023) studies white-box attacks and black-box query-based attacks. Some attacks are not possible in realistic scenarios where the attacker has only API access. Furthermore, existing evaluations use differing quality/performance metrics, making it difficult to compare the effectiveness between watermarking methods and between attacks.

Benchmarks for Image Watermarks. Before the advent of AIGC, there were significant benchmarks introduced that greatly accelerated the progress of watermark standardization Kutter & Petitcolas (1999); Tao et al. (2014); Petitcolas (2000). However, with the development of AIGC, the need to watermark images generated by AI has become urgent, as previous methods were weak in robustness and could not meet current requirements. Nowadays, more and more methods for watermarking images generated by AI have been proposed, but they all use different methods to evaluate robustness. Therefore, this paper proposes a benchmark for the AIGC era.

B Formalism of Watermark Detection and Identification

Invisible image watermarks, which are inspired by classical watermarks to protect the intellectual properties of creators, are now applied for a wider range of application scenarios. With the vast development of AI generative models, most current research focuses on applying invisible watermarks to (1) identify AI-generated images (AI Detection) (Saberi et al., 2023), and (2) identify the user who generated the image for source tracking (User Identification) (Fernandez et al., 2023).

To fairly evaluate the different watermark methods for different applications, we start from formulating a general, message-based watermarking protocol, partially adopting the notation of (Lukas et al., 2023), which generalizes most of the existing setups. Let θ_G denote an image generator, $\mathcal{M}$ the space of watermark messages, and $\mathcal{X}$ the domain of images. We assume $\mathcal{M}$ is a metric space with distance function $D(\cdot,\cdot)$. The choice of message space $\mathcal{M}$ can be very different depending on the watermarking algorithm: for Tree-Ring, messages are random complex Gaussians, while for the Stable Signature and StegaStamp, each message is a length-d binary string, where d denotes the length of the message. For watermarking algorithms following the encoder-decoder training approach, like Stable Signature and StegaStamp, the choice of message length d is fixed after training. Some methods, such as Tree-Ring, enjoy flexible message length at the time of injecting watermarks.

In addition to classifying images as watermarked or non-watermarked, a good detector will often provide a *p-value* for the watermark detection, which measures the probability that the level of watermark strength observed in an image could occur by random chance. The Tree-Ring watermark also includes an image location parameter τ to embed a message $m \in \mathcal{M}$, but we subsume this under the parameters of θ_G. We now introduce several important watermarking operations:

- $\mathsf{EMBED}: \theta_G \times \mathcal{M} \to \mathcal{X}$ is the generative procedure that creates a watermarked image given user-defined parameters of θ_G (such as prompt, guidance scale, etc. for a diffusion model) and a target message $m \in \mathcal{M}$.
- $\mathsf{DECODE}: \mathcal{X} \to \mathcal{M}$ is a recovery procedure of a message m embedded within a watermarked image $x = \mathrm{EMBED}(\theta_G, m)$. In particular, the recovery $m' = \mathrm{DECODE}(x)$ may be imperfect, i.e., $m' \neq m$.
- $\mathsf{VERIFY}_\alpha : \mathcal{M} \times \mathcal{M} \to \{0,1\}$ is conducted by the model owner to decide whether x was watermarked by inspecting $m' = \mathsf{DECODE}(x)$, where $x = \mathsf{EMBED}(\theta_G,m)$. For a decoded message m', we consider the following p-value (further discussed in Section C) for evaluating whether the image could have been watermarked using m. which is defined as

  $$p = \mathrm{P}_m\big(D(\omega,m') < D(m,m') \,|\, H_0\big),$$

  where, $D(\omega,m')$ is the similarity between an arbitrary message $\omega \sim \mathcal{M}$ (drawn uniformly at random) and m', and $D(m,m')$ is the similarity between the ground truth message m and the recovered message m'. H_0 denotes the null hypothesis that the image was generated without knowledge of the watermark (and therefore the recovered message is random). $\mathsf{VERIFY}_\alpha(m',m)$ returns 1 if $p < \alpha$, and 0 otherwise. In our experiments, we set $\alpha = 0.001$.

To establish a comprehensive evaluation toolbox, we consider two distinct problems that naturally arise during watermark analysis: detection and identification. Let $\mathcal{A} : \mathcal{X} \to \mathcal{X}$ represent an image attack function and denote by Q a fixed subset of messages independently drawn from $\mathcal{M}$ used by θ_G. Further, assume that the owner of θ_G will only embed messages contained within a finite subset Q drawn randomly from $\mathcal{M}$.

B.1 Detection

In the *watermark detection problem*, given $x = \mathrm{EMBED}(\theta_G,m)$, and an attack $x' = \mathcal{A}(x)$, the model owner is tasked with producing EMBED and DECODE protocols which satisfy the following,

(1) If $x = \mathsf{EMBED}(\theta_G,m)$ is a watermarked image, then $\mathsf{VERIFY}_\alpha(\mathsf{DECODE}(x')) = 1$.
(2) If $x = \mathsf{EMBED}(\theta_G,\mathsf{NULL})$ is an unwatermarked image, then $\mathrm{VERIFY}_\alpha(\mathsf{DECODE}(x')) = 0$.

For both conditions, a comparison of the extracted message $m' = \mathsf{DECODE}(x)$ is performed against all messages in Q. Failure of the above conditions is referred to as Type II and Type I errors, respectively. Exploration of the tradeoff between minimization of both error types is an interesting research topic in its own right Zhao et al. (2023a); Saberi et al. (2023).

B.2 Identification

While watermark detection requires only that $\mathsf{VERIFY}(\theta_G, x') = 1$, the *watermark identification problem* further requires that one can accurately determine which message from Q is embedded in the image. Rigorously, given $x = \mathsf{EMBED}(\theta_G,m)$, an attack $x' = \mathcal{A}(x)$, and $m' = \mathsf{DECODE}(\theta_G,x')$, the user requires the EMBED and DECODE to satisfy

$$\underset{m' \in Q}{\operatorname{argmin}} \ \mathrm{P}\big(D(\omega,m) < D(m',m) \,|\, H_0\big) = m,$$

for randomly drawn $\omega \sim \mathcal{M}$ if x.

The identification problem is useful in the scenario where the model owner wishes to identify the user who created an image (e.g., a user of DALL·E). Note that as $|Q| \to \infty$, the identification problem becomes difficult as Q will resemble $\mathcal{M}$ in distribution.

C Details on Performance Metrics

C.1 Clarifications on p-Value

Here, we clarify the definition of the p-value as follows.

Watermark injection and evaluation are often done by encoding a message m into the image, and later recovering the message m', which may be an imperfect recovery. In addition to classifying images as watermarked or non-watermarked, a good detector will often provide a *p-value* for the watermark detection, which measures the probability that the level of watermark strength observed in an image could happen by random chance. Rigorously, we have

$$p = \mathrm{P}_m\big(D(\omega,m') < D(m,m') \,|\, H_0\big),$$

where $D(\omega,m')$ is a dissimilarity metric between an arbitrary message $\omega \sim \mathcal{M}$ (selected uniformly at random) and recovered message m' from the image by the detector, and $D(m,m')$ denotes dissimilarity between the ground truth message m and the recovered message m'. H_0 denotes the null hypothesis that the image was generated without knowledge of the watermark (and therefore, the recovered message is random). The same hypothesis testing can also be applied to user identification.

As in some prior work (Fernandez et al., 2023), one may set a threshold on the estimated p-value to determine the detection result. However, this approach makes it difficult to compare different watermark methods fairly. Even if we set the same p-value threshold on all watermark methods, the distinct choice of message space $\mathcal{M}$, message distribution P_m, and hypothesis test may differ. Therefore, we seek to evaluate watermark methods mainly using metrics that are independent of the choice of p-value threshold and statistical test.

C.2 Performance Metrics for User Identification

For user identification, we also focus on metrics that do not depend on statistical testing and hyperparameters like p-value thresholds.

The user detection issue involving K users is aptly conceptualized as a K-way classification task. This can be reframed into a binary classification problem by designating the positive class as the correct user and the negative class as all other users. From this perspective, the TPR@x%FPR metric becomes applicable, defined for a specific FPR threshold and user count. In our study, we focus on TPR@0.1%FPR for a scenario involving 1,000 users. The identification performance results are shown in Section 3.3.

C.3 Other Performance Metrics

While this paper primarily focuses on the TPR@0.1%FPR metric, it's important to acknowledge other common metrics such as p-values, AUROC scores, mean accuracies, and bit accuracies.

However, we do not report p-values since their absolute values depend heavily on the chosen statistical test, making them less comparable across different watermark methods.

AUROC scores, although independent of the choice of p-value threshold and statistical test, have limitations used as a metric for evaluating watermark detection. In AI-generated image applications, labeling non-watermarked images as watermarked (false positive) are particularly detrimental. As a result, strict control of false positive rate (FPR) is crucial. However, a high AUROC does not guarantee a high true positive rate (TPR) at low false positive rate (FPR) levels.

Using message distances such as bit accuracy as a metric for evaluating watermarks' performance has several limitations:

(1) Insensitivity to error distribution: bit accuracy measures the proportion of correctly identified bits in the watermark but does not account for the distribution of errors. This means it treats all errors equally, regardless of their impact or pattern. In watermarking, certain types of errors (like clustered errors) might be more detrimental than others.
(2) Lack of contextual insight: bit accuracy alone doesn't provide insights into the types of errors (false positives or false negatives). In watermark detection, understanding the nature of errors is crucial, especially in differentiating between missing a watermark and incorrectly identifying one.
(3) Threshold dependency: the effectiveness of bit accuracy is dependent on the threshold chosen for determining a bit's value. Different thresholds can yield significantly different bit accuracies, making the metric somewhat arbitrary and less reliable for comparing different watermarking schemes.
(4) Non-representation of overall system performance: bit accuracy focuses narrowly on the correctness of individual bits, neglecting the broader context of the watermarking system's performance, such as its robustness against attacks, computational efficiency, or impact on image quality.
(5) Potential misleading results in imbalanced cases: in scenarios where the watermark bits are not evenly distributed (e.g., more 0s than 1s or vice versa), bit accuracy might give a skewed view of the system's performance. It could show high accuracy even if the system is only good at detecting the majority class. For these reasons, it's often more effective to use a combination of metrics that can provide a holistic view of the watermarking system's performance, considering aspects like error distribution, false positives/negatives, and overall impact on the media.

Although these metrics are not included in the paper, they are incorporated in the benchmark software and available for future research use.

D Design Choices of WAVES

D.1 Dataset Preparation

We utilize three datasets for the non-watermarked reference images in our evaluation: **DiffusionDB**, **MS-COCO**, and **DALL·E3**, each comprising 5000 reference images and prompts. **DiffusionDB** represents a diverse collection from the DiffusionDB dataset (Wang et al., 2022), focusing on images generated from the Stable Diffusion (Rombach et al., 2022) models. **MS-COCO** is derived from the well-known Microsoft COCO detection challenge (Lin et al., 2014), featuring a wide range of everyday scenes and objects. **DALL·E3**[1] includes images from the DALL·E3 model, showcasing another popular diffusion model trained on substantially different data. These datasets provide a comprehensive range of image types and contexts, ideal for robust watermark evaluation.

The three datasets are filtered subsets of the corresponding source dataset using the same filtering algorithm. The source dataset information is listed below.

- *DiffusionDB*: the 2m_random_100k split of DiffusionDB dataset (Wang et al., 2022), link.
- *MS-COCO*: the validation split of the 2017 Microsoft COCO detection challenge (Lin et al., 2014), link.
- *DALL·E3*: the train split of the *dalle-3-dataset* repository on HuggingFace, collected from the LAION share-dalle-3 discord channel, link.

