EDITSCAN: TOWARDS GENERALIZABLE DETECTION AND LOCALIZATION OF INSTRUCTION-DRIVEN IMAGE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Instruction-driven image editing models can realistically alter local image regions, posing severe challenges to the accuracy and generalization of image editing detection and localization. Existing forensic benchmarks for generative local editing either cover a limited range of instruction-driven editors or are primarily designed around mask-conditioned inpainting,leaving the generalization of forensic methods to rapidly evolving instruction-driven editing models insufficiently evaluated. To address this gap, we introduce **EditScan-Bench** together with a strong baseline approach, **EditScan**. EditScan-Bench is a purpose-built benchmark for instruction-driven image editing forensics, comprising 44K annotated image pairs at 1K resolution, with machine-produced masks that are carefully validated by human experts. It spans 3 widely used open-weight and 5 proprietary model families across four common manipulation types: *Add*, *Remove*, *Replace*, and *Attribute Change*. Building upon EditScan-Bench, we further propose EditScan, which connects vision foundation models with general-purpose segmentation models through specialized adaptation modules to enable both image-level detection and pixel-level localization. EditScan includes two complementary designs: (1) explicitly capturing local patch evidence to increase sensitivity to subtle edits, improving image-level detection; and (2) converting the foundation model's global-local features into forensic prompts that steer the segmentation model toward accurate manipulation localization. In cross-generator tests, EditScan raises average image-level detection F1 and Accuracy by 0.12 ($0.64 \rightarrow 0.76$) and 0.08 ($0.74 \rightarrow 0.82$), respectively, compared with the strongest baseline. For pixel-level localization, it achieves improvements of 0.22 F1 ($0.29 \rightarrow 0.51$) and 0.21 IoU ($0.26 \rightarrow 0.47$), while also showing stronger robustness to post-processing degradations. The benchmark dataset and code are available at `https://anonymous.4open.science/r/EditScan`.

1 INTRODUCTION

Recent advances in multimodal large language models and diffusion models enable high-quality, instruction-dirven local edits to real images. Compared with traditional splicing, inpainting, and copy-move, generative editing can flexibly change specified objects, backgrounds, or attributes while preserving unedited regions. As these tools spread across desktop and mobile platforms, edited content is increasingly common and can be misused (e.g., fake news or forged documents). This creates an urgent need to detect generative edits and localize edited regions at the pixel level.

Although substantial progress has been made in generated-image detection, three major challenges remain for emerging instruction-driven generative editing.

First, high-quality datasets for instruction-driven generative editing forensics remain scarce. Local generative editing typically follows two paradigms: (1) inpainting, where users predefine a region and the model fills it based on a text prompt; and (2) instruction-driven editing, where no region is preset and the model determines the edit area from the instruction. Most datasets are built for inpainting and do not reflect instruction-driven workflows. Existing instruction-based datasets cover few mainstream generators and often lack pixel-level masks of edited regions. Therefore, a large-

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.

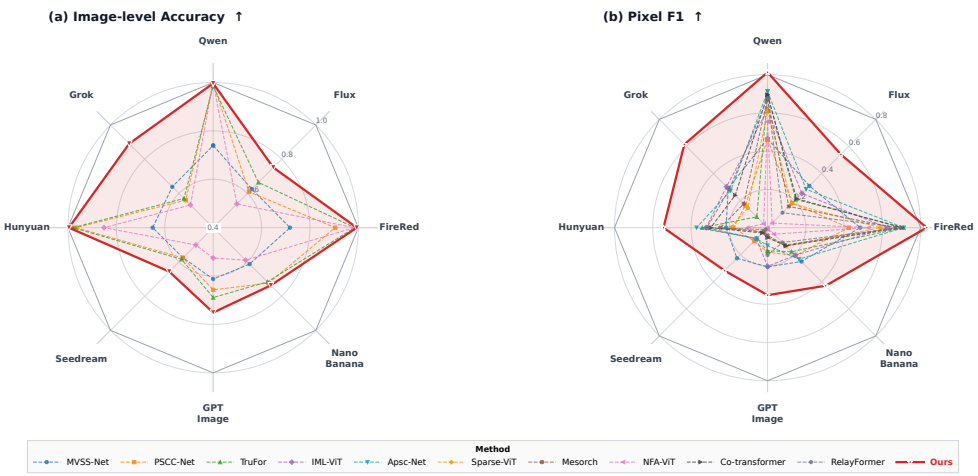


Figure 1: Cross-generator performance on eight generators:(a) image-level detection accuracy; (b) pixel-level F1 for edited-region localization.

scale dataset is needed that mirrors realistic instruction-driven editing, spans mainstream models and common edit operations, and provides accurate pixel-level annotations.

