Decoupling Defense Strategies for Robust Image Watermarking

Jiahui Chen[1], Zehang Deng[2], Zeyu Zhang[3], Chaoyang Li[1], Lianchen Jia[1], Lifeng Sun[1,4,*]
[1]Tsinghua University, [2] Swinburne University of Technology, [3] The Australian National University
[4]Key Laboratory of Pervasive Computing, Ministry of Education, * Corresponding Author
chenjiah22@mails.tsinghua.edu.cn, sunlf@tsinghua.edu.cn

Abstract

Deep learning-based image watermarking, while robust against conventional distortions, remains vulnerable to advanced adversarial and regeneration attacks. Conventional countermeasures, which jointly optimize the encoder and decoder via a noise layer, face 2 inevitable challenges: (1) decrease of clean accuracy due to decoder adversarial training and (2) limited robustness due to simultaneous training of all three advanced attacks. To overcome these issues, we propose AdvMark, a novel two-stage fine-tuning framework that decouples the defense strategies. In stage 1, we address adversarial vulnerability via a tailored adversarial training paradigm that primarily fine-tunes the encoder while only conditionally updating the decoder. This approach learns to move the image into a non-attackable region, rather than modifying the decision boundary, thus preserving clean accuracy. In stage 2, we tackle distortion and regeneration attacks via direct image optimization. To preserve the adversarial robustness gained in stage 1, we formulate a principled, constrained image loss with theoretical guarantees, which balances the deviation from cover and previous encoded images. We also propose a quality-aware early-stop to further guarantee the lower bound of visual quality. Extensive experiments demonstrate AdvMark outperforms with the highest image quality and comprehensive robustness, i.e. up to 29%, 33% and 46% accuracy improvement for distortion, regeneration and adversarial attacks, respectively.

1. Introduction

The advances of powerful generative models, such as Stable Diffusion [18] and Sora [16], necessitate robust mechanisms for tracing and authenticating AI-generated content (AIGC). Deep learning-based watermarking has emerged as a promising solution, embedding a secret message within an image that a corresponding decoder can later extract. To mitigate transmission distortions like JPEG compression, previous works [3] leverage joint adversarial optimization (JAT) on both encoder and decoder via a noise layer to simulate various attacks, as depicted in case 2 in Fig. 1. While effective against common attacks, recent diffusion-based image regeneration [29] and more advanced adversarial attacks like WEvade [9] have manifested much stronger evasion performance. We identify that naive joint training paradigm is inherently vulnerable due to 2 critical issues:

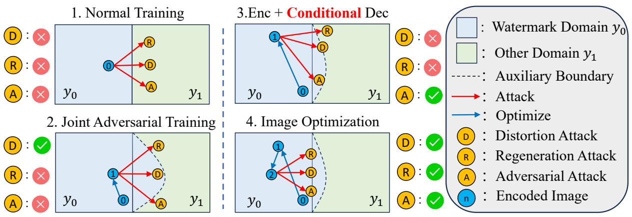


Figure 1. Four training paradigms. Adversarial training (2) degrades clean accuracy of y_1 and exhibits limited improvement, while moving image (3+4) suffices in both terms.

Challenge 1: *Decrease of clean accuracy.* It is well understood that decoder adversarial training gains robustness with inevitably lower clean accuracy [17, 25] on unattacked images due to tampered boundary distribution. As a result, JAT optimization also sacrifices accuracy for marginal robustness on each attack. We present an empirical experiment in Fig. 2 (refer to Section 4.1 for setup). Without robustness training, MBRS and DADW both exhibit low accuracy against Regen and WEvade. Applying JAT to MBRS successfully improves robustness but the clean accuracy instead decreases to 0.94 due to significant landscape modification, as shown in case 2 in Fig. 1. To guarantee both accuracy, our insight is to harness the power of encoder fine-tuning (EAT) to move the images towards the center (case 3 in Fig. 1), instead of expanding auxiliary boundary. As shown in Fig. 2, MBRS-EAT maintains high clean accuracy while achieving competitive robustness compared to MBRS-JAT.

Challenge 2: *Simultaneous training yields limited robustness.* While encoder-focused adversarial training (EAT) solves the clean accuracy problem, it is not a panacea. The

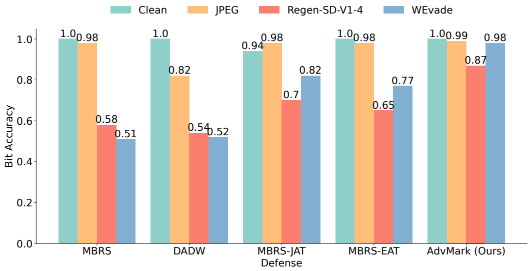


Figure 2. The bit accuracy ↑ of against three representative distortion, regeneration and adversarial attacks. JAT and EAT denote joint and encoder-based adversarial training.

accuracy against Regen (0.65) and WEvade (0.77) still exhibits gaps from JPEG. This is because these attacks have more intricate mechanisms, e.g. numerous sampling iterations of the diffusion model and derivatives from the decoder. Forcing a single model to simultaneously defend against all attacks within a monolithic training process leads to an inefficient and slow-converging optimization. To offload the training, our insight is to separate the defense stage with an additional direct image optimization on distortion and regeneration attacks (case 4 in Fig. 1), while we address the adversarial attack only via EAT due to two reasons: (1) it requires first-order derivatives of the image, and optimization on attacked image incurs higher-order derivatives which incur prohibitive computational and memory overhead; (2) past work has observed that ReLU neural networks are locally almost linear [7], which suggests that second derivatives may be close to zero in most cases and makes optimization hard to converge [5]. By decoupling the defense, we can achieve the highest clean and robust accuracy, as shown in Fig. 2.

Built upon the insights, we propose AdvMark, a framework that abandons the joint training paradigm in favor of a decoupling, two-stage fine-tuning process. In stage 1, we propose a novel adversarial training paradigm that focuses on encoder fine-tuning. This is guided by an improved defender-side adversarial attack construction. We optimize the message to deviate from the ground-truth rather than towards a random label. The decoder is only conditionally updated when robustness is below the threshold, ensuring that clean accuracy is not compromised. While in stage 2, we address other attacks via efficient and effective direct image optimization. To maintain the robustness obtained in stage 1, we propose a novel constrained image loss to not only enhance visual quality, but also limit the deviation from previously encoded image. The performance is guaranteed by a theorem based on practical and verifiable assumption. To further guarantee quality, we propose to improve upon the known PGD [14] optimization with a quality-aware early-stop, rather than the implicit ϵ-ball projection.

