LEARNING WHAT TO PROTECT: ROLE-AWARE REPRESENTATIONS FOR IMAGE SELF-RECOVERY AGAINST TAMPERING

Anonymous authors
Paper under double-blind review

ABSTRACT

With the rapid advancement of generative AI and image manipulation techniques, distinguishing authentic content from manipulated images has become increasingly challenging, raising growing forensic and security concerns. Existing image forensics methods can detect and localize manipulation but rarely recover the original content. Image self-recovery attempted to address this limitation by embedding recovery information into the image as a watermark. However, existing methods still face a longstanding *trade-off* between watermark imperceptibility and recovery fidelity. They typically protect information from the entire image without considering that different information contributes differently to recovery, resulting in redundant recovery information and a higher embedding burden. Hence, we propose **ProCast**, a novel self-recovery framework to tackle this fundamental challenge by jointly addressing three interdependent questions: *what to protect*, *where to embed*, and *how to recover*. ProCast learns to organize image information into components with different recovery roles and applies different protection strategies accordingly, reducing unnecessary recovery information and the embedding burden. It further adopts content-adaptive embedding, and reliability-based recovery to achieve high-fidelity recovery with improved watermark imperceptibility. Notably, although trained only on splicing, ProCast **generalizes to unseen AIGC and conventional manipulations**. Extensive experiments show that ProCast achieves higher watermark imperceptibility, more accurate tampering localization, and higher-fidelity recovery, **improving the quality** (PSNR) of watermarked image by **3.8 dB** and the quality of recovered image by around **2–8 dB** over the strongest existing self-recovery baselines.

1 INTRODUCTION

Advances in generative AI and image manipulation have made tampered content increasingly realistic and difficult to distinguish from authentic photographs (Huang et al., 2026). This poses growing risks to personal use, corporate applications, and law enforcement, where the authenticity of visual evidence is critical. Beyond detecting manipulations, there is a growing need to preserve sufficient information to recover the original content for forensic analysis and trustworthy use.

Research on image self-recovery aims to localize image manipulations and recover their original content by embedding recovery information into the image as an invisible watermark (Yu et al., 2026; An et al., 2026; Ying et al., 2023). This provides active image protection without relying on external copies. As shown in Fig. 1, a trusted content provider embeds an invisible watermark into an image before sharing it. Once published, the image may undergo diverse manipulations, ranging from conventional editing to AI-generated manipulations by (unseen) generative models. At the verifier side, only the received image is available. The embedded recovery information allows the verifier to localize tampered regions and further recover their original content.

Nevertheless, existing self-recovery methods (Kim & Seo, 2026; Liu et al., 2025) still face a fundamental trade-off between watermark imperceptibility and recovery fidelity. Embedding more comprehensive recovery information can improve recovery quality, but also introduces greater distortion to the protected image. Conversely, limiting the embedded information improves imperceptibility

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.

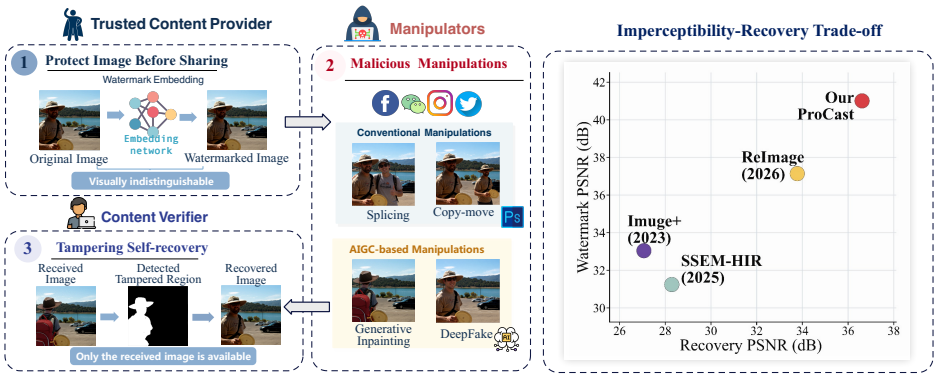


Figure 1: A typical usage scenario of **ProCast**. A trusted provider embeds recovery information before image sharing, enabling tampering localization and content recovery using only the received image. The right panel shows that ProCast achieves a better balance between recovery fidelity and watermark imperceptibility, with higher quality for both recovered and watermarked images.

but leaves insufficient information for high-fidelity recovery. Existing designs exhibit three key limitations: 1). **Redundant Recovery Information**. Most methods embed the entire image content as recovery information, although different image content does not contribute equally to recovery. This introduces unnecessary redundancy and increases the embedding burden. 2). **Lack of Content-Adaptive Protection**. Watermark residuals are typically generated through a global network, lacking adaptation to local content. This can introduce unnecessary distortion in perceptually sensitive regions while providing insufficient protection where stronger embedding is needed. And 3). **Vulnerable Single-Path Recovery**. Existing recovery processes mainly rely on a single decoding path to reconstruct the recovery reference. When the embedded recovery information is corrupted or inaccurately extracted, no alternative reference is available, potentially leading to recovery failure.

To address these limitations, we propose **ProCast**, a proactive framework for image tampering localization and self-recovery across diverse manipulations, including AIGC-based manipulations and conventional manipulations. ProCast addresses the three limitations by determining *what to protect, where to embed, and how to recover*. Specifically, Role-Aware Hierarchical Watermark Generation (RHWG) addresses redundant recovery information by learning different recovery roles for image information. These roles enable selective protection according to their contribution to recovery. **Content-Adaptive Wavelet Embedding (CAWE)** addresses the lack of content-adaptive protection by adapting watermark embedding to local image characteristics. Finally, **Reliability-Based Multi-Backup Recovery (RMBR)** mitigates single-path reliance by leveraging multiple recovery backups and manipulation evidence to select the more reliable reference for recovery.

We conduct extensive experiments across five datasets under various conventional and AIGC-based manipulations. The results demonstrate improved watermark imperceptibility, accurate tampering localization, high-fidelity recovery, and generalization to unseen manipulations and datasets. The main contributions are summarized as follows:

- We introduce a role-aware self-recovery strategy that learns **what to protect**. Through asymmetric recovery constraints, ProCast organizes image information into base, enhancement, and discardable auxiliary components, enabling selective protection with reduced embedding redundancy.
- Based on the learned recovery roles, CAWE further addresses **where to embed** by adaptively embedding role-specific recovery information across Haar subspaces. For **how to recover**, RMBR propagates manipulation evidence to estimate the spatial reliability of different backups and combines them through reliability-weighted fusion.
- Experiments on five datasets demonstrate high-quality recovery and strong generalization to unseen manipulations. Trained only on splicing, ProCast generalizes to copy-move and diverse AIGC-based manipulations, improving PSNR of the watermarked image by around **3.8 dB** and PSNR of the recovered image by around **2–8 dB** over the strongest self-recovery baseline.

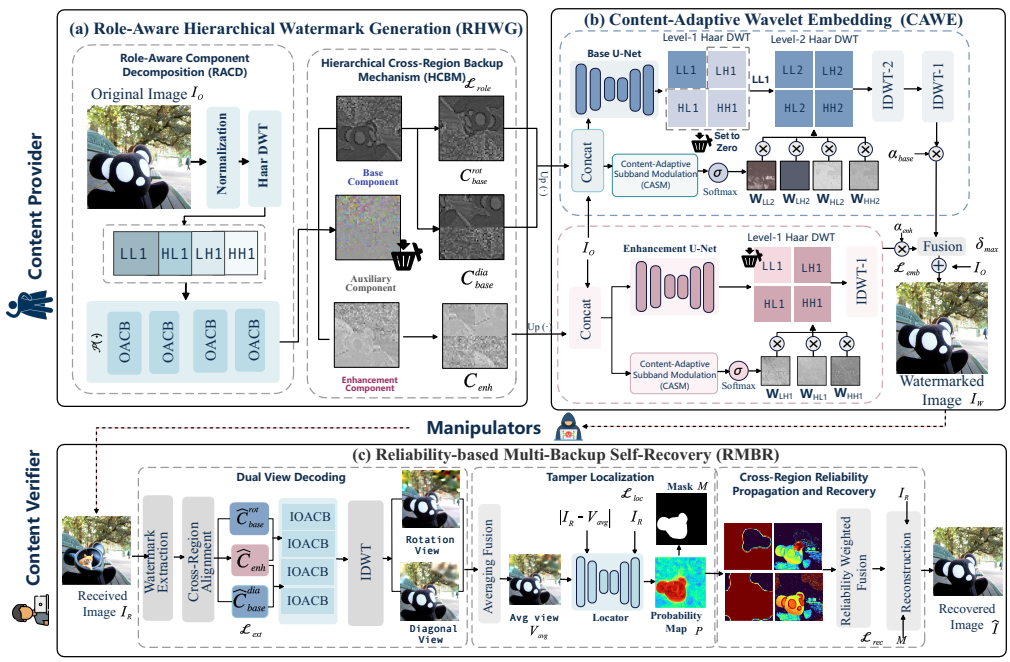


Figure 2: Overview of proposed ProCast.

2 Related Works

Deep Watermarking. Deep watermarking uses neural networks to embed watermarks into cover content. Initially developed for copyright protection, it has recently been extended to content authentication and digital forensics. Methods such as EditGuard(Zhang et al., 2024), OmniGuard(Zhang et al., 2025), and StableGuard(Yang et al., 2025) use watermarks to identify manipulated regions. However, localization cannot restore the tampered content, motivating watermark-based self-recovery(Liu et al., 2025), which embeds recovery information to jointly support tampering localization and content recovery.

Image Self-Recovery. Traditional methods embed recovery watermarks into image LSBs(Liu & Yuan, 2021), but are vulnerable to degradations. Deep learning has therefore been introduced to improve robustness. For face recovery, RecoverMark(An et al., 2026) embeds facial content into the surrounding background, while VeriFi(Yu et al., 2026) uses a compact semantic latent watermark. For general images, Imuge+(Ying et al., 2023) encodes RGB and edge information through an invertible network, SSEM-HIR(Liu et al., 2025) embeds a copy of the entire image in the spatial domain, and ReImage(Kim & Seo, 2026) embeds shuffled image information in the Haar wavelet domain. Despite improved robustness, these methods still preserve information from the entire image for recovery, introducing redundancy. In contrast, ProCast learns different recovery roles and applies role-specific protection, reducing redundant information and the embedding burden.