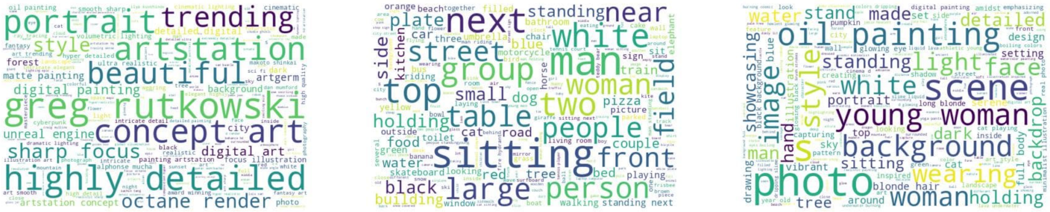
The filtering algorithm considers the following rules to subsample the 5,000 image subset:

- *Remove columns*: Remove irrelevant columns and only keep the reference images and prompt strings.
- *Filter prompts*: Tokenize the prompt strings by the Open Clip's tokenizer, and filter out samples with no tokens and more than 75 tokens. This is because Stable Diffusion (Rombach et al., 2022) truncates prompts at 75 tokens (Wang et al., 2022).
- *Rank images*: Rank the images by their aesthetics score, as defined by Xu et al. (2023), in descending order. We then select the top 5,000 images, along with their corresponding prompt strings. This approach is adopted because the DiffusionDB and DALL·E3 datasets,

[1] The DALL·E3 dataset is hosted at `https://huggingface.co/datasets/laion/dalle-3-dataset`.

sourced from chat-bots, contain some lower-quality images. We posit that watermarking holds greater utility for high-quality AI-generated images, as the copyright protection of low-quality generated images is less meaningful and practical.

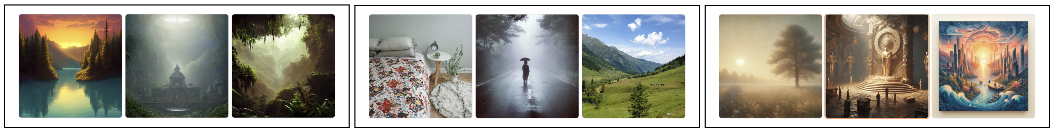
In our study, we examined three distinct datasets—DiffusionDB, MS-COCO, and DALL·E3—each characterized by a unique distribution of prompt words. As illustrated in the word-cloud plots (Figure 5), we observe notable differences. DiffusionDB predominantly features prompt words that emphasize the desired quality of the generated images, such as “beautiful” and “highly detailed.” In contrast, MS-COCO’s prompts mainly focus on describing the objects within the images. Meanwhile, DALL·E3’s prompts show a tendency towards describing aspects of fine arts.



(a) DiffusionDB prompts (b) MS-COCO prompts (c) DALL·E3 prompts

Figure 5. Word clouds of DiffusionDB, MS-COCO, and DALL·E3 prompts.

Image examples from the three datasets are illustrated in Figure 6. The reference images for DiffusionDB are produced by Stable Diffusion, MS-COCO includes real-world photographs, and DALL·E3 contains images generated by the DALL·E3 model. This choice of datasets effectively covers two popular generative models and the real-world scenario, highlighting their relevance in practical watermarking applications.



(a) DiffusionDB (b) MS-COCO (c) DALL·E3

Figure 6. Image examples of DiffusionDB, MS-COCO, and DALL·E3.

D.2 Selection of Watermark Representatives

Table 4. A list of alternative watermarking algorithms not tested by WAVES in this work.

Method	Known Weakness(es)
DwtDct (Al-Haj, 2007)	Distortion (Wen et al., 2023), Purification (Saberi et al., 2023)
DwtDctSvd (Al-Haj, 2007)	Distortion (Zhao et al., 2023a; Wen et al., 2023), Purification (Saberi et al., 2023), Regeneration (Zhao et al., 2023a)
RivaGan (Dong et al., 2023)	Regeneration (Zhao et al., 2023a), Purification (Saberi et al., 2023)
SSL (Fernandez et al., 2022)	Distortion(Zhao et al., 2023a), Regeneration (Zhao et al., 2023a)
WatermarkDM (Zhao et al., 2023b)	Purification (Saberi et al., 2023)

Our WAVES framework can be used to stress-test the robustness of any watermark. In this work, however, we focus on three methods: the *Stable Signature*, *Tree-Ring*, and *Stegastamp*. This is due to existing and extensive studies (Zhao et al., 2023a; Saberi et al., 2023; Wen et al., 2023) indicating these three methods are far more robust to simple off-the-shelf attacks than alternative watermarking algorithms listed in Appendix A. We list these competitors along with their documented vulnerabilities in Table 4.

E Evaluation Details

In this section, we provide more details on the evaluation scheme of WAVES.

E.1 Watermarking Protocol and Evaluation Workflow.

In-depth information on the applications of invisible image watermarks is provided, focusing on AI detection and user identification. We delve into the evolution of watermarks from classical copyright protection tools to their modern uses in AI scenarios. The appendix discusses the specific roles of AI detection in distinguishing AI-created images and user identification in tracing image origins, citing studies like (Saberi et al., 2023; Fernandez et al., 2023).

The formulation of our watermarking protocol is detailed, explaining the use of an image generator θ_G, a metric space of watermark messages $\mathcal{M}$, and an image domain $\mathcal{X}$. We elaborate on the variations in the choice of message space $\mathcal{M}$ across different watermark methods. For example, Tree-Ring uses random complex Gaussians, whereas Stable Signature and StegaStamp use binary strings. The implications of these choices on the flexibility and effectiveness of watermark methods are discussed.

An extensive analysis of the trade-off between watermark performance and image quality in the context of watermark attacks is provided. This includes the rationale for using Performance vs. Quality 2D plots for attack comparisons, highlighting the comprehensive perspective this offers over traditional performance-focused analyses. The methodology of our evaluation process is laid out in detail, describing how we compare watermarked images from model θ_G with a mixed set of real and AI-generated images to achieve a robust and unbiased assessment. This section also covers the specific metrics used, including TPR@0.1%FPR and various image quality metrics, and how they are integrated into a consolidated performance vs. quality analysis.

E.2 Performance Evaluation Metrics

The evaluation approach in WAVES addresses the challenges of using p-values for fair watermark method comparison. The diversity in message spaces $\mathcal{M}$, distributions P_m, and hypothesis tests can lead to biased results when traditional p-value thresholds are used. Our metrics, designed to be independent of these thresholds and tests, offer a balanced and thorough evaluation of watermark methods, focusing on their inherent strengths in encoding and recovering messages.

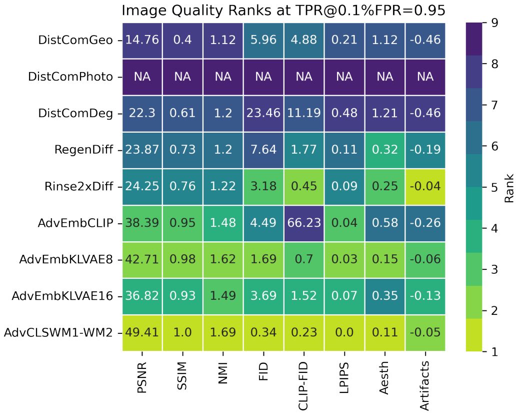
Emphasizing TPR@$x\%$FPR, particularly at the low FPR of 0.1%, sets WAVES apart in evaluating watermark methods. This novel approach, inspired by studies like Wen et al. (2023); Fernandez et al. (2023), challenges watermark methods beyond typical benchmarks such as TPR@1%FPR. Applied to a broader image dataset, it provides a more comprehensive evaluation of their effectiveness. In user identification, WAVES's multi-class classification approach assesses watermark methods' efficacy in correctly attributing users. The appendices detail the methodology's implementation and present additional results, demonstrating the effectiveness and accuracy of our approach in various user identification scenarios.

We treat the user identification problem as a multi-class classification task, as outlined in Section 2.1. This involves defining a set of ground-truth messages, each corresponding to a unique user. To avoid the exhaustive evaluation process (watermark encoding, attacking, and decoding) for varying numbers of users, we consistently watermark images with the same message, the ground-truth message of the first user, and generate a random set of ground-truth messages for the remaining users at the time of evaluation. This approach is feasible since the ground-truth messages for users other than the first do not influence the watermarking or attack phases. We conduct the identification assessment ten times with ten distinct random sets of ground-truth messages for the other users, and we report the mean multi-class classification accuracy.

E.3 Processing Results

A set of Performance vs. Quality 2D plots show the detailed evaluation results. We evaluate 3 watermarking methods under the 26 attacks, and report results across 3 datasets in Figure 24 to Figure 29. The quality of images post-attack is evaluated using 8 metrics and the detection performance of 3 methods is measured by TPR@0.1%FPR.

Different quality metrics yield similar ranking of attacks. Despite measuring different aspects of image quality, we observe that eight quality metrics consistently produce similar rankings for attacks, as illustrated in Figure 7. Since a strong attack should remove the watermark without sacrificing the image quality, we rank attack potency by ranking the post-attack quality, from best to worst, at a frozen performance threshold (e.g., TPR@0.1%FPR=0.95). Upon comparing the rankings derived



	PSNR	SSIM	NMI	FID	CLIP-FID	LPIPS	Aesth	Artifacts
DistComGeo	14.76	0.4	1.12	5.96	4.88	0.21	1.12	-0.46
DistComPhoto	NA	NA	NA	NA	NA	NA	NA	NA
DistComDeg	22.3	0.61	1.2	23.46	11.19	0.48	1.21	-0.46
RegenDiff	23.87	0.73	1.2	7.64	1.77	0.11	0.32	-0.19
Rinse2xDiff	24.25	0.76	1.22	3.18	0.45	0.09	0.25	-0.04
AdvEmbCLIP	38.39	0.95	1.48	4.49	66.23	0.04	0.58	-0.26
AdvEmbKLVAE8	42.71	0.98	1.62	1.69	0.7	0.03	0.15	-0.06
AdvEmbKLVAE16	36.82	0.93	1.49	3.69	1.52	0.07	0.35	-0.13
AdvCLSWM1-WM2	49.41	1.0	1.69	0.34	0.23	0.0	0.11	-0.05

Figure 7. **Ranking attacks with different quality metrics** on DiffusionDB images watermarked by Tree-Ring. Attack potency is ranked by image quality at 0.95 TPR@0.1%FPR. Colors indicate the ranks (1=best, 9=worst), and values show the measured quality. We use 'NA' to label an attack if its attack curve lies entirely above TPR=0.95; the attack is automatically ranked last.

from different quality metrics, we find that the variations in rank order are minimal. Consequently, we aggregate these metrics into a single, unified quality metric.

Unified Performance vs. Quality degradation 2D plots. We first set the "standardized" 0.1 and 0.9 points for each metric according to the distribution of measured values (as depicted in Figure 8). Subsequently, every metric's value is normalized to predominantly fall within the $[0.1, 0.9]$ range of the normalized quality metric (the detailed methodology is provided in Appendix E.4). We average these normalized quality scores to derive the *Normalized Quality Degradation*, with lower scores indicating lesser quality degradation caused by attacks, which is preferred. Furthermore, we aggregate the results across three distinct datasets. The Performance vs. Quality degradation 2D plots, as shown in Figure 2, visualize the unified evaluation results for each watermarking method. We use unified Performance vs. Quality degradation 2D plots to benchmark watermarks and attacks in the following sections.

E.4 Normalization and Aggregation of Quality Metrics

The eight quality metrics in WAVES exhibit unique range characteristics. To synthesize these into a single metric, we normalize each metric into a common interval, assigning the 10% quantile of all attacked images as the 0.1 point, and the 90% quantile as the 0.9 point. This normalization is based on a comprehensive dataset covering 26 attack methods, three watermark methods, and three datasets. Our focus is on specific applications, particularly attacking invisible image watermarks. The normalization process is informed by the cumulative distribution functions (CDFs) of these metrics, which exhibit a roughly linear distribution between the 10% and 90% quantiles, but a non-linear pattern outside this range. This observation is particularly evident in metrics like PSNR. The normalization method ensures values carry equivalent significance across different metrics. Figure 8 in this appendix provides a visual representation of the CDFs across all metrics. After normalization, metrics are aggregated by averaging to form the comprehensive quality metric, utilized in Section 3 for Performance vs Quality plots, watermark radar plots, and attack leaderboards. This section elaborates on the normalization and aggregation process, providing a foundation for understanding the metric's application and significance.

In Figure 8, the cumulative distribution functions (CDFs) for eight image quality metrics over all attacked watermarked images are presented. This illustration includes the metric values at the 10% and 90% quantiles, which are used as the boundaries for normalizing the metric values within the range of $[0.1, 0.9]$. Such normalization ensures that all normalized metrics exhibit a comparable statistical distribution over attacked watermarked images, facilitating an unbiased aggregated evaluation. To consolidate these normalized metrics, we first calculate the average within each of the four defined categories (image similarities, distribution distances, perception-based metrics, and image quality assessments) as delineated in Section 2.1. Subsequently, the average of these category averages is calculated to yield a single, consolidated normalized, and aggregated quality metric.