We conduct extensive experiments with 9 watermarking methods against 10 attacks to demonstrate that, AdvMark outperforms with up to 46% accuracy improvement and the highest image quality in terms of PSNR, SSIM and LPIPS.

We summarize our contributions as follows:

- To our best knowledge, we are the first to evaluate existing watermarking against distortion, regeneration, and adversarial attacks systematically. We demonstrate that typical joint optimization exhibits 2 challenges: (1) low clean accuracy due to decoder training and (2) limited robustness due to simultaneous training of non-trivial attacks. Then we derive two key insights through empirical evaluation.
- To bridge the gap, we propose AdvMark, a two-stage comprehensively robust image watermarking method. In stage 1, we propose an improved defender tailored adversarial attack, accompanied by a novel adversarial training paradigm that mainly fine-tunes the encoder. The decoder is only conditionally trained to guarantee clean accuracy. In stage 2, we leverage image optimization to address the distortion and regeneration attacks. To preserve adversarial robustness, we design a constrained image loss with theoretical guarantees. We also propose a quality-aware PGD to improve visual quality.
- Extensive experiments indicate the comprehensive robustness of AdvMark with the highest image quality. The ablation study further validates the efficiency of AdvMark.

2. Related Work

2.1. Image Watermarking

Image watermarking has long been studied in the context of intellectual property protection. Existing watermarking methods can be categorized into three types: (1) Traditional watermarking methods that leverage heuristic hand-crafted embedding like DwtDctSvd [2] in frequency domain. (2) Post-processing deep learning based alternatives [3, 12, 13] such as HiDDeN [30] and MBRS [8]. They leverage a noise layer between encoder and decoder to simulate various attack such as JPEG to enhance robustness. More recent VINE [11] leverages pretrained text-to-image diffusion model to defend against image editing attacks [27, 28]. (3) In-processing methods [1, 24] that directly embed watermarks as part of the image generation process. For example, Stable Signature [4] fine-tunes the decoder of the LDM to incorporate the message in latent features, leaving the diffusion component unchanged.

2.2. Threat Model And Attacks

The defender produces a watermarked image $x_w = E(x_o, m)$ from clean image x_o and message m via proprietary encoder E. The decoder D is subjected to API or model exposure. **The attacker** aims to construct perturbed image x_a such that the decoder fails to recover the correct message (i.e., $D(x_a) \neq m$), while ensuring x_a remains vi-

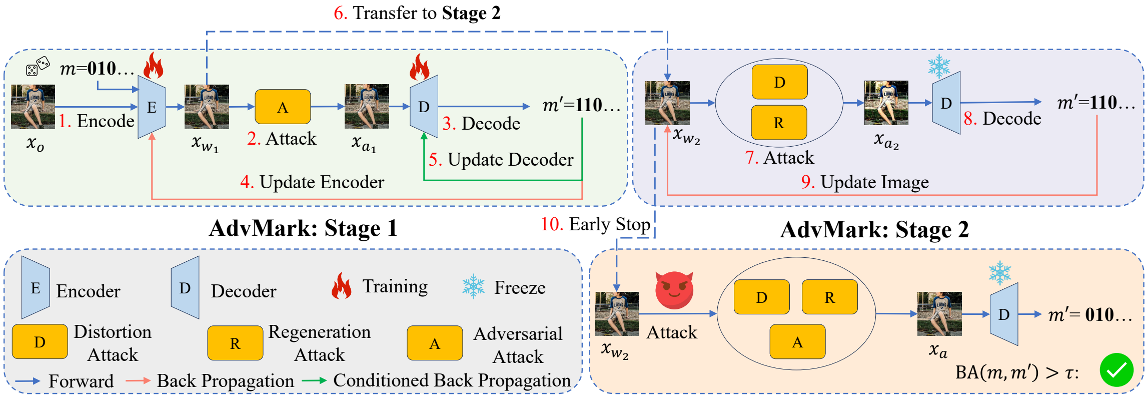


Figure 3. Overview of AdvMark. In stage 1 we mainly fine-tune the encoder with slight decoder training to tackle adversarial attack. In stage 2 we directly optimize the encoded image to address the rest two attacks while preserving adversarial robustness.

sually similar to x_w. The encoder cannot be accessed but the decoder's knowledge varies: (1) white-box with full access (e.g. the model leakage from third party [15]) or (2) black-box with only query capability. We consider three types of attacks:

Distortion Attack. We evaluate against heuristic transformations like JPEG compression, Gaussian noise, and various geometric attacks, which resemble real-world transmission loss.

Regeneration Attack. Given the remarkable generation performance from diffusion models [19, 29], we can destroy the watermark with semantic content reconstruction. Specifically, it first destructs a watermarked image by adding noise to its representation (the forward process) and then removes the watermark pattern via efficient denoising process like DDIM [20].

Adversarial Attack. For white-box with full access, we can construct malicious images to evade decoder D with minimal, often imperceptible modifications like WEvade [9]. It crafts an adversarial example such that the decoded message from D approaches a random target message m_t within a perturbation budget r, formulated as:

$$\min_{\delta} \ l(D(x_w + \delta), m_t), \ s.t. \|\delta\|_\infty < r \tag{1}$$

where l denotes the L_2 loss and x_w is the watermarked image. In the **black-box** setting, a query-based attack [9] perturbs and adjusts the image based on feedback from the decoder's output. Alternatively, Saberi et al. [19] proposes a transfer-based attack by training a surrogate decoder on watermarked images. Adversarial examples are then crafted and transferred to evade the target decoder.

3. Proposed Method: AdvMark

3.1. Overview

Built upon previous insights, we propose the two-stage framework overview of AdvMark in Fig. 3. In stage 1, we propose a novel encoder-based fine-tuning strategy to tackle adversarial attack. We encode the original image x_o and derive the attacked x_{a_1} via our defender tailored adversarial attack. The decoded message m_{a_1} incurs L_1 loss. After several rounds of fine-tuning on encoder, we conditionally update the decoder if the bit accuracy of m_{a_1} is below a prefixed threshold. While in stage 2, we directly optimize the encoded image x_{w_2} to preserve visual quality and previous adversarial robustness. Finally, we leverage a quality-aware metric to early stop the optimization and obtain the encoded image, which is capable of defending against all three kinds of attacks without degrading quality or clean accuracy.