3 Proposed Method

The overview of ProCast is shown in Fig. 2. On the provider side, RHWG determines what recovery information to protect, while CAWE adaptively embeds it into the original image $\boldsymbol{I}_O$ to generate the watermarked image $\boldsymbol{I}_W$. On the content verifier side, only the received image is available. RMBR localizes manipulated regions and recovers their content using the extracted watermark.

3.1 Role-aware hierarchical watermark generation

To reduce redundancy, RHWG decomposes the recovery information according to different functional roles. Specifically, a Role-Aware Component Decomposition (RACD) module is introduced to separate the recovery information into three components: Base Component

C_{base}, Enhancement Component C_{enh}, and Auxiliary Component C_{aux}. The three components play different roles in self-recovery. The C_{base} provides the primary information for self-recovery. The C_{enh} serves as a refinement component that provides complementary information for improving recovery. In contrast, the C_{aux} contributes little to recovery and is therefore regarded as redundant information. It can be discarded without being embedded into the image. Then, the Hierarchical Cross-Region Backup Mechanism (HCBM) applies different protection levels to the components according to their contribution to recovery.

Role-Aware Component Decomposition. Given an original image $\boldsymbol{I}_O$, we resize and normalize it to $[-1, 1]$ to obtain the preprocessed input $\boldsymbol{I}_P$. Then, the Haar DWT is applied to $\boldsymbol{I}_P$ to obtain the coefficients. However, this predefined subband organization reflects scale and orientation characteristics rather than their contributions to image self-recovery. Therefore, we employ a reversible projection $\mathcal{P}(\cdot)$ to reorganize them. The resulting features are split into three recovery-oriented components $\boldsymbol{C}_{\text{base}}$, $\boldsymbol{C}_{\text{enh}}$, and $\boldsymbol{C}_{\text{aux}}$. This decomposition only defines the capacity of the three components. Their recovery roles are learned through the role-aware constraints described below. The above process is formulated as: $[\boldsymbol{C}_{\text{base}}, \boldsymbol{C}_{\text{enh}}, \boldsymbol{C}_{\text{aux}}] = \text{Split}(\mathcal{P}(\text{Haar}(\boldsymbol{I}_P)))$.

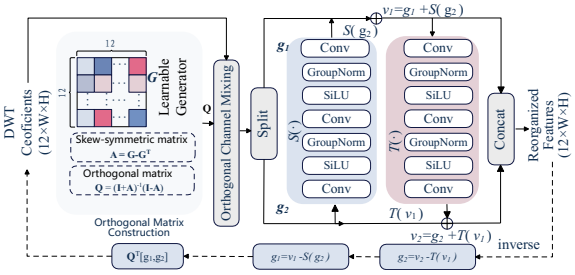


Figure 3: The structure of Orthogonal Additive Coupling Block (OACB).

The projection $\mathcal{P}(\cdot)$ consists of four Orthogonal Additive Coupling Blocks (OACBs), as illustrated in Fig. 3. Each OACB combines orthogonal channel mixing (Gomez et al., 2017; Kingma & Dhariwal, 2018) with additive coupling (Dinh et al., 2014) to reversibly reorganize information across Haar subbands. Given the input features, a learnable orthogonal matrix $\boldsymbol{Q}$ first mixes information across channels. The mixed features are then split into $\mathbf{g}_1$ and $\mathbf{g}_2$, followed by an additive coupling transformation:

$$\mathbf{v}_1 = \mathbf{g}_1 + \mathcal{S}(\mathbf{g}_2), \qquad \mathbf{v}_2 = \mathbf{g}_2 + \mathcal{T}(\mathbf{v}_1), \tag{1}$$

where $\mathcal{S}(\cdot)$ and $\mathcal{T}(\cdot)$ are nonlinear transformations. The outputs $\mathbf{v}_1$ and $\mathbf{v}_2$ are concatenated to form the reorganized features. As shown in Fig. 3, the inverse OACB (IOACB) reverses the additive coupling and then applies inverse channel mixing using $\boldsymbol{Q}^T$ to recover the input features. The construction of $\boldsymbol{Q}$ is provided in Appendix A.1.1.

Role-Aware Learning. To learn what recovery information each role should contain, we apply asymmetric recovery constraints during training. Specifically, the asymmetric recovery constraints are implemented through two decoding paths:

$$\widehat{\boldsymbol{I}}_{\text{base}} = \mathcal{D}(\boldsymbol{C}_{\text{base}}, \mathbf{0}, \mathbf{0}), \qquad \widehat{\boldsymbol{I}}_{\text{base+enh}} = \mathcal{D}(\boldsymbol{C}_{\text{base}}, \boldsymbol{C}_{\text{enh}}, \mathbf{0}), \tag{2}$$

where $\mathcal{D}(\cdot)$ denotes the inverse projection followed by the inverse Haar transform, and both outputs are supervised toward the $\boldsymbol{I}_O$. The $\boldsymbol{C}_{\text{base}}$ participates in both decoding paths and is required to recover $\boldsymbol{I}_O$ independently. This encourages it to preserve the primary information required for recovery. In contrast, the $\boldsymbol{C}_{\text{enh}}$ participates only in the second path together with the $\boldsymbol{C}_{\text{base}}$. It is therefore encouraged to preserve complementary information that further improves recovery. The $\boldsymbol{C}_{\text{aux}}$ is excluded from both decoding paths and additionally regularized toward zero. Therefore, reference recovery cannot rely on $\boldsymbol{C}_{\text{aux}}$, and recovery information in this component is suppressed. Further details of the asymmetric recovery constraints are provided in Appendix A.1.2.

Hierarchical Cross-Region Backup Mechanism. Local tampering can simultaneously destroy the image content and watermark carried by the same region. Therefore, HCBM spatially separates the watermark from its protected content. It assigns different levels of redundancy based on the contribution of each recovery role. Specifically, $\boldsymbol{C}_{\text{base}}$ is protected by two cross-region backups using diagonal and rotational mappings, producing $\boldsymbol{C}_{\text{base}}^{\text{dia}}$ and $\boldsymbol{C}_{\text{base}}^{\text{rot}}$. Their displacement patterns map recovery information from the same source region to different spatial regions. $\boldsymbol{C}_{\text{enh}}$ is assigned one diagonal backup $\boldsymbol{C}_{\text{enh}}^{\text{dia}}$, while $\boldsymbol{C}_{\text{aux}}$ is discarded. The produced backups $\boldsymbol{C}_{\text{base}}^{\text{dia}}, \boldsymbol{C}_{\text{base}}^{\text{rot}}, \boldsymbol{C}_{\text{enh}}^{\text{dia}}$ form the watermark payload for embedding.

3.2 CONTENT-ADAPTIVE WAVELET EMBEDDING

The produced backups have different embedding requirements due to their different roles. Therefore, we propose CAWE to assign them to different Haar subspaces. Their embedding allocation is further adaptively modulated according to the image content and watermark. As shown in Fig. 2-(b), the backups are first upsampled and concatenated with $\boldsymbol{I}_O$ to form role-specific embedding inputs:

$$\boldsymbol{X}_{\text{base}} = [\boldsymbol{I}_O, \text{Up}([\boldsymbol{C}^{\text{dia}}_{\text{base}}, \boldsymbol{C}^{\text{rot}}_{\text{base}}])], \quad \boldsymbol{X}_{\text{enh}} = [\boldsymbol{I}_O, \text{Up}(\boldsymbol{C}^{\text{dia}}_{\text{enh}})]. \tag{3}$$

Next, two U-Net networks respectively produce the corresponding raw residuals $\boldsymbol{R}^{\text{raw}}_{\text{base}}$ and $\boldsymbol{R}^{\text{raw}}_{\text{enh}}$ from $\boldsymbol{X}_{\text{base}}$ and $\boldsymbol{X}_{\text{enh}}$. According to their recovery roles, the $\boldsymbol{R}^{\text{raw}}_{\text{base}}$ and $\boldsymbol{R}^{\text{raw}}_{\text{enh}}$ are constrained to complementary Haar subspaces. The Base branch is restricted to the level-1 low-frequency subspace, with $\mathcal{S}_{\text{base}} = \{LL_2, LH_2, HL_2, HH_2\}$ obtained by further decomposing LL_1. The Enhancement branch uses the level-1 directional subbands $\mathcal{S}_{\text{enh}} = \{LH_1, HL_1, HH_1\}$.

For each branch $r \in \{\text{base}, \text{enh}\}$, Content-Adaptive Subband Modulation (CASM) predicts spatial weights over its available subbands:

$$\boldsymbol{W}_r = \text{Softmax}_s\left(\text{CASM}_r\left(\text{Down}_r(\boldsymbol{X}_r)\right)\right), \qquad \widetilde{\boldsymbol{H}}_{r,s} = \boldsymbol{W}_{r,s} \odot \boldsymbol{H}_{r,s}, \tag{4}$$

where $\boldsymbol{H}_{r,s}$ denotes the Haar coefficients of the raw residual in subband $s \in \mathcal{S}_r$. The weighted coefficients are transformed back to obtain $\boldsymbol{R}_{\text{base}}$ and $\boldsymbol{R}_{\text{enh}}$. Finally, the watermark residual $\boldsymbol{R}$ is obtained as

$$\boldsymbol{R} = \alpha_{\text{base}}\boldsymbol{R}_{\text{base}} + \alpha_{\text{enh}}\boldsymbol{R}_{\text{enh}}, \tag{5}$$

where α_{base} and α_{enh} control the embedding weights of the two branches.

Finally, the combined $\boldsymbol{R}$ is further scaled to satisfy the maximum perturbation budget $\delta_{\max}$ and added to $\boldsymbol{I}_O$ to obtain $\boldsymbol{I}_W$. In this way, CAWE achieves content-adaptive embedding. Further architecture details of CASM are provided in Appendix A.1.3.

3.3 RELIABILITY-BASED MULTI-BACKUP RECOVERY

Given a received image $\boldsymbol{I}_R$, RMBR localizes tampered regions and recovers their content from the embedded backups. As shown in Fig. 2-(c), RMBR consists of three stages: dual-view decoding, tamper localization, and cross-region reliability propagation and recovery.