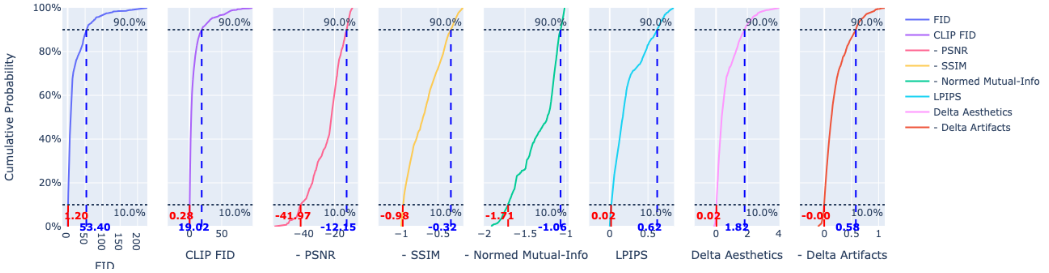


Figure 8. Cumulative distribution functions (CDFs) for eight image quality metrics across all attacked watermarked images. The horizontal dashed lines mark the 10% and 90% quantiles, and the intersecting vertical dashed lines delineate the bounds of the normalization intervals. Values at the lower bound are normalized to 0.1, and those at the upper bound to 0.9.

E.5 Details of Benchmarking Watermarks

When benchmarking watermark robustness in Figure 3 and Figure 17, we consider the following effective attacks. We select 21 attacks from 26 attacks. We include all distortion attacks. We select the two most effective single regeneration attacks and two rinsing attacks. For adversarial attacks, we do not include AdvEmbB-RN18, and AdvCls-Real&WM since they basically do not work. We also eliminate AdvCls-UnWM&WM and only use AdvCls-WM1&WM2 to represent surrogate detector attacks since AdvCls-UnWM&WM is based on an unrealistic assumption. For each type of attack, we compute Average TPR@0.1%FPR across all practical strength levels that cause quality degradation less than 0.8, and across all attacks in each category.

- *Distortion Single*: Dist-Rotation, Dist-RCrop, Dist-Erase, Dist-Bright, Dist-Contrast, Dist-Blur, Dist-Noise, Dist-JPEG.
- *Distortions Combination*: DistCom-Geo, DistCom-Photo, DistCom-Deg, DistCom-All.
- *Regeneration Single*: Regen-Diff, Regen-KLVAE.
- *Regeneration Rinsing*: Regen-2xDiff, Regen-4xDiff.
- *Adv Embedding Grey-box*: AdvEmbG-KLVAE8.
- *Adv Embedding Black-box*: AdvEmbB-CLIP, AdvEmbB-SdxlVAE, AdvEmbB-KLVAE16.
- *Adv Surrogate Detector*: AdvCls-WM1&WM2.

E.6 Details of Benchmarking Attacks

In addition to benchmarking watermarks, WAVES also facilitates the analysis from the perspective of attacks. Table 3 provides a leaderboard of individual attacks. A strong attack should result in low post-attack detection performance while simultaneously preserving image quality for practical uses. Therefore, we benchmark attacks according to both performance and quality degradation. Based on three Performance vs. Quality 2D plots in Figure 2, we first select two performance thresholds, TPR@0.1%FPR=0.95 and TPR@0.1%FPR=0.7, ensuring intersections with most attack curves. Then, we calculate the quality degradation for each attack at these two performance thresholds, denoted as Q@0.95P and Q@0.7P. Given that some attack curves do not intersect with either threshold, we also compute each attack's average performance and quality degradation across all strengths, termed as Avg P and Avg Q. We report these metrics — Q@0.95P, Q@0.7P, Avg P, and Avg Q — for attack comparison. Based on them, we also provide a ranking of 26 attacks for each watermarking method for reference. During this ranking process, we incorporate a 0.01 buffer for both P and Q, meaning that if the difference between any two values is less than 0.01, they are considered a tie in terms of ranking.

F Details of Attacks

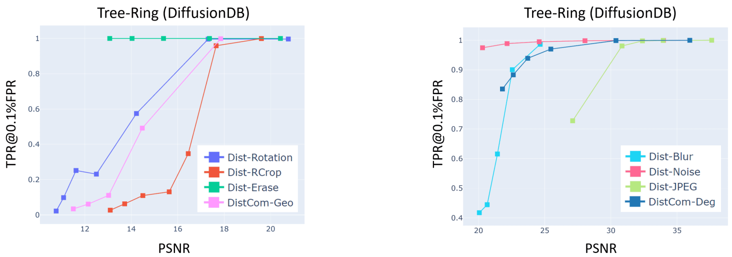
F.1 Distortion Attacks

For single distortions, we consider, as described in Section 2.2, eight types: rotation, resized-crop, random erasing, brightness adjustment, contrast adjustment, Gaussian blur, Gaussian noise, and JPEG compression. For each distortion, we consider five evenly distributed distortion strengths between minimum and maximum; the minimums and maximums are listed as follows.

- *Rotation*: rotate 9° to 45° clock-wise.
- *Resized-crop*: crop 10% to 50% of the image area.
- *Random erasing*: erase 5% to 25% of the image area and fill with gray color.
- *Brightness adjustment*: increase image brightness by 20% to 100%.
- *Contrast adjustment*: increase image contrast by 20% to 100%.
- *Gaussian blur*: blur with kernel size from 4 to 20 pixels.
- *Gaussian noise*: add Gaussian random noise with standard deviation from 0.02 to 0.1 (when pixel values normalized to [0, 1]).
- *JPEG compression*: compress with JPEG quality score from 90 to 10.

It is worth noting that our strength selections are more conservative than most of the watermark papers, such as (Wen et al., 2023; Fernandez et al., 2023). This is because we want to keep the image quality after distortion within a reasonable interval compared to the other attacks. While some watermark papers intentionally select unreasonably large distortion strength (for example, cropping 90% of image area in (Fernandez et al., 2023), or Gaussian blurring with kernel size 40 (Wen et al., 2023)) to demonstrate their robustness under some distortions. We implement the distortions following the standard image augmentations in the *torchvision* library.

For combinations of distortions (also called combo distortions in paper for short), we apply each single distortion with the same relative strength, where the relative strength is between 0 and 1, normalized with respect to the minimum and maximum strengths above. For combinations of geometric, photometric, and degradation distortions, we consider five evenly distributed normalized strengths from 0.05 to 0.45. For combinations of all distortions, we consider five evenly distributed normalized strengths from 0.05 to 0.20. The relative strengths are selected for reasonable image qualities after distortions again.



(a) Geometric distortions (PSNR ↑)

(b) Degradation distortions (PSNR ↑)

Figure 9. Distortions and their combinations. We combine three types of distortions: geometric, photometric, and degradation, both individually and collectively. By comparing quality-performance plots, we see combinations of distortions do not necessarily lead to better attacks.

F.2 Regeneration Attacks

Following the language of Section 2, regeneration attacks Zhao et al. (2023a) use off-the-shelf VAEs and diffusion models to transfer a target image $x \in \mathcal{X}$ to a latent representation followed by a restoration to $x' \in \mathcal{X}$ that is faithful to its original representation, i.e., $x' \approx x$. Since the chosen VAE or diffusion model will not be contained by the attacker's model of interest, the entire regeneration is likely to disrupt the latent representation of x, thereby damaging an embedded watermark. However, since the capacity of the attacker's regenerative model is inferior to the target model, x' will likely be of reduced quality. In this work, the target model is Stable Diffusion v2.1 while the surrogate model used for regeneration is Stable Diffusion v1.4.

Figure 12 demonstrates that a long diffusion or low-quality VAE attack will significantly reduce watermark detectability but at the expense of reduced image quality, which is clear by visual inspection of the sequence of images in Figure 10. Rising regenerations achieve similar reductions in detection, although too deep of rinsing regenerations (> 30 noising steps) significantly alter image quality as evidenced by Figure 11.

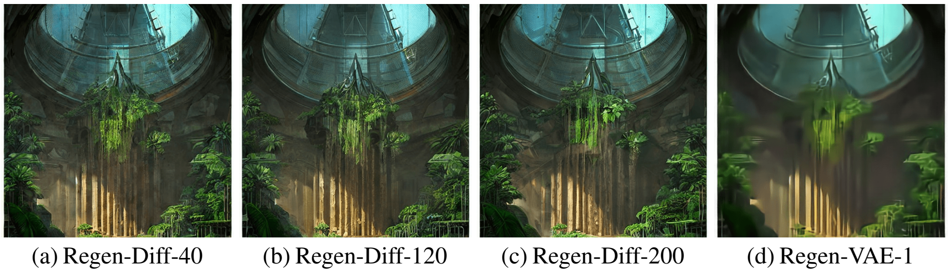

(a) Regen-Diff-40 (b) Regen-Diff-120 (c) Regen-Diff-200 (d) Regen-VAE-1

Figure 10. Regenerative diffusion with varying depth of noising steps and a VAE regeneration with a low quality factor.

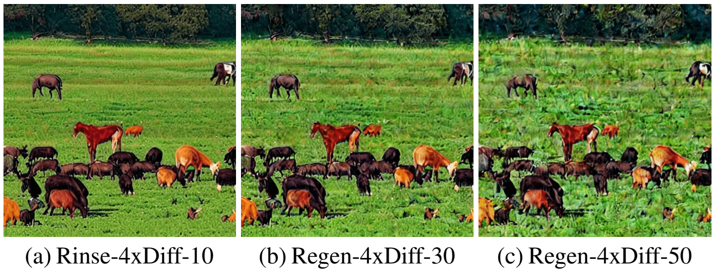

(a) Rinse-4xDiff-10 (b) Regen-4xDiff-30 (c) Regen-4xDiff-50

Figure 11. 4x rinsing regeneration with varying depth of noising steps per diffusion.

F.2.1 Prompted Regeneration

We propose a simple variation on a regenerative diffusion attack: if an image is produced via a known prompt, then an attacker uses the prompt to guide the diffusion of their surrogate model. This type of attack is reasonable and realistic for users of online generative models such as DALL·E or Midjourney. Figure 12 and Tables 5 & 3 indicate that this type of attack, labeled Regen-DiffP is slightly stronger than conventional Regen-Diff.

F.2.2 Mixed Regeneration

Mixed regeneration refers to any style of attack that uses a regenerative diffusion on an image followed by VAE-style regeneration for the purposes of denoising. In Figure 12, we label examples of such attacks as RinseD-VAE and RegenD-KLVAE, which respectively denote VAE and KLVAE denoising following a 4x rinsing regeneration with 50 steps (Rinse-4xDiff-50). According to Figure 12, such a combination improves PSNR and CLIP-FID, as opposed to a Rinse-4xDiff alone. The restorative effects of mixed regeneration are visually observable for shallower (i.e., 2x or 3x) rinsing regenerations, as depicted in Figure 13. We do not extensively study or rank such attacks in this work, but include them as a future topic of research.

All tested regeneration attacks are summarized as follows, with five evenly divided strengths between the listed minimum and maximum unless specified otherwise:

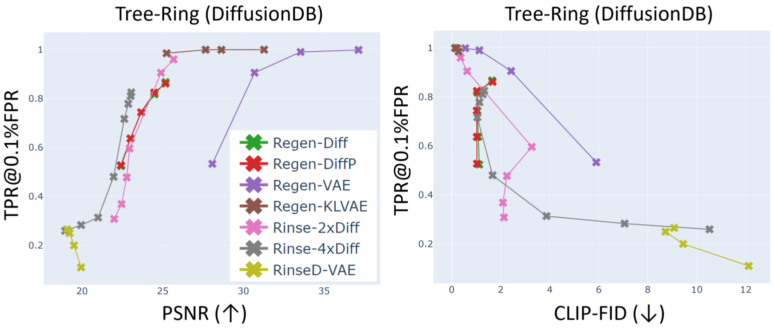
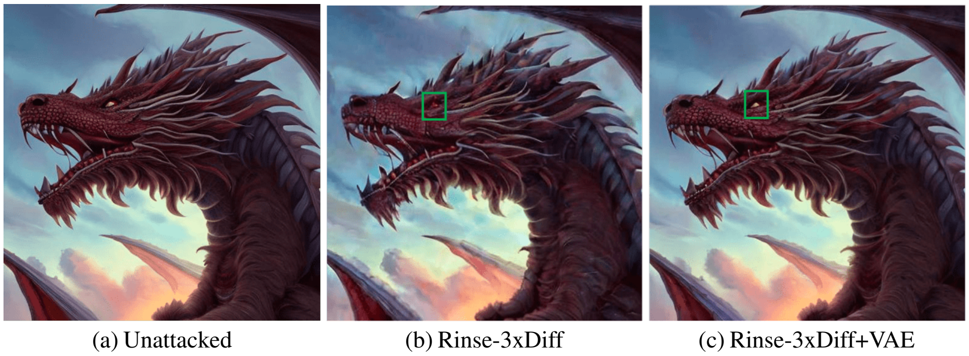


Figure 12. Regeneration attacks on Tree-Ringk. Regen-Diff is a single diffusive regeneration and Rinse-[N]xDiff is a rinsing one with N repeated diffusions, with the number of noising steps as attack strength. Regen-VAE uses a pre-trained VAE with quality factor as strength and Regen-KLVAE uses pre-trained KL-VAEs with bottleneck size as strength. RinseD-VAE applies a VAE as a denoiser after Rinse-4xDiff.


