EDITR²: ROBUST IMAGE WATERMARKING BY LEARNING HOW GENERATIVE EDITING TRANSFORMS THE WATERMARK

Anonymous authors
Paper under double-blind review

ABSTRACT

Instruction-guided generative editing replaces objects or repaints scenes, often without visible traces—challenging invisible watermarking, whose value depends on recovering provenance and ownership after modification. Robust watermarking trains an embed–distort–extract pipeline against simulated attacks in a differentiable distortion layer. A generative editor, however, cannot take that place: its sampling pipeline resists stable gradient-based training. A natural remedy is to simulate the editor, yet what matters for recovery is not the edited image's appearance but how editing transforms the watermark perturbation. We therefore identify the learning target as the editor's response to the embedded perturbation and introduce DERS, a Differentiable Editing-Response Simulator. Trained on paired clean and watermarked edits, this differentiable surrogate predicts only this response, adds it to the clean edit—content generation inherited, not simulated—to take the editor's place in the distortion layer. The watermark, however, is embedded at 256×256, while the editor works at native resolution on its own working grid, where naive resizing breaks the watermark's message-bearing structure; we therefore introduce Content- and Geometry-aware Residual Transfer (CART) to adapt it to image content and the editor's preprocessing geometry. We present EditR², an end-to-end framework integrating the two in two stages: the watermark is first trained against common distortions, and DERS is then pretrained on the paired edits and frozen. The embedder, extractor, and CART are trained against this fixed simulator—faithful to the editor's observed response rather than drifting with the watermark it judges. On five Qwen-Image-Edit operations, EditR² reaches 90.76% mean bit accuracy, an absolute gain of 12.00 percentage points over the strongest evaluated baseline, while maintaining 47.33 dB PSNR. It remains stable across the tested native resolutions and transfers without retraining, reaching 87.76% and 84.40% mean bit accuracy on JoyAI-Image-Edit and Doubao-Seedream-4.5.

1 INTRODUCTION

Instruction-guided generative editing has drastically lowered the barrier to modifying visual content: a single text instruction can replace objects or repaint scenes (Wu et al., 2025; Qwen Team, 2025), and the results are seamless, leaving no visual trace. As editing becomes routine in content production, a photograph that looks intact no longer proves it has not been altered—its provenance and ownership can no longer be judged by eye. Invisible watermarks serve as a key technical anchor: they embed provenance and ownership information imperceptibly, and their value rests on whether the embedded message can still be recovered after modification. Yet generative editing attacks exactly this: rather than perturbing pixels like traditional distortions, it reconstructs pixels according to instruction, erasing the subtle changes that encode the message.

Existing robust watermarking follows the embed–distort–extract paradigm (Zhu et al., 2018; Jia et al., 2021): an embedder encodes a message, a distortion layer simulates anticipated distortions (JPEG, noise, blur), and an extractor recovers the message from the distorted output; recovery error backpropagates through the distortion layer to optimize the embedder. The paradigm's success depends on how well the distortion layer covers post-embedding operations. Instruction-guided

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

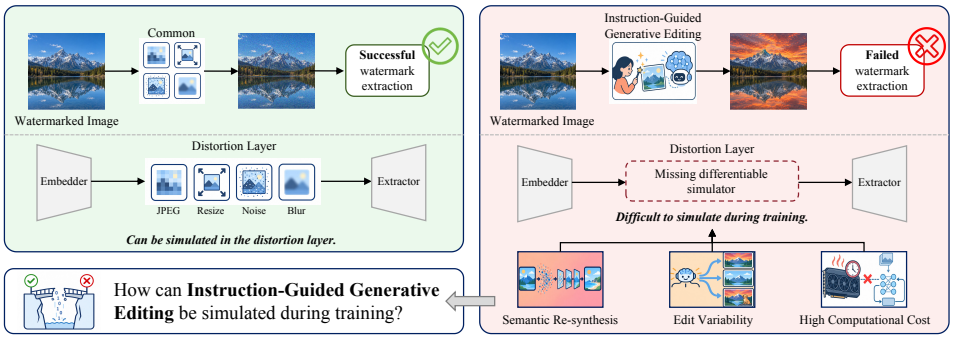

Figure 1: Common distortions can be included in watermark training directly or through differentiable approximations. Instruction-guided editing instead reconstructs content, varies with the instruction and sampling, and is costly to run at every optimization step.

generative editing undermines this premise on two levels (Figure 1). First, irreproducibility: editing output depends on the image, the instruction, and sampling stochasticity; even with a fixed prompt, different sampling paths yield different results, so no fixed distortion layer can fully cover the editor's effect. Second, inaccessibility: repeatedly invoking large editors is computationally prohibitive, and black-box interfaces provide no input gradients, making it impossible to integrate the editor directly into the training pipeline. Together, these two failures mean that the editor itself can neither be simulated by a fixed distortion layer nor be used directly as a differentiable distortion layer. A viable path—and the one we pursue—is a learnable surrogate that approximates the editor's effect while remaining differentiable to the embedder's input.

However, a critical design question arises: what should this surrogate learn? A naive approach would train it to reconstruct the editor's full visual output—a task that, at first glance, appears well within the reach of existing generative models. Yet this is fundamentally misaligned with the watermarking objective. The editor's output appearance and the message's survivability are decoupled: visual reconstruction fidelity $\neq$ perturbation response fidelity. Message recovery depends not on how faithfully the surrogate reproduces the edited image's pixels, but on how accurately it captures the transformation that the watermark perturbation undergoes during editing—a perturbation-level response that visual reconstruction alone cannot isolate. This insight dictates what the surrogate must learn: the editor's response to the watermark perturbation, not the editor's visual generation.

Based on this insight, we introduce the **Differentiable Editing-Response Simulator (DERS)**. Rather than reconstructing the editor's full visual output, DERS models how the watermark perturbation propagates through editing. It is supervised via paired editing: an unwatermarked source image and its watermarked counterpart are edited under the same instruction and the same random seed; the paired difference exposes the editor's response to the watermark perturbation. Once pretrained, DERS is frozen, serving as a differentiable distortion layer, through which message recovery error backpropagates to the embedder, turning post-edit robustness into an optimizable objective.