Dual-View Decoding. The embedded watermark is first extracted from $\boldsymbol{I}_R$ using a ConvNeXt-Base initialized with pretrained weights to obtain the diagonal and rotational backups. These backups are inversely mapped to their original coordinates, producing the aligned components $\widehat{\boldsymbol{C}}^{\text{rot}}_{\text{base}}$, $\widehat{\boldsymbol{C}}^{\text{dia}}_{\text{base}}$, and $\widehat{\boldsymbol{C}}^{\text{dia}}_{\text{enh}}$. Each Base backup is combined with the $\widehat{\boldsymbol{C}}^{\text{dia}}_{\text{enh}}$ and decoded through the IOACB and IDWT, producing two views for recovery $\boldsymbol{V}_{\text{dia}}$ and $\boldsymbol{V}_{\text{rot}}$.

Tamper Localization. The two decoded views provide independent recovery from different backups, so we first average them to an initial estimate: $\boldsymbol{V}_{\text{avg}} = (\boldsymbol{V}_{\text{dia}} + \boldsymbol{V}_{\text{rot}})/2$. Tampered regions are expected to differ from their recovered content. Therefore, we localize tampering using $\boldsymbol{I}_R$, $\boldsymbol{V}_{\text{avg}}$, and their absolute difference:

$$\boldsymbol{P} = \text{Locator}\left([\boldsymbol{I}_R, \boldsymbol{V}_{\text{avg}}, |\boldsymbol{I}_R - \boldsymbol{V}_{\text{avg}}|]\right), \tag{6}$$

where $\boldsymbol{P}$ is the probability map of manipulation and is binarized to obtain the detection mask $\boldsymbol{M}$.

Cross-Region Reliability Propagation and Recovery. The predicted $\boldsymbol{P}$ is further used to estimate the reliability of each recovery backup. We first derive a spatial confidence map $\boldsymbol{F} = 1 - \text{Down}(\boldsymbol{P})$. Since each recovery view originates from different cross-region backups, their reliability depends on whether the corresponding source locations are tampered. Therefore, we propagate $\boldsymbol{F}$ according to the inverse mapping of each backup:

$$\boldsymbol{F}_k = \text{Up}\left(\text{InverseMap}_k(\boldsymbol{F})\right), \qquad k \in \{\text{dia}, \text{rot}\}. \tag{7}$$

The resulting $\boldsymbol{F}_k$ provides reliability information for views $\boldsymbol{V}_k$.

Based on the reliability information, we further assess the quality of each recovery view. For each $\boldsymbol{V}_k$, we construct a candidate recovery image

$$\boldsymbol{J}_k = (1 - \boldsymbol{P}) \odot \boldsymbol{I}_R + \boldsymbol{P} \odot \boldsymbol{V}_k, \tag{8}$$

Here, $\boldsymbol{P}$ combines the $\boldsymbol{I}_R$ and each recovery view according to the predicted tampering probability. A learnable scoring network then evaluates the quality of each view from $[\boldsymbol{J}_k, \boldsymbol{V}_k, |\boldsymbol{J}_k - \boldsymbol{V}_k|, \boldsymbol{F}_k]$. The predicted scores are combined with $\boldsymbol{F}_k$ and normalized to obtain $\boldsymbol{S}_{\mathrm{dia}}$ and $\boldsymbol{S}_{\mathrm{rot}}$. Then, the two views are combined through reliability-weighted fusion: $\boldsymbol{I}_{\mathrm{ref}} = \boldsymbol{S}_{\mathrm{dia}} \odot \boldsymbol{V}_{\mathrm{dia}} + \boldsymbol{S}_{\mathrm{rot}} \odot \boldsymbol{V}_{\mathrm{rot}}$. Finally, the fused reference is used by a U-Net to reconstruct the localized tampered regions, producing the recovered image $\hat{\boldsymbol{I}}$.

3.4 Training objectives

ProCast is optimized using five grouped objectives that jointly supervise watermark embedding, payload extraction, role-aware learning, manipulation localization, and content recovery:

$$\mathcal{L}_{\mathrm{total}} = \lambda_{\mathrm{emb}}\mathcal{L}_{\mathrm{emb}} + \lambda_{\mathrm{ext}}\mathcal{L}_{\mathrm{ext}} + \lambda_{\mathrm{ref}}\mathcal{L}_{\mathrm{ref}} + \lambda_{\mathrm{loc}}\mathcal{L}_{\mathrm{loc}} + \lambda_{\mathrm{rec}}\mathcal{L}_{\mathrm{rec}}. \tag{9}$$

Here, $\mathcal{L}_{\mathrm{emb}}$ controls watermark distortion and regularizes content-adaptive subband modulation. $\mathcal{L}_{\mathrm{ext}}$ supervises the extraction of the protected Base and Enhancement components. $\mathcal{L}\mathrm{ref}$ supervises role-aware reference decoding and reliability-weighted fusion. Finally, $\mathcal{L}\mathrm{loc}$ supervises manipulation localization, while $\mathcal{L}_{\mathrm{rec}}$ optimizes the final recovery fidelity. Detailed loss formulations and weighting parameters are provided in Appendix A.2.

4 Experimental Results

Implementation Details are provided in Appendix A.3.1.

Cross-Dataset Evaluation. ProCast is only trained on the COCO training set. We evaluate within-dataset performance on COCO Val (Lin et al., 2014) and cross-dataset generalization on ADE20K (Zhou et al., 2017) and CelebA-HQ (Karras et al., 2018). Additional dataset results on ILSVRC (Russakovsky et al., 2015) and ValAGE (Zhang et al., 2024) are provided in the Appendix A.3.2.

Unseen Manipulation Evaluation. ProCast is trained using splicing only and evaluated on unseen conventional and AIGC-based manipulations. The unseen manipulations include copy-move, diffusion-based generative inpainting using SDXL (Podell et al., 2024) and HD-Painter (Manukyan et al., 2025), and GAN-based deepfake manipulation using E4S (Liu et al., 2023). Results on the seen splicing manipulation are provided in Appendix A.3.2.

Baselines. We compare ProCast with a broad range of existing methods. For self-recovery, we compare general-image methods, including Imuge+ (Ying et al., 2023), SSEM-HIR (Liu et al., 2025), and ReImage (Kim & Seo, 2026), as well as face-specific recovery methods, including RecoverMark (An et al., 2026) and VeriFi (Yu et al., 2026). We further compare localization-only methods, including EditGuard (Zhang et al., 2024), StableGuard (Yang et al., 2025), and OmniGuard (Zhang et al., 2025). For methods with publicly available code, we use their official implementations and evaluate them on the same manipulated images under the same degradation settings. Since the implementation of VeriFi is not publicly available, its visual results are taken from the original paper.

Evaluation Metrics. We use F1-score, IoU, and AUC to evaluate tampering localization. PSNR, SSIM, and LPIPS are used to evaluate the quality of watermarked images and recovered images.

4.1 Comparison with Self-Recovery methods

Within-Domain Self-Recovery Comparison under Unseen manipulations. We first compare ProCast with existing self-recovery methods on COCO under two unseen manipulations: generative inpainting using HD-Painter and conventional copy-move. Table 1 reports watermark imperceptibility, tampering localization accuracy, and recovery fidelity. As shown, ProCast achieves a better trade-off between watermark imperceptibility and recovery fidelity. It achieves significantly higher recovery fidelity in terms of PSNR and SSIM and more accurate tampering localization, while introducing the least perceptible watermark distortion.

Cross-Domain Self-Recovery Comparison under Unseen Manipulation. Table 2 evaluates cross-dataset generalization on ADE20K and CelebA-HQ, representing general-image and face domains, respectively. Two unseen AIGC manipulations are evaluated: SDXL-based generative inpainting and E4S-based deepfake manipulation.

Table 1: Self-recovery evaluation in a *Within-Dataset* setting (COCO) under unseen AI inpainting (HD-Painter), and conventional copy-move. SSEM-HIR achieves slightly better LPIPS, but this comes at the cost of considerably lower PSNR and SSIM.

Methods	Venue	Watermark Imperceptibility			Localization						Self-Recovery					
					HD-Painter			Copy-Move			HD-Painter			Copy-Move		
		PSNR	SSIM	LPIPS	F1	AUC	IoU	F1	AUC	IoU	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Imuge+	TPAMI'23	32.10	0.924	0.033	0.89	0.9962	0.82	0.933	0.998	0.882	25.31	0.84	0.150	25.84	0.83	0.138
SSEM-HIR	TIFS'25	29.54	0.813	**0.009**	0.97	0.9994	0.94	0.974	**0.999**	0.953	25.90	0.79	0.034	26.19	0.79	**0.032**
ReImage	CVPRF'26	36.54	0.941	0.019	0.95	0.9985	0.91	0.178	0.608	0.109	32.18	0.92	0.045	22.10	0.83	0.135
Ours		**40.53**	**0.982**	0.010	**0.99**	**0.9999**	**0.98**	**0.983**	**0.999**	**0.972**	**34.22**	**0.95**	**0.031**	**34.53**	**0.95**	0.033

Table 2: Self-recovery evaluation in a *Cross-Dataset* setting (ADE20K and CelebA-HQ) under unseen AI manipulations: SDXL inpainting and E4S Deepfake manipulation. SSEM-HIR achieves slightly better LPIPS, but this comes at the cost of considerably lower PSNR and SSIM.