Second, local editing is difficult to detect because its generative traces are weak and spatially scattered. Most generated-image detection targets fully synthetic images. Local edits change only parts of an authentic image, so small edited regions may be missed when global features dilute generative traces, while focusing only on a few salient patches can ignore overall context. Effective detection should combine stable global semantics with explicit patch-level evidence to improve the final prediction.

Third, pretrained visual representation and segmentation models lack forensic adapters, hindering use of their complementary strengths for generative editing forensics. Vision foundation models such as DINOv3 Siméoni et al. (2025) capture global semantics and dense local details, and segmentation foundation models such as SAM Kirillov et al. (2023) provide robust general-purpose segmentation. However, efficient adapters to integrate them for generative image editing forensics remain unexplored.

To address these challenges, our main contributions in dataset construction, local generative editing detection, and localization are summarized as follows:

- We construct an annotation pipeline and benchmark dataset, **EditScan-Bench**, for instruction-driven editing. The pipeline uses a vision-language model to interpret source images and propose edit regions/objects. It supports four mask-free edits and auto-generates pixel-level masks. EditScan-Bench includes 44K annotated 1K-resolution image pairs spanning eight major instruction-driven image editing model families.
- We propose **Patch-Artifact-Aware Detection Enhancement** for generative editing detection. Based on the vision foundation model DINOv3, we introduce Patch Evidence Scoring to quantify local traces, Patch-Aware Fusion to aggregate patch evidence into a patch logit, and Learnable Residual Scaling to refine the global CLS logit with a learnable weight conditioned on the patch logit, which improves image-level detection performance.
- We propose a **Localization Prompt Generator** that efficiently connects DINOv3 and SAM to localize edited regions. The module uses the DINOv3 CLS token to condition learnable queries, applies cross-attention over patch tokens to gather local evidence for likely edited regions, and maps the result into SAM's prompt space as DINO-guided prompts. Unlike methods that predict a coarse mask before generating box/point prompts, it directly converts DINOv3 global-local features into SAM-ready forensic prompts, yielding a tighter, more accurate link between the models.

2 Related Work

2.1 Image Manipulation Datasets and Benchmarks

Early benchmarks for image manipulation, including Columbia Hsu & Chang (2006), CASIA Dong et al. (2013), and IMD2020 Novozámskỳ et al. (2020), largely focus on traditional tampering operations such as splicing, copy-move, and object removal. As diffusion models and text-guided generation became prevalent, later benchmarks broadened image forensics to cover generative, local editing Guillaro et al. (2023); Jia et al. (2023); Sun et al. (2024); Mareen et al. (2024); Chen et al. (2025). More recent datasets push the scale and variety of generative edits even further Wang et al. (2025b); Huang et al. (2025a); Cai et al. (2026); Zhang et al. (2025b). AutoEditForge Lek et al. (2026) and Impostor Li et al. (2026) present automated pipelines that track and adapt to evolving generative models, whereas LocateEdit-Bench Wu et al. (2026) concentrates on instruction-following editing. In this work, we develop a multi-generator benchmark that spans both open-weight and proprietary editors, and we provide a unified evaluation of image-level detection, pixel-level localization, and cross-generator generalization.

2.2 Image Manipulation Detection and Localization

Image Manipulation Detection and Localization (IMDL) requires both image-level authenticity prediction and pixel-level localization. Early methods rely on low-level forensic cues (e.g., boundaries, noise, compression, frequency artifacts) Chen et al. (2021); Kwon et al. (2021); Liu et al. (2022); Guillaro et al. (2023). Transformer-based approaches capture long-range dependencies and fine-grained evidence via self-attention, multi-scale features, and non-semantic representations Hao et al. (2021); Wang et al. (2022); Ma et al. (2023); Qu et al. (2024); Su et al. (2025); Zhu et al. (2025); Nam et al. (2025). Recent work further combines local evidence with global context to detect subtle edits Cai et al. (2026); Zhang et al. (2026); Huang et al. (2026).