DERS captures the editing response, but a prerequisite remains: the perturbation must survive the resampling pipeline to the grid where the editor consumes it. The embedder operates on a fixed canonical grid, while each image lives at its own native resolution and the editor resamples it once more to its working grid; a residual—the watermark perturbation itself—merely resized between these grids—as in current fixed-resolution watermarking (Bui et al., 2025)—loses the fine spatial structure that carries the message : each resampling recomputes pixels as weighted sums of neighbors, smoothing away the perturbation's message-bearing pattern. The perturbation is designed on a fixed grid but consumed on the editor's, so the embedder optimizes the former while the latter determines survival. We address this with **Content- and Geometry-aware Residual Transfer (CART)**, which refines the residual at the image's native resolution conditioned on content and size, aligns it with the editor's preprocessing geometry, and—since the editor itself encodes the image in a variational autoencoder (VAE) latent space—matches the watermark-induced latent difference across canonical and native views, preserving the message-bearing structure through both resamplings.

We present **EditR²**—named for the two quantities it models, the editor's Response and the watermark Residual—an end-to-end framework for editing-robust invisible watermarking at native resolution. The two components close a loop: CART shapes the watermark residual on the grid the editor actually processes, DERS measures whether the message survives editing, and the embedder optimizes against DERS's measurement—so the whole framework trains end to end without invoking the editor in the loop.

Our main contributions are summarized as follows:

1. We recast post-edit watermark robustness as learning the editor's response to the watermark perturbation via a differentiable surrogate (DERS), supervised by paired editing—turning a previously unoptimizable objective into one amenable to gradient-based training.
2. We identify the grid-mismatch failure mode of editing-aware watermarking: a residual produced on a fixed canonical grid and merely resized to native resolution loses its message-bearing structure. CART resolves this by content- and size-conditioned refinement at native resolution, alignment with the editor's preprocessing geometry, and consistency of the watermark-induced latent difference across the editor's VAE representations, enabling editing-aware watermarking at native resolution without per-resolution retraining.
3. We validate EditR² extensively on Qwen-Image-Edit: it attains 90.76% mean bit accuracy across five editing operations, 12.00 percentage points above the best-performing baseline in our evaluation, at 47.33 dB PSNR; the same model stays robust across tested native resolutions and transfers without retraining to JoyAI-Image-Edit and Doubao-Seedream-4.5, demonstrating cross-editor generalization.

2 Related Work

Watermark robustness through distortion simulation. Earlier watermarking approaches include DWT–DCT (Al-Haj, 2007). HiDDeN established an end-to-end framework in which embedding and extraction networks are trained around a simulated distortion layer (Zhu et al., 2018). StegaStamp extended this framework to physical capture, while MBRS combined real and simulated JPEG compression (Tancik et al., 2020; Jia et al., 2021). Many subsequent methods follow the embed–distort–extract pipeline (Fernandez et al., 2022; Fang et al., 2022; 2023; Bui et al., 2023; Guo et al., 2024; Rezaei et al., 2025; Sun et al., 2025; Ma et al., 2025b;a; Sander et al., 2025; Xu et al., 2025; Fernandez et al., 2024; Souček et al., 2025). These learning-based methods incorporate anticipated distortions into watermark training.

Watermarking under instruction-guided generative editing. Instruction-guided editing can introduce substantial, instruction-dependent content changes that are difficult to model with a conventional distortion layer. Robust-Wide incorporates partially differentiable denoising into watermark training (Hu et al., 2025). VINE uses blur as a proxy for editing-induced frequency changes and generative priors for embedding (Lu et al., 2025). These methods approximate editing during training. DERS instead learns from matched real edits how a watermark perturbation changes the editor's output. Its Clean-Edit bypass carries the precomputed clean edit to the output, where the predicted response is added to estimate the watermarked edit differentiably.

Watermarking at native resolution. A residual learned at fixed resolution must survive transfer to the source image's native grid and the editor's subsequent resampling. TrustMark addresses native-resolution transfer by resizing the watermark residual and adding it to the unwatermarked source image (Bui et al., 2025). In instruction-guided editing, however, Qwen-Image-Edit resamples the resulting watermarked image again to its preprocessing grid. Native-grid interpolation alone does not explicitly account for this second grid change. This gap motivates CART, which refines the residual using source-image content and the Qwen reference preprocessing grid.

3 Method

3.1 Overview

As illustrated in Figure 2, EditR² comprises a watermark embedder $E_\omega(\cdot)$, a distortion layer $\mathcal{A}(\cdot)$, and a watermark extractor $X_\psi(\cdot)$. The embedder is a U-Net-style convolutional network, whereas