Methods	Watermark Imperceptibility						Localization						Self-Recovery					
	ADE20K			CelebA-HQ			ADE20K (SDXL)			CelebA-HQ (E4S)			ADE20K (SDXL)			CelebA-HQ (E4S)		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	F1	AUC	IoU	F1	AUC	IoU	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Imuge+	35.14	0.922	0.076	34.15	0.922	0.039	0.761	0.9686	0.6763	0.93	0.98	0.89	28.07	0.8555	0.154	22.81	0.78	0.25
SSEM-HIR	31.26	0.848	**0.006**	29.19	0.811	**0.010**	0.8966	0.9955	0.822	0.95	0.97	0.90	28.73	0.8425	**0.040**	19.45	0.72	0.20
ReImage	38.31	0.939	0.028	37.94	0.941	0.026	0.9717	0.9976	0.9472	**0.97**	**0.99**	**0.95**	34.16	0.925	0.054	32.51	0.91	0.09
Ours	**42.29**	**0.974**	0.018	**41.58**	**0.980**	0.018	**0.9947**	**0.9999**	**0.9895**	0.95	**0.99**	0.92	**36.5**	**0.9549**	**0.040**	**35.19**	**0.96**	**0.05**

1) *Watermark Imperceptibility*. ProCast consistently achieves the **highest PSNR and SSIM**, with an average PSNR above 40 dB. Please note that SSEM-HIR's slightly higher LPIPS comes with lower PSNR and SSIM and more noticeable image distortions, due to its use of perceptual loss to explicitly optimize feature-level similarity. Fig. 4 further visualizes the embedding distortion by amplifying the difference between $\boldsymbol{I}_O$ and $\boldsymbol{I}_W$ by $5\times$. ProCast shows the smallest visible perturbations, consistent with the quantitative results. For *tampering localization*, ProCast maintains high accuracy under the cross-dataset setting, achieving an F1 score above 0.99 on ADE20K with SDXL. The accuracy is slightly lower on CelebA-HQ with E4S because some GT masks include visually unchanged regions, such as parts of the ears. These regions remain visually consistent with the original content and may be missed by ProCast, but have limited impact on subsequent recovery.

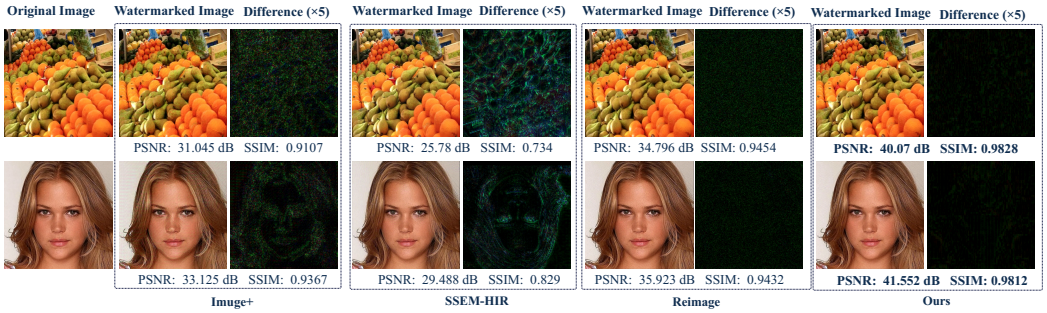
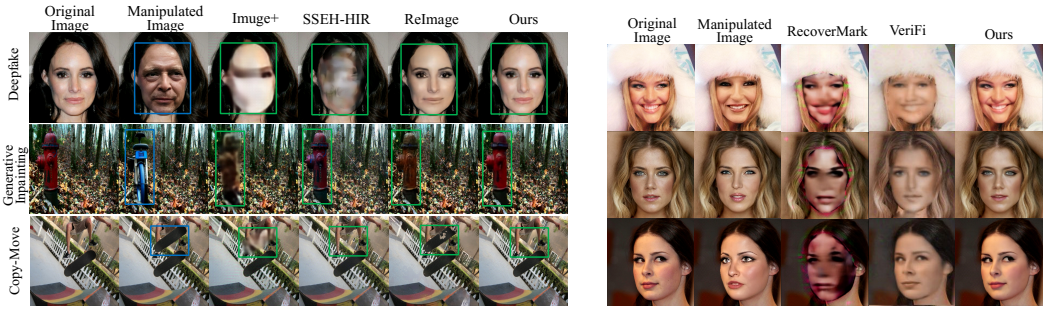


Figure 4: Visual comparison of existing self-recovery methods on watermark imperceptibility.

2) *Recovery Fidelity.* ProCast achieves the **highest PSNR and SSIM** across all evaluated settings. The improvement is particularly evident on CelebA-HQ, where face manipulations are often concentrated around the central facial region. SSEM-HIR relies on a single spatially symmetric backup generated by a 180° rotation, causing the original content and its backup to potentially fall within the same manipulated region. This reduces the recovery information available to the decoder. In contrast, ProCast distributes recovery information across multiple cross-region backups, allowing alternative information to remain available when one backup is corrupted. This reduces the risk of simultaneous corruption and improves recovery fidelity for face manipulations.

Self-Recovery visual comparison under general-purpose unseen AI and copy-move manipulations are shown in Fig. 5a. ProCast shows higher visual consistency with the original image, whereas other methods exhibit more noticeable color and content discrepancies. Fig. 5b compares ProCast with face-specific self-recovery methods, RecoverMark and VeriFi, on CelebA-HQ under unseen deepfake manipulation. Although ProCast is not designed specifically for facial content, we still achieve better visual recovery quality compared to face-specific recovery methods in the face domain.



(a) General-purpose manipulations.

(b) Face-specific manipulations.

Figure 5: Visual comparison of self-recovery methods on recovery fidelity for unseen manipulations.

Evaluation on large model. We further evaluate ProCast on semantic manipulations generated by the recently developed large-scale **Qwen**-Image-Edit-2511 model (Wu et al., 2025). Qwen can introduce explicit semantic changes to image content through text instructions, such as color changes and facial modifications. As shown in Fig. 6, ProCast successfully recovers the original content from these manipulations and produces results that are more consistent with the original image than ReImage (existing SOTA method). This further demonstrates ProCast's applicability to local manipulations generated by recent image editing models.

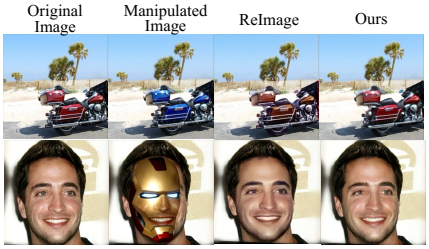


Figure 6: Visual comparison of recovery under semantic manipulations generated by large scale model **Qwen**.

4.2 More Comparison with Localization-Only Methods

To examine whether ProCast maintains competitive watermark imperceptibility and localization accuracy while additionally supporting image recovery, we further compare it with active watermark-based SOTA localization methods, including EditGuard, StableGuard, and OmniGuard. Table 3 reports imperceptibility and localization performance under unseen manipulations in both within- and cross-dataset settings. OmniGuard achieves higher watermark imperceptibility but less consistent localization performance, while EditGuard achieves better localization with lower image quality. In contrast, ProCast maintains high localization accuracy while achieving competitive watermark imperceptibility, demonstrating a better trade-off between the two objectives.

Notably, ProCast shows a particularly clear advantage under the more challenging copy-move. Since the copied content originates from the same watermarked image, it may carry valid watermark information, making localization difficult based on watermark evidence alone. The F1 of EditGuard, StableGuard, and OmniGuard decrease to 0.053, 0.002, and 0.070, respectively, while ProCast achieves 0.983 by additionally exploiting the discrepancy between the received and recovered content. Importantly, ProCast achieves these results **while also supporting self-recovery**, which is NOT provided by the compared localization-only methods.

Table 3: Comparison with localization-only methods under within-domain and cross-domain manipulations. Watermark imperceptibility is averaged over COCO, ADE20K, and CelebA-HQ. OmniGuard achieves higher imperceptibility *at the expense of localization accuracy*. While we achieve a better trade-off between the two objectives and additionally supports self-recovery.

Methods	Venue	Recovery	Avg. Imperceptibility			Within-Domain Localization						Cross-Domain Localization					
						HD-Painter (COCO)			Copy-Move (COCO)			ADE20K (SDXL)			CelebA-HQ (E4S)		
			PSNR	SSIM	LPIPS	F1	AUC	IoU	F1	AUC	IoU	F1	AUC	IoU	F1	AUC	IoU
EditGuard	CVPR'24	×	32.43	0.8470	0.0563	0.91	0.9989	0.84	0.053	0.639	0.050	0.9557	0.9984	0.9167	**0.96**	**0.99**	**0.92**
StableGuard	NeurIPS'25	×	29.12	0.8653	0.0783	0.47	0.9420	0.35	0.002	0.646	0.001	0.586	0.937	0.450	0.78	0.94	0.65
OmniGuard	CVPR'25	×	**50.72**	**0.9937**	**0.0003**	0.83	0.9904	0.76	0.070	0.781	0.04	0.5821	0.9496	0.4863	0.56	**0.99**	0.44
Ours		✔	41.47	0.9787	0.0153	**0.99**	**0.9999**	**0.98**	**0.983**	**0.999**	**0.972**	**0.9947**	**0.9999**	**0.9895**	0.95	**0.99**	**0.92**

4.3 Robustness and Runtime Analysis.

Robustness Comparison. To evaluate robustness against post-processing degradations, we test recovery performance under noise addition (Gaussian and Poisson), JPEG compression, resizing, and blurring (Gaussian and median). As shown in Fig. 7, ProCast maintains competitive recovery performance across different degradation types and strengths. It achieves higher PSNR than the compared methods in most settings.

Runtime Analysis. ProCast maintains a computational cost comparable to existing methods. Detailed comparisons of parameters, FLOPs, memory, and MAdds are provided in Appendix A.3.2.

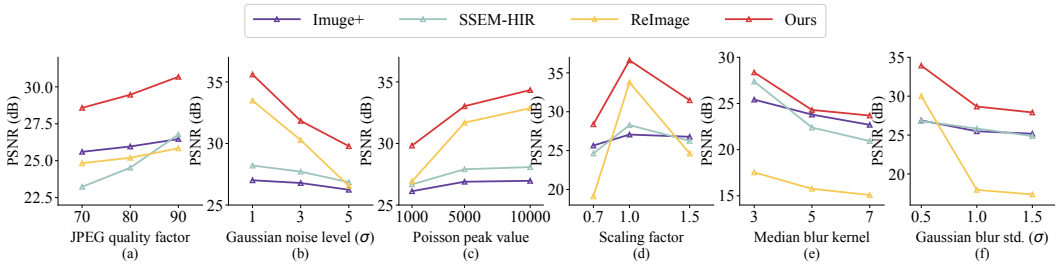


Figure 7: Robustness comparison across different degradation types and strengths

4.4 Ablation Study

Role-Aware Learning. Table 4 shows the effect of role-aware learning on different components. Without role-aware learning, adding Enhancement to Base improves the PSNR of the reference by +6.910 dB (17.968→24.879 dB). With role-aware learning, this gain decreases to only +0.383 dB (24.816→25.199 dB). This shows that Base learns the primary recovery information, while Enhancement provides complementary information. For Auxiliary, its contribution decreases from +0.244 dB to only +0.060 dB, showing that role-aware learning effectively suppresses its recovery information. Embedding Auxiliary would also increase the payload by 20%. Therefore, Auxiliary is discarded to reduce the payload with little loss in recovery quality.

