SELF-HEALING IMAGES: UNIFYING MANIPULATION DETECTION AND RECOVERY VIA LATENT SELF-EMBEDDING

Anonymous authors
Paper under double-blind review

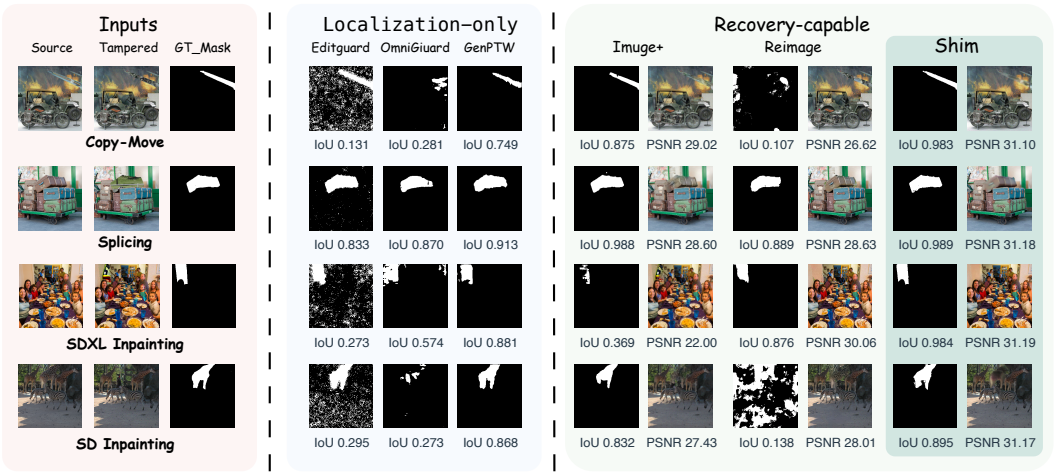


Figure 1: Qualitative comparison across four manipulations. Localization-only methods predict masks, whereas recovery-capable methods reconstruct source content.

ABSTRACT

Recovering overwritten image content requires source information that survives editing. SHIM (**S**elf-**H**ealing **IM**ages) stores this information as a compact latent embedded in the image before distribution. A frozen autoencoder encodes the source, and a spatial permutation redistributes its latent tokens before embedding. The receiver extracts one latent and decodes it twice. Direct decoding exposes manipulation-related inconsistencies; inverse-shuffle decoding produces an estimate aligned with the source. Both views and the received image feed a coarse-to-fine locator. Recovery combines a locally corrected source estimate with a globally restored received image, using fusion weights constrained by the predicted mask. We evaluate copy-move, splicing, generative inpainting, and local blur, with and without channel degradation. Input ablations show that the direct view improves localization when added to the aligned view in the trained locator. The complete recovery module also reduces reconstruction error relative to direct composition. Experiments on COCO, ImageNet-1K, and CelebA-HQ show improved localization and recovery under the tested degradation conditions, with higher average F1, PSNR, and SSIM.

1 INTRODUCTION

Local image editing can replace an object or alter a scene while leaving much of the image intact. Generative editing (Korshunova et al., 2017; Sheynin et al., 2024; Ling et al., 2021; Yu et al., 2025; Fu et al., 2024; Zhao et al., 2024; Kawar et al., 2023) and text-guided synthesis (Rombach et al., 2022; Podell et al., 2023; Esser et al., 2024; Saharia et al., 2022; Betker et al., 2023) make such changes readily accessible. Image manipulation detection and localization (IMDL) seeks

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.

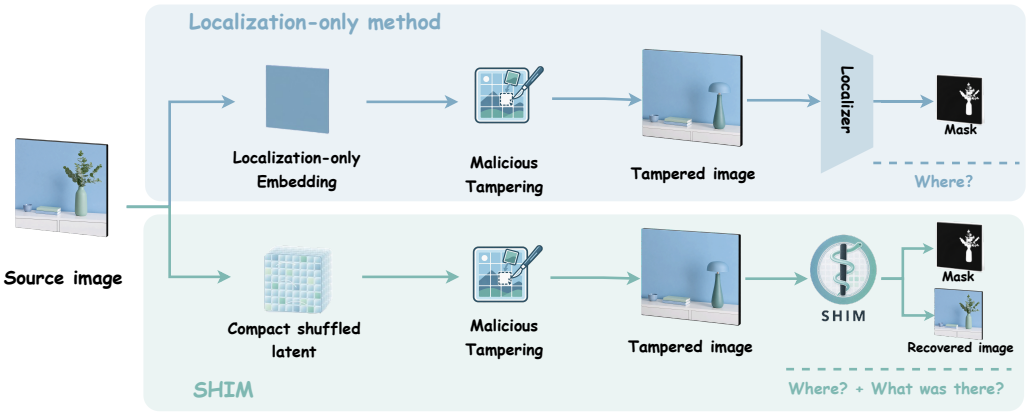


Figure 2: Proactive image protection. Localization-only methods embed cues for predicting tamper masks. SHIM embeds source content as a shuffled latent used for both localization and recovery.

to identify the affected pixels. Traditional methods use compression inconsistencies or keypoint correspondences (Chandana et al., 2024; Wang et al., 2025); neural methods learn spatial and noise features (Zhang et al., 2025a; Liu et al., 2022; Wu et al., 2019; Chen et al., 2021; Guillaro et al., 2023). These passive methods inspect the received image without retaining the content that an edit may have removed.

After inpainting or splicing, the received image need not determine the original content: different sources can produce the same edited observation. We use *self-healing* to mean reconstruction from source information embedded before distribution. The receiver must locate an unknown edit and recover the overwritten content from the protected image, including after benign noise or compression. The practical question is how much source information can be retained without visibly changing the image, and how that information can remain useful after editing.

Learned image hiding and watermarking embed auxiliary images or messages (Yu et al., 2023; Jing et al., 2021; Zhu et al., 2018; Guan et al., 2023; Fernandez et al., 2023). Self-recovery methods embed information derived from the source itself (Ying et al., 2021; 2023; Cao et al., 2024; Kim & Seo, 2026). ReImage, for example, shuffles an image-scale source representation (Kim & Seo, 2026). SHIM separates source encoding from spatial dispersion. A frozen autoencoder first produces a compact latent; a permutation then redistributes its spatial tokens before an invertible network embeds them in the source image.

At the receiver, SHIM decodes the extracted latent in two orders. Direct decoding retains concentrated manipulation-related inconsistencies. Decoding after inverse shuffling restores source alignment and allows comparison with the received image. A locator uses both views. The recovery model corrects the aligned estimate and combines it with a globally restored received image. Its learned fusion weights are restricted to a dilated prediction of the edited region. Both tasks use the same extracted latent, with no per-image source copy sent to the receiver.

The paper makes three contributions: (i) a compact latent self-embedding design with separate source encoding and spatial permutation; (ii) dual-order decoding for localization and source-aligned reconstruction; and (iii) a recovery model that combines local source correction, global restoration, and mask-constrained fusion. We train on COCO (Lin et al., 2014) and evaluate protection quality, localization, and recovery on valAGE-Set (Zhang et al., 2024; Kim & Seo, 2026). Additional experiments examine decoder inputs, recovery, source representations, and transfer across datasets and channels. The appendix records training and evaluation details, including the limits of each comparison.

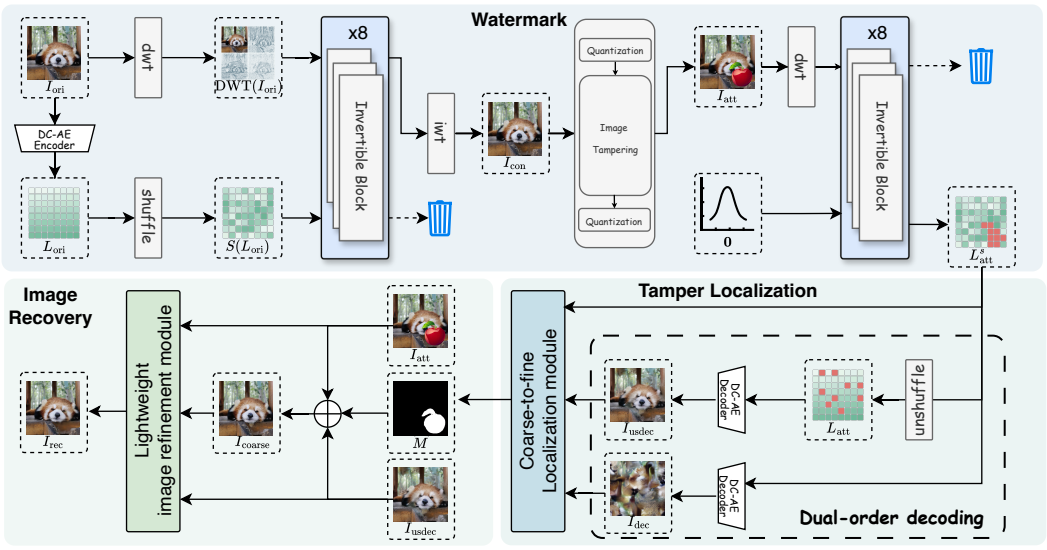


Figure 3: SHIM data flow. A shared INN embeds and extracts a shuffled source latent. Two decoding orders provide image evidence for localization. Recovery combines local source correction and global restoration through learned mask-constrained fusion. Image panels illustrate the stages; the auxiliary extraction input is a zero tensor.