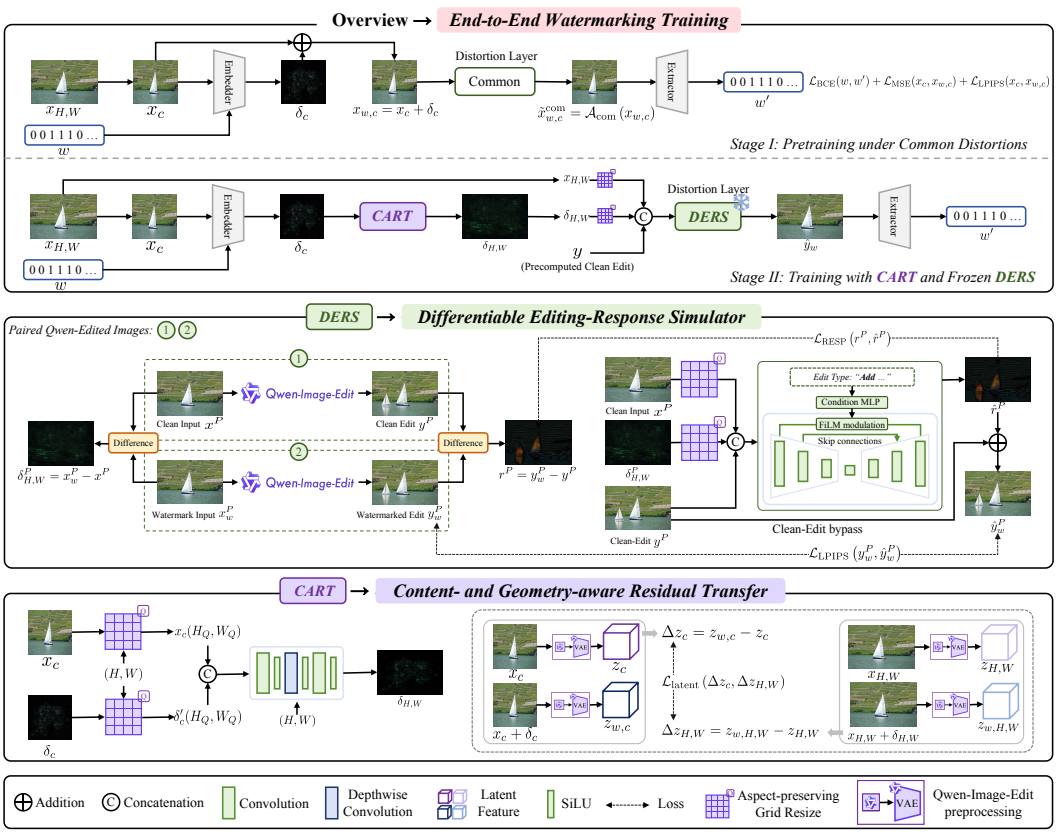

Figure 2: Overview of EditR². The top panel presents the end-to-end watermark embedding pipeline: Stage I pretrains the embedder and extractor under common distortions, whereas Stage II trains the embedder, extractor, and CART through frozen DERS. The middle panel details the DERS architecture and its pretraining procedure. The bottom panel details CART.

the extractor is a ConvNeXt-Base backbone with a 100-bit prediction head (Appendix D.2). Let $x_{H,W} \in [0,1]^{3\times H\times W}$ be a native-resolution unwatermarked source image and $w \in \{0,1\}^B$ a $B = 100$ bit message. The canonical unwatermarked image is $x_c = \mathrm{Resize}_{256,256}(x_{H,W})$; the embedder generates its watermark residual δ_c on this 256×256 grid. The distortion layer uses common distortions in Stage I and frozen DERS in Stage II. Training against the real editor directly is not an option, so EditR² adopts a two-stage protocol.

Stage I: pretraining under common distortions. Stage I trains the embedder and extractor at 256×256 resolution under common distortions listed in Appendix E. Given x_c and w, the embedder generates $\delta_c = E_\omega(x_c, w)$ and forms watermarked image $x_{w,c} = x_c + \delta_c$. $x_{w,c}$ passes through the distortion layer, $\widetilde{x}^{\mathrm{com}}_{w,c} = \mathcal{A}_{\mathrm{com}}(x_{w,c})$. The extractor then predicts the embedded watermark, $w' = X_\psi(\widetilde{x}^{\mathrm{com}}_{w,c})$, where $w' \in (0,1)^B$ denotes the predicted bit probabilities. Stage I minimizes

$$\mathcal{L}_{\mathrm{I}} = \lambda_b\,\mathrm{BCE}(w, w') + \lambda_m\,\mathrm{MSE}(x_c, x_{w,c}) + \lambda_p\,\mathrm{LPIPS}(x_c, x_{w,c}). \tag{1}$$

Here BCE supports message recovery after distortion, whereas MSE and LPIPS preserve visual fidelity. The coefficients $\lambda_b, \lambda_m, \lambda_p$ are shared by both training stages.

Stage II: training for instruction-guided generative editing. Stage II adds CART and the pretrained, frozen DERS to the Stage-I pipeline. For a native-resolution source $x_{H,W}$, it forms $\delta_c = E_\omega(x_c, w)$, $\delta_{H,W} = \mathrm{CART}_\theta(x_c, \delta_c; H, W)$, and $x_w = x_{H,W} + \delta_{H,W}$. Given edit category a, DERS models the editor's response to the watermark perturbation, conditioned on the source, the residual, and the precomputed clean edit y. Adding this response to y gives $\hat{y}_w$, from which $w' = X_\psi(\hat{y}_w)$ is extracted. Stage II minimizes

$$\mathcal{L}_{\mathrm{II}} = \lambda_b\,\mathrm{BCE}(w, w') + \lambda_m\,\mathrm{MSE}(x_{H,W}, x_w) + \lambda_p\,\mathrm{LPIPS}(x_{H,W}, x_w) + \lambda_z\mathcal{L}_{\mathrm{latent}}(\Delta z_c, \Delta z_{H,W}). \tag{2}$$

The latent coefficient λ_z is used only in Stage II; its value is given in Section 4.1 and its ablation in Table 6. We optimize the embedder, CART, and the extractor while keeping DERS frozen. Unlike the simulator, CART has no ground-truth native-resolution residual; it is optimized jointly with the embedder under the message recovery, image-fidelity, and latent-difference consistency terms in Eq. 2. DERS and the VAE are training-time components and are absent at deployment.

Design principle. EditR^2 separates simulation from adaptation. DERS (Section 3.2) models the editor's response to the watermark residual and provides a differentiable training signal; CART (Section 3.3) adapts the residual to native content and geometry to preserve its message-bearing structure through resampling.

Using the real editor in training is impractical: its generative sampling pipeline resists stable gradient-based training, and per-step inference is prohibitively expensive. The distortion layer must be a learned surrogate. We pretrain DERS on paired real edits and freeze it in Stage II. Its explicit residual input lets the recovery loss update the embedder and CART through the frozen simulator.

3.2 DERS: Differentiable Editing-Response Simulator

Learning a surrogate that directly predicts the complete watermarked edit would require it to reproduce the editor's large, instruction-dependent content change before it could provide useful watermark supervision. DERS instead carries y to the output through a Clean-Edit bypass and predicts only the editor's pixel-space editing response to the watermark perturbation. The central construction is a matched pair of editor calls. To form an offline paired sample $\mathcal{P}$, we resize the unwatermarked source image x^P to 256×256, generate its residual with the frozen Stage-I embedder, bicubically interpolate that residual to the native $H \times W$ grid, and add it to x^P to obtain x_w^P. Thus $\delta_{H,W}^P = x_w^P - x^P$. Let $Q_{p,\xi}$ be Qwen-Image-Edit under instruction p and seed ξ. Applying the same editing instruction, sampling seed, output dimensions, and inference settings to x^P and x_w^P gives

$$y^P = Q_{p,\xi}(x^P), \qquad y_w^P = Q_{p,\xi}(x_w^P), \qquad r^P = y_w^P - y^P. \tag{3}$$

Matched instructions, seeds, output dimensions, and inference settings control the editing conditions. We use r^P as the observed paired-edit response associated with the watermark perturbation. DERS receives x^P, $\delta_{H,W}^P$, and y^P as separate inputs, while r^P supplies response supervision.