CAWE and RMBR. To evaluate the effectiveness of CAWE, we replace the adaptive subband embedding with fixed subband embedding. As shown in Table 5, this decreases the PSNR of watermarked image by nearly 2 dB and also reduces the recovery PSNR. This shows that adaptive subband embedding benefits both watermark imperceptibility and content recovery. To evaluate RMBR, we replace reliability-weighted fusion with average fusion. The recovery PSNR decreases by nearly 4 dB, showing the benefit of using the reliability of different backups for recovery.

Table 4: Component contribution analysis of RHWG in terms of Ref. PSNR (dB).

Base	Enh.	Aux.	Payload	w/o Role L.	w/ Role L.
✓	×	×	27648	17.968	**24.816**
×	✓	×	6912	7.901	**11.745**
✓	✓	×	34560	24.879	**25.199**
✓	✓	✓	41472	25.123	**25.259**

Table 5: Ablation study of the key designs in CAWE and RMBR.

Design	Ablation Setting	WM PSNR	Ref. PSNR	Rec. PSNR
CAWE	Fixed subband	38.744	23.367	32.482
RMBR	Average fusion	40.002	21.542	30.698
ProCast	**Full**	**40.601**	**24.902**	**34.554**

5 Conclusion and Limitations

We propose ProCast, a proactive framework for image tampering localization and self-recovery. Extensive experiments demonstrate that ProCast achieves high watermark imperceptibility, accurate tampering localization, and high-fidelity content recovery across diverse manipulation types and datasets. A current limitation is that ProCast is designed primarily for local manipulations and cannot effectively handle global manipulations that may destroy all the embedded recovery information. In future work, we will explore more robust self-recovery mechanisms for such scenarios.

AI USE STATEMENT

In this work, we used generative AI tools to generate AIGC-based image manipulations for experimental evaluation, including generative inpainting and deepfake manipulation. Generative AI tools were not used for method or experiment design, method implementation, result interpretation, or theoretical and mathematical development. Large language models (LLMs) were used only for grammar checking and language polishing to improve the readability of the paper. All AI-assisted content and experimental images were reviewed by the authors, who take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed information to support the reproducibility of our work. The model architecture and training objectives are described in the main paper, with additional method details and loss formulations provided in Appendix A.1 and Appendix A.2, respectively. The experimental settings, including datasets, manipulation generation, degradation settings, and training configurations, are detailed in Appendix A.3. Additional quantitative and qualitative results are also provided in the Appendix. Anonymous source code is also provided in `https://anonymous.4open.science/r/ProCast--ICLR-2027-C590/full_system.py`

REFERENCES

Haonan An, Xiaohui Ye, Guang Hua, Yihang Tao, Hangcheng Cao, Xiangyu Yu, and Yuguang Fang. Recovermark: Robust watermarking for localization and recovery of manipulated faces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8587–8597, June 2026.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network: Backpropagation without storing activations. *Advances in neural information processing systems*, 30, 2017.

Jiahao Huang, Tong Liu, Fangyuan Lei, Xiuli Bi, and Xiaochen Yuan. Omniaggregation topology perception networks for aigc-manipulation detection and localization. *IEEE Transactions on Industrial Informatics*, 22(9):8608–8619, 2026. doi: 10.1109/TII.2026.3694161.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Hk99zCeAb`.

Minyoung Kim and Paul Hongsuck Seo. Robust image self-recovery against tampering using watermark generation with pixel shuffling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 8877–8886, June 2026.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

Tong Liu and Xiaochen Yuan. Adaptive feature calculation and diagonal mapping for successive recovery of tampered regions. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7):2617–2630, 2021. doi: 10.1109/TCSVT.2020.3032455.

Tong Liu, Xiaochen Yuan, Wei Ke, Chan-Tong Lam, Sio-Kei Im, and Pedro Martins. A symmetric self-embedding mechanism for high-fidelity image recovery against tampering. *IEEE Transactions on Information Forensics and Security*, 20:12857–12870, 2025. doi: 10.1109/TIFS.2025.3638170.

Zhian Liu, Maomao Li, Yong Zhang, Cairong Wang, Qi Zhang, Jue Wang, and Yongwei Nie. Fine-grained face swapping via regional gan inversion. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8578–8587. IEEE, 2023.

Hayk Manukyan, Andranik Sargsyan, Barsegh Atanyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Hd-painter: High-resolution and prompt-faithful text-guided image inpainting with diffusion models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 96301–96330, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/f0156a82b6af6a4e838923ce9c124424-Paper-Conference.pdf.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

Haoxin Yang, Bangzhen Liu, Xuemiao Xu, Cheng Xu, Yuyang Yu, Zikai Huang, Yi Wang, and Shengfeng He. Stableguard: Towards unified copyright protection and tamper localization in latent diffusion models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 14692–14720. Curran Associates, Inc., 2025. doi: 10.52202/085713-0495. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/15972a9575e0f03bf82f00aebeb40774-Paper-Conference.pdf.

Qichao Ying, Hang Zhou, Zhenxing Qian, Sheng Li, and Xinpeng Zhang. Learning to immunize images for tamper localization and self-recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13814–13830, 2023.

Peipeng Yu, Jinfeng Xie, Chengfu Ou, Xiaoyu Zhou, Jianwei Fei, Yunshu Dai, Zhihua Xia, and Chip Hong Chang. High-fidelity face content recovery via tamper-resilient versatile watermarking, 2026. URL https://arxiv.org/abs/2603.23940.

Xuanyu Zhang, Runyi Li, Jiwen Yu, Youmin Xu, Weiqi Li, and Jian Zhang. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11964–11974. IEEE, 2024.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3008–3018, 2025. doi: 10.1109/CVPR52734.2025.00286.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5122–5130, 2017. doi: 10.1109/CVPR.2017.544.

A APPENDIX

A.1 ADDITIONAL METHOD DETAILS

A.1.1 REVERSIBLE PROJECTION

The reversible projection $\mathcal{P}(\cdot)$ consists of four OACBs. Each OACB sequentially performs orthogonal channel mixing and additive coupling, as illustrated in Fig. 3.

Orthogonal Matrix Construction. The first operation reorganizes information across different Haar subbands using a learnable orthogonal matrix. Specifically, a learnable matrix $\boldsymbol{G}$ is introduced to construct a skew-symmetric matrix $\boldsymbol{A} = \boldsymbol{G} - \boldsymbol{G}^T$. The orthogonal matrix $\boldsymbol{Q}$ is then obtained using the Cayley transform:

$$\boldsymbol{Q} = (\boldsymbol{I} + \boldsymbol{A})^{-1}(\boldsymbol{I} - \boldsymbol{A}), \tag{10}$$

where $\boldsymbol{I}$ denotes the identity matrix. Since $\boldsymbol{A}^T = -\boldsymbol{A}$, the orthogonality of $\boldsymbol{Q}$ can be verified as

$$\boldsymbol{Q}^T\boldsymbol{Q} = (\boldsymbol{I} + \boldsymbol{A})(\boldsymbol{I} - \boldsymbol{A})^{-1}(\boldsymbol{I} + \boldsymbol{A})^{-1}(\boldsymbol{I} - \boldsymbol{A}) \quad = \boldsymbol{I}. \tag{11}$$

Therefore, $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^T$, allowing the channel mixing operation to be exactly reversed.

The mixed features are then passed to the additive coupling layer. They are first divided into two groups, $\mathbf{g}_1$ and $\mathbf{g}_2$. Two nonlinear transformations, $\mathcal{S}(\cdot)$ and $\mathcal{T}(\cdot)$, exchange information between the two groups according to Eq. (1).

Inverse OACB. Given the outputs $\mathbf{v}_1$ and $\mathbf{v}_2$, this process can be inverted in reverse order:

$$\mathbf{g}_2 = \mathbf{v}_2 - \mathcal{T}(\mathbf{v}_1), \qquad \mathbf{g}_1 = \mathbf{v}_1 - \mathcal{S}(\mathbf{g}_2). \tag{12}$$

The reconstructed groups are finally transformed by $\boldsymbol{Q}^T$ to recover the input features. Since both operations are invertible, each OACB is invertible, and stacking four OACBs forms the reversible projection $\mathcal{P}(\cdot)$.

A.1.2 DETAILS OF ROLE-AWARE LEARNING

As described in the main paper, the recovery roles in RACD are jointly learned through asymmetric decoding. Here, we provide the training constraints that guide the information distribution among the three components.

Base-Only Supervision. Since Base is protected by two cross-region backups, both views are required to independently recover the original image:

$$\mathcal{L}_{\text{base}} = \frac{1}{2} \sum_{v \in \{\text{dia},\text{rot}\}} \left\| \mathcal{D}(\boldsymbol{C}^v_{\text{base}}, \mathbf{0}, \mathbf{0}) - \boldsymbol{I}_O \right\|_1 . \tag{13}$$

Since Enhancement and Auxiliary are unavailable in this decoding path, Base is encouraged to carry the primary recovery information.

Joint Supervision. Each Base view is further decoded together with Enhancement:

$$\widehat{\boldsymbol{I}}^v_{\text{base+enh}} = \mathcal{D}(\boldsymbol{C}^v_{\text{base}}, \boldsymbol{C}_{\text{enh}}, \mathbf{0}), \qquad v \in \{\text{dia}, \text{rot}\}. \tag{14}$$

The two decoded references are supervised by

$$\mathcal{L}_{\text{joint}} = \frac{1}{2} \sum_{v \in \{\text{dia},\text{rot}\}} \left\| \widehat{\boldsymbol{I}}^v_{\text{base+enh}} - \boldsymbol{I}_O \right\|_1 . \tag{15}$$

Since Base is already required to recover the primary information independently, Enhancement is encouraged to provide complementary information that further improves reference quality.

Auxiliary Suppression. Auxiliary is excluded from all reference decoding paths and is further regularized by

$$\mathcal{L}_{\text{aux}} = \left\| \boldsymbol{C}_{\text{aux}} \right\|_1 . \tag{16}$$

Reference recovery therefore cannot depend on Auxiliary, while its recovery information is explicitly suppressed.