2 Related Work

Passive localization. Passive IMDL detects forensic artifacts and low-level inconsistencies. IML-ViT combines a high-resolution Transformer with multiscale features and edge supervision (Ma et al., 2023); TruFor fuses RGB content with a noise-sensitive fingerprint (Guillaro et al., 2023). HDF-Net uses RGB and SRM streams, whereas Mesorch combines CNN and Transformer features across scales (Han et al., 2024; Zhu et al., 2025). SIDA and FakeShield additionally generate textual forensic explanations (Huang et al., 2025; Xu et al., 2025). These methods analyze the received image. They do not embed a source representation before editing and therefore have no stored copy of overwritten content to decode.

Proactive watermarking. Learned watermarking and image hiding balance fidelity, capacity, and robustness. HiDDeN trains an encoder and decoder with simulated noise (Zhu et al., 2018); RoSteALS represents messages in a frozen autoencoder's latent space (Bui et al., 2023); HiNet hides a full-resolution secret image with an invertible network (Jing et al., 2021). EditGuard combines localization with copyright bits, while OmniGuard adds degradation-aware passive extraction (Zhang et al., 2024; 2025b). WAM localizes watermarked regions and decodes their messages (Sander et al., 2025). StableGuard and GenPTW combine watermark verification or provenance tracing with localization (Yang et al., 2025; Gan et al., 2026). These embedded signals support localization or attribution, but are not source-content backups for image recovery.

Self-recovery. Self-recovery preserves source information before dissemination. Imuge jointly learns immunization, localization, and recovery (Ying et al., 2021), while Imuge+ connects these tasks through an invertible network (Ying et al., 2023). W-RAE combines reversible adversarial examples with self-embedding watermarking (Cao et al., 2024). ReImage embeds a shuffled image-scale source representation (Kim & Seo, 2026); RecoverMark places facial content in the surrounding background for face-specific recovery (An et al., 2026). SHIM uses a compact autoencoder latent as its source representation and permutes its spatial tokens before embedding. The receiver uses both the directly decoded view and the inverse-shuffled, source-aligned view.

3 METHOD

3.1 PRELIMINARIES

Compression autoencoder. A frozen DC-AE encoder E and decoder D (Chen et al., 2025) provide a compact source representation. For $I \in \mathbb{R}^{H\times W\times 3}$,

$$L = E(I), \qquad L \in \mathbb{R}^{H/32\times W/32\times 32}. \tag{1}$$

The decoder reconstructs $\widetilde{I} = D(L)$. At 256×256 resolution, the payload contains $8\times8\times32 = 2{,}048$ latent values. This is an internal continuous representation, not a serialized bitstream or an external source-image copy.

Invertible network. An invertible neural network (INN) provides forward and reverse mappings with shared parameters (Dinh et al., 2014; Jing et al., 2021). SHIM uses the forward path to embed the source latent and the reverse path to extract it from the received image. Replacing the discarded auxiliary output during extraction makes recovery approximate even though the network mapping itself is invertible.

3.2 COMPACT LATENT PROTECTION

For a source image I_{ori}, the encoder produces $L_{ori} = E(I_{ori})$. A fixed spatial permutation S, shared with the receiver, disperses its tokens:

$$L_{sh} = S(L_{ori}). \tag{2}$$

The permutation changes token positions while preserving their channel values. SHIM transforms the source into Haar-wavelet subbands and jointly processes the image and shuffled-latent branches:

$$(X_{con}, Z_{con}) = G_\theta(\mathrm{DWT}(I_{ori}), L_{sh}), \qquad I_{con} = \mathrm{IWT}(X_{con}). \tag{3}$$

The auxiliary output Z_{con} is discarded. Only the protected image I_{con} is released; no per-image side information is transmitted. Before evaluation, the image is clamped and converted to RGB UINT8.

3.3 DUAL-ORDER TAMPER LOCALIZATION

The received image I_{att} may contain a local edit and subsequent channel degradation. The reverse INN uses a fixed zero auxiliary input to extract

$$(\cdot, L_{att}) = G_\theta^{-1}(\mathrm{DWT}(I_{att}), \mathbf{0}). \tag{4}$$

The receiver decodes this estimate in two spatial orders:

$$I_{dec} = D(L_{att}), \qquad I_{usdec} = D\big(S^{-1}(L_{att})\big). \tag{5}$$

Direct decoding retains concentrated manipulation-related inconsistencies. Inverse shuffling restores source alignment and redistributes affected tokens. The aligned view can then be compared with the received image to estimate where source content has changed.

A coarse-to-fine locator outputs a probability map P, thresholded to obtain $\hat{M}$. Its coarse branch receives I_{dec}; its fine branch combines the received image, both decoded views, residual evidence, and the coarse probability. The threshold is fixed across attacks and channels. Appendix A gives the architecture and post-processing.

3.4 CONTEXT-AWARE SOURCE RECOVERY

The aligned view I_{usdec} provides an estimate of the original content, but extraction errors can leave artifacts. Following the recovery panel in Fig. 3, we summarize recovery through a coarse composition and a lightweight refinement module:

$$\begin{aligned} I_{coarse} &= (1 - M) \odot I_{att} + M \odot I_{usdec}, \\ I_{rec} &= \mathcal{R}_\phi(I_{coarse}, I_{att}, I_{usdec}), \end{aligned} \tag{6}$$

where M is the predicted repair support after post-processing and $\odot$ denotes elementwise multiplication. The coarse composition uses aligned source content inside the predicted repair region and

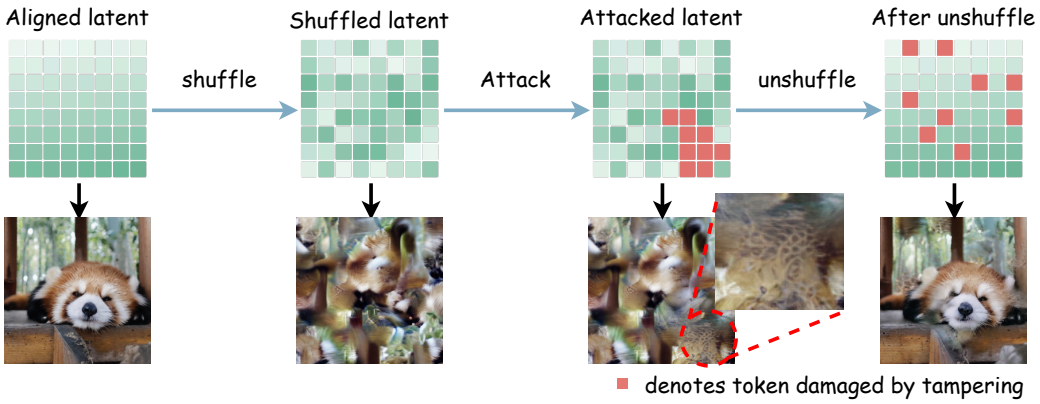


Figure 4: Dual-order decoding of a corrupted latent. Direct decoding retains concentrated corruption; inverse shuffling restores source alignment and disperses affected tokens. Red cells mark damaged tokens. The two views provide complementary localization evidence.

received context outside it. This combines the complementary roles of the two images: I_{usdec} supplies an estimate of overwritten content, while I_{att} provides the surrounding structure and appearance.

The lightweight NAFNet-style module $\mathcal{R}_\phi$ (Chen et al., 2022) represents learned recovery. It uses source evidence and received context to correct extraction artifacts and channel degradation, producing the final image I_{rec}. The predicted support guides the contribution of source evidence to the repaired region.

All recovery inputs are available at the receiver; ground-truth masks are not used at inference. Raw fusion uses I_{coarse} directly as a comparison without learned refinement. Architecture and post-processing details appear in Appendix A.

3.5 Two-Stage Training

We first freeze DC-AE and train the INN for source-latent embedding and extraction. We then freeze the carrier and train the receiver. The deployed locator is selected on a disjoint COCO validation split; after joint preparation, final refiner training restores and freezes this locator and uses predicted masks. This separation prevents recovery objectives from changing the carrier or the deployed localization checkpoint. Key hyperparameters and checkpoint-selection criteria appear in Appendix A.

4 Experiments

4.1 Experimental Setup

Data and protocol. We train on COCO (Lin et al., 2014). Tables 1 and 2 evaluate all 1,000 image–mask pairs in valAGE-Set (Zhang et al., 2024) at 256×256. We generate Copy-Move, Splicing, SD Inpainting, SDXL Inpainting, and Local Blur within the annotated masks. Protected images undergo clamping and RGB UINT8 conversion before editing. *Clean* has no subsequent channel; *Degradation* adds one of Gaussian noise, Poisson noise, or JPEG compression. Stochastic edit batches differ across methods, so these comparisons are protocol-aligned but not paired. Table 3 uses a separate cross-dataset protocol, defined in Section 4.3.

Metrics and implementation. PSNR, SSIM, and LPIPS measure container and full-image recovery fidelity; F1 and IoU measure localization. Metrics are computed per image on a common canvas, averaged within each attack, and then averaged equally across the five attacks. Recovery uses each method's predicted mask. Thresholds remain fixed across attacks and conditions. We omit cross-method AUC because several checkpoints do not expose continuous scores. Appendix D.1 specifies checkpoints, post-processing, channel parameters, and generation provenance.

Table 1: Protocol-aligned self-recovery on valAGE-Set under *Clean* and *Degradation*. Recovery uses each method's predicted mask. Best and second-best results are **bold** and underlined.