Given edit category a, DERS predicts

$$\hat{r}^P = \mathrm{DERS}_\phi\big(\mathcal{Q}(x^P), \mathcal{Q}(\delta_{H,W}^P), y^P; a\big), \qquad \hat{y}_w^P = y^P + \hat{r}^P. \tag{4}$$

This equation also makes the division of labor explicit. The clean edit y^P reaches the output unchanged through a Clean-Edit bypass and supplies the complete edited content; the learned branch only predicts the additive response $\hat{r}^P$. Its three image inputs play different roles: x^P supplies the unwatermarked source content, $\delta_{H,W}^P$ supplies the embedding perturbation, and y^P supplies the result of the requested content edit. Before channelwise concatenation, the aspect-preserving grid resize $\mathcal{Q}$ maps x^P and $\delta_{H,W}^P$ to a common (H_Q, W_Q) grid determined by the native source size (H, W) (Section 3.3). The resulting tensors align with the spatial grid of y^P. An MLP encodes the edit category a for feature-wise linear modulation (FiLM), allowing a single simulator to represent multiple edit types. DERS architecture details appear in Appendix D.3.

DERS combines perceptual reconstruction with an energy-normalized response-matching loss:

$$\mathcal{L}_{\mathrm{LPIPS}} = \mathrm{LPIPS}(y^P + \hat{r}^P, y_w^P), \qquad \mathcal{L}_{\mathrm{RESP}} = \frac{\|\hat{r}^P - r^P\|_2^2}{\|r^P\|_2^2 + \varepsilon}, \quad r^P = y_w^P - y^P, \tag{5}$$

where $\varepsilon > 0$. Its pretraining objective is $\mathcal{L}_{\mathrm{DERS}} = 0.10\,\mathcal{L}_{\mathrm{LPIPS}} + 0.25\,\mathcal{L}_{\mathrm{RESP}}$. The perceptual term constrains the output $y^P + \hat{r}^P$ against the real watermarked edit. The response-matching term measures squared ℓ_2 error against the editor's response to the watermark perturbation, normalized by the target response energy. The constant ε stabilizes the loss when that response is weak.

During Stage II, frozen $\mathrm{DERS}_{\phi^\star}$ predicts $\hat{r} = \mathrm{DERS}_{\phi^\star}(\mathcal{Q}(x_{H,W}), \mathcal{Q}(\delta_{H,W}), y; a)$, after which $\hat{y}_w = y + \hat{r}$ and $w' = X_\psi(\hat{y}_w)$. Although $\phi^\star$ is fixed, gradients with respect to $\delta_{H,W}$ are retained, so the recovery loss updates CART and the embedder. Algorithm 1 in Appendix C gives the full pretraining procedure.

3.3 CART: Content- and Geometry-aware Residual Transfer

CART transfers the message-bearing watermark residual learned at 256×256 to native-resolution images such as 1080p, 2K, and 4K inputs. Resizing a residual can match the input dimensions, but the editor subsequently resamples the watermarked image to its working grid. We therefore condition CART's native-resolution refinement on image content and editor-facing geometry.

Instruction-guided editors often resize inputs to an internal working grid. We denote the purple *Aspect-preserving Grid Resize* blocks in Figure 2 by $\mathcal{Q}$. Given native size (H, W), $\mathcal{Q}$ selects a grid (H_Q, W_Q) using the source aspect ratio and a target pixel area of 1024^2, rounding each dimension to the nearest multiple of 32, following Qwen-Image-Edit's VAE-input sizing rule (Qwen Team, 2025). The native size fixes this grid for the spatial inputs to CART and DERS within each sample.

CART applies $\mathcal{Q}$ to its canonical inputs, $x_{c(H_Q,W_Q)} = \mathcal{Q}(x_c)$ and $\delta'_{c(H_Q,W_Q)} = \mathcal{Q}(\delta_c)$, then predicts the native-resolution residual $\delta_{H,W} = \mathrm{CART}_\theta([x_{c(H_Q,W_Q)}, \delta'_{c(H_Q,W_Q)}]; H, W)$ for addition to the source image. Here $[\cdot,\cdot]$ denotes channelwise concatenation. Grid alignment exposes the spatial arrangement before editing, while image content guides residual refinement.

To regularize residual transfer in the editor's representation space, we apply the *Qwen-Image-Edit preprocessing* blocks in Figure 2 identically to unwatermarked and watermarked images in both canonical and native branches. These blocks perform image-size processing followed by frozen-VAE encoding; $\mathcal{Q}$ denotes only the purple grid resize for CART and DERS spatial inputs. Let z_c and $z_{w,c}$ be the resulting frozen-VAE features of x_c and $x_c + \delta_c$; likewise, $z_{H,W}$ and $z_{w,H,W}$ correspond to $x_{H,W}$ and $x_{H,W} + \delta_{H,W}$. The watermark-induced latent differences are

$$\Delta z_c = z_{w,c} - z_c, \qquad \Delta z_{H,W} = z_{w,H,W} - z_{H,W}. \tag{6}$$

Because the two latent grids may differ in spatial size, we bilinearly resize the canonical difference to the native latent grid before matching (implementation details appear in Appendix D.4):

$$\mathcal{L}_{\mathrm{latent}}(\Delta z_c, \Delta z_{H,W}) = \mathrm{L1}(\mathrm{sg}(\mathcal{I}_{\mathrm{bil}}(\Delta z_c)), \Delta z_{H,W}), \tag{7}$$

Here $\mathcal{I}_{\mathrm{bil}}$ bilinearly resizes to the spatial size of $\Delta z_{H,W}$, L1 is mean absolute error, and the canonical difference is a detached reference. This encourages preservation of the watermark-induced VAE latent difference after geometric adaptation; the VAE is used only during training.