3.2. Stage 1: Adversarial Encoder Fine-tuning

Adversarial Examples. We first fine-tune the pretrained encoder to handle adversarial attack. We denote original image $x_o \in \mathbb{R}^{3\times H\times W}$, H, W are height and width of an image, a secret message of length n is $m \in \{0,1\}^n$, the encoded image from encoder E is $x_{w_1} = E(x_o, m) \in \mathbb{R}^{3\times H\times W}$, while the extracted message logits after decoder D is $y_{w_1} = D(x_{w_1}) \in \mathbb{R}^n$, and $m_{w_1} = clamp(round(y_{w_1}), 0, 1)$. The final bit accuracy $BA(m, m_{w_1})$ is the fraction of matched bits. To construct adversarial examples $x_{a_1} = x_{w_1} + \delta$, we improve optimization in Equ. 1 and propose a defender tailored loss. Since we have the ground-truth secret message m, we can directly render m_{a_1} random to m (i.e. $BA(m_{a_1}, m) \to 0.5$):

$$\min_{\delta} \ |0.5 - l(clamp(D(x_{w_1} + \delta), 0, 1), m)|, \ s.t. \|\delta\|_\infty < r \tag{2}$$

where l denotes the mean-squared-error (MSE) loss throughout the paper, and r is the perturbation budget to maintain image quality. $clamp(D(x_{w_1} + \delta), 0, 1)$ represent the bit possibility of m_{a_1} aligned with m. Note that we do not optimize m_{a_1} due to zero gradient from $round(*)$ operation. The final δ^* is derived via PGD [14] optimization.

Optimization Loss. With adversarial example x_{a_1}, we tend to minimize the adversarial loss such that m_{a_1} approaches

Algorithm 1: Stage 1 of AdvMark

```
Input: pretrained Encoder $\theta_E$ and Decoder $\theta_D$,
       image datasets $I$, other parameters $iter_E$,
       $\alpha_E$, $\alpha_D$, $r$, $\lambda_{w_1}$, $\lambda_{i_1}$ and $\tau_1$
Output: fine-tuned Encoder $\theta_E^*$ and Decoder $\theta_D^*$
for batched images $x_o$ in $I$ do
    sample random message $m \in \{0,1\}^n$;
    for $i \leftarrow 1$ to $iter_E$ do
        $x_{w_1} \leftarrow E(x_o, m)$; $x_{a_1} \leftarrow x_{w_1} + \delta^*(r)$;
        $L_1 \leftarrow L_{a_1} + \lambda_{w_1} L_{w_1} + \lambda_{i_1} L_{i_1}$ in Equ. 6;
        $\theta_E \leftarrow \theta_E - \alpha_E \cdot \frac{\partial L_1}{\partial \theta_E}$;
        if $i == iter_E$ then
            $m_{a_1} \leftarrow clamp(round(D(x_{a_1})), 0, 1)$;
            if $BA(m_{a_1}, m) < \tau_1$ then
                $\theta_D \leftarrow \theta_D - \alpha_D \cdot \frac{\partial L_1}{\partial \theta_D}$;
        end
    end
end
return $\theta_E, \theta_D$
```

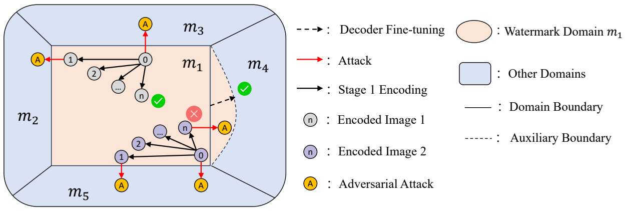


Figure 4. Illustration of stage 1. We constantly fine-tune the encoder to map the image (No. 1) towards the non-attackable center (No. n). The decoder is updated only when the final image fails to suffice (No. n of image 2).

m under budget r, formulated as L_{a_1}:

$$L_{a_1} = l(D(x_{a_1}), m),\ x_{a_1} = x_{w_1} + \delta^* \tag{3}$$

To maintain clean accuracy we derive L_{w_1} formulated as:

$$L_{w_1} = l(D(x_{w_1}), m) \tag{4}$$

To enhance image visual quality, we leverage the MSE distance and LPIPS [26] loss to guarantee both pixel and semantic similarity, formulated as:

$$L_{i_1} = (l(x_{w_1}, x_o) + LPIPS(x_{w_1}, x_o))/2 \tag{5}$$

Together we have the total loss as:

$$L_1 = L_{a_1} + \lambda_{w_1} L_{w_1} + \lambda_{i_1} L_{i_1} \tag{6}$$

where λ_{w_1} and λ_{i_1} denote the weights for clean accuracy and image quality, respectively.

Fine-tuning. We propose to mainly fine-tune the encoder for $iter_E$ rounds. If the final robustness accuracy $BA(m_{a_1}, m)$ is below a fixed threshold τ_1, the decoder is updated once with L_1 loss. We present the algorithm of Stage 1 of AdvMark in Algorithm 1.

Example. We illustrate stage 1 in Fig. 4. The encoder is fine-tuned to mitigate adversarial attack by moving the image into a safe zone. If the encoded image n remains vulnerable anyway, we slightly update the decoder once to expand auxiliary boundary.

3.3. Stage 2: Quality-aware Image Optimization

Optimization Loss. Following **Insight 2**, we propose to directly optimize the encoded image x_{w_1} to enhance robustness against distortion and regeneration attacks. We first derive the attack loss as follows:

$$L_{a_2} = \frac{\sum_{k=2}^{K+2} \lambda_{a_k} \cdot l(D(x_{a_k}), m)}{\sum_{k=2}^{K+2} \lambda_{a_k}} \tag{7}$$

where $x_{a_k} = x_{w_2} + \delta x_{a_k},\ k \in [2, K+2]$, K denotes the distortion number, $x_{a_{K+2}}$ denotes the regeneration attack and λ_{a_k} denotes optimization weights for different attacks. Note that for each attack we apply the differentiable implementation to ensure optimization of x_{a_k}. Similar to Equ. 4, we also maintain the clean accuracy of x_{w_2} as follows:

$$L_{w_2} = l(D(x_{w_2}), m) \tag{8}$$

To preserve the adversarial robustness in stage 1, we propose a novel constrained image loss that additionally includes the deviation distance between (x_{w_1}, x_{w_2}), which is formulated as:

$$L_{i_2} = \frac{l(x_{w_2}, x_o) + LPIPS(x_{w_2}, x_o) + 2 \cdot l(x_{w_2}, x_{w_1})}{4} \tag{9}$$

Together the total loss for stage 2 is in Equ. 10, where λ_{w_2} and λ_{i_2} represent weights for clean accuracy and image quality, respectively.

$$L_2 = L_{a_2} + \lambda_{w_2} L_{w_2} + \lambda_{i_2} L_{i_2} \tag{10}$$

To demonstrate the effectiveness of $l(x_{w_2}, x_{w_1})$ on robustness preservation, we propose theorem 1 based on an assumption.