Method	Container I_{con}			Recovered Image I_{rec}																			
				CM			SP			SDXL			SD			LB			**Macro**		
	P↑	S↑	L↓	P	S	L	P	S	L	P	S	L	P	S	L	P	S	L	P	S	L
(a) Clean *UINT8 RGB, no post-tamper degradation*																							
Imuge*	32.43	.879	.083	24.93	.871	.136	24.89	.871	.136	24.98	.871	.136	25.07	.871	.136	25.35	.874	.133	25.04	.872	.135
Imuge+	34.80	.942	.044	25.70	.890	.146	23.13	.874	.164	28.70	.907	.134	28.36	.905	.134	29.90	.913	.126	27.16	.898	.141
ReImage	37.26	.935	.024	21.88	.839	.131	21.82	.831	.140	**32.90**	.920	**.050**	**33.18**	.921	**.047**	**34.14**	.923	**.044**	28.78	.887	.082
ReImage‡	34.97	.902	.048	26.83	.862	.100	20.79	.823	.138	30.51	.884	.080	31.14	.886	.074	32.32	.889	.069	28.32	.869	.092
SHIM (Ours)	**39.13**	**.958**	**.017**	**31.38**	**.924**	**.059**	**30.02**	**.919**	**.066**	31.29	**.924**	.059	31.27	**.923**	.059	33.35	**.930**	.054	**31.46**	**.924**	**.059**
(b) Degradation *UINT8 RGB + post-tamper channel perturbation*																							
Imuge*	32.43	.879	.083	23.95	.845	.159	23.91	.845	.160	23.96	.844	.160	24.02	.843	.160	24.48	.848	.156	24.07	.845	.159
Imuge+	34.80	.942	.044	20.59	.783	.227	19.90	.776	.233	21.09	.784	.232	22.24	.786	.225	26.21	.811	.201	22.01	.788	.224
ReImage	37.26	.935	.024	20.79	.736	.212	20.81	.732	.218	23.91	.782	.167	24.20	.770	.175	27.57	.793	.146	23.46	.763	.184
ReImage‡	34.97	.902	.048	23.70	.770	.168	20.16	.742	.195	25.41	.784	.157	26.33	.786	.150	28.83	.803	.133	24.88	.777	.161
SHIM (Ours)	**39.13**	**.958**	**.017**	**28.37**	**.895**	**.088**	**27.37**	**.890**	**.093**	**28.41**	**.895**	**.087**	**28.23**	**.892**	**.088**	**31.12**	**.904**	**.077**	**28.70**	**.895**	**.087**

Notes. P/S/L: PSNR/SSIM/LPIPS; CM/SP/SDXL/SD/LB: copy-move/splicing/SDXL inpainting/SD inpainting/local blur; Macro: unweighted mean. $*$ Completed epoch-50 Imuge reproduction; ‡ our cumulative 50k-step UINT8-aware ReImage adaptation; the released implementation omits the UINT8 round trip. Protocol details are in Appendix D.1.

4.2 Main Comparison

Image recovery. Among the evaluated methods, SHIM has the highest container PSNR (39.13 dB) and lowest LPIPS (.017). It also has the best macro recovery PSNR, SSIM, and LPIPS in both conditions. Recovery PSNR is 31.46 dB under *Clean* and 28.70 dB under *Degradation*, exceeding the next-best results by 2.68 and 3.82 dB.

The per-attack results qualify this aggregate advantage. Under *Clean*, ReImage has higher PSNR and lower LPIPS on SDXL Inpainting, SD Inpainting, and Local Blur, while SHIM has higher SSIM. After degradation, SHIM leads all three metrics on every attack. Splicing remains its hardest recovery case (30.02/27.37 dB under Clean/Degradation); Local Blur is easiest (33.35/31.12 dB). Channel degradation reduces its per-attack PSNR by 2.23–3.04 dB. ReImage‡ improves degraded macro recovery over the released checkpoint but reduces container and clean recovery fidelity. Since the two checkpoints use different stochastic edit batches, this difference is not a paired adaptation effect. Figure 5 compares their visible embedding residuals on shared sources.

Full-image recovery metrics reflect both localization and reconstruction: missed edits remain visible, while changes outside the edited region also affect fidelity. They should therefore be read alongside F1/IoU. The appendix additionally reports undilated ground-truth-mask diagnostics. These diagnostics change each receiver's recovery support and are not assumed to be oracle upper bounds or used for cross-method ranking.

Manipulation localization. SHIM achieves .910 F1 and .862 IoU after degradation and ranks first on every attack. Its macro gains over Imuge, the strongest recovery-capable baseline, are .135 F1 and .164 IoU; the gains over EditGuard-Degrade, the strongest localization-only proactive checkpoint, are .333 and .346. SD Inpainting is hardest for SHIM (.792 IoU), compared with .880 for Copy-Move and .902 for Local Blur. Complete *Clean* results appear in Appendix D.

Figure 6 illustrates predicted masks and SHIM's direct decoded view I_{dec}, which supplies localization evidence. Recovery uses the source-aligned view I_{usdec}, not the displayed direct decode. These examples are illustrative; all reported metrics use the complete split.

4.3 Cross-Dataset Robustness

We test frozen checkpoints on 100 shared sources each from COCO, ImageNet-1K (IN1K), and CelebA-HQ, totaling 300 source images. A fixed seed selects each subset from the existing 1,000-image pool. Selection is independent of model outputs. Every condition includes all sampled sources; low localization or recovery scores never trigger image exclusion or replacement. Masks, donors, and attack seeds are fixed; edits are generated from each method's protected images. ReImage uses fixed positional interpolation, and SHIM uses the PSNR39 checkpoint with a 16×16 latent grid;

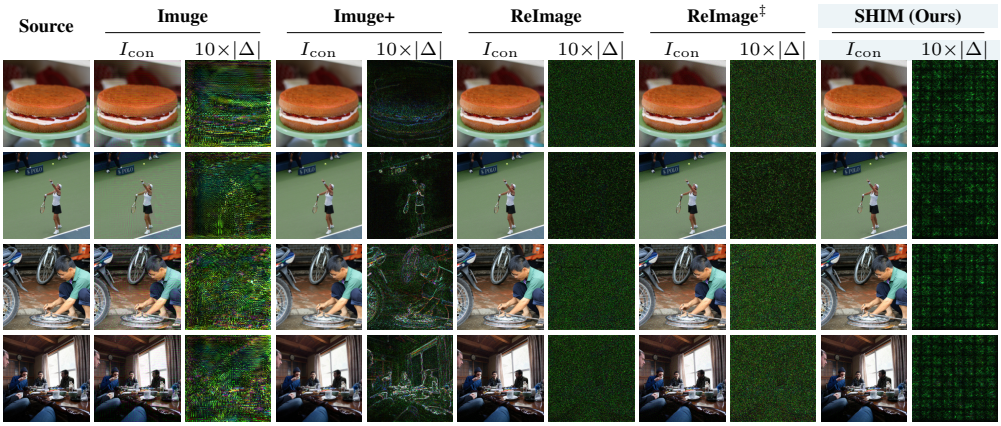


Figure 5: UINT8 containers and fixed-scale residuals for four shared sources. Residuals show $10\times|I_{con} - I_{ori}|$ and are clipped to $[0, 255]$ without per-image normalization. The examples are illustrative; Table 1 reports full-set metrics.

Table 2: valAGE-Set localization F1/IoU under *Degradation*; best per-attack results are in **bold**.

Method	CM		SP		SDXL		SD		LB		**Macro**	
	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU
Passive localization methods												
MVSSNet (Chen et al., 2021)	.435	.346	.436	.349	.377	.303	.189	.141	.190	.151	.325	.258
TruFor (Guillaro et al., 2023)	.696	.628	.668	.607	.532	.466	.258	.207	.355	.314	.502	.444
HiFiNet (Guo et al., 2023)	.261	.194	.340	.261	.112	.077	.040	.025	.024	.015	.155	.114
IML-ViT (Ma et al., 2023)	.725	.664	.705	.653	.584	.522	.240	.185	.299	.264	.511	.458
SparseViT (Su et al., 2025)	.722	.670	.736	.690	.644	.590	.343	.290	.438	.400	.576	.528
SAFIRE (Kwon et al., 2025)	.535	.460	.613	.544	.479	.404	.237	.177	.388	.334	.450	.384
Localization-only proactive methods												
EditGuard-Clean (Zhang et al., 2024)	.316	.216	.152	.085	.399	.284	.399	.285	.401	.288	.333	.232
EditGuard-Degrade (Zhang et al., 2024)	.423	.381	.005	.003	.834	.751	.835	.750	.789	.697	.577	.516
OmniGuard (Zhang et al., 2025b)	.319	.270	.076	.046	.512	.426	.480	.393	.435	.352	.364	.297
GenPTW (Gan et al., 2026)	.174	.132	.046	.031	.695	.610	.674	.592	.671	.587	.452	.391
Recovery-capable proactive methods												
Imuge (Ying et al., 2021)	.804	.730	.819	.746	.775	.693	.717	.633	.761	.687	.775	.698
Imuge+ (Ying et al., 2023)	.247	.190	.191	.141	.337	.269	.259	.196	.194	.148	.246	.189
ReImage (Kim & Seo, 2026)	.181	.107	.228	.146	.550	.437	.449	.343	.484	.373	.379	.281
ReImage‡ (Kim & Seo, 2026)	.432	.378	.110	.077	.626	.551	.618	.546	.747	.659	.507	.443
SHIM (Ours)	**.923**	**.880**	**.894**	**.845**	**.932**	**.890**	**.865**	**.792**	**.935**	**.902**	**.910**	**.862**

Notes. CM/SP/SDXL/SD/LB: copy-move/splicing/SDXL inpainting/SD inpainting/local blur; Macro is the unweighted mean across these five attacks. Appendix D.1 defines the condition. ‡ denotes our cumulative 50k-step adaptation of ReImage; EditGuard-Clean/Degrade are distinct checkpoints.

weights and thresholds are frozen. ReImage applies channels before tampering, while SHIM and Imuge+ apply them afterward. Appendix E details these adapters.