4 Experiments

4.1 Implementation and Evaluation Details

We train EditR^2 in two stages. Stage I pretrains the embedder and extractor on 100,000 randomly sampled COCO2017 images (Lin et al., 2014), resized to 256×256 and embedded with random 100-bit messages. For each of three seen edit categories (object addition, background replacement, and recoloring), DERS uses 400 paired-editing tuples for pretraining and 50 held-out tuples for validation. Each tuple uses an independent payload embedded by the frozen Stage-I model; Appendix D.1 details pair construction. Stage II uses 1,000 source images per category, split into 800 for training and 200 for testing. This split is separate from the 300-image main evaluation benchmark. Both stages use $(\lambda_b, \lambda_m, \lambda_p) = (1.5, 1.0, 1.0)$ for BCE, MSE, and LPIPS. Stage II additionally sets $\lambda_z = 1.0$ and trains the embedder, extractor, and CART through frozen DERS. DERS is used only for training. Training uses one NVIDIA RTX PRO 6000 GPU, and Doubao-Seedream-4.5 inference uses its official API.

We compare EditR^2 with DWT-DCT (Al-Haj, 2007), MBRS (Jia et al., 2021), VideoSeal (Fernandez et al., 2024), Robust-Wide (Hu et al., 2025), InvisMark (Xu et al., 2025), VINE (Lu et al., 2025), TrustMark-Q (Bui et al., 2025), and PixelSeal (Souček et al., 2025), using their officially released pretrained models. On Qwen-Image-Edit (Wu et al., 2025; Qwen Team, 2025), all methods share images, prompts, and seeds across object addition, replacement, removal, background replacement, and recoloring. Prompt templates are given in Appendix A.

We report PSNR, SSIM (Wang et al., 2004), LPIPS, and SIFID (Zhang et al., 2018; Shaham et al., 2019) for perceptual quality, and mean bit accuracy (ACC, %) for robustness and generalization. The main editing and fidelity benchmarks each contain 300 images: 150 2K images from

DIV2K (Agustsson & Timofte, 2017) and 150 4K images from UHRSD (Xie et al., 2022). This balanced construction gives the standard 2K and ultra-high-resolution 4K regimes equal weight while keeping a common evaluation set for all methods. For the resolution study, we evaluate 50 images at each of six native resolutions for fidelity, common-distortion robustness, and Qwen-Image-Edit robustness. JoyAI-Image-Edit (Song et al., 2026) and Doubao-Seedream-4.5 (ByteDance Seed, 2025) are evaluated without retraining; *None* denotes a watermarked input without additional distortion.

4.2 Comparison with State-of-the-Art Methods

Perceptual Quality. Table 1 evaluates the visual fidelity of the native-resolution watermarked images. EditR^2 ranks among the leading methods on the reported fidelity metrics, indicating that adapting the message-bearing structure to the native grid need not introduce conspicuous visual artifacts. Figure 3 qualitatively compares methods on a DIV2K example. The residual display exposes a regular grid for MBRS and a more prominent periodic pattern for Robust-Wide, whereas the EditR^2 residual is less conspicuous. This content-aligned residual pattern illustrates how CART adapts the message-bearing structure at native resolution rather than applying one spatial pattern uniformly.

Table 1: Perceptual quality. PSNR, SSIM, LPIPS, and SIFID are measured against the unwatermarked source images. Bold and underlined values are best and second-best.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	SIFID ↓
DWT-DCT (Al-Haj, 2007)	39.4361	0.9700	0.0229	0.016788
MBRS (Jia et al., 2021)	42.8718	0.9945	0.0061	0.000286
VideoSeal (Fernandez et al., 2024)	45.4738	0.9960	0.0017	**0.000011**
Robust-Wide (Hu et al., 2025)	40.2927	0.9946	0.0041	0.000147
InvisMark (Xu et al., 2025)	**49.0195**	0.9949	0.0021	0.000208
VINE (Lu et al., 2025)	34.9224	0.9932	0.0077	0.001062
TrustMark-Q (Bui et al., 2025)	44.7237	<u>0.9968</u>	<u>0.0014</u>	0.000036
PixelSeal (Souček et al., 2025)	46.3887	0.9963	0.0015	<u>0.000019</u>
$\mathbf{EditR}^2_{(\mathrm{ours})}$	<u>47.3290</u>	**0.9979**	**0.0012**	0.000037

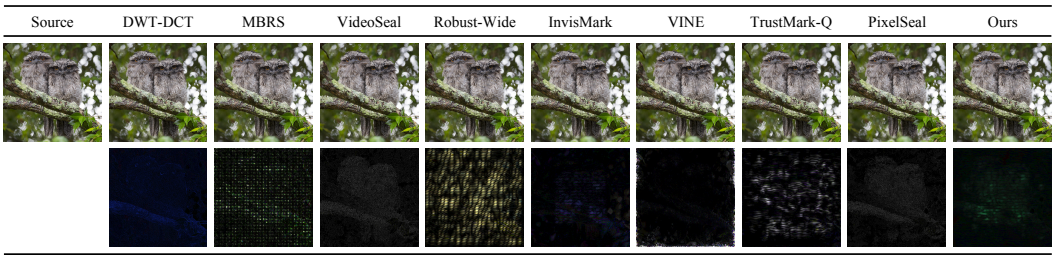

Figure 3: Qualitative comparison of perceptual distortion on a DIV2K example. The top row shows the unwatermarked source and watermarked images for each method. The bottom row shows the corresponding pixel-wise residuals, amplified by $15\times$ with a common display range.

Robustness and Generalization. Table 2 shows that EditR^2 leads all five Qwen-Image-Edit categories, reaching 90.76% mean bit accuracy, 12.00 percentage points above VINE, the strongest evaluated baseline. This pattern follows the response-focused design of DERS: paired edits supervise how the editor transforms the watermark perturbation, while the clean edit supplies the scene content, so the learned simulator does not need to encode one particular edited appearance. CART complements this separation by maintaining the residual on the native and fixed Qwen reference grids. Without retraining, EditR^2 also transfers directly to JoyAI-Image-Edit and Doubao-Seedream-4.5, with the cross-editor results summarized in Table 3. The transfer suggests that the combined DERS and CART design captures editing effects that remain useful beyond the editor used for paired supervision. Figure 4 visualizes representative editing effects.