Assumption 1 *Assume the constrained image loss Equ. 9 is minimized such that* $\|x_{w_2} - x_{w_1}\| \leq \delta\ < \alpha$.

Theorem 1 *Given a robust* x_{w_1}, *i.e.* $f(x_{w_1}) = f(x_{w_1} + \eta_1), \forall\ \|\eta_1\| \leq \alpha$, *where* f *maps images into messages. Let assumption hold, then* x_{w_2} *is also robust with an adjusted budget, i.e.* $f(x_{w_2}) = f(x_{w_2} + \eta_2), \forall\ \|\eta_2\| \leq \alpha - \delta$.

Empirical Results. We verify the assumption with L_2: $\alpha = 0.012$, $\delta = 0.007$ and the actual η_2 bound is $0.010 > \alpha - \delta$. The technical proof is trivial because based on assumption

```
Algorithm 2: Stage 2 of AdvMark
Input: original image $x_o$, fine-tuned encoder $E$,
       message $m$, other parameters $iter_o$, $\alpha_x$, $p$,
       $\lambda_{w_2}$, $\lambda_{i_2}$, $\lambda_{a_k}$, $k \in [2, K+2]$, $\tau_2$
Output: full-scale robust images $x_{w_2}$
$x_{w_2} \leftarrow E(x_o, m)$;
for $i \leftarrow 1$ to $iter_o$ do
    for $k$ in $[2 \sim K+2]$ do
        $x_{a_k} \leftarrow x_{w_2} + \delta x_{a_k}$;
        $m_{a_k} \leftarrow clamp(round(D(x_{a_k})), 0, 1)$;
        if $BA(m_{a_k}, m) \geq \tau_2$ then del $x_{a_k}$;
    end
    $L_2 \leftarrow L_{a_2} + \lambda_{w_2} L_{w_2} + \lambda_{i_2} L_{i_2}$ in Equ. 10;
    $x_{w_2} \leftarrow I(x_{w_2} - \alpha_x \cdot sign(\frac{\partial L_2}{\partial x_{w_2}}))$;
    if $PSNR(x_{w_2}, x_o) \leq p$ then break;
end
return $x_{w_2}$
```

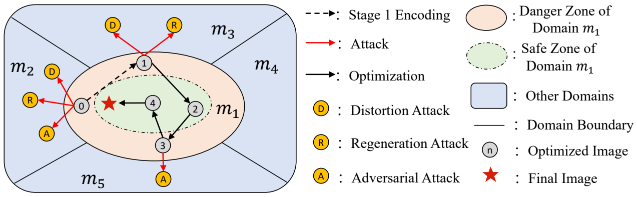


Figure 5. Illustration of stage 2. 0 → 1: initialize image 1 from encoder, which exhibits only adversarial robustness; 2: comprehensive robustness but low visual quality; 3: high quality yet vulnerable to A attack again; 4: similar to image 2; $\star$: high quality and robustness.

1, we have $f(x_{w_1}) = f(x_{w_2})$, while we also have $\|x_{w_2} + \eta_2 - x_{w_1}\| \leq \|x_{w_2} - x_{w_1}\| + \|\eta_2\| \leq \delta + \alpha - \delta = \alpha$, hence $f(x_{w_2} + \eta_2) = f(x_{w_1}) = f(x_{w_2})$.

Direct Optimization. To minimize L_2 loss, we improve upon the PGD optimization with quality-aware mapping as follows:

$$x_{w_2} = I(x'_{w_2}, x_{w_2}), x'_{w_2} = x_{w_2} - \alpha_x \cdot sign(\frac{\partial L_2}{\partial x_{w_2}}) \quad (11)$$

$$I(x'_{w_2}, x_{w_2}) = \begin{cases} x'_{w_2} & \text{if } PSNR(x'_{w_2}, x_o) \geq p, \\ x_{w_2} & \text{if } PSNR(x'_{w_2}, x_o) < p. \end{cases} \quad (12)$$

where α_x is the learning rate. We replace the typical ϵ-ball projection $\prod_{x_o,\epsilon}$ with $I(*,*)$ mapping. In this way, we directly constrain the lower bound of visual quality via budget p. During the optimization, we disregard attacks in L_{a_2} whose bit accuracy of x_{a_k} is above the threshold τ_2 to guarantee comprehensive robustness. Finally we early stop the optimization if PSNR is below budget p. The summarized steps for stage 2 are in Algorithm 2.

Table 1. Visual quality of watermarked images, n is the secret message length. Blue and Red denote the best and worst.

Defense	Image Size	n	MS-COCO			DiffusionDB		
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DwtDctSvd	256 × 256	30	35.8	0.98	0.03	36.8	0.98	0.05
HiDDeN	128 × 128	30	30.3	0.94	0.14	30.8	0.94	0.16
MBRS	128 × 128	30	32.1	0.95	0.09	31.8	0.94	0.14
Stable Signature	128 × 128	48	30.5	0.94	0.12	30.8	0.94	0.14
PIMoG	128 × 128	30	35.8	0.95	0.08	35.6	0.95	0.10
DADW	128 × 128	30	33.4	0.98	0.05	33.6	0.98	0.04
StegaStamp	256 × 256	64	29.8	0.93	0.15	31.7	0.94	0.14
EditGuard	256 × 256	64	36.3	0.95	0.04	36.1	0.94	0.04
VINE	256 × 256	100	35.5	0.98	0.01	35.2	0.99	0.01
AdvMark (Ours)	128 × 128	30	37.0	0.99	0.01	36.9	0.99	0.01
	128 × 128	48	37.2	0.99	0.01	37.0	0.99	0.01
	256 × 256	64	38.4	0.99	0.01	38.3	0.99	0.01
	256 × 256	100	38.9	0.99	0.01	38.8	0.99	0.01

Example. We illustrate stage 2 in Fig. 5. The initial image 1 gains only adversarial robustness from vulnerable image 0. During 1 → 2 we optimize L_{a_2} to enhance overall robustness, which comes with low visual quality and clean accuracy. Hence we further optimize $L_{i_2} + L_{w_2}$ to obtain image 3, which again remains vulnerable to adversarial attack. During 3 → 4 we optimize $l(x_{w_2}, x_{w_1})$ to approach image 1 and gain previous robustness. Finally, we optimize L_{i_2} to further enhance image quality while maintaining comprehensive robustness.