Role Formation. The three roles are jointly formed rather than assigned by separate losses. Base-only supervision encourages Base to carry the primary recovery information. Joint supervision encourages Enhancement to complement Base. Excluding and suppressing Auxiliary makes it suitable for discarding. These constraints are included in the overall reference objective with $\lambda_{\text{joint}} = 1$, $\lambda_{\text{base}} = 0.5$, and $\lambda_{\text{aux}} = 0.05$.

A.1.3 Details of CASM

The Base and Enhancement branches use two independent CASMs with the same architecture but separate parameters. Each CASM consists of a 3×3 convolution, GroupNorm, SiLU, a 3×3 depthwise convolution, GroupNorm, SiLU, and a 1×1 convolution, with 16 intermediate channels. The Base CASM outputs four 64×64 weight maps for $\{LL_2, LH_2, HL_2, HH_2\}$, while the Enhancement CASM outputs three 128×128 maps for $\{LH_1, HL_1, HH_1\}$. Softmax is applied along the subband dimension so that the weights at each spatial location sum to one.

The final convolution is initialized with zero weights and log-prior biases, using $\boldsymbol{\pi}_{\text{base}} = [0.15, 0.30, 0.30, 0.25]$ and $\boldsymbol{\pi}_{\text{enh}} = [0.35, 0.35, 0.30]$. The allocation maps are regularized by

$$\mathcal{L}_{\text{budget}} = \sum_r \left\| \text{Mean}(\boldsymbol{W}_r) - \boldsymbol{\pi}_r \right\|_1, \qquad \mathcal{L}_{\text{smooth}} = \sum_r \text{TV}(\boldsymbol{W}_r), \tag{17}$$

where $r \in \{\text{base}, \text{enh}\}$. The two terms constrain the average subband allocation and spatial variation, respectively.

A.1.4 Reliability Scoring.

The two recovery views share the same scoring network. For each view $k \in \{\text{dia}, \text{rot}\}$, the network takes $[\boldsymbol{J}_k, \boldsymbol{V}_k, |\boldsymbol{J}_k - \boldsymbol{V}_k|, \boldsymbol{F}_k]$ as input and predicts a spatial score s_k. It consists of two 3×3 convolutional layers with 32 channels, each followed by GroupNorm and SiLU, and a final 1×1 convolution.

The score is combined with the propagated reliability $\boldsymbol{F}_k$ to obtain the fusion weight:

$$\boldsymbol{S}_k = \text{Softmax}_k \left[s_k + \log \max(\boldsymbol{F}_k, \epsilon) \right], \qquad \epsilon = 10^{-6}. \tag{18}$$

For supervision, we compute the local reconstruction error $\boldsymbol{E}_k$ between $\boldsymbol{V}_k$ and $\boldsymbol{I}_O$ using a 3×3 average filter. The oracle fusion weight is defined as

$$\boldsymbol{S}_k^* = \text{Softmax}_k \left[-\boldsymbol{E}_k / \tau + \log \max(\boldsymbol{F}_k^{\text{gt}}, \epsilon) \right], \qquad \tau = 0.05, \tag{19}$$

where $\boldsymbol{F}_k^{\text{gt}}$ is obtained from the GT manipulation mask using the same inverse mapping. The selection loss is

$$\mathcal{L}_{\text{sel}} = -\sum_k \boldsymbol{S}_k^* \log \boldsymbol{S}_k. \tag{20}$$

The loss is applied only where at least one recovery view is reliable and the local errors of the two views are sufficiently different.

A.2 Training Objectives

Embedding Loss. The embedding loss constrains watermark distortion and regularizes the content-adaptive subband modulation:

$$\mathcal{L}_{\text{emb}} = \text{MSE}(\boldsymbol{I}_O, \boldsymbol{I}_W) + 0.01\mathcal{L}_{\text{zero}} + 0.05\mathcal{L}_{\text{budget}} + 0.01\mathcal{L}_{\text{smooth}}, \tag{21}$$

where $\mathcal{L}_{\text{zero}}$ minimizes the absolute spatial mean of the watermark residual $\boldsymbol{R}$. $\mathcal{L}_{\text{budget}}$ and $\mathcal{L}_{\text{smooth}}$ are defined in (17).

Extraction Loss. The extraction loss supervises the recovery information extracted from the received image:

$$\mathcal{L}_{\text{ext}} = \mathcal{L}_{\text{base}}^{\text{ext}} + 0.5\mathcal{L}_{\text{enh}}^{\text{ext}} + 0.05\mathcal{L}_{\text{grad}}^{\text{ext}}. \tag{22}$$

Here, $\mathcal{L}_{\text{base}}^{\text{ext}}$ and $\mathcal{L}_{\text{enh}}^{\text{ext}}$ are reliability-weighted ℓ_1 losses between the extracted and target Base and Enhancement components, respectively. The losses of the two Base views are computed separately

and then averaged. $\mathcal{L}_{\text{grad}}^{\text{ext}}$ measures the ℓ_1 difference between the horizontal and vertical gradients of the extracted and target Base components.

Reference Loss. The reference loss supervises role-aware decoding, recovery candidates, and reliability-weighted fusion:

$$\begin{aligned}\mathcal{L}_{\text{ref}} = \mathcal{L}_{\text{fused}}^{\text{ref}} + \mathcal{L}_{\text{joint}} + 0.5\mathcal{L}_{\text{base}} + 0.05\mathcal{L}_{\text{aux}} \\ + 0.5\mathcal{L}_{\text{cand}}^{\text{ref}} + 0.1\mathcal{L}_{\text{sel}}.\end{aligned} \tag{23}$$

The role-aware terms $\mathcal{L}_{\text{base}}$, $\mathcal{L}_{\text{joint}}$ and $\mathcal{L}_{\text{aux}}$ are defined in (13), (15), and (16).After manipulation, $\mathcal{L}_{\text{cand}}^{\text{ref}}$ applies a GT reliability-weighted ℓ_1 loss between each decoded recovery view and $\boldsymbol{I}_O$, while $\mathcal{L}_{\text{fused}}^{\text{ref}}$ applies a GT reliability-weighted ℓ_1 loss between the fused reference $\boldsymbol{I}_{\text{ref}}$ and $\boldsymbol{I}_O$. $\mathcal{L}_{\text{sel}}$ is defined in (20)

Localization Loss. The localization loss supervises the predicted manipulation map using the GT mask $\boldsymbol{M}_{gt}$:

$$\mathcal{L}_{\text{loc}} = \mathcal{L}_{\text{Dice}} + 0.25\mathcal{L}_{\text{BCE}}, \tag{24}$$

where $\mathcal{L}_{\text{Dice}}$ and $\mathcal{L}_{\text{BCE}}$ denote the Dice loss and binary cross-entropy loss, respectively.

Recovery Loss. The recovery loss supervises the final recovered image:

$$\mathcal{L}_{\text{rec}} = \|\hat{\boldsymbol{I}} - \boldsymbol{I}_O\|_1 + 2\mathcal{L}_{\text{masked}} + 0.1\mathcal{L}_{\text{Y-grad}}. \tag{25}$$

Here, $\mathcal{L}_{\text{masked}}$ applies an ℓ_1 loss within the manipulated regions, while $\mathcal{L}_{\text{Y-grad}}$ constrains the luminance gradients around these regions to preserve local edges and textures.

Overall, the weights for the embedding, extraction, reference, localization, and recovery losses are set to 0.15, 1.25, 1.50, 1.00, and 1.00, respectively.

A.3 Additional Experimental Details

A.3.1 Implementation Details

Implementation Details. All experiments are implemented in PyTorch on a single GeForce RTX 3090. Images are resized to 256×256, and the resolution of $\boldsymbol{I}_P$ is 96×96. The $\boldsymbol{C}_{\text{base}}$, $\boldsymbol{C}_{\text{enh}}$, and $\boldsymbol{C}_{\text{aux}}$ contain 6, 3, and 3 channels. The embedding weights are set to $\alpha_{\text{base}} = 0.65$ and $\alpha_{\text{enh}} = 0.35$, with $\delta_{\max} = 0.04$. We use AdamW with a learning rate of 5×10^{-5} and a weight decay of 1×10^{-4}. The batch size is 4. The model is first pretrained for 20 epochs without degradations, followed by 150 epochs of fine-tuning with progressively introduced JPEG, noise, blur, and resizing.

Database. We use 11K images from COCO for training. For evaluation, we use the full COCO validation set of 5,000 images and all 1,000 images from ValAGE. We randomly select 1,000 images from ILSVRC and CelebA-HQ, and 500 images from ADE20K. During training, we select object masks covering 10%–35% of the image from the COCO segmentation annotations. The same criterion is used to select masks from the ADE20K annotations for evaluation. For CelebA-HQ and ValAGE, we directly use the provided manipulation masks.

Manipulation Generation. We evaluate ProCast under both conventional and AIGC-based manipulations. All manipulations are applied to the watermarked image, and the corresponding manipulated regions are used as ground-truth masks for localization.

1) *Splicing.* A region from another donor image is pasted into the watermarked image. Only the pasted region is considered manipulated, and its binary mask is used as the ground truth for localization.

2) *Copy-Move.* A region is copied from the watermarked image and pasted into another non-overlapping location within the same image. The source region remains unchanged, while only the pasted destination region is considered manipulated and used as the ground truth.

3) *AIGC Inpainting.* We use SDXL Inpainting and HD-Painter to generate local AIGC-based manipulations. The watermarked image, a binary editing mask, and a text prompt are provided to the generative model. The manipulation is restricted to the masked region, while the content outside the mask remains unchanged.

4) *E4S Deepfake Manipulation.* We use E4S to generate deepfake face manipulations. The watermarked target face and another donor face are provided to E4S to transfer facial information from the donor to the target. The manipulation is restricted to the corresponding face mask, which is also used as the ground truth for localization.

5) *Qwen-Based Editing.* We use Qwen-Image-Edit-2511 to generate diverse semantic manipulations. The manipulation is restricted to the specified editing mask, while the content outside the mask remains unchanged. The editing mask is used as the ground truth for localization.