(a) Unattacked

(b) Rinse-3xDiff

(c) Rinse-3xDiff+VAE

Figure 13. An image of a dragon attacked using a 3x rinsing regeneration. Pushing the image through a VAE restores image quality, noticeable in the eye color of the dragon (indicated by the green box). Image is drawn from the Gustavosta Stable Diffusion dataset available @ `https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts`.

- *Regeneration via diffusion*: passes an image through Stable Diffusion v1.4 with strength as the number of noise/de-noising steps timesteps, 40 to 200.
- *Regeneration via prompted diffusion*: passes an image through Stable Diffusion v1.4 conditioned on its generative prompt with strength as the number of noise/de-noising steps timesteps, 40 to 200.
- *Regeneration via VAE*: Image is encoded then decoded by a pre-trained VAE (bmshj2018) Ballé et al. (2018) with strength as quality level from 1 to 7.
- *Regeneration via KL-VAE*: Image is encoded and then decoded by a pre-trained KL-regularized autoencoder with strength as bottleneck sizes 4, 8, 16, or 32.
- *Rinsing generation 2x*: an image is noised then de-noised by Stable Diffusion v1.4 two times with strength as number of timesteps, 20-100 (per diffusion).
- *Rinsing generation 4x*: an image is noised then de-noised by Stable Diffusion v1.4 two times with strength as number of timesteps, 10-50 (per diffusion).
- *Mixed Regeneration via VAE*: an image passed through a rinsing regeneration 4x (for 50 timesteps each) and then a VAE with strength as quality level from 1-7.
- *Mixed Regeneration via KL-VAE*: an image passed through a rinsing regeneration 4x (for 50 timesteps each) and then a KL-VAE with strength as bottleneck sizes 4, 8, 16, or 32.

F.3 Adversarial Attacks

In this section, we detail our adversarial attacks. Figure 15 visually summarizes these methods.

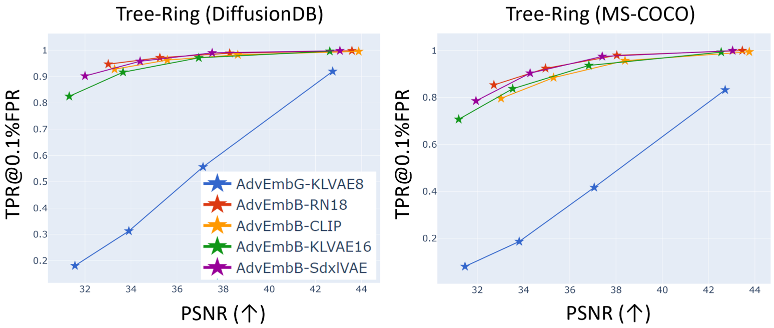


Figure 14. Adversarial embedding attacks target Tree-Ring at strengths of {2/255, 4/255, 6/255, 8/255}. Tree-Ring shows vulnerability to embedding attacks, especially when the adversary can access the VAE being used.

(A) Embedding Attacks. Watermark detection can be thwarted by perturbations on image embedding. Such attacks have been used against Multimodal Large Language Models like GPT-4V (Dong et al., 2023) and shown good transferability (Inkawhich et al., 2019). We examine if attacks on off-the-shelf embedding models can transfer to watermark detectors. Given an encoder $f:\mathcal{X}\rightarrow\mathcal{Z}$ mapping images to latent features, we craft an adversarial image x_{adv} to diverge its embedding from the original watermarked image x, within an l_∞ perturbation ball limit: $\max_{x_{adv}} \|f(x_{adv})-f(x)\|_2$, s.t. $\|x_{adv}-x\|_\infty \leq \epsilon$. We approximately solve this using the PGD (Madry et al., 2017) algorithm (see details in Appendix F.3.1), and see if the adversarial image transfers to real watermark detectors.

We evaluate five off-the-shelf encoders. **AdvEmbB-RN18** uses a pre-trained ResNet18 (He et al., 2016), targeting the pre-logit feature layer. **AdvEmbB-CLIP** employs CLIP's (Radford et al., 2021) image encoder. **AdvEmbG-KLVAE8** utilizes the encoder of KL-VAE (f8) which is used in the victim latent diffusion model. This is a grey-box setting but reflects the use of public VAEs in proprietary models (for example, DALLE·3 uses a public KL-VAE according to `https://cdn.openai.com/papers/dall-e-3.pdf`). Further, we do ablation studies on KL-VAE (f16), which has a different architecture but is trained on the same data, and on SDXL-VAE (Podell et al., 2023), an enhanced version of KL-VAE (f8). They are black-box attacks and are labeled **AdvEmbB-KLVAE16** and **AdvEmbB-SdxlVAE**.

As shown in Figure 14, Tree-Ring is vulnerable to embedding attacks, particularly under the grey-box condition where TPR@0.1%FPR can drop to nearly zero, effectively removing most watermarks. This is because the detection process of Tree-Ring first maps the image to the latent representation through the encoder of KL-VAE (f8), then conducts inverse DDIM to retrieve the watermark. The embedding attack changes the latent representation severely; therefore, watermark retrieval becomes very difficult. Using similar yet distinct VAEs, attack effectiveness diminishes but still manages to remove some watermarks, with KL-VAE (f16), trained on the same images, demonstrating the highest transferability. CLIP-based attacks also achieve some success, especially on natural images like MS-COCO, likely due to CLIP being trained on natural images akin to those in MS-COCO, enhancing the transferability. Conversely, Stable Signature and StegaStamp demonstrate robustness against embedding attacks (Figure 2), likely because their detectors are trained independently from generative models, differing significantly from standard classifiers and VAEs. Hence, our attacks fail to effectively transfer to their detectors.

(B) Surrogate Detector Attacks. Watermark detection hinges on a detector that decodes and verifies messages from watermarked images. Adversaries might acquire numerous watermarked and non-watermarked images to train a surrogate detector, and transfer attacks on it to the actual watermark detector. Figure 15 explores our various settings. **AdvCls-UnWM&WM** trains a surrogate detector with both watermarked and non-watermarked images from the victim generative model, as per Saberi et al. (2023). Note that this is an unrealistic setting for proprietary models since all their outputs are assumed to be watermarked. **AdvCls-Real&WM** trains the surrogate watermark detector with watermarked and non-watermarked images, where non-watermarked images are sampled from the ImageNet dataset (not from the generative model). This approach is more applicable to proprietary models. **AdvCls-WM1&WM2** only uses watermarked images. It actually trains a surrogate watermark message classifier to distinguish two users. Suppose the system assigns a particular message to each user for identification purposes, the adversary can collect the training data from two users' outputs, with an identical set of prompts. Adversarial attacks on this surrogate model

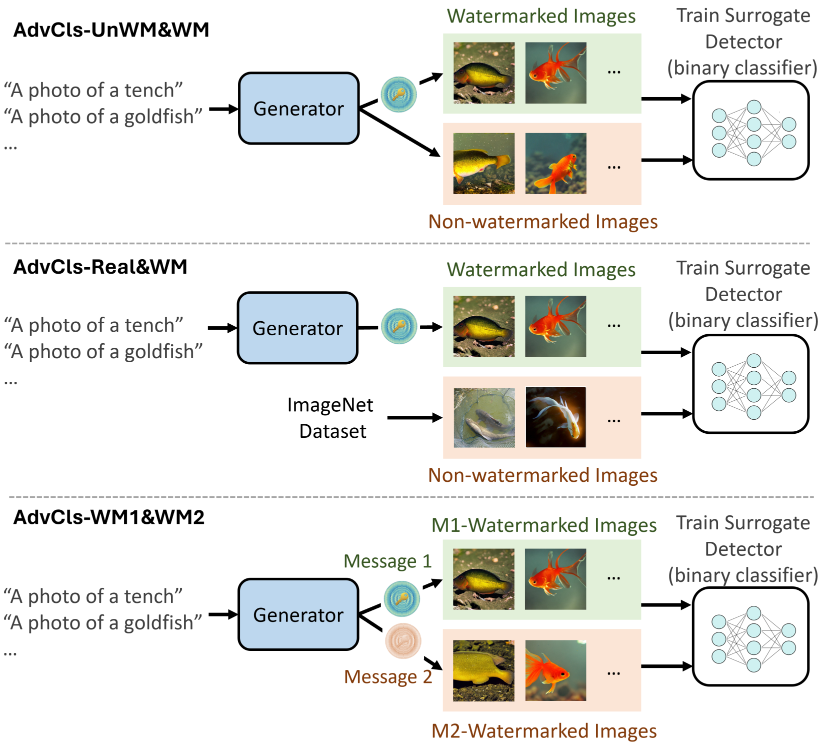


Figure 15. Three settings for training the surrogate detector. The Generator is the victim generator under attack. We externalize the watermarking process for simplicity, but it could be in-processing watermarks. After training the surrogate detectors, the adversary performs PGD attacks on them to flip the labels.

aim at user misidentification. All surrogate detectors are fine-tuned on ResNet18. We use ImageNet text prompts "A photo of a {*class name*}" to generate training images (see details in Appendix F.3.2).

With the trained surrogate detector $f : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{Y} = \{0, 1\}$, adversaries launch targeted attacks. The goal is to craft an adversarial image x_{adv} from an original image x so that f incorrectly predicts the target label y_{target} (i.e., wrong label), minimizing the following with cross-entropy loss: $\min_{x_{adv}} L(f(x_{adv}), y_{target})$, s.t. $\|x_{adv} - x\|_\infty \leq \epsilon$. It enables adversaries to erase watermarks from marked images or implant them into clean images in the first two settings, and to disrupt user identification as well as watermark detection in the third setting. We solve it with the PGD algorithm.

Figure 18 shows Tree-Ring's vulnerability to surrogate detector-based attacks. In **AdvCls-UnWM&WM**, the adversary accessing non-watermarked images has good transferability and removes watermarks effectively. However, it fails to add watermarks to clean images (spoofing attack), as detailed in Figure 19. The reason behind this is explored in Appendix G.2, where we find the attacker disrupts the entire latent space, not just the watermark (as shown in Figure 20). Conversely, the spoofing attack fails to embed the precise watermark. **AdvCls-Real&WM** attack fails entirely, likely due to the surrogate model appearing to differentiate real from generated images, using broader features than the watermark. The newly proposed **AdvCls-WM1&WM2** successfully attacks Tree-Ring using only watermarked images. Like the first scenario, the surrogate model fails to precisely locate watermarks but learns the mapping to the latent feature space, allowing a PGD attack to remove the watermark by disturbing the entire latent space (see Figure 21). In user identification tasks (Figure 22), the attack doesn't consistently mislead the detector into misidentifying User1's watermarked images as User2's (targeted misidentification). Instead, imprecise perturbations often lead to incorrect attribution of User1's images to others.

Figure 2 shows that Stable Signature and StegaStamp are robust to these attacks. Even with high surrogate classifier accuracy in AdvCls-UnWM&WM, adversarial examples fail to transfer to the true detector, possibly due to reliance on different features than those used by the true detector.

F.3.1 Embedding Attack

The embedding attacks use off-the-shelf encoders and perform untargeted attacks. We use the Projected Gradient Descent (PGD) algorithm (Madry et al., 2017) to optimize the adversarial examples. We conduct the attack using a range of perturbation budgets ϵ, specifically {2/255, 4/255, 6/255, 8/255}. All the attacks are configured with a step size of $\alpha = 0.05 * \epsilon$ and the number of total iterations of 200. The attacks are on the watermarked images, aiming to remove the watermarks by perturbing their latent representations.