4. Experiments

4.1. Experimental Setup

All our experiments are conducted on NVIDIA RTX 4090.

Datasets and Metrics. We evaluate AdvMark on MS-COCO [10] and DiffusionDB [23]. For image visual quality, we adopt the well-known PSNR, Structural Similarity score (SSIM) [22], and semantic metric Learned Perceptual Image Patch Similarity (LPIPS) [26]. For watermarking performance, we use the bit accuracy $BA(m_1, m_2)$, i.e. the fraction of matched bits.

Parameter Setup. For distortion attacks, we follow previous work in [8, 9] and adopt their default setting. For regeneration attacks, we adopt 2 Stable Diffusion checkpoints for image reconstruction. Adversarial attacks include WEvade and the black-box query-based (Black-Q) from [9] with target threshold $\tau' = 0.75$. We also follow [19] for surrogate-based attack (Black-S).

For all watermarking methods, we set both thresholds $\tau_1 = \tau_2 = 0.95$ throughout the paper. For AdvMark, we fine-tune the pretrained MBRS models [8] with the following parameters: In stage 1, encoder fine-tuning iteration $iter_E = 10$, learning rate $\alpha_E = \alpha_D = 5e^{-4}$, adversarial attack budget $r = \frac{20}{255}$, $\lambda_{w_1} = 1$, $\lambda_{i_1} = 3$. While in stage 2, we consider $K = 4$ distortion attacks and set JPEG weight $\lambda_{a_2} = 1$, the rest $\lambda_{a_k} = 0.1, k \in [3, 6]$, optimization iteration $iter_o = 10$, learning rate $\alpha_x = 5e^{-2}$, loss weight

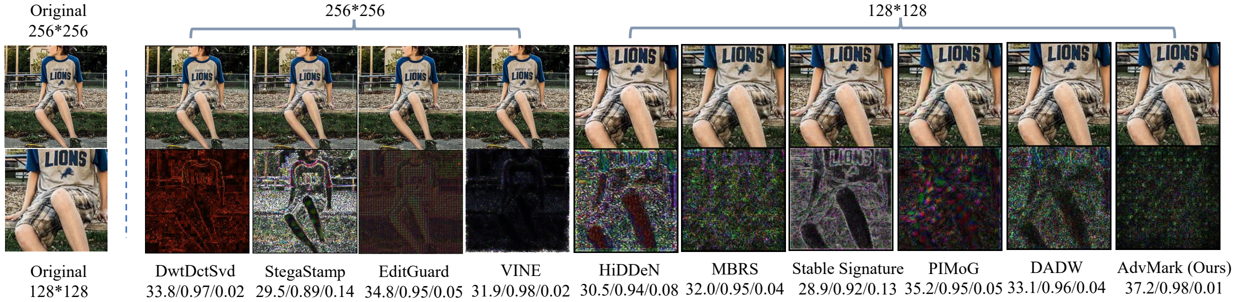


Figure 6. The original x_o, watermarked x_w and residual images x_r of different watermarking methods where $x_r = |x_w - x_o| \times 10$. We present below each watermarking method as PSNR↑/SSIM↑/LPIPS↓.

Table 2. Robust accuracy↑. $1, 2, 3, 4, 1 \sim 4$ represent JPEG, Gaussian noise, Gaussian blur, brightness and 4 attacks combined respectively. * denotes unknown attacks not included in training. **Geometric and advanced combined attacks are in Table 5.**

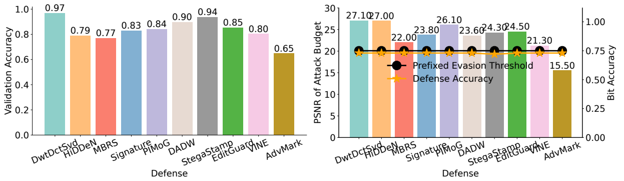
Datasets	Attack / Defense	Clean	Distortion					Regeneration		Adversarial		
			1	2	3	4	(1∼4)*	V1-4	V1-5*	WEvade	B-S*	B-Q*
COCO	DwtDctSvd	0.99	0.88	0.96	0.99	0.60	0.54	0.62	0.63	/	0.63	0.73
	HiDDeN	0.88	0.70	0.75	0.87	0.85	0.64	0.57	0.56	0.57	0.68	0.73
	MBRS	0.93	0.98	1.00	1.00	1.00	0.76	0.70	0.70	0.82	1.00	0.73
	Stable Signature	0.90	0.80	0.88	0.91	0.94	0.70	0.65	0.65	0.78	0.87	0.73
	PIMoG	0.93	0.85	0.91	1.00	0.98	0.68	0.66	0.66	0.77	0.80	0.73
	DADW	1.00	0.82	0.88	1.00	0.98	0.58	0.54	0.54	0.52	0.55	0.73
	StegaStamp	0.92	0.94	0.95	1.00	0.88	0.79	0.71	0.71	0.79	0.56	0.72
	EditGuard	1.00	0.82	0.75	0.94	0.95	0.61	0.59	0.59	0.48	0.75	0.73
	VINE	1.00	0.96	0.94	0.93	0.96	0.78	0.74	0.75	0.51	0.80	0.73
	AdvMark (Ours)	1.00	0.99	1.00	1.00	1.00	0.83	0.87	0.87	0.98	1.00	0.73
DB	DwtDctSvd	1.00	0.90	0.95	0.99	0.58	0.53	0.58	0.59	/	0.62	0.72
	HiDDeN	0.86	0.70	0.72	0.85	0.84	0.64	0.56	0.57	0.55	0.65	0.73
	MBRS	0.91	0.98	1.00	1.00	1.00	0.72	0.69	0.69	0.80	1.00	0.73
	Stable Signature	0.88	0.72	0.83	0.86	0.90	0.66	0.64	0.64	0.76	0.82	0.71
	PIMoG	0.92	0.84	0.90	1.00	0.97	0.66	0.62	0.62	0.73	0.82	0.72
	DADW	1.00	0.81	0.88	1.00	0.96	0.57	0.53	0.53	0.51	0.57	0.72
	StegaStamp	0.91	0.95	0.96	0.91	0.76	0.79	0.72	0.72	0.78	0.59	0.72
	EditGuard	1.00	0.80	0.74	0.94	0.93	0.60	0.55	0.56	0.51	0.70	0.72
	VINE	1.00	0.95	0.91	0.91	0.94	0.75	0.76	0.76	0.53	0.78	0.73
	AdvMark (Ours)	1.00	0.98	1.00	1.00	1.00	0.83	0.85	0.85	0.96	1.00	0.74