We test Clean and eight separate channels: JPEG Q90, a 150% scaling round trip, Gaussian and median blur, hue, brightness, contrast, and AWGN. Clean includes local tampering without an additional channel. Both blur filters use 3×3 support; Gaussian blur has $\sigma = 1$ and AWGN has $\sigma = 1/255$. Recovery uses predicted masks. F1 and full-image PSNR/SSIM are evaluated at 512, with image metrics against the shared enlarged source and no metric downsampling. We average per-image metrics within each attack, then equally across five attacks and three datasets. Avg is the equally weighted mean over the eight degradation channels, excluding Clean.

SHIM has the highest F1 and SSIM in all nine conditions and the highest PSNR in eight. Its degradation averages are .901 F1, 32.02 dB PSNR, and .916 SSIM. ReImage leads JPEG recovery PSNR (32.28 dB versus 31.74 dB for SHIM). These are descriptive results on the fixed subset. Operating resolution and metric targets are shared, but training histories, resolution adapters, channel order, and JPEG implementations differ. Upsampling adds no source detail, and shared seeds need not produce identical generative edits. Aggregate rankings need not hold for every dataset, and the experiment does not isolate any single design choice. Protected-image PSNR before editing is

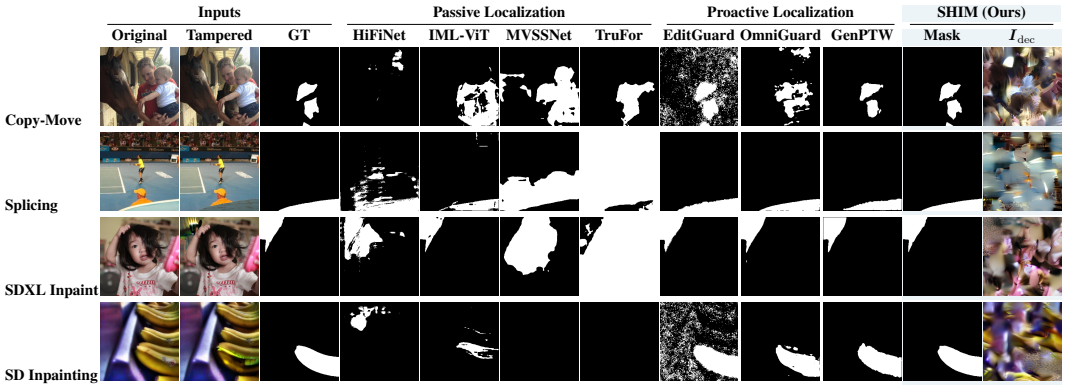


Figure 6: Qualitative localization and direct-decoding examples. Rows 2–3 use *Clean* (UINT8 RGB only), whereas rows 1 and 4 add Poisson noise after tampering. I_{dec} is diagnostic evidence, not the final recovered image.

Table 3: Cross-dataset robustness at 512×512 with frozen models. Values average three datasets and five attacks (100 sources per dataset). All metrics use 512 resolution; PSNR/SSIM share the unmodified source target. **Bold** marks the highest observed value per column and metric. Clean: tampering only. Avg: mean over JP–AWGN, excluding Clean.

Method	Metric	Clean	JP	SC	GF	MF	Hue	Brightness	Contrast	AWGN	Avg
Imuge+	F1	.794	.800	.809	.789	.778	.764	.787	.777	.781	.786
	PSNR	31.34	30.67	31.19	30.86	30.59	28.62	29.30	28.41	31.17	30.10
	SSIM	.910	.887	.904	.898	.891	.887	.906	.871	.906	.894
ReImage	F1	.664	.857	.573	.303	.341	.619	.638	.605	.665	.575
	PSNR	32.50	**32.28**	25.24	19.53	19.65	29.16	29.96	28.95	32.28	27.13
	SSIM	.907	.900	.865	.738	.737	.888	.904	.877	.900	.851
SHIM (Ours)	F1	**.938**	**.873**	**.904**	**.864**	**.906**	**.903**	**.925**	**.902**	**.934**	**.901**
	PSNR	**33.52**	31.74	**33.73**	**33.19**	**33.32**	**30.07**	**30.77**	**29.73**	**33.62**	**32.02**
	SSIM	**.917**	**.908**	**.936**	**.937**	**.934**	**.897**	**.915**	**.881**	**.920**	**.916**

36.90 dB (Imuge+), 39.65 dB (ReImage), and 39.30 dB (SHIM). Repeated attacks and channels reuse the same 300 sources: the 40,500 tuples do not represent independent source images. Column maxima do not establish statistical significance, and resampling may change the estimates.

4.4 Ablation and Model Analysis

Table 4 examines locator evidence, refinement, and source representations. Blocks (a)–(b) hold the deployed checkpoints and attacked inputs fixed; Block (c) compares separately optimized systems.

Complementary localization evidence. In Block (a), the aligned view supplies most of the trained locator's evidence. Adding the direct view raises Clean F1/IoU from .951/.922 to .955/.927 and degraded F1/IoU from .897/.846 to .910/.862. The improvement is larger after degradation. The attacked image alone gives .537/.450 and .484/.388. We keep the locator weights fixed in these comparisons, so the results measure its use of each input. Independently trained single-view locators would be needed to compare the architectures themselves.

Learned recovery versus raw composition. Block (b) fixes the aligned decode and predicted support. The complete recovery module improves PSNR by 3.78/4.08 dB and SSIM by .029/.098 under Clean/Degradation. Degraded LPIPS decreases from .123 to .087; Clean LPIPS increases slightly from .058 to .059. This comparison includes local correction, global restoration, and learned fusion. It does not isolate the contribution of any one branch. In particular, full-image gains after degradation can include restoration outside the predicted repair region.

Table 4: Analyses of (a) locator evidence with fixed weights, (b) raw mask fusion versus the complete recovery module, and (c) separately trained autoencoder systems. Shading marks the evaluated SHIM configuration.

(a) Dual-order localization

Locator evidence	Clean		Degradation	
	F1↑	IoU↑	F1↑	IoU↑
Attacked image only	.537	.450	.484	.388
Attacked + direct	.596	.515	.548	.457
Attacked + aligned	.951	.922	.897	.846
Both decodes (full)	.955	.927	.910	.862

(b) Raw fusion vs. learned recovery

Channel	Variant	Recovery quality		
		PSNR↑	SSIM↑	LPIPS↓
Clean	Raw fusion	27.68	.895	.058
	Refined	31.46	.924	.059
Degradation	Raw fusion	24.62	.797	.123
	Refined	28.70	.895	.087

(c) Autoencoder variants

Backbone	Latent shape / values	Container PSNR↑	Localization IoU↑		Recovery PSNR↑	
			Clean	Degr.	Clean	Degr.
SD 1.5	$32^2 \times 4$ / 4,096	34.74	.923	.830	29.57	27.42
SDXL	$32^2 \times 4$ / 4,096	34.70	.904	.799	28.91	26.69
FLUX	$32^2 \times 16$ / 16,384	34.80	.905	.776	28.15	26.00
DC-AE (dep.)	$\mathbf{8^2 \times 32}$ **/ 2,048**	**39.13**	**.927**	**.862**	**31.46**	**28.70**

"dep." denotes the deployed configuration. Only metrics archived for every autoencoder variant are shown. Blocks (a)–(b) share attacked inputs and fixed upstream outputs. Raw fusion directly composites the aligned decode inside the predicted support. Refined recovery includes local correction, global restoration, and learned fusion. Block (c) separately trains protection and receiver modules for each backbone; it does not isolate the autoencoder.

Compact source representations. Block (c) compares complete systems, not interchangeable autoencoders inside a fixed model. DC-AE uses 2,048 latent values, half the SD 1.5/SDXL payload and one eighth of the FLUX payload. It nevertheless improves container PSNR by 4.33–4.43 dB, localization IoU by .004–.023 under Clean and .032–.086 under Degradation. Recovery gains are 1.89–3.31 dB under Clean and 1.28–2.70 dB under Degradation. DC-AE gives the strongest results among these trained variants. Architecture, latent statistics, and training change together, so the comparison does not identify payload size as the cause of the gains. Appendix C provides grid-size analysis and a separately trained full-resolution pixel-payload experiment.

4.5 Inference Cost

On an RTX 4090 at 256×256 and batch size 1, protection takes 17.95 ms and the full receiver takes 36.54 ms, using 0.90 GiB peak allocated GPU memory. Timings include GPU post-processing but exclude data loading and file I/O. The system loads 366.74M distinct parameters, including frozen DC-AE. The frozen DC-AE accounts for 312.25M of these parameters. Appendix A.3 gives the timing protocol and component costs.

5 Conclusion and Limitations

SHIM uses a shuffled source latent for both localization and recovery. The two decoding orders supply different image evidence, and learned recovery improves reconstruction over direct composition. The experiments also have limits: valAGE-Set baseline batches are not paired by edit content, training-seed stability is untested, and the threshold comes from an earlier benchmark-level sweep. Cross-dataset comparisons retain different training histories, resolution adapters, and channel orders. Enlarging existing sources does not establish performance on native high-resolution images or show that resolution alone explains the observed gains. The autoencoder study does not isolate payload size; a payload-matched pixel-space comparison remains needed. Recovery is lossy, and permutation provides no encryption guarantee. Global replacement, stronger channels, and broader image domains remain outside the tested setting.