F.3.2 Surrogate Detector Attack

Figure 15 illustrates the three settings of training the surrogate detectors. In all three settings, we train the surrogate detectors by fine-tuning the ResNet18[2] for 10 epochs with a learning rate of 0.001 and a batch size of 128. The training images are either generated by the victim generator with the ImageNet text prompts "A photo of a {*ImageNet class name*}," or real ImageNet images. We randomly shuffle those images and build the binary training set according to each setting. In the AdvCls-UnWM&WM setting, we train the surrogate detector with 3000 images (1500 images per class) since we find a larger training set might have the overfitting problem. In the AdvCls-Real&WM and AdvCls-WM1&WM2 settings, we train the surrogate detector with 15000 images (7500 images per class). The watermarked images in AdvCls-WM1&WM2 are embedded with two distinct messages. One message is the one used in the test watermarked images. The other one is randomly generated. In all three settings, we use 5000 images (2500 images per class) for validation (derived from the same source as the training set), and the training yields nearly 100% validation accuracy in all cases.

After completing the training phase, the adversary executes a Projected Gradient Descent (PGD) attack on the surrogate detector using the testing data (DiffusionDB, MS-COCO, DALL·E3). In all three settings, we conduct the attack using a range of perturbation budgets ϵ, specifically {2/255, 4/255, 6/255, 8/255}. The attack is configured with a step size of $\alpha = 0.01 * \epsilon$ and the number of total iterations of 50. By flipping the label, the adversary can either try to remove the watermarks or add the watermarks. The analyses of results appear in Appendix G.2.

G Additional Results

G.1 More Results for Identification

Figure 16 shows the Performance vs. Quality degradation plots under the user identification setting. Table 5 presents the ranking of attacks in the identification setup. Figure 17 is a radar plot of identification accuracy.

G.2 More Analyses on Surrogate Detector Attacks

The AdvCls-UnWM&WM attack leverages a surrogate model to distinguish between images that are watermarked and those that are not. As demonstrated in Figure 18, the PGD attack is effective in removing watermarks by flipping the label of watermarked images. This raises a question: Is it possible to similarly ‘add’ watermarks to clean images by flipping their labels? This process, commonly referred to as a spoofing attack, which demonstrates a false detection of watermarks in clean images, is explored in our study.

However, as illustrated in Figure 19, our attempts to add watermarks to clean images by simply flipping the labels were unsuccessful. In this experiment, detailed in Figure 19, we focus exclusively on unwatermarked images, aiming to introduce watermarks, while leaving already watermarked images untouched. Despite employing the most intensive perturbations, we were unable to artificially add watermarks to these images. This outcome leads to an intriguing inquiry: Why is the technique effective in removing watermarks but not in adding them? We delve into the underlying reasons for this asymmetry in Figure 20.

The insights from Figure 20 reveal that the surrogate model does not exactly remove the watermark. Instead, it perturbs the watermark along with other features within the latent space. The disturbance

[2] https://pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html

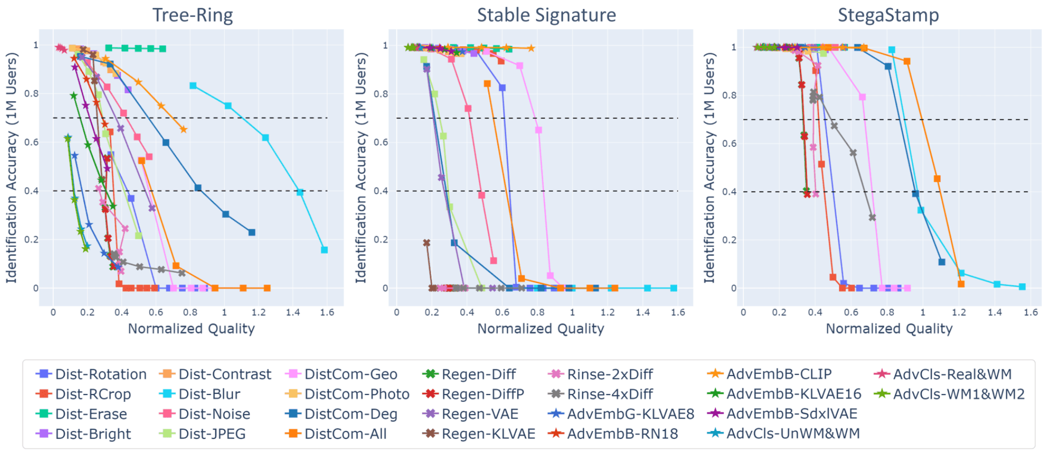


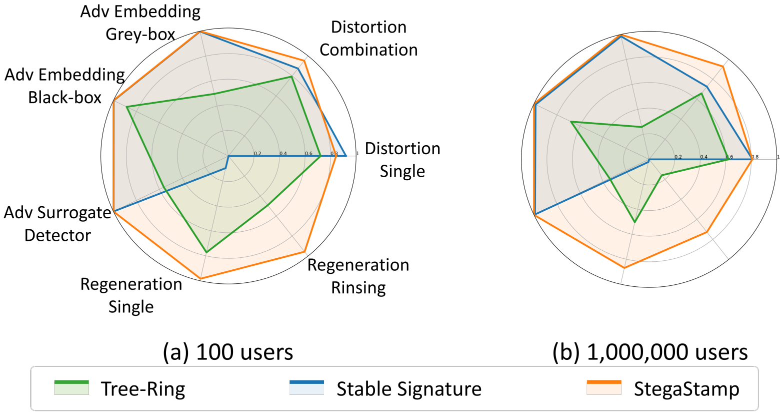
Figure 16. **Aggregated performance vs. quality degradation 2D plots under identification setup (one million users).** We evaluate each watermarking method under various attacks. Two dashed lines show to thresholds used for ranking attacks.

Table 5. **Comparison of attacks across three watermarking methods under the identification setup (one million users).** Q denotes the normalized quality degradation and P denotes the performance as derived from aggregated 2D plots. Q@0.7P measures quality degradation at a 0.7 performance threshold where "inf" denotes cases where all tested attack strengths yield performance above 0.7, and "-inf" where all are below. Q@0.4P is defined analogously. Avg P and Avg Q are the average performance and quality over all the attack strengths. The lower the performance and the smaller the quality degradation, the stronger the attack. For each watermarking method, we rank attacks by Q@0.7P, Q@0.4P, Avg P, Avg Q, in that order, with lower values (↓) indicating stronger attacks. The top 5 attack of each watermarking method are highlighted in red.

Attack	Tree-Ring					Stable Signature					StegaStamp				
	Rank	Q@0.7P	Q@0.4P	Avg P	Avg Q	Rank	Q@0.7P	Q@0.4P	Avg P	Avg Q	Rank	Q@0.7P	Q@0.4P	Avg P	Avg Q
Dist-Rotation	8	-inf	0.434	0.131	0.648	12	0.613	0.642	0.400	0.650	4	0.454	0.500	0.288	0.616
Dist-RCrop	11	-inf	0.592	0.094	0.463	24	inf	inf	0.972	0.461	6	0.602	0.602	0.494	0.451
Dist-Erase	26	inf	inf	0.986	0.490	25	inf	inf	0.988	0.489	25	inf	inf	1.000	0.483
Dist-Bright	22	inf	inf	0.913	0.304	23	inf	inf	0.982	0.305	22	inf	inf	0.995	0.317
Dist-Contrast	23	inf	inf	0.949	0.243	20	inf	inf	0.979	0.243	17	inf	inf	0.994	0.231
Dist-Blur	21	1.105	1.437	0.551	1.221	5	-inf	-inf	0.000	1.204	9	0.897	0.970	0.280	1.198
Dist-Noise	16	0.427	inf	0.728	0.395	8	0.415	0.480	0.633	0.390	24	inf	inf	1.000	0.360
Dist-JPEG	17	0.499	0.499	0.700	0.284	9	0.485	0.485	0.540	0.284	21	inf	inf	0.995	0.263
DistCom-Geo	9	-inf	0.559	0.105	0.768	13	0.788	0.835	0.519	0.767	7	0.676	0.717	0.359	0.733
DistCom-Photo	23	inf	inf	0.947	0.242	20	inf	inf	0.981	0.243	17	inf	inf	0.994	0.239
DistCom-Deg	18	0.556	0.864	0.570	0.694	7	0.216	0.281	0.183	0.679	8	0.870	0.957	0.737	0.664
DistCom-All	10	-inf	0.575	0.123	0.908	11	0.550	0.623	0.176	0.900	10	0.995	1.096	0.682	0.870
Regen-Diff	6	-inf	0.307	0.258	0.323	1	-inf	-inf	0.000	0.300	2	0.333	inf	0.766	0.327
Regen-DiffP	6	-inf	0.308	0.256	0.327	1	-inf	-inf	0.000	0.303	1	0.336	0.356	0.763	0.329
Regen-VAE	19	0.578	0.578	0.701	0.348	10	0.545	0.545	0.340	0.339	23	inf	inf	1.000	0.343
Regen-KLVAE	14	0.257	inf	0.810	0.233	6	-inf	-inf	0.047	0.206	17	inf	inf	0.999	0.240
Rinse-2xDiff	5	-inf	0.270	0.220	0.357	3	-inf	-inf	0.000	0.332	3	0.390	0.402	0.778	0.366
Rinse-4xDiff	1	-inf	-inf	0.110	0.466	4	-inf	-inf	0.000	0.438	5	0.488	0.676	0.687	0.477
AdvEmbG-KLVAE8	4	-inf	0.168	0.259	0.253	20	inf	inf	0.985	0.249	17	inf	inf	1.000	0.232
AdvEmbB-RN18	15	0.288	inf	0.811	0.218	17	inf	inf	0.990	0.212	14	inf	inf	1.000	0.196
AdvEmbB-CLIP	20	0.697	inf	0.798	0.549	26	inf	inf	0.991	0.541	25	inf	inf	1.000	0.488
AdvEmbB-KLVAE16	12	0.158	0.309	0.540	0.238	19	inf	inf	0.983	0.233	14	inf	inf	1.000	0.206
AdvEmbB-SdxlVAE	13	0.214	inf	0.692	0.221	17	inf	inf	0.986	0.219	14	inf	inf	1.000	0.204
AdvCls-UnWM&WM	2	-inf	0.123	0.352	0.145	14	inf	inf	0.991	0.101	11	inf	inf	1.000	0.101
AdvCls-Real&WM	25	inf	inf	0.986	0.047	14	inf	inf	0.990	0.092	11	inf	inf	1.000	0.106
AdvCls-WM1&WM2	2	-inf	0.118	0.343	0.139	14	inf	inf	0.991	0.084	13	inf	inf	1.000	0.129

alone is sufficient to confuse the detector, making it challenging to recognize the watermark. In contrast, successfully adding watermarks requires precise modifications in the latent space, rather than mere perturbations, which proves to be a more challenging task. The relative imprecision of this attack may stem from the 'transferable gap' between the surrogate model and the ground-truth detector. Notably, for the purpose of watermark removal, perturbing the latent space proves to be adequately effective.

These findings have led to the development of our proposed AdvCls-WM1&WM2 attack, which utilizes images watermarked with different messages (e.g., collected from two users, User1 and User2). The essential requirement for this approach is the surrogate model's ability to map images to the generator's latent space. This mapping allows the attacker to perturb the latent space, removing the watermark. In contrast to the AdvCls-UnWM&WM approach, which uses both watermarked and non-watermarked images for training (differing only in the latent space), AdvCls-WM1&WM2



(a) 100 users

(b) 1,000,000 users

Figure 17. Identification accuracy of three watermarks after attacks.