Vision foundation models are being explored for forensics due to transferable global and dense representations. DINOv3 Siméoni et al. (2025) provides complementary global/local features that support transfer and cross-distribution generalization Huang et al. (2025b); Zhou et al. (2026); Yu et al. (2026). Vision-language and multimodal models have also been studied for forensic representation, localization, and interpretation Li et al. (2024); Xu et al. (2025); Huang et al. (2025a); Sun et al. (2025).

SAM Kirillov et al. (2023) is also promising for manipulation localization. Prior methods adapt SAM using spatial prompts, model tuning, or automatically generated forensic prompts Lai et al. (2023); Su et al. (2024); Wang et al. (2025a). SAFIRE Kwon et al. (2025) uses point prompts, while IMDPrompter Zhang et al. (2025a), Detective SAM Lek et al. (2026), and SARIF Moon et al. (2026) generate prompts from forensic evidence; LISA Lai et al. (2024) instead maps high-level semantics to embedding-level prompts.

Nonetheless, detecting and localizing generative editing remains challenging, as it requires careful analysis of both global and local generative artifacts in the target images, along with effective integration and use of vision foundation models and segmentation models pre-trained on large-scale datasets to achieve better detection and localization performance.

3 EditScan-Bench

3.1 Dataset Construction Pipeline

As illustrated in Figure 2, the data generation and annotation pipeline consists of Three stages: source preparation, instruction-driven mask-free image editing, and pixel-level annotation and human check.

Source preparation. The source images are drawn from SA-1B Kirillov et al. (2023) and FFHQ Karras et al. (2019). SA-1B covers diverse scenes, but many faces are mosaicked for privacy protection; we therefore include FFHQ to supplement high-resolution human-centered scenes. To reduce interference from pre-existing visual artifacts, we automatically remove low-resolution

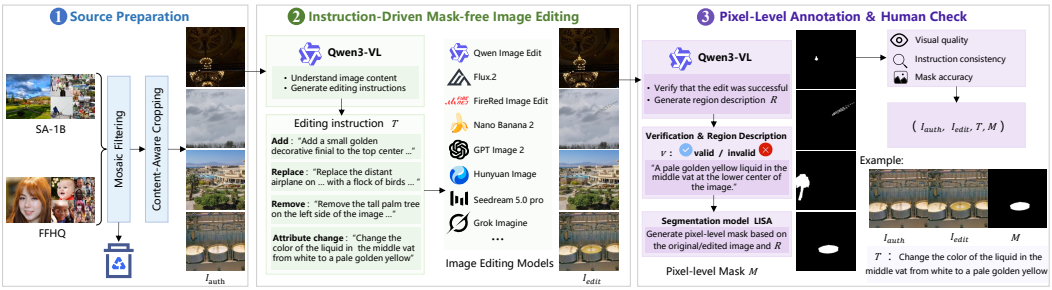


Figure 2: EditScan-Bench construction pipeline.

images and samples containing large privacy mosaics, and standardize all images to RGB. We then perform content-aware cropping by scoring candidate regions according to edges, corners, and intensity variation, suppressing large smooth backgrounds while retaining detail-rich content. All outputs are finally standardized to 1024×1024 to form the authentic image set.

Instruction-driven mask-free image editing. For each authentic image I_{auth}, we use the vision-language model Qwen3-VL Bai et al. (2025) to understand the scene, objects, attributes, and spatial relations, and to generate a natural-language local editing instruction T consistent with the image content. The instructions cover four edit types: *Add*, *Remove*, *Replace*, and *Attribute Change*. We then feed I_{auth} and T to an editing model G to obtain the edited image I_{edit}. No pixel-level mask is provided during editing; the actual edited region is determined solely by each editor's instruction understanding and generation mechanism.

Pixel-level annotation and human check. We prompt Qwen3-VL to jointly analyze the authentic image I_{auth}, edited image I_{edit}, and instruction T. It first verifies whether the visual difference between I_{auth} and I_{edit} is consistent with T, thereby checking whether the requested edit was successfully executed. For valid samples, Qwen3-VL generates a spatial and semantic description R of the actual changed region, which is then used to guide the instruction-following segmentation model LISA to produce a pixel-level mask M. For *Remove*, the removed content exists only in the original image, so I_{auth} is used as the segmentation reference; all other edit types are segmented on I_{edit}. Expert reviewers further verify visual quality, instruction consistency, and mask accuracy, discarding invalid samples. Each retained sample is represented as $(I_{\text{auth}}, I_{\text{edit}}, T, M)$.