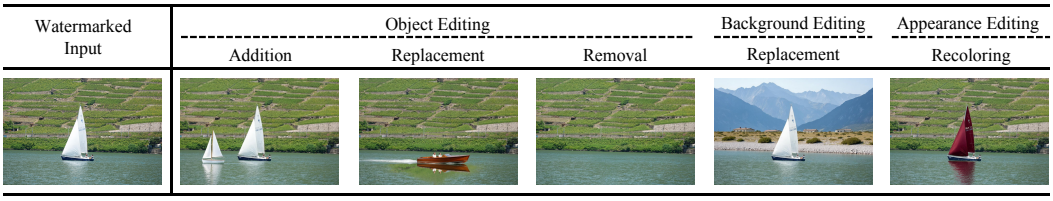

Figure 4: Visual examples of Qwen-Image-Edit operations used to evaluate post-edit bit accuracy.

Table 2: Bit accuracy (%) after instruction-guided editing with Qwen-Image-Edit. Figure 4 provides visual examples of the evaluated editing operations. Mean averages the edit categories; best results are bold. † denotes edit categories unseen during training and evaluated only at test time.

Method	Object Editing			Background	Appearance	Mean
	Addition	Replacement†	Removal†	Replacement	Recoloring	
DWT-DCT (Al-Haj, 2007)	50.54	50.36	50.44	50.92	50.47	50.55
MBRS (Jia et al., 2021)	62.07	58.21	57.74	53.88	58.67	58.11
VideoSeal (Fernandez et al., 2024)	70.43	67.06	66.50	59.44	70.37	66.76
Robust-Wide (Hu et al., 2025)	81.03	74.65	74.24	62.43	79.53	74.38
InvisMark (Xu et al., 2025)	65.58	58.08	57.26	54.74	56.36	58.40
VINE (Lu et al., 2025)	89.66	78.44	80.11	67.84	77.77	78.76
TrustMark-Q (Bui et al., 2025)	62.32	62.21	61.79	55.93	67.08	61.87
PixelSeal (Souček et al., 2025)	69.94	65.72	65.24	59.42	69.56	65.98
$\mathbf{EditR}^2_{(\text{ours})}$	**94.24**	**90.17**	**90.13**	**85.38**	**93.87**	**90.76**

Table 3: Cross-editor generalization without retraining. Bit accuracy (%) of EditR² across JoyAI-Image-Edit (open-source) and Doubao-Seedream-4.5 (closed-source).

Target editor	Object Editing			Background	Appearance	Mean
	Addition	Replacement	Removal	Replacement	Recoloring	
JoyAI-Image-Edit	95.55	88.90	86.17	78.70	89.47	87.76
Doubao-Seedream-4.5	91.56	83.83	79.11	75.72	91.78	84.40

Table 4: Bit accuracy (%) under common distortions. Distortion abbreviations and parameter settings are provided in the Appendix E. Bold and underlined values are best and second-best.

Method	None –	Flip H/V	Crop 30–50%	GB 3/5/7	JPEG 10–60	GN $\sigma = 0.04$	B/C ± 0.2	Resize 20–80%	Posterize 4 bits	Rotation $[15°–45°]$
DWT-DCT (Al-Haj, 2007)	87.16	50.10	50.29	68.13	50.94	88.12	86.12	80.78	88.77	92.87
MBRS (Jia et al., 2021)	99.99	49.81	49.86	99.99	99.75	99.98	98.36	99.87	99.11	96.69
VideoSeal (Fernandez et al., 2024)	98.24	75.50	81.00	97.64	95.07	97.48	96.17	96.81	94.02	90.83
Robust-Wide (Hu et al., 2025)	**100.00**	45.96	49.87	**100.00**	99.89	**100.00**	99.41	**100.00**	99.65	99.04
InvisMark (Xu et al., 2025)	**100.00**	99.27	99.35	99.55	94.71	**100.00**	99.80	**100.00**	99.81	99.67
VINE (Lu et al., 2025)	99.98	50.05	49.80	99.97	**99.97**	99.97	**99.95**	99.97	**99.97**	**99.97**
TrustMark-Q (Bui et al., 2025)	97.33	73.66	77.50	97.13	92.82	96.87	93.64	96.77	92.08	96.97
PixelSeal (Souček et al., 2025)	99.43	74.88	96.21	99.10	96.40	98.95	97.99	97.85	96.38	69.98
$\mathbf{EditR}^2_{(\text{ours})}$	**100.00**	**99.85**	**99.42**	99.97	98.52	99.97	99.75	99.97	99.90	**99.97**

Common distortions test whether robustness to instruction-guided generative editing coexists with reliable recovery under ordinary processing (Table 4). EditR² remains above 98% in every setting and leads flipping, cropping, and rotation. MBRS's 99.75% JPEG accuracy but 58.11% Qwen-Image-Edit mean confirms that common distortions and generative editing test different failure modes; EditR² performs strongly in both.

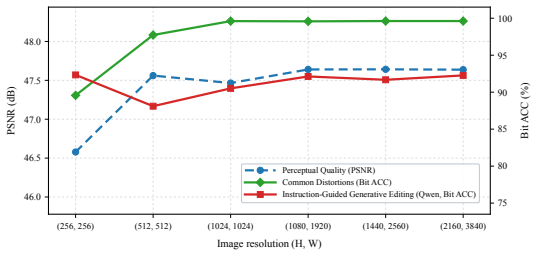

Figure 5: Performance across native resolutions. ACC denotes mean bit accuracy. Qwen-Image-Edit and common-distortion settings follow Tables 2 and 4.

Performance across image resolutions. Figure 5 evaluates EditR² at six native resolutions from 256×256 to 4K. Across this range, the model maintains high image fidelity and bit accuracy under both common distortions and Qwen-Image-Edit, without resolution-specific retraining. Together, CART's residual transfer and DERS's editing-response feedback enable this cross-resolution stability.

4.3 Ablation Study

We examine the individual and joint contributions of DERS and CART, compare response prediction with direct full-edit prediction, and ablate CART's content, grid-alignment, and latent-difference consistency components. All variants use the same Qwen-Image-Edit evaluation split and have comparable source-image PSNR before editing.