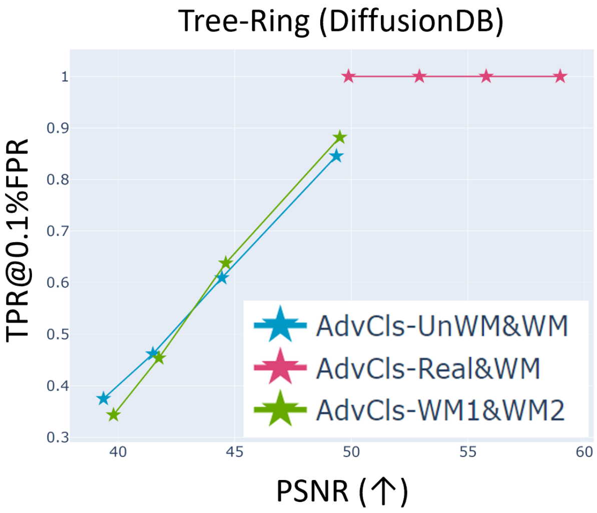


Figure 18. Adv. surrogate detector attacks on Tree-Ring.

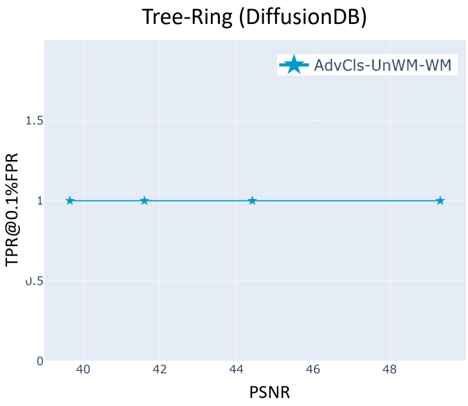


Figure 19. The spoofing attack fails for AdvCls-UnWM&WM.

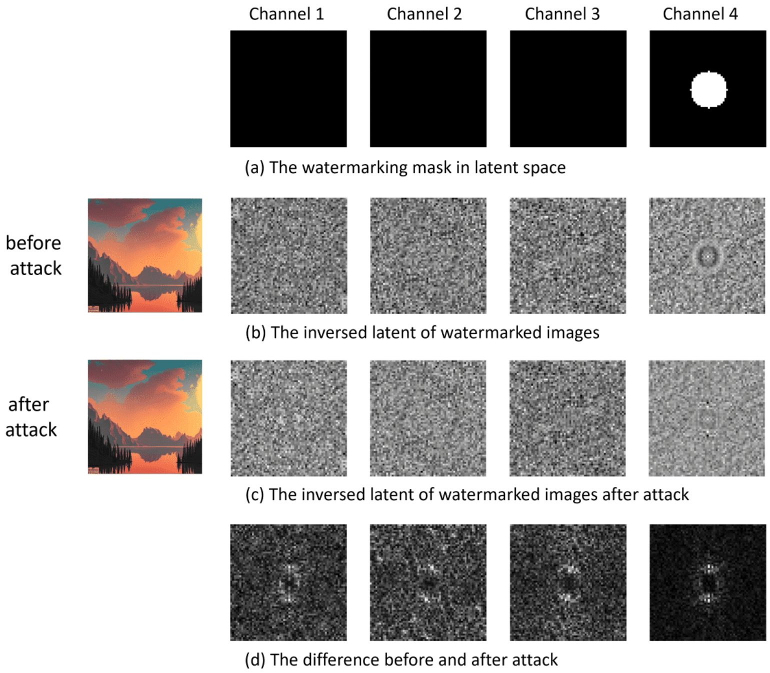


Figure 20. Visualization of AdvCls-UnWM&WM attack. (a) shows the watermarking mask of Tree-Ring where there are four channels, and we only watermark the last channel. The watermark message is the rings, which contain ten complex numbers that are not shown in the figure. (b) and (c) show the inversed latent before and after the attack in the Fourier space. We only show the real part of the latent. Clearly, the rings exist before the attack and vanish after the attack. (d) shows the magnitude of the element-wise difference before and after the attack. The attack not only perturbs the watermark part but also other features. The average magnitude change of the watermark-part and non-watermark-part is around 2:1. The attack successfully disturbs the watermark, albeit in an imprecise manner.

uses two sets of images, each embedded with a distinct watermark message (differing only in the latent space as well). Figure 21 shows that AdvCls-WM1&WM2 attack effectively disrupts the latent features of the images, including the watermarks. However, it lacks the precision to interchange the embedded watermark message. Consequently, while this attack can remove watermarks and mislead user identification—mistaking an image originally generated by User1 as belonging to another user—it cannot accurately manipulate the identification to frame User2 as desired by the attacker. The identification results in Figure 22 also support this finding. Although AdvCls-WM1&WM2 aims to misidentify images as belonging to User2, it often leads to misidentification as users other than User2. However, in a system with fewer users, like 100 users, and under intense attack conditions (e.g., strength=8), AdvCls-WM1&WM2 demonstrates a targeted identification success rate of 0.7%, showing a potential direction for attacks aimed at targeted user identification.

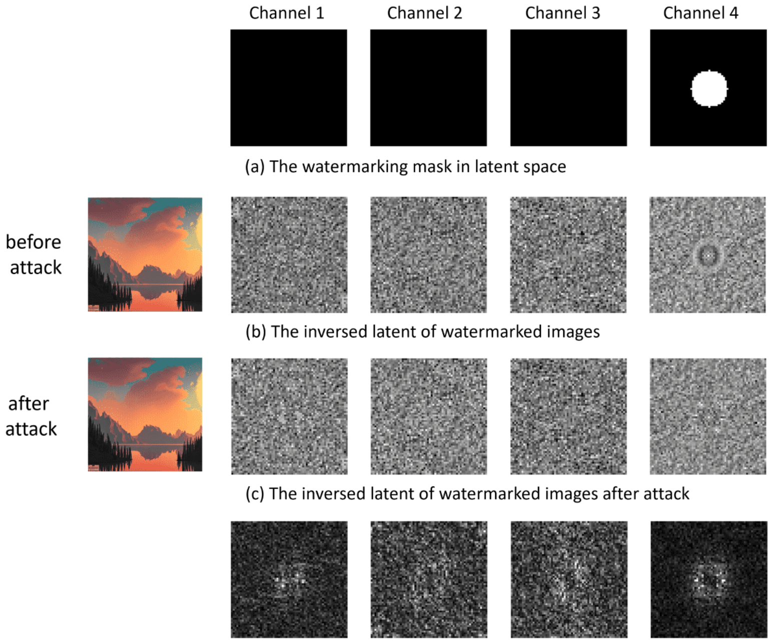
G.3 Visualization of Attacks

In Figure 23, we present visualizations of several attacks included in the WAVES benchmark. Prefix indicates the attack strategy, while suffix indicates the strength.

G.4 Full Results on DiffusionDB, MS-COCO and DALL·E3

H Limitations

We only stress-test the Tree-Ring, Stable Signature, and Stegastamp watermarking algorithms. We curated these watermarks for WAVES after an extensive literature review indicated these three techniques to be the most powerful and practical candidates for deployment in the wild. However, we



(a) The watermarking mask in latent space

(b) The inversed latent of watermarked images

(c) The inversed latent of watermarked images after attack

(d) The difference before and after attack

Figure 21. Visualization of AdvCls-WM1&WM2 attack. (a) and (b) are the same as that in Figure 20. (c) shows the inversed latent after the attack, where the watermark vanishes instead of changing to another watermark. (d) shows the magnitude of the element-wise difference before and after the attack. The attack not only perturbs the watermark part but also other features. The average magnitude change of the watermark-part and non-watermark-part is also around 2:1. Although the surrogate detector is trained to classify two different watermark messages. The attack based on it cannot change the watermark message from one to another but can effectively disturb the watermark.

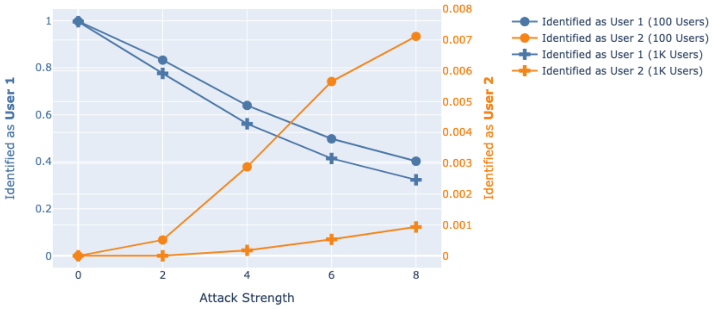
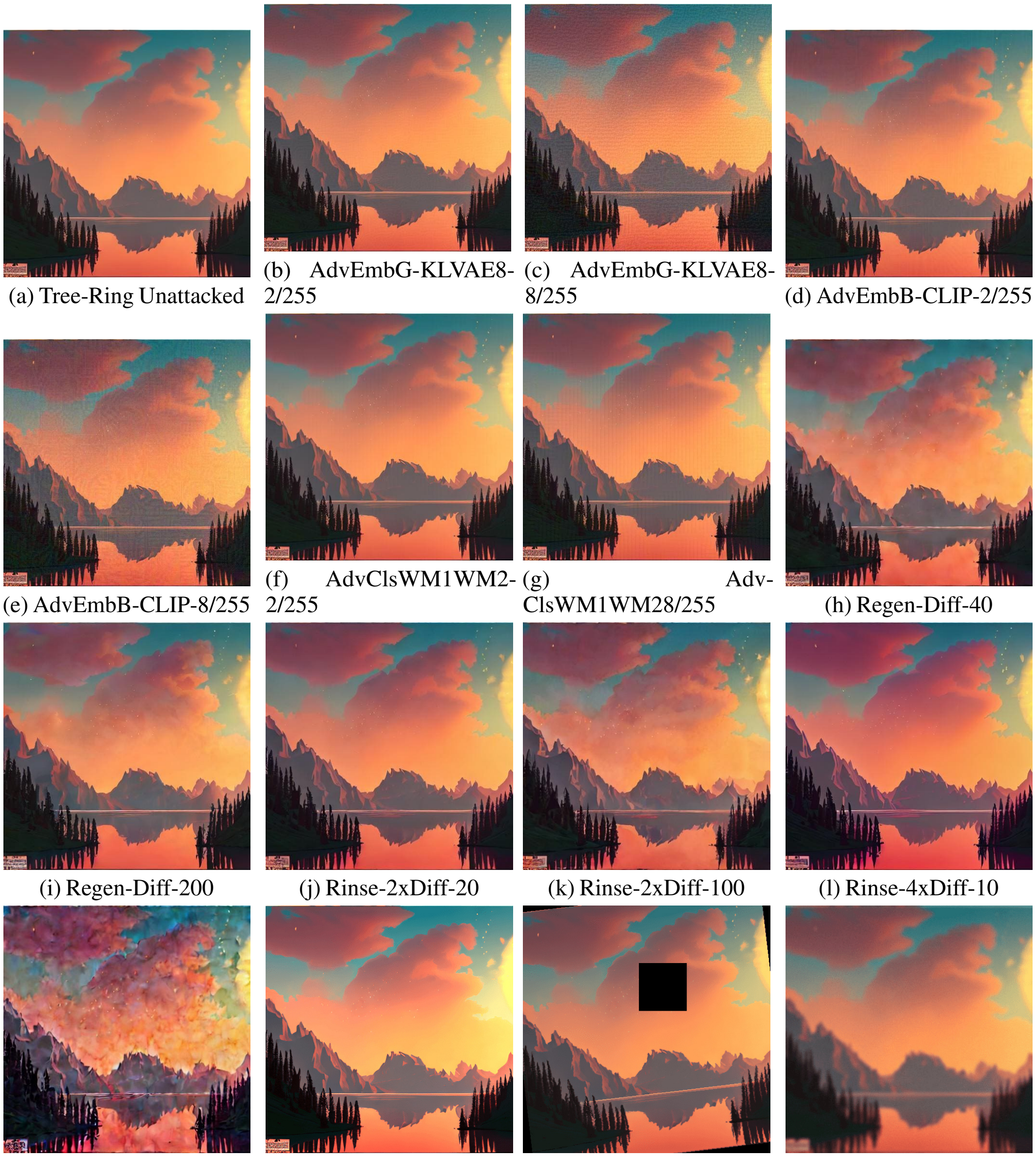


Figure 22. The user identification results for Tree-Ring under AdvCls-WM1&WM2 attacks. The original watermarked images are embedded with User1's message. AdvCls-WM1&WM2 tries to disrupt the latent feature of those images so that they can be misidentified as User2 generated. We simulate two settings: 100 users and 1000 users in total. The blue curves represent the proportion of images correctly identified as belonging to User1, while the orange curves show those misidentified as User2's. Note that, the axes for blue and orange curves have different ranges in the figure. With increasing attack strengths, the likelihood of correctly identifying them as User1's decreases significantly under both 100 and 1K user scenarios. However, misidentification as User2's images occurs notably only when the total number of users is small (e.g., 100 users).