Degradation Settings During training, multiple degradations can be applied jointly. Differentiable JPEG compression is applied with a probability progressively increased from 0.2 to 0.4, with quality factors randomly sampled from 60 to 95. Gaussian and median blur are applied with probabilities of 0.2 and 0.15, respectively. Gaussian blur uses a kernel size of 3 with $\sigma \in [0.1, 0.5]$, while median blur uses a kernel size of 3, 5, and 7. Gaussian noise is applied with a probability of 0.15 and $\sigma = 0.05$. Resizing is applied with a probability of 0.15 using a scaling factor of 70% or 150%. Poisson noise is not used during training.

For robustness evaluation, we use real JPEG compression with quality factors of 70, 80, and 90. Gaussian and median blur are evaluated at different strengths. Gaussian blur uses a kernel size of 7 with $\sigma = 0.5$, 1.0, and 1.5, while median blur uses kernel sizes of 3, 5, and 7. Gaussian noise is evaluated with $\sigma = \{1, 3, 5\}/255$, and Poisson noise with scales of 1000, 5000, and 10000. Resizing is evaluated at 70% and 150%, with 100% shown as the no-resizing reference.

A.3.2 Additional Experimental Results

Recovery under Different Tampering Ratios. We further evaluate localization and recovery performance on the ValAGE dataset with tampering ratios ranging from 5% to 35% in steps of 5%. As shown in Fig. 8, localization remains consistently accurate across all tampering ratios, with F1 scores above 0.998. Recovery quality gradually decreases as the tampering ratio increases, with PSNR dropping from 36.12 dB at 5% to 28.40 dB at 35%. This is expected because larger tampered regions destroy more embedded recovery information.

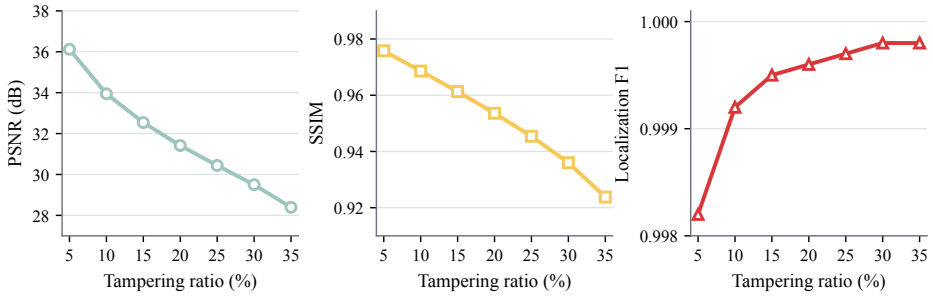


Figure 8: Localization and recovery performance under different tampering ratios.

Table 6: Recovery quality comparison within tampered regions using masked PSNR (M-PSNR). The best results are highlighted in bold.

Method	Conventional Manipulation		AIGC Manipulation		
	Splicing	Copy-Move	SDXL	HD-Painter	E4S
IMUGE+	12.950	18.840	19.490	17.200	18.660
SSEM-HIR	22.280	24.380	25.320	22.438	15.150
ReImage	21.570	11.624	26.707	23.800	28.690
Ours	**27.229**	**24.940**	**29.037**	**25.000**	**31.100**

To exclude the influence of watermark distortion on recovery quality, we further report the masked PSNR (M-PSNR), which measures recovery quality only within the tampered regions. As shown in Table 6, our method achieves the highest M-PSNR across all manipulation settings, improving

over the second-best results by 0.56–4.95 dB. These results further demonstrate the recovery quality within the tampered regions.

Computational Efficiency. Table 7 compares the computational complexity and memory consumption of different self-recovery methods. ProCast has 194M parameters and requires 214G FLOPs, while its memory consumption is 743 MB, the lowest among the compared methods.

Table 8 further compares the runtime of different methods on an NVIDIA RTX 3090 GPU. The runtime is separately reported for watermark embedding and the subsequent localization and recovery stages to provide a more detailed comparison of the complete self-recovery pipeline. ProCast requires 14.788 ms for watermark embedding, which is the fastest among the compared methods. Although SSEM-HIR has the lowest runtime for localization and recovery, ProCast achieves the lowest overall runtime of 41.540 ms when all stages are considered. This indicates that the additional role-aware decomposition, content-adaptive embedding, and multi-backup recovery do not result in excessive inference time. Together with the memory results in Table 7, these results demonstrate that ProCast provides an efficient complete pipeline while jointly supporting watermark embedding, tampering localization, and content recovery.

Table 7: Comparison of computational complexity and memory consumption.

Method	FLOPs	Params	Memory	MAdd
IMUGE+	190.88G	32.0M	1580MB	191.03G
ReImage	64.76G	9.94M	4570MB	36.44G
SSEM-HIR	1224G	200M	763MB	612G
Ours	214G	194M	743.06MB	107G

Table 8: Runtime comparison on an NVIDIA RTX 3090 GPU. All runtimes are measured in milliseconds (ms).

Method	Embedding	Localization + Recovery	Total
IMUGE+	31.560	59.444	91.004
ReImage	35.000	129.000	164.000
SSEM-HIR	40.425	12.545	52.970
Ours	14.788	26.752	41.540

Additional Results on ILSVRC and ValAGE

To further evaluate the generalization ability of the proposed method, we conduct additional experiments on ILSVRC and ValAGE. These experiments cover both conventional splicing on ILSVRC and AIGC-based manipulation using SDXL on ValAGE. We report watermark imperceptibility and tampering localization performance in Table 9, and self-recovery performance in Table 10. The proposed method consistently achieves strong localization and recovery performance on both datasets, demonstrating its effectiveness across different image distributions and manipulation types.

Table 9: Tampering localization performance on ILSVRC and ValAGE. The best results are highlighted in bold, and the second-best are underlined.

Methods	Venue	Watermark Imperceptibility						Localization					
		ILSVRC			ValAGE			ILSVRC (Splicing)			ValAGE (SDXL)		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	F1 ↑	AUC ↑	IoU ↑	F1 ↑	AUC ↑	IoU ↑
EditGuard	CVPR'24	32.08	0.868	0.055	32.34	0.865	0.048	0.91	0.998	0.85	<u>0.93</u>	<u>0.9994</u>	0.88
StableGuard	NeurIPS'25	25.82	0.769	0.112	26.14	0.792	0.100	0.52	0.949	0.40	0.57	0.9574	0.45
Imuge+	TPAMI'23	32.73	0.926	0.035	33.04	0.928	0.036	0.94	0.998	0.92	0.76	0.9834	0.68
SSEM-HIR	TIFS'25	28.83	0.802	**0.010**	31.24	0.849	**0.004**	<u>0.96</u>	<u>0.999</u>	<u>0.94</u>	0.90	0.9953	0.84
ReImage	CVPRF'26	<u>37.07</u>	<u>0.944</u>	0.021	<u>37.14</u>	<u>0.941</u>	0.021	0.94	0.997	0.88	0.94	0.9937	<u>0.90</u>
Ours		**40.95**	**0.982**	<u>0.012</u>	**41.01**	**0.981**	<u>0.012</u>	**0.99**	**1.000**	**0.99**	**0.98**	**0.9998**	**0.97**

Additional Visualization Results.

We provide additional qualitative results to complement the quantitative evaluation in the main paper. The results include complete examples of the ProCast pipeline, tampering localization comparisons, and recovery comparisons across different manipulation types.

Table 10: Self-recovery performance on ILSVRC and ValAGE. The best results are highlighted in bold, and the second-best are underlined.

Methods	ILSVRC (Splicing)			ValAGE (SDXL)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Imuge+	25.19	0.82	0.148	26.58	0.85	0.119
SSEM-HIR	27.30	0.81	**0.028**	28.28	0.82	**0.015**
ReImage	<u>30.81</u>	<u>0.91</u>	0.077	<u>33.77</u>	<u>0.92</u>	0.042
Ours	**35.67**	**0.95**	<u>0.035</u>	**36.61**	**0.95**	<u>0.028</u>

Fig. 9 presents complete examples of ProCast under both conventional and AIGC-based manipulations, including splicing, copy-move, generative inpainting, and deepfake. For each example, we show the original image, watermarked image, manipulated image, ground-truth mask, predicted tampering mask, and recovered image. The watermarked images remain visually close to the original images. Meanwhile, the predicted masks closely match the manipulated regions, and the recovered images restore the corresponding original content. These examples visually demonstrate the complete process from watermark embedding to tampering localization and content recovery.

Fig. 10 provides additional localization comparisons with existing methods across different manipulation types. ProCast produces masks that closely follow the ground-truth manipulated regions under both conventional and AIGC-based manipulations. Compared with other methods, it shows fewer missed or incorrectly localized regions, particularly under challenging manipulations such as copy-move. These results are consistent with the localization results reported in the main paper.

Fig. 11 further compares the recovered images produced by different self-recovery methods. Across different manipulation types, ProCast reconstructs the manipulated regions with better consistency with the original content. In comparison, other methods may exhibit incomplete recovery, color differences, or content inconsistencies.