AI USE STATEMENT

Generative AI tools were used for literature discovery and checking, experiment-design feedback, code and text editing, experiment orchestration and bookkeeping, analysis, table and formatting assistance, and language polishing. All reported measurements were produced by executed evaluation code on the stated datasets rather than synthesized by a language model, and AI tools did not determine scientific claims autonomously. The authors reviewed and verified all AI-assisted text, citations, code changes, experimental results, and claims, and take full responsibility for the manuscript.

ETHICS STATEMENT

SHIM is intended for research on image integrity, manipulation localization, and content recovery. These capabilities are dual-use and could be misapplied to conceal or alter evidence; the method should not be used to obscure manipulation or evade forensic analysis. Our experiments use publicly available research images, do not perform identity tracking, and do not involve human subjects.

REPRODUCIBILITY STATEMENT

The appendix specifies the evaluated modules, key training hyperparameters, checkpoint-selection criteria, attacks, channel parameters, thresholds, and metric implementations. It also identifies differences between the comparison protocols and the settings used for historical ablations.

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. URL https://openaccess.thecvf.com/content/CVPR2026/html/An_RecoverMark_Robust_Watermarking_for_Localization_and_Recovery_of_Manipulated_Faces_CVPR_2026_paper.html.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. Technical report, OpenAI, 2023. URL https://cdn.openai.com/papers/dall-e-3.pdf.

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.

Xu Cao, Ju Liu, Jinghui Yin, Xuejun Cheng, Jing Li, Hao Ma, and Guanghui Luo. Reversible adversarial examples based on self-embedding watermark for image privacy protection. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2024. doi: 10.1109/IJCNN60899.2024.10650535.

S Chandana, C R Nagarathna, A Amrutha, and A Jayasri. Detection of image forgery using error level analysis. In *2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, pp. 1–5, 2024. doi: 10.1109/IITCEE59897.2024.10467523.

Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f062da1973ac9ac61fc6d44dd7fa309f-Abstract-Conference.html.

Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European Conference on Computer Vision*, pp. 17–33, 2022. doi: 10.1007/978-3-031-20071-7_2.

Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14185–14193, October 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Chen_Image_Manipulation_Detection_by_Multi-View_Multi-Scale_Supervision_ICCV_2021_paper.html.

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation, 2014. URL https://arxiv.org/abs/1410.8516.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

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

Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=S1RKWSyZ2Y.

Zhenliang Gan, Chunya Liu, Yichao Tang, Binghao Wang, Shiwen Cui, Weiqiang Wang, and Xinpeng Zhang. GenPTW: Latent image watermarking for provenance tracing and tamper localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4085–4093, 2026. doi: 10.1609/aaai.v40i5.37412.

Zhenyu Guan, Junpeng Jing, Xin Deng, Mai Xu, Lai Jiang, Zhou Zhang, and Yipeng Li. Deepmih: Deep invertible network for multiple image hiding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):372–390, 2023. doi: 10.1109/TPAMI.2022.3141725.

Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20606–20615, 2023.

Xiao Guo, Xiaohong Liu, Zhiyuan Ren, Steven Grosz, Iacopo Masi, and Xiaoming Liu. Hierarchical fine-grained image forgery detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3155–3165, 2023.

Ruidong Han, Xiaofeng Wang, Ningning Bai, Yihang Wang, Jianpeng Hou, and Jianru Xue. Hdf-net: Capturing homogeny difference features to localize the tampered image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10005–10020, 2024. doi: 10.1109/TPAMI.2024.3432551.

Zhenglin Huang, Jinwei Hu, Xiangtai Li, Yiwei He, Xingyu Zhao, Bei Peng, Baoyuan Wu, Xiaowei Huang, and Guangliang Cheng. Sida: Social media image deepfake detection, localization and explanation with large multimodal model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28831–28841, 2025.

Junpeng Jing, Xin Deng, Mai Xu, Jianyi Wang, and Zhenyu Guan. HiNet: Deep image hiding by invertible network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4733–4742, 2021.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6007–6017, 2023.

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.

URL https://openaccess.thecvf.com/content/CVPR2026F/html/Kim_Robust_Image_Self-Recovery_against_Tampering_using_Watermark_Generation_with_Pixel_CVPRF_2026_paper.html.

Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 3677–3685, 2017.

Myung-Joon Kwon, Wonjun Lee, Seung-Hun Nam, Minji Son, and Changick Kim. SAFIRE: Segment any forged image region. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 4437–4445, 2025. doi: 10.1609/aaai.v39i4.32467.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48.

Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *Advances in Neural Information Processing Systems*, 34: 16331–16345, 2021.

Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7505–7517, 2022. doi: 10.1109/TCSVT.2022.3189545.

Xiaochen Ma, Bo Du, Zhuohang Jiang, Xia Du, Ahmed Y. Al Hammadi, and Jizhe Zhou. Iml-vit: Benchmarking image manipulation localization by vision transformer. *arXiv preprint arXiv:2307.14863*, 2023.

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, 2023. URL https://arxiv.org/abs/2307.01952.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo-Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494, 2022. doi: 10.52202/068431-2643. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/ec795aeadae0b7d230fa35cbaf04c041-Abstract-Conference.html.

Tom Sander, Pierre Fernandez, Alain Oliviero Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. In *International Conference on Learning Representations*, 2025.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8871–8879, 2024.

Lei Su, Xiaochen Ma, Xuekang Zhu, Chaoqun Niu, Zeyu Lei, and Ji-Zhe Zhou. Can we get rid of handcrafted feature extractors? sparsevit: Nonsemantics-centered, parameter-efficient image manipulation localization through sparse-coding transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7024–7032, 2025. doi: 10.1609/aaai.v39i7.32754. URL https://ojs.aaai.org/index.php/AAAI/article/download/32754/34909.

Xiangyang Wang, Huiying Zhang, Dawei Wang, and Panpan Niu. Adaptive copy move forgery detection based on new keypoint feature and matching. *Applied Intelligence*, 55(12):864, July 2025. ISSN 0924-669X. doi: 10.1007/s10489-025-06735-4. URL https://doi.org/10.1007/s10489-025-06735-4.

Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9535–9544, 2019. doi: 10.1109/CVPR.2019.00977.

Zhipei Xu, Xuanyu Zhang, Runyi Li, Zecheng Tang, Qing Huang, and Jian Zhang. Fakeshield: Explainable image forgery detection and localization via multi-modal large language models. In *International Conference on Learning Representations*, 2025.

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 *Advances in Neural Information Processing Systems*, volume 38, pp. 14692–14720, 2025. doi: 10.52202/085713-0495. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/15972a9575e0f03bf82f00aebeb40774-Abstract-Conference.html.

Qichao Ying, Zhenxing Qian, Hang Zhou, Haisheng Xu, Xinpeng Zhang, and Siyi Li. From image to imuge: Immunized image generation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3565–3573, 2021. doi: 10.1145/3474085.3475520.

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. doi: 10.1109/TPAMI.2023.3301958.

Jiwen Yu, Xuanyu Zhang, Youmin Xu, and Jian Zhang. Cross: Diffusion model makes controllable, robust and secure image steganography. *Advances in Neural Information Processing Systems*, 36: 80730–80743, 2023.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26125–26135, 2025.

Bolin Zhang, Qilin Yin, Wei Lu, and Xiangyang Luo. Deepfake detection and localization using multi-view inconsistency measurement. *IEEE Transactions on Dependable and Secure Computing*, 22(2):1796–1809, 2025a. doi: 10.1109/TDSC.2024.3472064.

Xuanyu Zhang, Runyi Li, Jiwen Yu, Youmin Xu, Weiqi Li, and Jian Zhang. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11964–11974, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Zhang_EditGuard_Versatile_Image_Watermarking_for_Tamper_Localization_and_Copyright_Protection_CVPR_2024_paper.html.

Xuanyu Zhang, Zecheng Tang, Zhipei Xu, Runyi Li, Youmin Xu, Bin Chen, Feng Gao, and Jian Zhang. Omniguard: Hybrid manipulation localization via augmented versatile deep image watermarking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3008–3018, June 2025b.

Haozhe Zhao, Xiaojian Shawn Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024.

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

Xuekang Zhu, Xiaochen Ma, Lei Su, Zhuohang Jiang, Bo Du, Xiwen Wang, Zeyu Lei, Wentao Feng, Chi-Man Pun, and Ji-Zhe Zhou. Mesoscopic insights: Orchestrating multi-scale & hybrid architecture for image manipulation localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11022–11030, 2025. doi: 10.1609/aaai.v39i10.33198.

Table 5: Key hyperparameters of the final training stages. Epoch counts refer to the final continuation of each component, rather than its total training history.

Component	Optimizer	Learning rate	Epochs
INN	Adam	Cosine: 5×10^{-9} to 5×10^{-10}	10
Locator	AdamW	Fine: 4×10^{-6}; coarse: 2×10^{-6}	6
Refiner	AdamW	Local: 5×10^{-7}; global: 3×10^{-6}; fusion: 10^{-5}	30

A Implementation Details

A.1 Model Configuration

Training and the main valAGE-Set evaluation use 256×256 images. The separate cross-dataset retest uses 512×512 images, with the resolution adapters given in Appendix E. At 256×256, the frozen DC-AE checkpoint (Sana 1.1, f32c32) (Chen et al., 2025) produces an $8 \times 8 \times 32$ latent. An 8-block INN embeds this latent in the Haar-wavelet representation of the source image. The 64 spatial tokens are shuffled with a fixed permutation (seed 97), shared by sender and receiver. Protected images are projected to RGB UINT8, and the reverse INN uses a zero auxiliary input.