$\lambda_{w_2} = 0.1$ and $\lambda_{i_2} = 5$, PSNR budget $p = 36$. Due to inconsistent image size and message length among baselines, we also train MBRS with different setup.

Regarding baselines, we include 7 methods: traditional scheme DwtDctSvd [2], learning-based HiDDeN [30], MBRS [8], PIMoG [3], StegaStamp [21], DADW [12], EditGuard [27] (degrade version), VINE [11] (robust version) and in-processing method Stable Signature [4]. To ensure a fair comparison, we fine-tune all the baselines via typical joint optimization with all attacks, except for EditGuard and VINE due to their specific design in tampering detection and editing defense.

4.2. Image Visual Quality

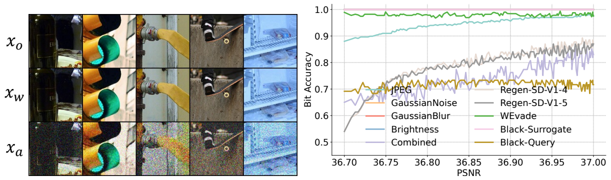
As shown in Table 1, the baselines exhibit limited quality due to complex joint training for robustness against all the attacks. Our method surpasses the original MBRS with significant improvement (32.1 to 37.0), thanks to our two-stage design and optimization in Equ. 5 and 9. To better perceive the visual quality, we present an example to compare the residual images in Fig. 6. Most methods embed watermark inside the background to mitigate noticeable artifacts, while AdvMark outperforms by reducing overall noises to improve visual similarity.



(a) Validation accuracy↓.

(b) PSNR↓ of Black-Q attacks.

Figure 7. Attack performance of Black-S and Black-Q.



(a) Image of Black-Q attacks.

(b) Accuracy↑ in stage 2.

Figure 8. Example of Black-Q attack and accuracy optimization.

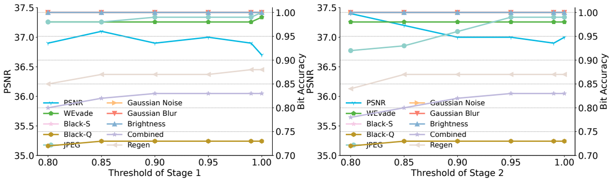
4.3. Watermark Robustness

Table 2 depicts the comprehensive robustness across all attack where ours maintains the highest clean accuracy. For distortion and regeneration attacks: AdvMark outperforms the SOTA methods by up to 29% against JPEG and 33% against V1-4 attacks. This stems from our efficient attack optimization in Equ. 7. In comparison, MBRS achieves high JPEG robustness but yields a low accuracy of 0.76 against combined distortions compared to 0.83.

For adversarial attacks, AdvMark outperforms baselines with 16%-46% accuracy improvement. Note that DwtDctSvd is composed of non-differentiable transformations and henceforth is not applicable. MBRS ranks the second with only 0.82 accuracy due to poor joint training. Thanks to our fine-tuning in stage 1 where encoded images are moved into non-attackable regions, it mitigates Black-S attack by confusing surrogate models with fine-tuned images distribution. As shown in Fig. 7 (a), we can find that lower

Table 3. Ablation of AdvMark. AdvMark w/ $\lambda_{i_2} = 7$ achieves better robustness but at the cost of more optimization overheads. AdvMark w/ different sizes and n still outperforms baselines.

Attack / Defense	PSNR	Distortion					Regeneration		Adversarial		
		1	2	3	4	1∼4	V1-4	V1-5	WEvade	B-S	B-Q
AdvMark (Ours)	37.0	0.99	1.00	1.00	1.00	0.83	0.87	0.87	0.98	1.00	0.73
w/o Stage 1	34.7	1.00	1.00	1.00	1.00	0.85	0.88	0.88	0.50	1.00	0.73
w/o Stage 2	36.7	0.88	1.00	1.00	1.00	0.65	0.54	0.54	0.99	1.00	0.69
$iter_E$=5	36.5	0.99	1.00	1.00	1.00	0.82	0.87	0.87	0.91	1.00	0.73
$iter_E$=20	36.1	0.90	1.00	1.00	1.00	0.75	0.85	0.85	1.00	1.00	0.73
$\lambda_{i_2} = 3$	36.4	1.00	1.00	1.00	1.00	0.83	0.88	0.88	0.98	1.00	0.73
$\lambda_{i_2} = 7$	37.2	1.00	1.00	1.00	1.00	0.85	0.89	0.89	0.99	1.00	0.73
size=128,n=48	37.2	1.00	1.00	1.00	1.00	0.84	0.87	0.88	0.98	1.00	0.73
size=256,n=64	38.4	1.00	1.00	1.00	1.00	0.89	0.92	0.92	1.00	1.00	0.73
size=256,n=100	38.9	1.00	1.00	1.00	1.00	0.89	0.93	0.93	1.00	1.00	0.73



(a) Stage 1

(b) Stage 2

Figure 9. PSNR ↑ and accuracy ↑ for different thresholds.

validation accuracy generally aligns with higher bit accuracy (robustness) e.g., accuracy of 0.65 and 0.77 from AdvMark and MBRS correspond to both 1.00 in Table 2.

For Black-Q attack, it initializes an image with random noise, and then iteratively optimizes the sample towards the target image but always guarantees successful evasion. Therefore the bit accuracy is always below threshold $\tau' = 0.75$ and only the perturbation budget differs. We present the PSNR of the attacked image in Fig. 7 (b). As expected, AdvMark extracts a significant quality loss (∼15 PSNR) from Black-Q attack, thanks to our image moving strategy. We also present some examples of the Black-Q attacked images from AdvMark in Fig. 8 (a).