DERS ablation. The *Baseline* and *Only DERS* variants use bicubic interpolation to transfer the 256×256 residual to native resolution. DERS improves post-edit bit accuracy under both bicubic and CART transfer, with a larger gain when paired with CART (Table 5). We next compare DERS's response prediction with direct prediction of the complete watermarked edit. *Full-Edit Prediction+CART* and *DERS+CART* share the clean edit y as a condition, the U-Net backbone, CART, paired data, training losses, and training budget. They differ only in output parameterization: the former predicts the complete watermarked edit, whereas the latter predicts a response added to y (Appendix D.3). We evaluate both using the same validation watermark-response error, subtracting y from the full-edit prediction before comparison with the observed pixel-space editing response. During matched pretraining, the full-edit variant starts with higher error and remains well above DERS after both curves level off (Figure 6(a)), indicating that full-edit prediction is harder for learning the watermark response. The same ordering holds after real editing: at comparable source-image PSNR (47.3 versus 47.33 dB), Full-Edit Prediction+CART achieves lower bit accuracy in every category (Figure 6(b)). Accuracy rises through 400 training tuples per seen category but falls slightly at 500 (Figure 6(c)); we therefore use 400 training and 50 held-out validation tuples per seen category.

CART ablation. Without DERS, CART raises mean bit accuracy from 60.23% to 62.15%; with DERS, it raises accuracy from 77.81% to 90.76% (Table 5). Removing content (C), grid alignment (G), or latent-difference consistency (L) lowers accuracy to 80.72%, 85.56%, or 80.24%, respectively (Table 6). These results support transferring the residual with source content and editor-facing geometry while preserving its message-bearing latent structure.

Table 5: EditR^2 component ablations.

Variant	DERS	CART	PSNR↑	ACC↑	Δ (pp)
Baseline	✗	✗	46.22	60.23	−30.53
Only DERS	✓	✗	46.81	77.81	−12.95
Only CART	✗	✓	46.63	62.15	−28.61
$\mathbf{EditR}^2_{(\mathrm{ours})}$	✓	✓	47.33	**90.76**	0.00

Table 6: CART component ablations.

Variant	C	G	L	PSNR↑	ACC↑	Δ (pp)
Grid + latent	✗	✓	✓	46.54	80.72	−10.04
Content + latent	✓	✗	✓	46.76	85.56	−5.20
Content + grid	✓	✓	✗	46.33	80.24	−10.52
CART ($\mathbf{EditR}^2_{(\mathrm{ours})}$)	✓	✓	✓	47.33	**90.76**	0.00

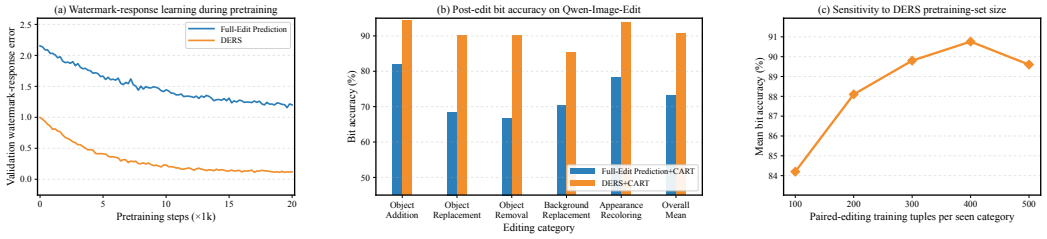

Figure 6: DERS ablation. (a) Validation watermark-response error during matched pretraining of DERS and Full-Edit Prediction. (b) Bit accuracy after real editing for both predictors with CART. (c) Bit accuracy versus the number of DERS pretraining tuples per seen category.

5 Conclusion

Instruction-guided generative editing poses a distinct challenge to watermark recovery because the editor reconstructs image content. EditR^2 learns the editor's response to the watermark perturbation from paired edits and adapts the residual to native content and the editor's preprocessing geometry. The clean edit already reflects the requested modification; DERS adds only its predicted watermark-induced response. Across five Qwen-Image-Edit operations, EditR^2 achieves 90.76% mean bit accuracy, 12.00 percentage points above the strongest evaluated baseline, at 47.33 dB source-image PSNR. It remains effective across tested resolutions and two other editors without retraining.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing of the manuscript, including improving language expression, clarity, readability, and organization. We did not use generative AI tools to generate the core research ideas, methodology, experimental results, or conclusions of this work. All AI-assisted text was manually reviewed, verified, and revised by the authors to ensure its correctness and consistency with the authors' intended meaning. We take full responsibility for the final content of this work, including text produced with the assistance of generative AI.

REFERENCES

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1122–1131, 2017. doi: 10.1109/CVPRW.2017.150. URL https://openaccess.thecvf.com/content_cvpr_2017_workshops/w12/html/Agustsson_NTIRE_2017_Challenge_CVPR_2017_paper.html.

Ali Al-Haj. Combined DWT-DCT digital image watermarking. *Journal of Computer Science*, 3(9): 740–746, 2007. doi: 10.3844/jcssp.2007.740.746.

Tu Bui, Shruti Agarwal, Ning Yu, and John Collomosse. RoSteALS: Robust steganography using autoencoder latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 933–942, 2023. URL https://openaccess.thecvf.com/content/CVPR2023W/WMF/html/Bui_RoSteALS_Robust_Steganography_Using_Autoencoder_Latent_Space_CVPRW_2023_paper.html.

Tu Bui, Shruti Agarwal, and John Collomosse. TrustMark: Robust watermarking and watermark removal for arbitrary resolution images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18629–18639. IEEE, 2025. doi: 10.1109/ICCV51701.2025.01731. URL https://openaccess.thecvf.com/content/ICCV2025/papers/Bui_TrustMark_Robust_Watermarking_and_Watermark_Removal_for_Arbitrary_Resolution_Images_ICCV_2025_paper.pdf.

ByteDance Seed. Seedream 4.5. Official model page, 2025. URL https://seed.bytedance.com/en/seedream4_5. Accessed 2026-09-22.