The coarse-to-fine U-Net locator has base widths 16 and 32. The recovery module uses local and global NAFNet-style branches (Chen et al., 2022) with widths 32 and 24, respectively, and a fusion head of width 16. The localization threshold is 0.03; an 8-pixel dilation defines the repair support. Both settings remain fixed across attacks and conditions. Recovery uses predicted masks and blends the local and global estimates within this support; outside it, the output is the global estimate.

A.2 Training Setup

The downstream training set contains 4,000 COCO source images, with another 1,000 reserved for validation. Each source is paired with five attacks, giving 20,000 training and 5,000 validation records. These splits are disjoint from valAGE-Set. DC-AE remains frozen throughout training. Receiver training freezes the INN, and final refiner training also freezes the selected locator.

Training uses BF16. The INN and refiner use a global batch size of 64. Adam uses $\beta = (0.5, 0.9)$ and weight decay 10^{-5}; locator AdamW uses $\beta = (0.9, 0.999)$ and weight decay 10^{-4}, while refiner AdamW uses weight decay 0.01. The locator learning rate follows cosine decay without warm-up; the refiner uses a 0.25-epoch warm-up. Seeds are 97 for the INN and locator, and 497 for the final refiner.

Channel augmentation. Each attack record has a Clean view and a Degradation view. We generate the latter after tampering with one of three equally likely operations: Gaussian noise with $\sigma \sim \mathcal{U}(1, 9)$ in UINT8 units, Poisson noise with $\alpha = 4$, or JPEG compression at quality 90. Assignments are resampled each training epoch and fixed for validation. The locator averages the losses of the paired views; final refiner training samples the degraded view with probability 0.75. Protection training uses UINT8 quantization with a straight-through estimator, without additional channel degradation.

Checkpoint selection. We select checkpoints on held-out COCO data. For the INN, we minimize grid artifacts subject to fidelity and latent-reconstruction constraints. For the locator, we maximize mean Clean/Degradation macro IoU. The refiner maximizes Degradation macro full-image PSNR among the last two predicted-mask epochs. The selected epochs are 9, 4, and 28 for the INN, locator, and refiner, respectively. We do not use valAGE-Set localization or recovery scores for checkpoint selection.

A.3 Efficiency and Payload

We profile inference at 256×256 with batch size 1 on an NVIDIA GeForce RTX 4090. DC-AE uses FP16; the remaining modules and image/latent tensors use FP32. Latency is the median of three run means, each measured with 50 warm-up and 200 CUDA-event timed iterations. Data loading and file I/O are excluded.

Table 6: Measured inference cost of the deployed SHIM model. Lower is better for every column.

Inference path	Loaded params. (M)↓	fvcore net. GFLOPs↓	Latency (ms/image)↓	Peak alloc. VRAM (GiB)↓
Protection (sender)	321.75	209.29	17.95	0.74
Full localization receiver	331.57	588.39	29.75	0.77
Refiner only (evidence given)	35.17	20.74	10.00	0.20
Full receiver (localization + recovery)	366.74	609.13	36.54	0.90

Notes. Parameter counts include the complete loaded DC-AE. GFLOPs use fvcore's supported-operator count, with one fused multiply–add counted as one operation; unsupported operators and pre/post-processing are excluded from this count but included in latency. Peak allocated VRAM is measured separately for each path. The full receiver is timed directly, rather than by adding component latencies.

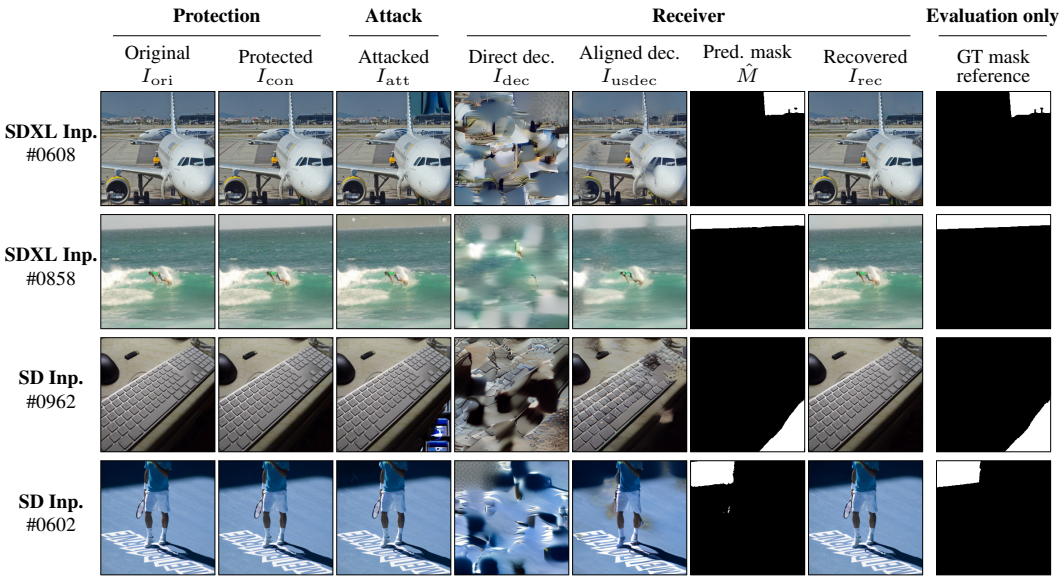


Figure 7: SHIM outputs for four selected valAGE-Set examples under Clean/UINT8. Columns show the original image I_{ori}, protected image I_{con}, attacked image I_{att}, direct decode I_{dec}, inverse-shuffled decode I_{usdec}, predicted mask $\hat{M}$, and recovered image I_{rec}. The last column shows the ground-truth mask for reference; it is not an inference input. These examples were selected rather than randomly sampled.

At 256×256, each image carries 2,048 latent activations ($8 \times 8 \times 32$), occupying 8 KiB as an internal FP32 tensor. The latent is embedded in the protected image; we transmit no separate per-image bitstream or side information.

B Qualitative Results

Figure 7 shows selected SHIM outputs under Clean/UINT8.

C Additional Experiments

C.1 Latent Shuffle Granularity

Figure 8 compares shuffle grids with fixed weights and edit content. Increasing the grid from 1×1 (no shuffle) to 8×8 lowers container PSNR from 37.137 to 36.137 dB and raises localization IoU from 0.870/0.820 to 0.932/0.884 under Clean/Degradation. Recovery PSNR peaks at 4×4: 30.785/28.927 dB, compared with 30.559/28.783 dB at 8×8.

We use 8×8 in the trained model and all main-table results because it gives the highest localization IoU in both conditions. This grid permutes all 64 spatial locations of the $8 \times 8 \times 32$ latent as individual

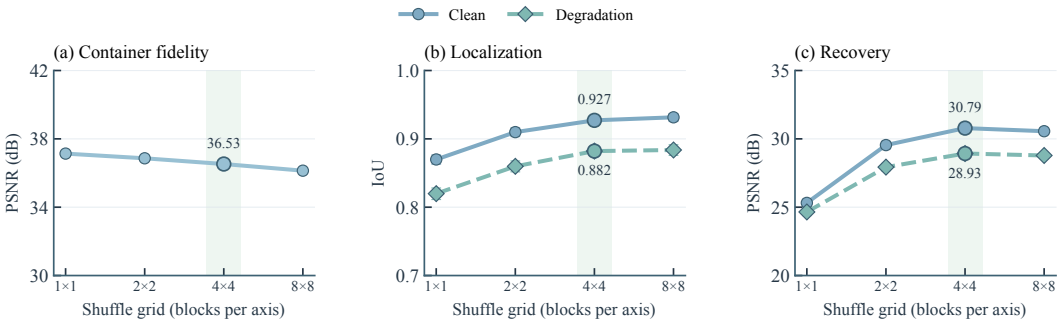


Figure 8: Effect of shuffle grid size on valAGE-Set. We vary the grid at test time while fixing the INN, pixel-space locator, and NAFNet refiner. Each point averages five attacks equally, with 1,000 images per attack. Clean and Degradation follow the main-table protocols; container quality is shared across them. Error bars show paired source-level 95% bootstrap confidence intervals. Shading marks the 4×4 grid with the highest recovery PSNR. The final system uses an 8×8 grid.

Table 7: Full-resolution pixel-payload reconstruction on 1,000 held-out COCO images at 256×256, using the final trained checkpoint. Clean denotes UINT8 storage without an additional channel.

Channel	Reconstruction PSNR (dB)↑
Clean UINT8	28.83
Gaussian noise ($\sigma = 16/255$)	17.56
JPEG (quality 90, 4:4:4)	25.69
Gaussian blur (3×3, $\sigma = 1$)	29.58
Median filter (3×3)	27.51

1×1 tokens. Switching to 4×4 at test time gains 0.226/0.144 dB in recovery PSNR at an IoU cost of 0.004/0.002 under Clean/Degradation. The sweep uses an earlier checkpoint than the final 39.13-dB model, so its scores should be compared within the sweep.