(a) Tree-Ring Unattacked

(b) AdvEmbG-KLVAE8-2/255

(c) AdvEmbG-KLVAE8-8/255

(d) AdvEmbB-CLIP-2/255

(e) AdvEmbB-CLIP-8/255

(f) AdvClsWM1WM2-2/255

(g) AdvClsWM1WM28/255

(h) Regen-Diff-40

(i) Regen-Diff-200

(j) Rinse-2xDiff-20

(k) Rinse-2xDiff-100

(l) Rinse-4xDiff-10

(m) Rinse-4xDiff-50

(n) DistCom-Photo-0.15

(o) DistCom-Geo-0.15

(p) DistCom-Deg-0.15

Figure 23. A visual demonstration of various adversarial, regeneration, and distortion attacks on a Tree-Ring watermarked image. **Figure (a)** is the base unattacked image. The base prompt, drawn from DiffusionDB, is "digital painting of a lake at sunset surrounded by forests and mountains," along with further styling details.

emphasize our framework is extensible to any watermarking method. Additionally, our attack ranking method relies on author-selected TPR thresholds and image quality metrics that we believe will fairly capture attack potency based on existing literature and experimental studies. The use of other quality metrics (MSE, Watson-DFT, etc.) and differing TPR thresholds may affect attack rankings.

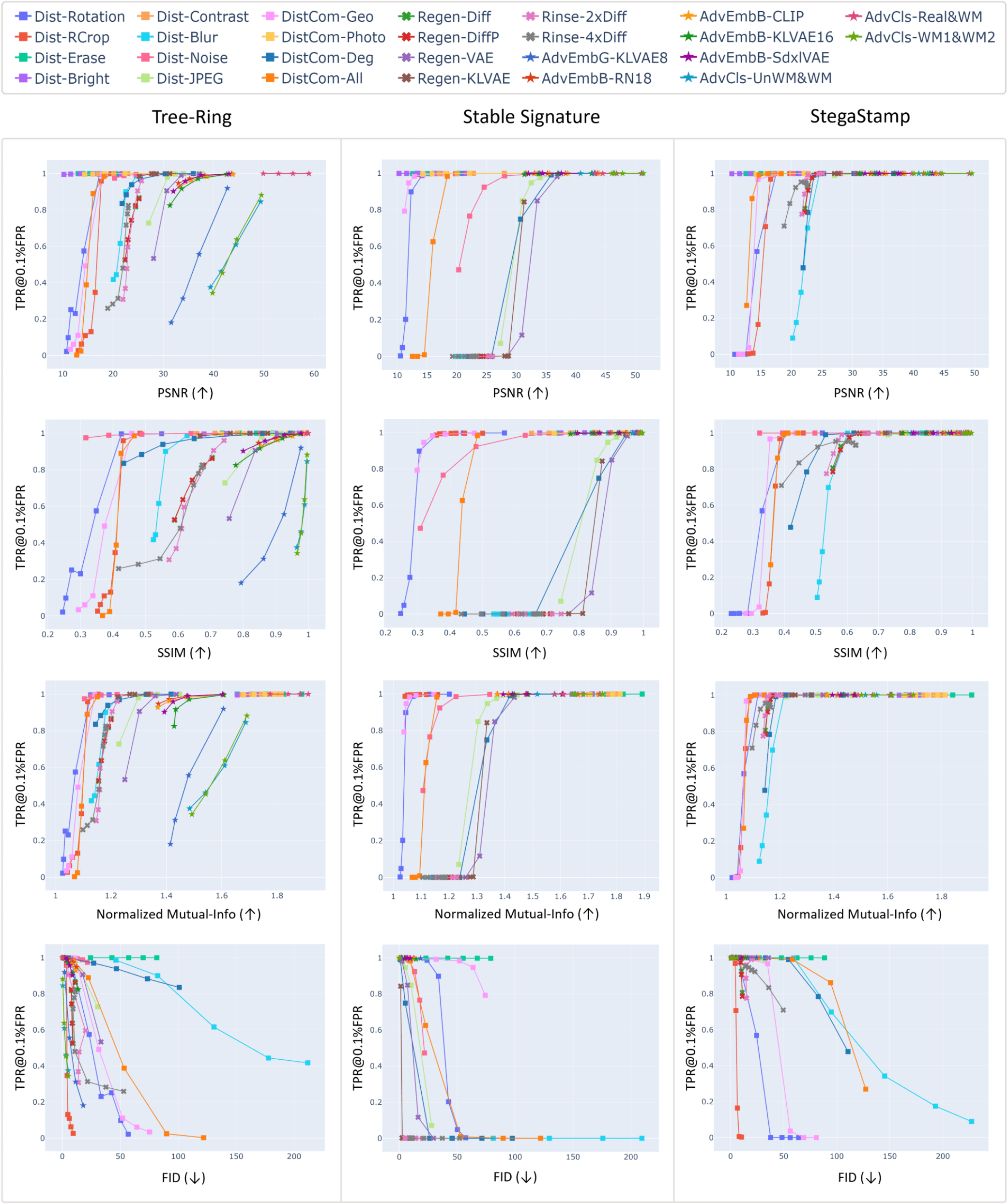


Figure 24. Evaluation on DiffusionDB dataset under the detection setup (part 1).

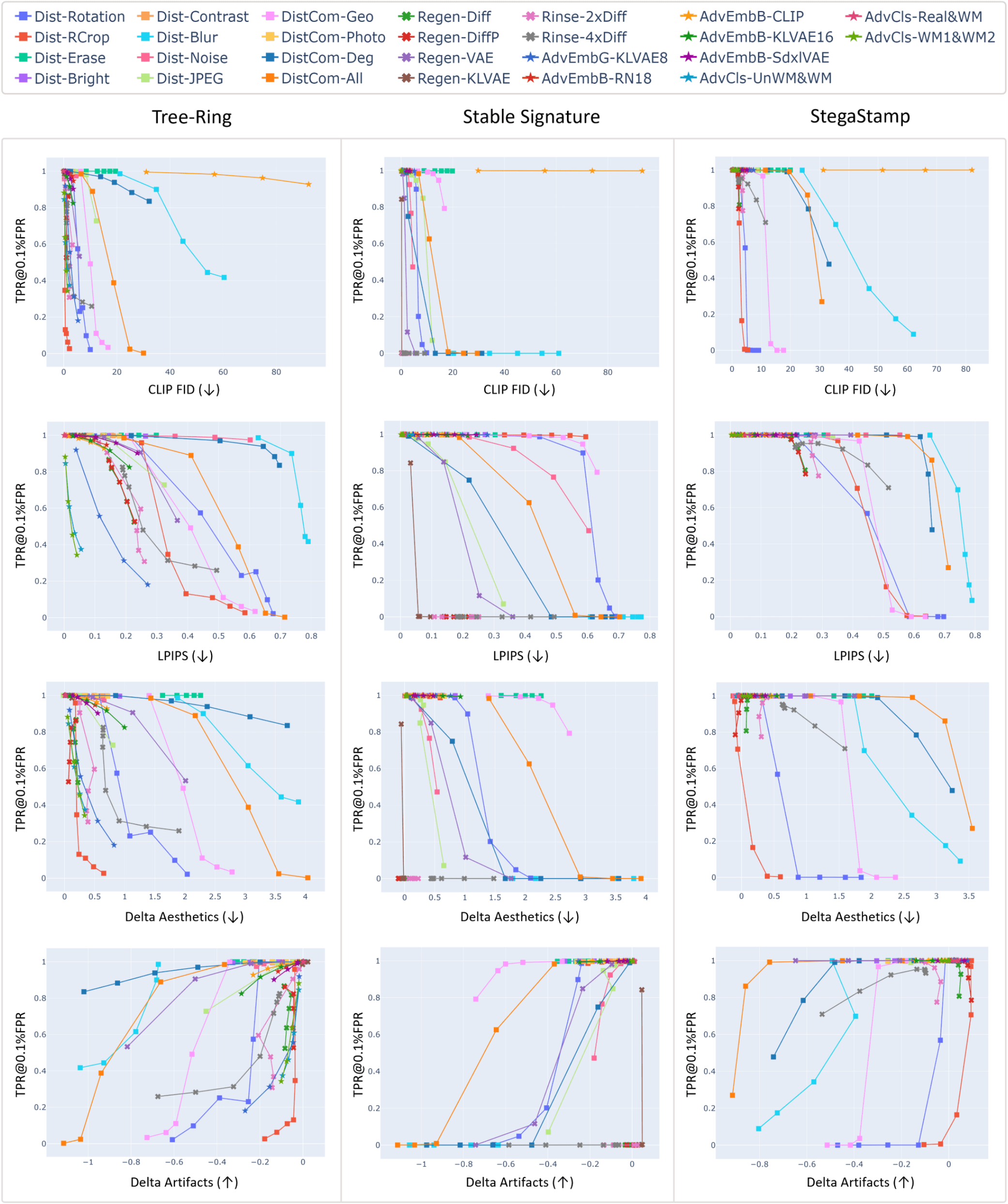


Figure 25. Evaluation on DiffusionDB dataset under the detection setup (part 2).

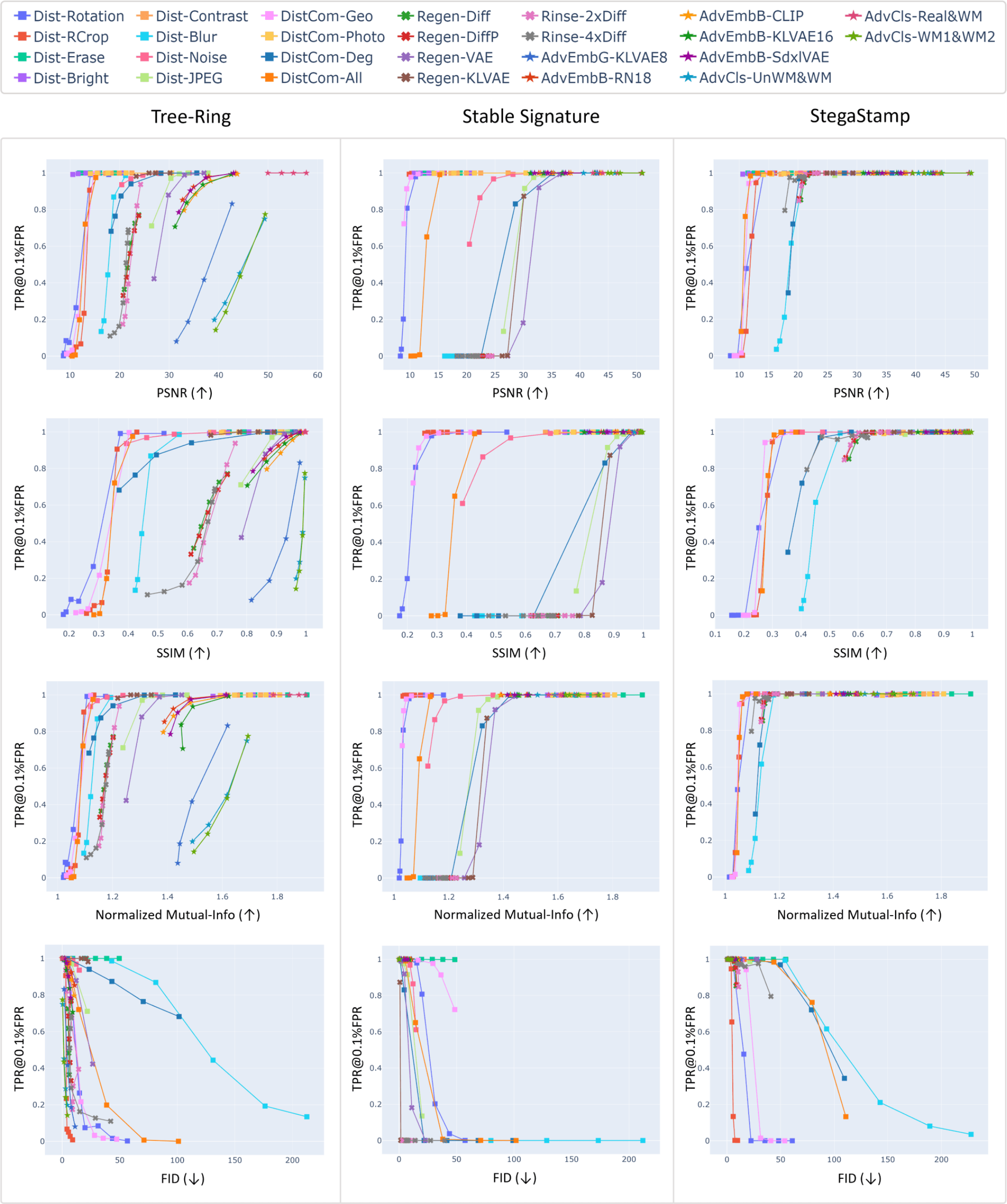


Figure 26. Evaluation on MS-COCO dataset under the detection setup (part 1).