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*, pp. 2267–2275, 2022. doi: 10.1145/3503161.3548049.

Han Fang, Zhaoyang Jia, Yupeng Qiu, Jiyi Zhang, Weiming Zhang, and Ee-Chien Chang. De-END: Decoder-driven watermarking network. *IEEE Transactions on Multimedia*, 25:7571–7581, 2023. doi: 10.1109/TMM.2022.3223559.

Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3054–3058, 2022. doi: 10.1109/ICASSP43922.2022.9746058.

Pierre Fernandez, Hady Elsahar, I. Zeki Yalniz, and Alexandre Mourachko. Video Seal: Open and efficient video watermarking. *arXiv preprint arXiv:2412.09492*, 2024. doi: 10.48550/arXiv.2412.09492. URL https://arxiv.org/abs/2412.09492.

Yiyang Guo, Ruizhe Li, Mude Hui, Hanzhong Guo, Chen Zhang, Chuangjian Cai, Le Wan, and Shangfei Wang. FreqMark: Invisible image watermarking via frequency based optimization in latent space. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3564. URL https://papers.nips.cc/paper_files/paper/2024/hash/cbc4912b67d57e3932f56f3fa99faab3-Abstract-Conference.html.

Runyi Hu, Jie Zhang, Ting Xu, Jiwei Li, and Tianwei Zhang. Robust-Wide: Robust watermarking against instruction-driven image editing. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, volume 15080 of *Lecture Notes in Computer Science*, pp. 20–37. Springer Nature Switzerland, 2025. doi: 10.1007/978-3-031-72670-5_2. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/03220.pdf.

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*, pp. 41–49. Association for Computing Machinery, 2021. doi: 10.1145/3474085.3475324. URL https://doi.org/10.1145/3474085.3475324.

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*, volume 8693 of *Lecture Notes in Computer Science*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.

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. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d077bc9ea82a2998ca6b2d0158b5ac6e-Abstract-Conference.html.

Linfeng Ma, Han Fang, Tianyi Wei, Zijin Yang, Zehua Ma, Weiming Zhang, and Nenghai Yu. A geometric distortion immunized deep watermarking framework with robustness generalizability. In *Computer Vision – ECCV 2024*, volume 15125 of *Lecture Notes in Computer Science*, pp. 268–285. Springer, 2025a. doi: 10.1007/978-3-031-72855-6_16.

Zehua Ma, Han Fang, Xi Yang, Kejiang Chen, and Weiming Zhang. RoPaSS: Robust watermarking for partial screen-shooting scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19332–19339, 2025b. doi: 10.1609/aaai.v39i18.34128.

Qwen Team. Qwen-Image-Edit-2511. Hugging Face model card, 2025. URL https://huggingface.co/Qwen/Qwen-Image-Edit-2511. Accessed 2026-08-31.

Ahmad Rezaei, Mohammad Akbari, Saeed Ranjbar Alvar, Arezou Fatemi, and Yong Zhang. LaWa: Using latent space for in-generation image watermarking. In *Computer Vision – ECCV 2024*, volume 15147 of *Lecture Notes in Computer Science*, pp. 118–136. Springer, 2025. doi: 10.1007/978-3-031-73024-5_8.

Tom Sander, Pierre Fernandez, Alain Oliviero Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c5ee1911dffe4886367eee7fca314235-Abstract-Conference.html.

Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SinGAN: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4570–4580, 2019. doi: 10.1109/ICCV.2019.00467.

Lin Song, Wenbo Li, Guoqing Ma, Wei Tang, Bo Wang, Yuan Zhang, Yijun Yang, Yicheng Xiao, Jianhui Liu, Yanbing Zhang, Guohui Zhang, Wenhu Zhang, Hang Xu, Nan Jiang, Xin Han, Haoze Sun, Maoquan Zhang, Haoyang Huang, and Nan Duan. JoyAI-Image: Awaking spatial intelligence in unified multimodal understanding and generation. *arXiv preprint arXiv:2605.04128*, 2026. doi: 10.48550/arXiv.2605.04128. URL https://arxiv.org/abs/2605.04128.

Tomáš Souček, Pierre Fernandez, Hady Elsahar, Sylvestre-Alvise Rebuffi, Valeriu Lacatusu, Tuan Tran, Tom Sander, and Alexandre Mourachko. Pixel Seal: Adversarial-only training for invisible image and video watermarking. *arXiv preprint arXiv:2512.16874*, 2025. doi: 10.48550/arXiv.2512.16874. URL https://arxiv.org/abs/2512.16874.

Nan Sun, Han Fang, Yuxing Lu, Chengxin Zhao, and Hefei Ling. END^2: Robust dual-decoder watermarking framework against non-differentiable distortions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 773–781, 2025. doi: 10.1609/aaai.v39i1.32060.

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. doi: 10.1109/CVPR42600.2020.00219. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Tancik_StegaStamp_Invisible_Hyperlinks_in_Physical_Photographs_CVPR_2020_paper.html.

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. doi: 10.1109/TIP.2003.819861.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025. doi: 10.48550/arXiv.2508.02324. URL https://arxiv.org/abs/2508.02324.

Chenxi Xie, Changqun Xia, Mingcan Ma, Zhirui Zhao, Xiaowu Chen, and Jia Li. Pyramid grafting network for one-stage high resolution saliency detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11717–11726, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Xie_Pyramid_Grafting_Network_for_One-Stage_High_Resolution_Saliency_Detection_CVPR_2022_paper.html.

Rui Xu, Mengya Hu, Deren Lei, Yaxi Li, David Lowe, Alex Gorevski, Mingyu Wang, Emily Ching, and Alex Deng. InvisMark: Invisible and robust watermarking for AI-generated image provenance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 909–918. IEEE, 2025. doi: 10.1109/WACV61041.2025.00098. URL https://openaccess.thecvf.com/content/WACV2025/html/Xu_InvisMark_Invisible_and_Robust_Watermarking_for_AI-Generated_Image_Provenance_WACV_2025_paper.html.

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*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. HiDDeN: Hiding data with deep networks. In *Computer Vision – ECCV 2018*, volume 11219 of *Lecture Notes in Computer Science*, pp. 682–697. Springer, 2018. doi: 10.1007/978-3-030-01267-0_40.