C.2 Full-Resolution Pixel-Payload Embedding

We train a self-embedding variant from scratch that carries the full RGB image in the pixel domain. We divide each 256×256 source into an 8×8 grid of 32×32 patches, shuffle them with seed 97, and apply a lossless Haar transform to obtain a $12 \times 128 \times 128$ payload. An eight-block INN with 6.67M parameters embeds and extracts the payload. Inverse Haar transformation and inverse shuffling then reconstruct the source. This variant uses no autoencoder. During extraction, the inverse INN receives zeros for the discarded auxiliary branch.

We split COCO train2017 into 117,287 training images and 1,000 held-out images using seed 9701, and resize them to 256×256. We train for 16 epochs (234,560 updates) on four GPUs with BF16, a global batch size of 8, and seed 197. Adam's learning rate follows a cosine schedule from 10^{-4} to 10^{-6}. The first epoch uses only clean UINT8 images. We increase channel severity over the next two epochs, then use full-strength augmentation. The four sampled channels are Gaussian noise with $\sigma \in [1, 16]/255$, JPEG at quality 75–95, 3×3 Gaussian blur with $\sigma = 1$, and 3×3 median filtering. The loss weights are 10 for carrier L1, 10 for payload L1, 4 for Alex-LPIPS, and 0.05 for the hinge adversarial term. When carrier PSNR falls below 37 dB, we scale down the payload weight, using a minimum multiplier of 0.05.

We evaluate the final epoch-16 checkpoint on all 1,000 held-out images (Table 7). Images are quantized to UINT8 before and after the channel. We use BF16 inference, clip reconstructed images to the valid RGB range, and average their full-image PSNR against the original sources. Protected images reach 38.37 dB PSNR. This test measures payload extraction under channel degradation, without local tampering, a localization network, or a learned image refiner.

Reconstruction reaches 28.83 dB under Clean UINT8; Gaussian noise causes the largest drop among the tested channels. The pixel payload contains 196,608 values, 96 times SHIM's 2,048 latent values. Because payload size, architecture, and training all differ, these results cannot isolate the effect of representation. They also do not measure recovery from local tampering.

D Benchmark Details

D.1 Evaluation Conditions

We evaluate on the released 1,000-image valAGE-Set split introduced by EditGuard (Zhang et al., 2024) and later used by ReImage (Kim & Seo, 2026). The release stores the images and masks as `valAGE-Set` and `valAGE-Set-Mask`.[1] We use all 1,000 image–mask pairs, without subsampling or selection, for each of five manipulations: copy-move, splicing, SDXL inpainting, Stable Diffusion inpainting, and local blur.

Shared input preparation. Before tampering, we clamp each protected or neutral image, convert it to UINT8 RGB, and decode it on the common 256×256 canvas. We then apply the manipulation within the annotated mask.

Clean. We evaluate the manipulated image without further channel degradation.

Degradation. After tampering, each image receives one channel perturbation, sampled uniformly from Gaussian noise with $\sigma \sim \mathcal{U}(1, 9)$, Poisson noise with $\alpha = 4$, and JPEG compression at quality 90. Assignments are fixed within each generation batch.

Generation batches. Imuge+ and the passive and proactive localization baselines use results from an earlier batch with tamper/degradation seeds 20260714/20260713. We exclude that batch's SHIM, local W-RAE, and released-ReImage results from the current ranking. We retest the released ReImage checkpoint with the paired 20260731/20260732 seeds used for final SHIM and ReImage‡. The epoch-50 Imuge reproduction was evaluated in separate runs. All batches use the same 1,000 image–mask pairs, nominal attack definitions, channel distributions, and metrics. Separately generated batches can differ in their stochastic edit content and per-image degradation assignments, so comparisons across batches match the evaluation protocol but are not paired per image. We resize native-resolution outputs to the common canvas before scoring.

Localization thresholds. We hold each threshold fixed across attacks and conditions. SHIM uses 0.03, chosen in an earlier benchmark-level threshold/postprocessing sweep with a previous locator and transferred to the current model. The thresholds are 0.20/0.40 for EditGuard-C/EditGuard-D and 0.50 for Imuge, Imuge+, ReImage, ReImage‡, OmniGuard, GenPTW, MVSSNet, TruFor, HiFiNet, IML-ViT, SparseViT, and SAFIRE.

Metric implementation. We compute full-reference metrics on clamped RGB tensors in $[0, 1]$, average per-image scores within each attack, and then macro-average across attacks. PSNR uses full-image MSE with data range 1. SSIM uses the TorchMetrics Gaussian 11×11 window with $\sigma = 1.5$, $k_1 = 0.01$, and $k_2 = 0.03$. LPIPS uses version 0.1 with the AlexNet backbone after mapping RGB values to $[-1, 1]$. RGB inputs are resized to the common canvas with bicubic interpolation and masks with nearest-neighbor interpolation.

Evaluated methods. The localization benchmark covers 13 method families and 15 checkpoints or variants under two conditions and five attacks: 150 checkpoint–condition–attack combinations and 150,000 image evaluations. These include the earlier baseline results and the complete evaluations of our epoch-50 Imuge reproduction, final SHIM, and the final 50k-step ReImage‡ adaptation. Each checkpoint is evaluated on all 1,000 images in each of the ten condition–attack combinations. ReImage‡ adapts the released ReImage checkpoint (Kim & Seo, 2026); we exclude the local W-RAE reproduction from the ranking because no official checkpoint is available, and exclude Mesorch and RITA because no usable checkpoint is available. As with the full-reference metrics, we average localization scores per image within each attack and then across attacks.

ReImage‡ adaptation. Starting from the authors' released 200k-step ReImage checkpoint (Kim & Seo, 2026), we fine-tune for 50,000 additional updates on all 118,287 COCO-train2017 images at

[1] Official release instructions: `https://github.com/xuanyuzhang21/EditGuard`

Table 8: Per-attack localization IoU. Macro is the unweighted average over copy-move (CM), splicing (SP), SDXL inpainting (SDXL), Stable Diffusion inpainting (SD), and local blur.

	Clean						Degradation					
Method	CM	SP	SDXL	SD	Blur	**Macro**	CM	SP	SDXL	SD	Blur	**Macro**
Passive localization methods												
MVSSNet	0.3419	0.3432	0.4042	0.2264	0.4105	0.3452	0.3460	0.3488	0.3028	0.1405	0.1513	0.2579
TruFor	0.6335	0.6086	0.5027	0.2278	0.4649	0.4875	0.6276	0.6075	0.4655	0.2071	0.3144	0.4444
HiFiNet	0.3987	0.4688	0.4062	0.1526	0.3950	0.3643	0.1944	0.2605	0.0771	0.0246	0.0155	0.1144
IML-ViT	0.6714	0.6820	0.5754	0.2152	0.4247	0.5137	0.6644	0.6535	0.5220	0.1848	0.2639	0.4577
SparseViT	0.6890	0.7069	0.7215	0.5786	0.6068	0.6606	0.6695	0.6900	0.5896	0.2898	0.3999	0.5278
SAFIRE	0.6070	0.6551	0.6007	0.4617	0.5991	0.5847	0.4596	0.5442	0.4044	0.1774	0.3339	0.3839
Localization-only proactive methods												
EditGuard-Clean	0.5766	0.0257	0.8742	0.8843	0.8953	0.6512	0.2157	0.0849	0.2843	0.2855	0.2877	0.2316
EditGuard-Degrade	0.3872	0.0015	0.7540	0.7530	0.6915	0.5174	0.3811	0.0025	0.7509	0.7497	0.6965	0.5162
OmniGuard	0.4415	0.0680	0.7918	0.9024	0.3683	0.5144	0.2699	0.0457	0.4256	0.3932	0.3516	0.2972
GenPTW	0.1345	0.0003	0.8751	0.8755	0.7081	0.5187	0.1322	0.0309	0.6102	0.5923	0.5872	0.3906
Recovery-capable proactive methods												
Imuge	0.8266	0.8383	0.8110	0.7780	0.8197	0.8147	0.7301	0.7459	0.6933	0.6331	0.6869	0.6979
Imuge+	0.6797	0.4409	0.9200	0.8373	0.9154	0.7587	0.1901	0.1406	0.2692	0.1963	0.1479	0.1888
ReImage	0.1296	0.1946	0.9057	0.9020	0.9164	0.6097	0.1071	0.1463	0.4367	0.3429	0.3725	0.2811
ReImage‡	0.5626	0.1198	0.8450	0.8501	0.8744	0.6504	0.3784	0.0771	0.5514	0.5463	0.6595	0.4425
SHIM	0.9345	0.9002	0.9381	0.8910	0.9696	0.9267	0.8798	0.8445	0.8899	0.7919	0.9020	0.8616

256×256 (seed 97; global batch size 8). We train in three segments of 20k, 10k, and 20k updates, carrying over the weights but resetting Adam and its scheduler between segments. Adam uses $\beta = (0.9, 0.5)$, weight decay 10^{-12}, and gradient clipping at 1. The first segment starts at 10^{-5} and multiplies the rate by 0.5 at 10k and 15k updates; the latter two segments use a constant 2.5×10^{-6}. We retain the released objective and loss weights, adding only a clamp–round–dequantize UINT8 STE immediately after IWT and before the released degradation/tampering pipeline. We log the quantization error but do not add it to the loss. We use the 50k adaptation checkpoint, selected in advance rather than from intermediate valAGE-Set results.