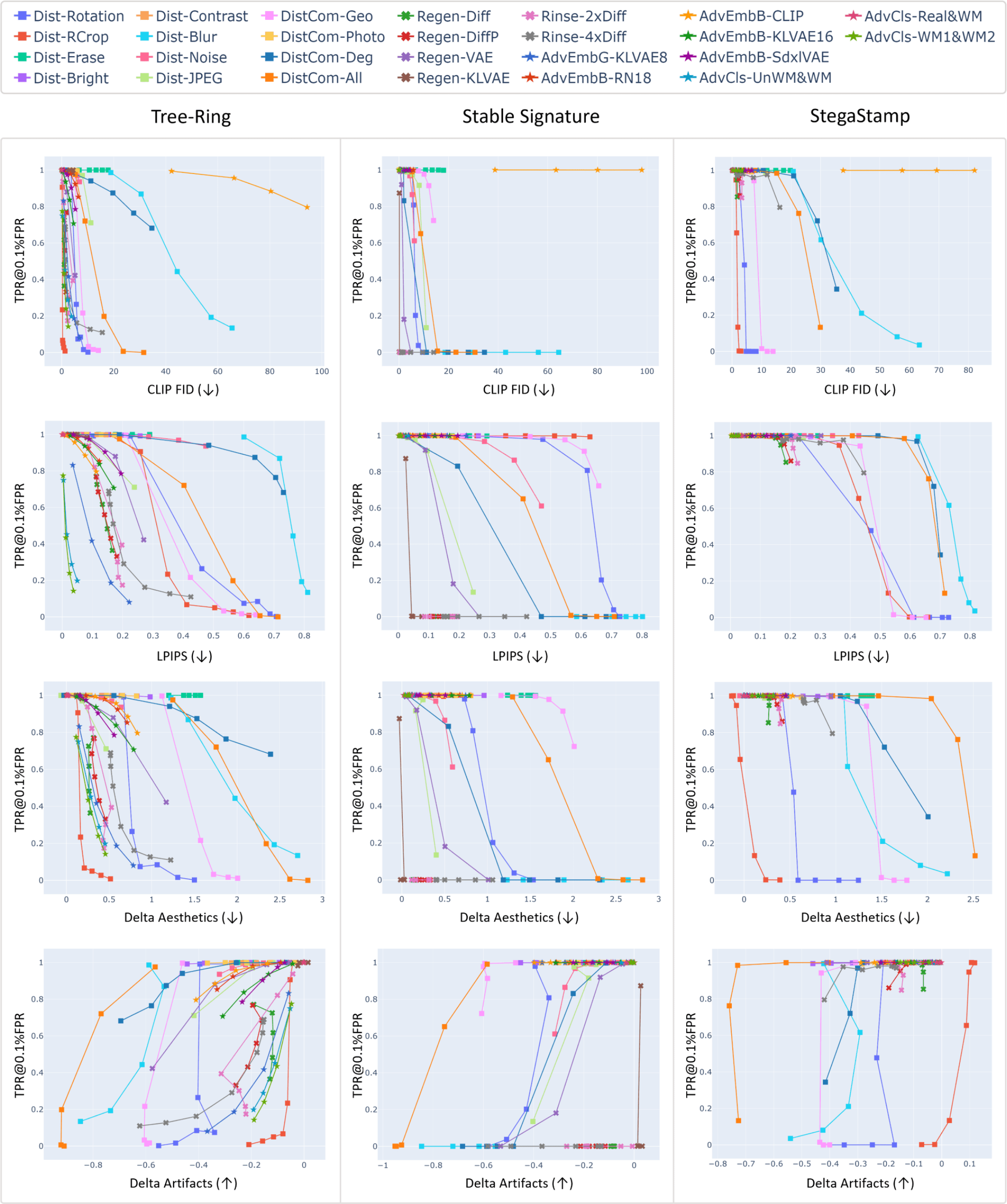


Figure 27. Evaluation on MS-COCO dataset under the detection setup (part 2).

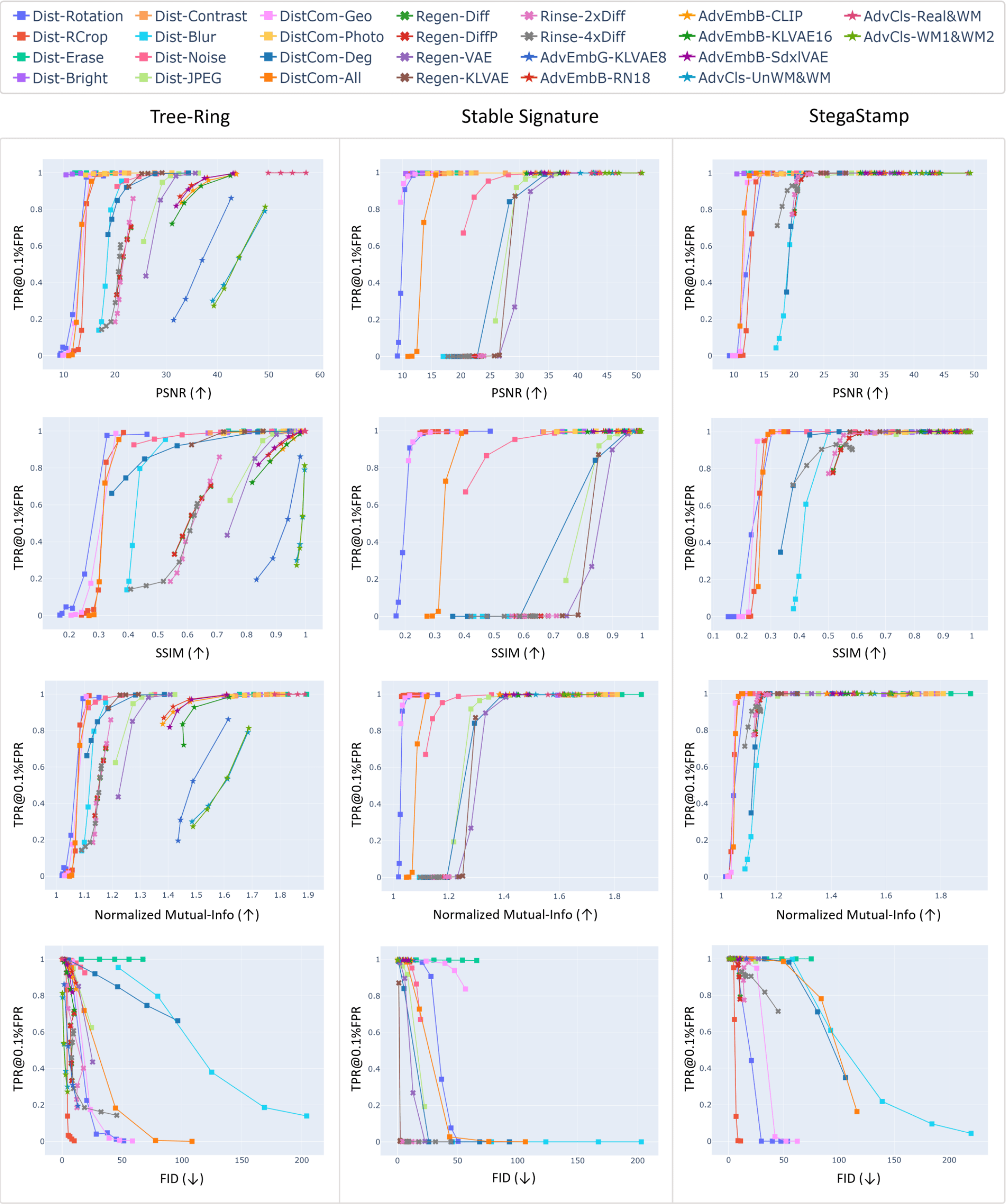


Figure 28. Evaluation on DALL·E3 dataset under the detection setup (part 1).

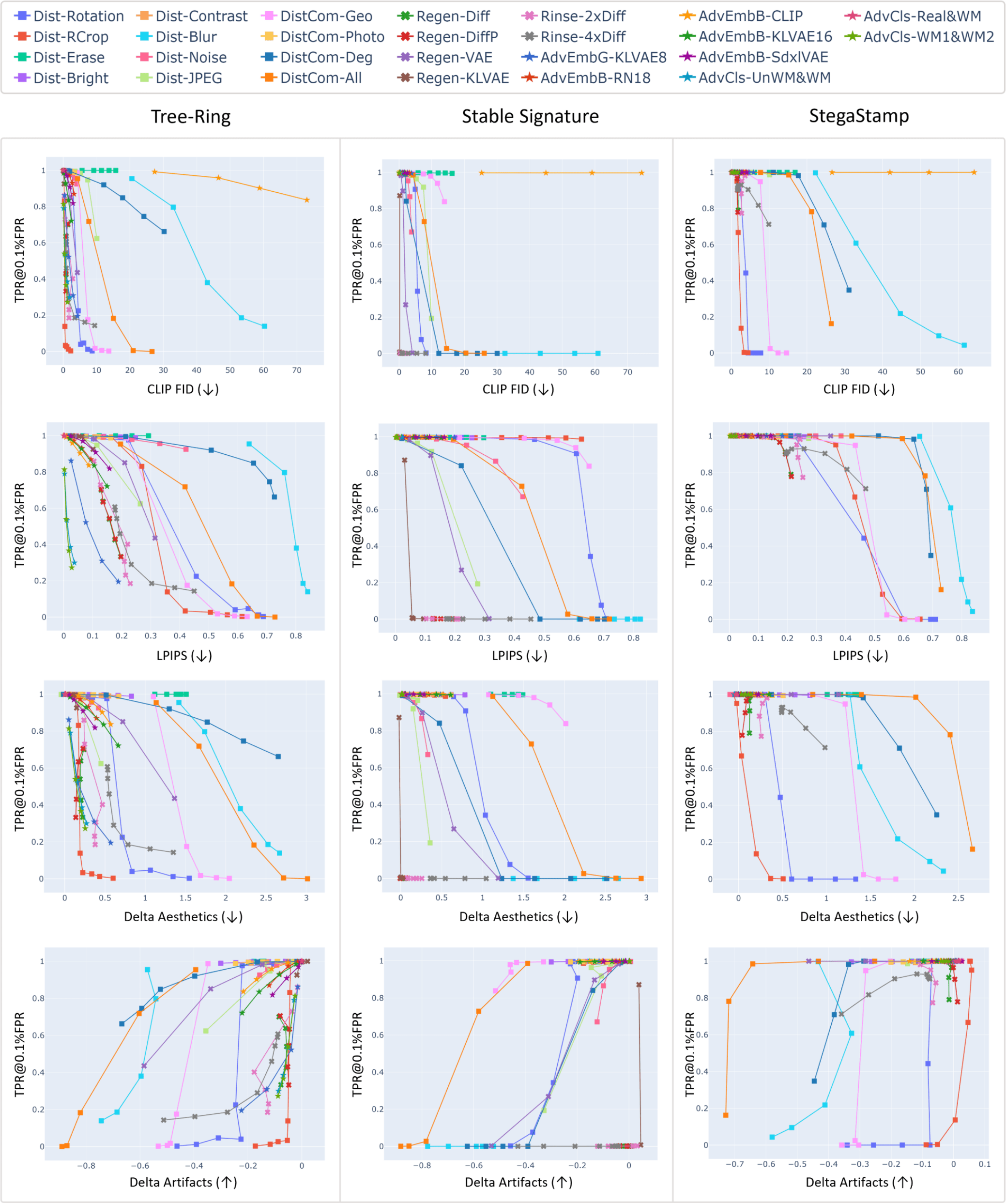


Figure 29. Evaluation on DALL·E3 dataset under the detection setup (part 2).