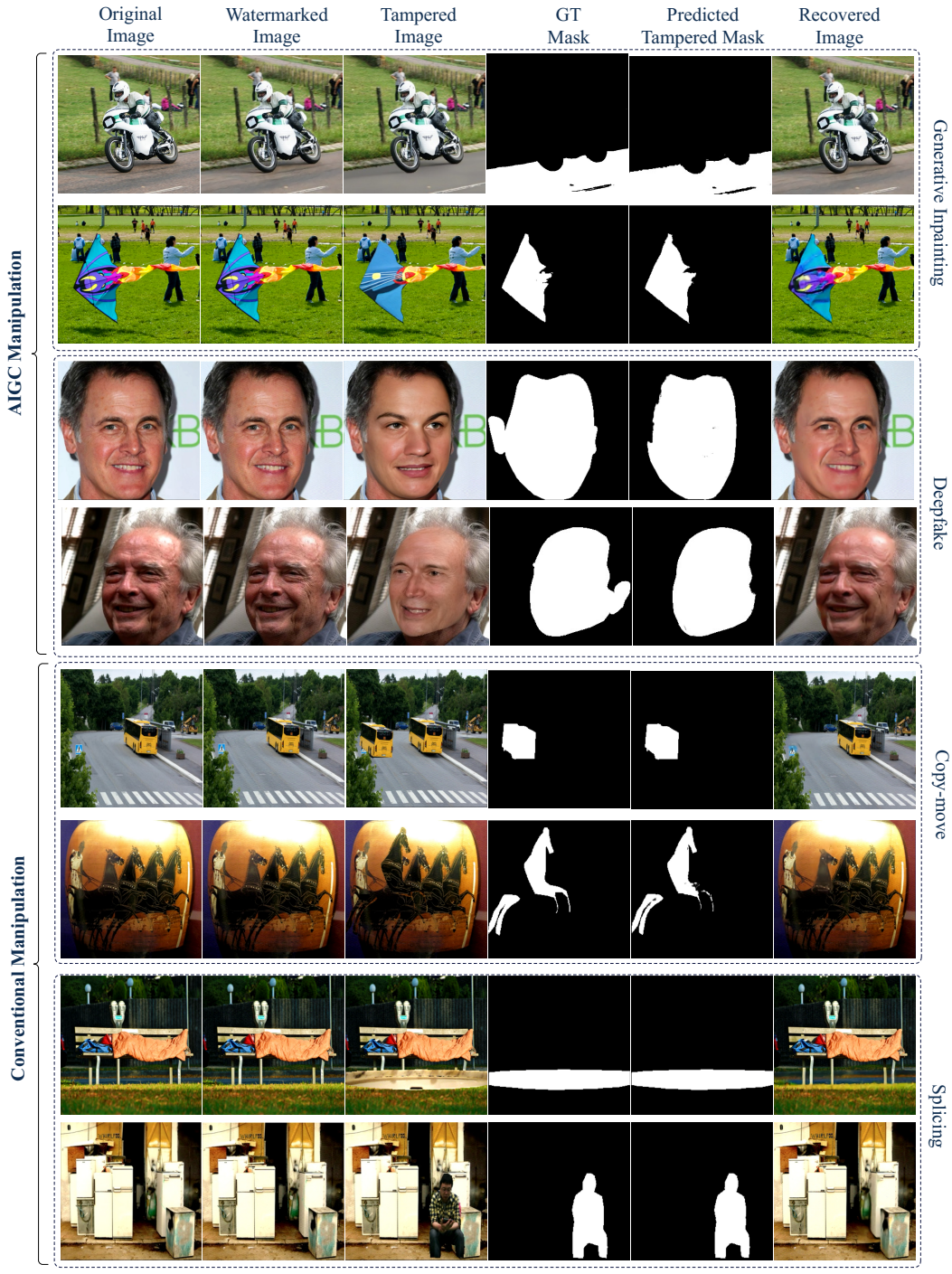


Figure 9: Additional examples of ProCast under different conventional and AIGC-based manipulations. From left to right: original image, watermarked image, tampered image, ground-truth mask, predicted tampering mask, and recovered image.

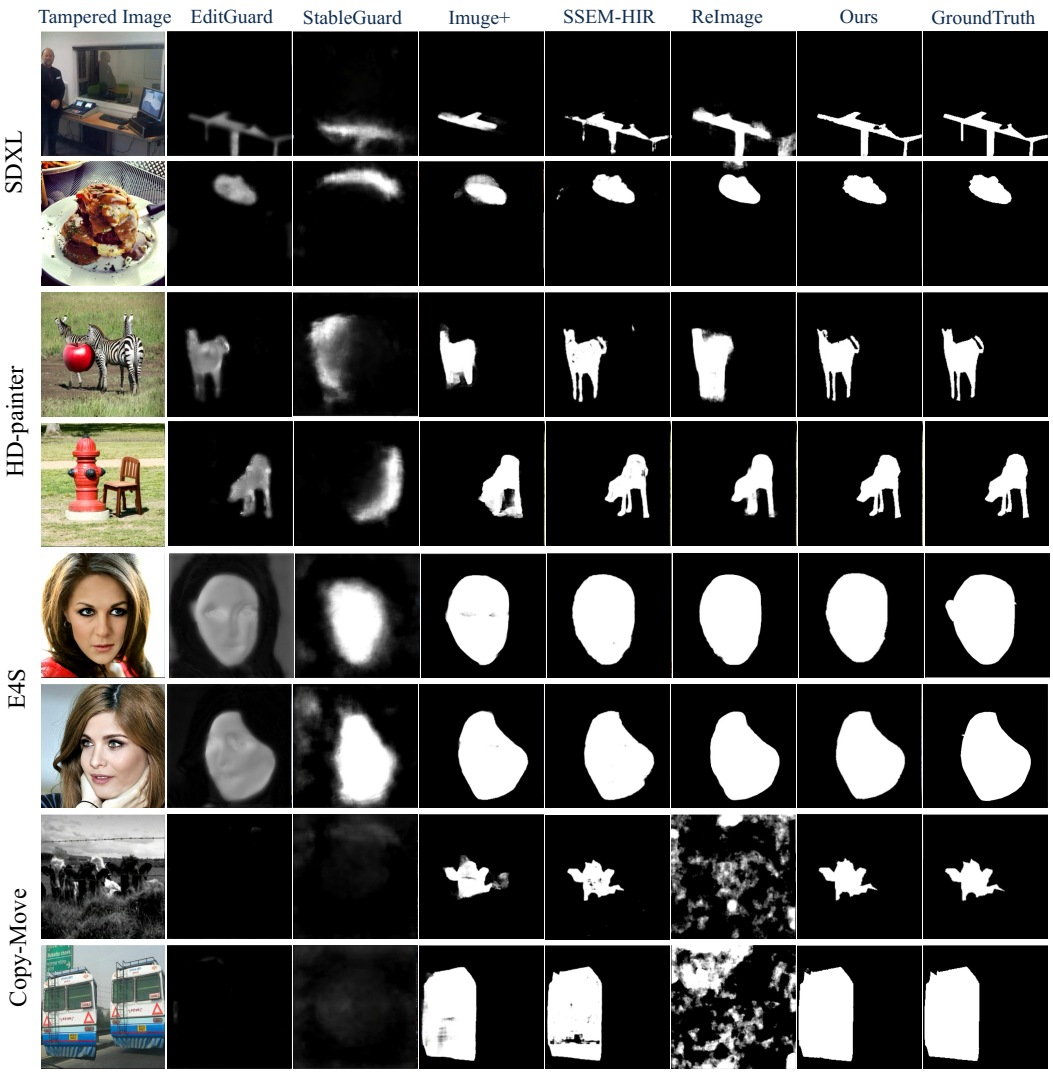


Figure 10: Additional visual comparison of tampering localization across different manipulation types.

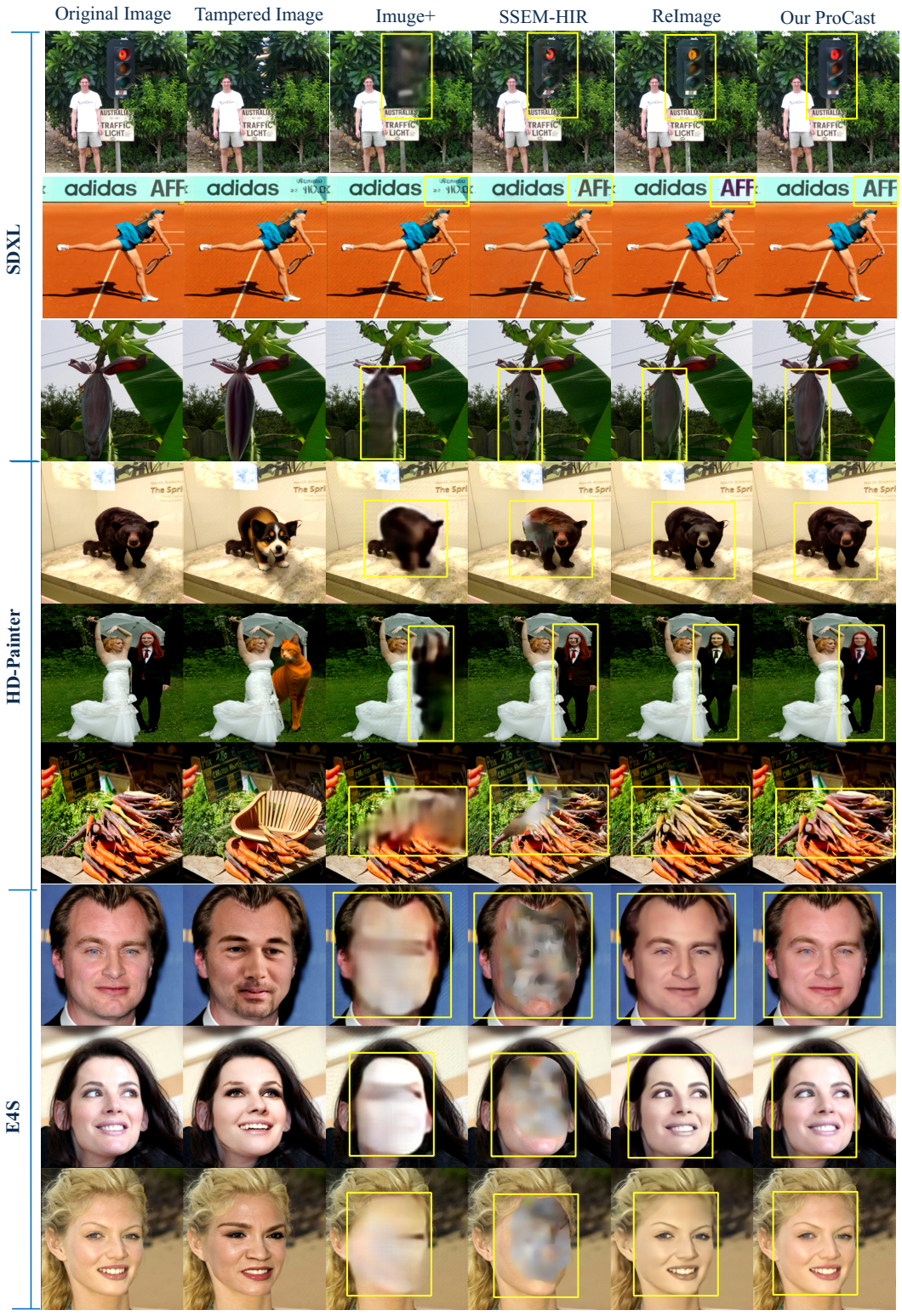


Figure 11: Additional visual comparison of recovery performance across different manipulation types.