4.4. Ablation Study

Effectiveness of two stages As shown in Table 3. We first skip stage 1 and replace the fine-tuned model with original MBRS. Separately we also skip stage 2 to evaluate the robustness against distortion and regeneration attacks. As expected, without stage 1, image quality degrades and AdvMark fails to defend against WEvade without encoder fine-tuning. Without stage 2, AdvMark only enhances adversarial robustness and deteriorates distortion robustness, i.e. only 0.88 and 0.65 accuracy against JPEG and combined attacks, which implies the necessity of image optimization to restore robustness balance.

To better understand how stage 2 preserves previous robustness, we present the bit accuracy change during optimization in Fig. 8 (b). We can find that stage 1 of AdvMark generates high quality images (PSNR=36.7) but with poor robustness. In stage two's optimization, AdvMark constantly enhances overall accuracy with slight quality improvement, while also maintaining adversarial robustness, thanks to our constrained image loss $l(x_{w_2}, x_{w_1})$ in Equ. 9.

Table 4. Ablation study of baselines. We present watermarking PSNR ↑ and robustness accuracy ↑. Baselines with stage 2 are enhanced, demonstrating our optimization effectiveness.

Attack / Defense	PSNR	Distortion					Regeneration		Adversarial		
		1	2	3	4	1∼4	V1-4	V1-5	WEvade	B-S	B-Q
HiDDeN w/ Stage 2	30.0	0.88	0.98	1.00	1.00	0.77	0.73	0.73	0.54	0.77	0.73
Stable Signature w/ Stage 2	30.7	0.92	1.00	1.00	1.00	0.79	0.78	0.79	0.76	1.00	0.73
PIMoG w/ Stage 2	36.4	0.96	1.00	1.00	1.00	0.81	0.81	0.81	0.74	1.00	0.72
AdvMark+HiDDeN (Ours)	30.7	0.85	1.00	1.00	1.00	0.75	0.80	0.80	0.91	1.00	0.73
HiDDeN	30.3	0.70	0.75	0.87	0.85	0.64	0.57	0.56	0.57	0.68	0.73
MBRS, S=0.6	38.0	0.87	0.98	0.99	0.98	0.68	0.60	0.61	0.72	1.00	0.72
MBRS, S=1.0	32.1	0.98	1.00	1.00	1.00	0.76	0.70	0.70	0.82	1.00	0.73
MBRS, S=2.0	28.1	1.00	1.00	1.00	1.00	0.83	0.78	0.78	0.89	1.00	0.73

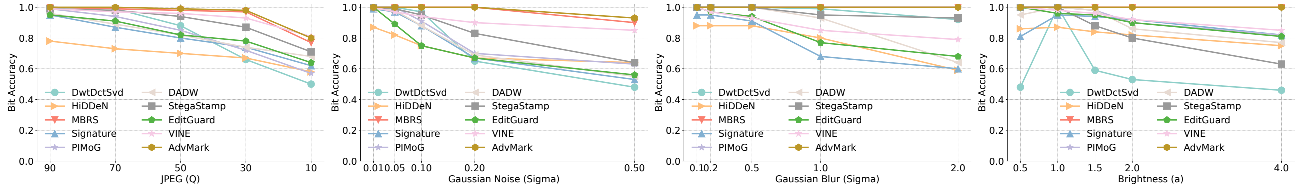
Parameter impact We retrain AdvMark with different encoder fine-tuning iterations $iter_E$ in stage 1 and image loss weight λ_{i_2} in stage 2. In Table. 3, regarding $iter_E$, we find that fewer fine-tuning iterations yield inferior adversarial robustness (0.91 WEvade accuracy) and lower visual quality since the encoder is not fully optimized. However, high iterations tend to overfit adversarial attacks, which in turn degrades image quality (PSNR budget p=36) to gain JPEG and regeneration attack robustness. Regarding λ_{i_2}, generally high image weight guarantees the optimization is fully explored until PSNR quality budget, therefore AdvMark achieves higher image quality with similar robustness at the cost of more computation overhead in stage 2.

To demonstrate the generalization of AdvMark, we also retrain with different image size and message length n in Table 3, which aligns with other baselines in Table 1. We can find that regardless of the parameters, AdvMark still outperforms with the highest quality and robustness, e.g. AdvMark (size=256, n=64, PSNR=38.4) surpasses StegaStamp with the same setting.

To demonstrate the impact of two thresholds τ_1 and τ_2, we present the PSNR and robustness changes in Fig. 9. Generally, when threshold 1 is higher than 0.98, it degrades WEvade robustness due to insufficient training from optional decoder. With lower threshold 2, there is a risk of insufficient optimization, e.g. lower JPEG and Regenerative robustness. In the end, we empirically choose both thresholds as 0.95 for best robustness.

Ablation on baselines To demonstrate the effectiveness of stage 2 on baselines, we apply image fine-tuning directly after the baseline models (HiDDeN, Stable Signature and PIMoG) and present the results in Table 4. It is solid that the robustness against distortion and regeneration attacks is enhanced, compared with the original performance in Table 2, thanks to our direct optimization.

To demonstrate the effectiveness of AdvMark on baselines, we apply the two-stage enhancement on another base

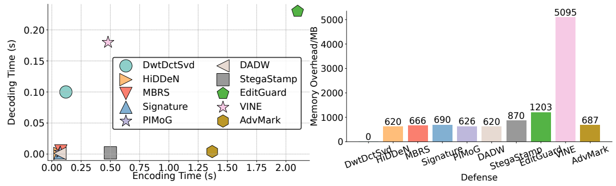


(a) JPEG (Train Q=50) (b) GaussianNoise (Train σ=0.1) (c) GaussianBlur (Train σ=0.5) (d) Brightness (Train a=1.5)

Figure 10. Bit accuracy ↑ of watermarking methods against distortion attacks with different testing parameters.

Table 5. Performance of geometric and advanced attacks. We present the robustness accuracy ↑. Although MBRS achieves similar results as AdvMark, it fails to defend against regeneration and adversarial attack, as shown in Table 2.