D.2 Results by Attack

ReImage‡ denotes the final checkpoint of our 50k-step UINT8-aware adaptation of the released ReImage model. The predicted-mask columns use each method's main evaluation procedure; SHIM uses the 8×8 token permutation. As described in Appendix D.1, these tables combine the earlier baseline results, the separately evaluated epoch-50 Imuge reproduction, and the 20260731/20260732 evaluations of released ReImage, final SHIM, and ReImage‡. The batches follow the same protocol but do not share identical stochastic edits.

The GT-mask columns test recovery with the undilated annotation. Imuge reruns recovery after replacing its predicted mask; Imuge+, ReImage, and ReImage‡ composite their recovery candidates inside the annotation; and SHIM uses the annotation in place of its predicted mask and 8-pixel dilation. These columns examine the effect of the mask within each method. They are not necessarily upper bounds on recovery quality and are not used to rank methods.

Table 9: Per-attack localization F1. Macro is the unweighted average over copy-move (CM), splicing (SP), SDXL inpainting (SDXL), Stable Diffusion inpainting (SD), and local blur.

	Clean						Degradation					
Method	CM	SP	SDXL	SD	Blur	**Macro**	CM	SP	SDXL	SD	Blur	**Macro**
Passive localization methods												
MVSSNet	0.4320	0.4322	0.4914	0.2940	0.4916	0.4282	0.4348	0.4363	0.3765	0.1892	0.1904	0.3255
TruFor	0.7011	0.6708	0.5690	0.2803	0.5141	0.5471	0.6959	0.6678	0.5320	0.2581	0.3554	0.5018
HiFiNet	0.5080	0.5832	0.5218	0.2147	0.5010	0.4657	0.2607	0.3397	0.1124	0.0398	0.0236	0.1552
IML-ViT	0.7297	0.7317	0.6390	0.2682	0.4689	0.5675	0.7254	0.7047	0.5843	0.2401	0.2990	0.5107
SparseViT	0.7388	0.7525	0.7706	0.6330	0.6505	0.7091	0.7217	0.7359	0.6439	0.3430	0.4377	0.5764
SAFIRE	0.6793	0.7195	0.6706	0.5353	0.6576	0.6525	0.5347	0.6133	0.4787	0.2372	0.3878	0.4504
Localization-only proactive methods												
EditGuard-Clean	0.6341	0.0473	0.9308	0.9365	0.9425	0.6982	0.3158	0.1519	0.3985	0.3994	0.4011	0.3333
EditGuard-Degrade	0.4252	0.0029	0.8355	0.8365	0.7838	0.5768	0.4227	0.0049	0.8340	0.8349	0.7885	0.5770
OmniGuard	0.5045	0.1190	0.8565	0.9438	0.4465	0.5741	0.3186	0.0758	0.5118	0.4800	0.4354	0.3643
GenPTW	0.1634	0.0004	0.9221	0.9219	0.7693	0.5554	0.1737	0.0464	0.6951	0.6739	0.6711	0.4520
Recovery-capable proactive methods												
Imuge	0.8852	0.8950	0.8743	0.8465	0.8760	0.8754	0.8039	0.8189	0.7754	0.7174	0.7613	0.7754
Imuge+	0.7444	0.5137	0.9507	0.8919	0.9457	0.8093	0.2469	0.1913	0.3373	0.2587	0.1941	0.2456
ReImage	0.2095	0.2764	0.9406	0.9380	0.9473	0.6623	0.1813	0.2282	0.5501	0.4493	0.4839	0.3786
ReImage‡	0.6121	0.1517	0.8998	0.9018	0.9197	0.6970	0.4321	0.1098	0.6263	0.6183	0.7470	0.5067
SHIM	0.9596	0.9339	0.9632	0.9351	0.9807	0.9545	0.9226	0.8937	0.9324	0.8652	0.9345	0.9097

Table 10: Per-attack full-image recovery PSNR (dB) using predicted and undilated ground-truth masks.

	Predicted mask						GT mask					
Method	CM	SP	SDXL	SD	Blur	**Macro**	CM	SP	SDXL	SD	Blur	**Macro**
Clean												
Imuge	24.928	24.890	24.979	25.069	25.348	25.043	25.460	25.460	25.460	25.460	25.460	25.460
Imuge+	25.699	23.126	28.698	28.356	29.899	27.156	25.872	23.367	28.950	28.582	29.953	27.345
ReImage	21.876	21.821	32.900	33.179	34.140	28.783	34.393	33.367	34.853	34.855	34.859	34.465
ReImage‡	26.827	20.790	30.511	31.137	32.324	28.318	33.063	32.212	33.106	33.108	33.089	32.916
SHIM	31.383	30.022	31.294	31.271	33.348	31.464	30.616	29.639	30.047	28.985	32.186	30.295
Degradation												
Imuge	23.955	23.915	23.956	24.022	24.485	24.066	24.607	24.605	24.609	24.610	24.608	24.608
Imuge+	20.595	19.897	21.093	22.237	26.212	22.007	21.648	20.823	22.275	23.461	27.609	23.163
ReImage	20.790	20.810	23.914	24.197	27.566	23.455	30.497	29.962	30.655	30.655	30.678	30.489
ReImage‡	23.701	20.156	25.415	26.325	28.825	24.885	30.118	29.582	30.144	30.155	30.162	30.032
SHIM	28.367	27.370	28.411	28.232	31.116	28.699	27.893	27.212	27.590	26.645	30.566	27.981

E CROSS-DATASET RETEST PROTOCOL

In Table 3, we reevaluate all three methods on 100 sources each from COCO, ImageNet-1K, and CelebA-HQ (300 sources total). We sample without replacement from each existing 1,000-image pool by ranking source identifiers with SHA-256 using seed 20260925 and taking the first 100; we fix this shared subset and record its hash before inference. All sources are center-cropped 256×256 RGB images, enlarged identically to 512×512 with PIL bilinear interpolation and uint8 rounding before protection. Masks use nearest-neighbor enlargement; copy-move coordinates and local-blur radii are doubled. Original masks, donor mappings, prompts, and attack seeds are retained. Splicing may use donors outside the evaluated subset. Three methods, three datasets, five attacks, and nine conditions give 405 combinations of 100 images, for 40,500 unique image tuples. An independent check before aggregation verified that every combination was complete, no tuple was duplicated, all metrics were finite, and shard hashes matched.

Inference at 512. Imuge+ uses released Rerun3/16999 generator and localizer weights, with floating-point protection at 512. Localization uses $p > 0.5$, scored before the released 16×16 erosion and 32×32 dilation; recovery uses the full inverse RGB output. ReImage uses released 200000_G weights and the BGR path at 512. Its absolute positional embeddings are bicubically interpolated from 64×64 to 128×128 with aligned corners disabled; relative positions use the upstream interpolation. To reduce memory use, we evaluate global attention in blocks of 256 queries, each attending to all keys. We use neither image tiles nor smaller attention windows and checked that the blocked computation is numerically equivalent. We apply no additional clamping or quantization during ReImage protection. Its recovery blends the attacked image and inverse candidate with the predicted soft probability; $p > 0.5$ is used only for localization. We report true F1, whereas the released ReImage quantity named F1 computes recall.

SHIM uses the fixed PSNR39 Grid-0.1 epoch-9 checkpoint, locator, and NAFNet recovery weights. At 512, its latent is $32 \times 16 \times 16$, with all 256 spatial tokens permuted using seed 97. The localization threshold remains 0.03, predicted-mask dilation is 8 pixels, and coarse-mask pooling is 8×8. SHIM's protected images are delivered as uint8. We do not fit weights or thresholds to this subset. Imuge+ was trained at 512; the existing ReImage and SHIM checkpoints transfer from 256 to 512. All methods use predicted support. ReImage applies channels before local tampering, whereas SHIM and Imuge+ apply them afterward.

Attacks and channels. The five edits are copy-move, splicing, SDXL inpainting, SD 1.5 inpainting, and local blur. Masks average 10.74% of the sampled image area. We regenerate the generative edits from each method's protected 512 image using the same models, prompts, 30 steps, guidance 7.5, strength 0.99, and per-source seeds. We reuse each generated replacement across channel conditions and test the nine conditions separately: Clean; JPEG Q90; a 150% bilinear scaling round trip; Gaussian blur (3×3, $\sigma = 1$); median blur (3×3); hue $\mathcal{U}[-0.10, 0.10]$; brightness $\mathcal{U}[0.90, 1.10]$; contrast $\mathcal{U}[0.70, 1.30]$; and AWGN with $\sigma = 1/255$. Clean includes tampering without an additional channel. SHIM and Imuge+ retain the uint8/PIL channel path, including JPEG without chroma subsampling; ReImage retains floating-point channels and released differentiable JPEG with 4:2:0 subsampling and nine-term Fourier rounding on its BGR path. We retain the separate channel conditions of our benchmark; these differ from the original methods' full random or composite channel suites.

Metrics and scope. We compute all localization and image metrics at 512×512 without downsampling. Per-image true F1 uses each method's fixed threshold. For full-image PSNR and SSIM, we clamp image tensors to $[0, 1]$ and compare recovered RGB with the same unmodified 512 source. We use common targets and a common SSIM implementation rather than each method's official choices. We average metrics within each attack, then equally across the five attacks and three datasets. Avg is the mean of JP through AWGN before rounding for display; it excludes Clean and requires no additional evaluations. Protected-image PSNR on this subset is 36.90 dB for Imuge+, 39.65 dB for ReImage, and 39.30 dB for SHIM. Although resolution and metric targets are shared, training, resolution adapters, channel order, and JPEG implementation still differ. Upsampling adds no source detail, and generative edits can differ pixelwise between methods. The results therefore compare these fixed inference settings, not the effect of any single design choice.