Attack / Defense	Geometry Attack						Advanced Combined	
	Crop	Resize	Dropout	SaltPepper	Rotation	Hue	Reg+Adv	Adv+Reg
HiDDeN	0.86	0.83	0.90	0.89	0.85	0.88	0.57	0.52
DADW	0.91	0.85	0.96	0.95	0.90	0.93	0.54	0.49
MBRS	0.97	1.00	1.00	1.00	0.94	1.00	0.70	0.65
AdvMark	1.00	1.00	1.00	1.00	1.00	1.00	0.87	0.81



(a) Time overhead ↓. (b) GPU Memory overhead ↓.

Figure 11. Overhead comparison.

model HiDDeN in Table. 4. It is as expected that AdvMark+HiDDeN significantly improves comprehensive robustness against all 10 attacks with only 0.4dB PSNR improvement. This is because HiDDeN itself is extremely vulnerable to even simple distortion attacks, therefore AdvMark exchanges more quality for higher robustness as much as up to 34% accuracy improvement.

To evaluate whether we can exchange robustness of image quality, we fine-tune MBRS with joint adversarial training under different strength factors S=0.6, 1.0, 2.0 in Table. 4. It depicts that MBRS with higher S yields lower visual quality with higher robustness. Nonetheless, AdvMark still outperforms the S=2.0 baseline with 9% accuracy improvement against regeneration and adversarial attacks, let alone our significantly high image quality. MBRS with S=0.6 shows higher quality yet exhibits poor robustness against all three kinds of attacks. As a result, solely relying on adjusted strength factor cannot suffice compared with AdvMark.

More attack settings To demonstrate adaptation to different attack strength, we present the bit accuracy against distortion attacks with different parameters in Fig. 10. The results indicate that AdvMark still outperforms in all the scenarios, e.g. ∼0.8 and ∼0.6 accuracy against JPEG with Q=10 for AdvMark and other baselines.

Finally, we include more geometric (following DWSF [6]) and combined attacks in Table 5. The parameters are Crop (p=0.035), Resize (r=0.7), Dropout(p=0.3), Salt-Pepper(p=0.1), Rotation(a=30°) and Hue(δ=0.2). In general, AdvMark still outperforms with all 1.0 accuracy with MBRS ranking the second, which validates the high robustness of the base model MBRS. For combined attacks, Reg+Adv is basically the same as Reg alone due to significant regeneration quality degradation that breaks the adversarial budget.

4.5. Overhead Analysis

We also evaluate AdvMark in terms of time and GPU memory overhead. For training overhead, AdvMark exhibits only O(1/2*N) and typical JAT requires O(N) to update both encoder and decoder, where N is the total iterations. For inference, the results are presented in Fig. 11. AdvMark achieves real time decoding and only incurs acceptable overhead from image optimization. EditGuard utilizes an inverse neural network to encode both secret images and bits simultaneously, which leads to longer processing time. While VINE requires high GPU memory due to heavy diffusion model. In summary, AdvMark can achieve SOTA performance within reasonable overhead.

5. Conclusion

In this paper, we first reveal the challenges of typical joint optimization against distortion, regeneration and adversarial attacks. Built upon the insights, we propose AdvMark which decouples a two-stage defense method. In stage one, we propose a defender tailored adversarial attack, along with a novel conditional training paradigm to mainly fine-tune the encoder to maintain clean accuracy. While in stage two, we directly optimize the encoded image to address other attacks. We propose a novel constrained image loss to preserve adversarial robustness with verifiable theoretical guarantees. To further guarantee visual quality, we improve upon the PGD optimization with a metric-aware early-stop. Extensive experiments demonstrate our effectiveness.

6. Acknowledgements

This work was supported by Key Laboratory of Data Intelligence, Beijing.

References

[1] Kasra Arabi, Benjamin Feuer, R Teal Witter, Chinmay Hegde, and Niv Cohen. Hidden in the noise: Two-stage robust watermarking for images. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[2] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan kaufmann, 2007. 2, 6

[3] Han Fang, Zhaoyang Jia, Zehua Ma, Ee-Chien Chang, and Weiming Zhang. Pimog: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2267–2275, 2022. 1, 2, 6

[4] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22466–22477, 2023. 2, 6

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 2

[6] Hengchang Guo, Qilong Zhang, Junwei Luo, Feng Guo, Wenbin Zhang, Xiaodong Su, and Minglei Li. Practical deep dispersed watermarking with synchronization and fusion. In *Proceedings of the 31st ACM international conference on multimedia*, pages 7922–7932, 2023. 8

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 2

[8] Zhaoyang Jia, Han Fang, and Weiming Zhang. Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In *Proceedings of the 29th ACM international conference on multimedia*, pages 41–49, 2021. 2, 5, 6

[9] Zhengyuan Jiang, Jinghuai Zhang, and Neil Zhenqiang Gong. Evading watermark based detection of ai-generated content. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1168–1181, 2023. 1, 3, 5

[10] 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*, pages 740–755. Springer, 2014. 5

[11] Shilin Lu, Zihan Zhou, Jiayou Lu, Yuanzhi Zhu, and Adams Wai-Kin Kong. Robust watermarking using generative priors against image editing: From benchmarking to advances. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 6

[12] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13548–13557, 2020. 2, 6

[13] Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie. Towards blind watermarking: Combining invertible and non-invertible mechanisms. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1532–1542, 2022. 2

[14] 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. 2, 3

[15] Meta. Meta's powerful ai language model has leaked online. `https://www.theverge.com/2023/3/8/23629362/meta-ai-language-model-llama-leak-online-misuse.`, 2023. 3

[16] OpenAI. Sora openai: Creating video from text. `https://openai.com/sora`, 2024. 1

[17] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. *ICLR*, 2021. 1

[18] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[19] 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. *ICLR*, 2024. 3, 5

[20] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[21] 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*, pages 2117–2126, 2020. 6

[22] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[23] 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. 5

[24] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[25] Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019. 1

[26] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of

deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4, 5

[27] Xuanyu Zhang, Runyi Li, Jiwen Yu, Youmin Xu, Weiqi Li, and Jian Zhang. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11964–11974, 2024. 2, 6

[28] Xuanyu Zhang, Zecheng Tang, Zhipei Xu, Runyi Li, Youmin Xu, Bin Chen, Feng Gao, and Jian Zhang. Omniguard: Hybrid manipulation localization via augmented versatile deep image watermarking. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3008–3018, 2025. 2

[29] Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, C Kruegel, G Vigna, YX Wang, and L Li. Invisible image watermarks are provably removable using generative ai. *arXiv*, 2023. 1, 3

[30] 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)*, pages 657–672, 2018. 2, 6