3.2 Dataset Composition

EditScan-Bench contains 44K annotated pairs at 1K resolution, covering eight editor families and four editing operations (Table 1).

The 30K Qwen-Image-Edit pairs are split 8:1:1 into 24K training, 3K validation, and 3K in-domain test samples. Each of the other seven editor families contributes 2K samples to an Out-Of-Domain (OOD) test subset. Authentic source images are disjoint between training and all OOD subsets to avoid content overlap. Evaluation protocols for image-level detection and pixel-level localization are described in Section 5.1.

4 EditScan for Detecting and Localizing Generative Editing

4.1 Overall Framework

As shown in Figure 3, EditScan is built on a shared DINOv3 backbone with two task-specific branches:one for image-level detection and another for pixel-level localization. For a given input image, DINOv3 produces a global CLS token $\mathbf{C} \in \mathbb{R}^d$ along with local patch tokens $\mathbf{P} \in \mathbb{R}^{N \times d}$. The detection branch strengthens the CLS-based prediction through Patch-Artifact-Aware Detection Enhancement, whereas the localization branch maps the combined global and local features into SAM-compatible forensic prompts.

Table 1: Composition of EditScan-Bench.

Access Type	Provider	Editing Models	# Pairs	Usage
Open-weight	Qwen	Qwen-Image-Edit-2509; Qwen-Image-Edit-2511	30,000	Train/Val/Test
	Black Forest Labs	FLUX.2 [dev]; FLUX.2 [klein]	2,000	OOD Test
	RedNote	FireRed-Image-Edit-1.1	2,000	OOD Test
Proprietary	Google	Nano Banana; Nano Banana Pro; Nano Banana 2	2,000	OOD Test
	OpenAI	GPT Image 1.5; GPT Image 2	2,000	OOD Test
	Tencent	Hunyuan Image 3.0	2,000	OOD Test
	XAI	Grok-imagine	2,000	OOD Test
	ByteDance	Seedream 4.0; Seedream 5.0 Pro	2,000	OOD Test

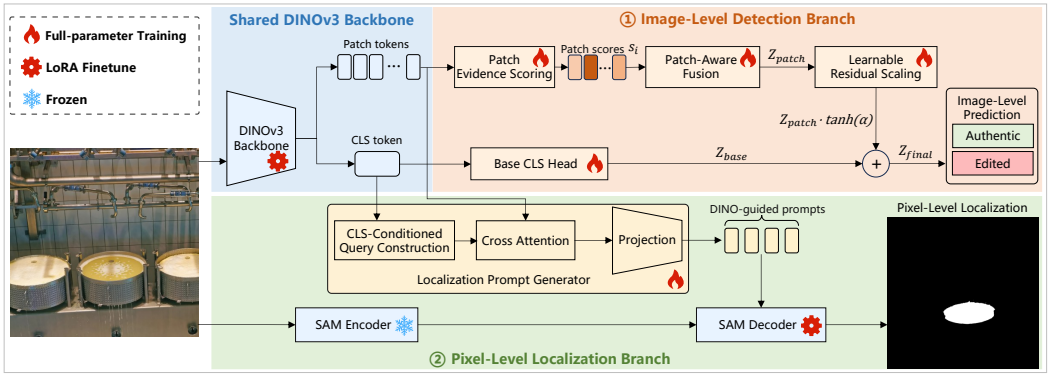


Figure 3: EditScan overview: a shared DINOv3 backbone extracts a global CLS token and local patch tokens. The image-level detection branch scores patch evidence, fuses it, and applies Learnable Residual Scaling to refine the CLS prediction. The pixel-level localization branch uses a Localization Prompt Generator to turn global-local features into forensic prompts that guide SAM to predict the edited-region mask. Flame, gear, and snowflake indicate full-parameter training, LoRA Hu et al. (2021) adaptation, and frozen modules, respectively.

4.2 Patch-Artifact-Aware Detection Enhancement

Local edits often occupy only a small region, causing their forensic traces to be diluted in global representations. We therefore complement the CLS-based prediction with explicit patch-level forensic evidence. The CLS token produces a base logit $z_{\text{base}} = \mathbf{w}_{\text{cls}}^{\top}\mathbf{C} + b_{\text{cls}}$, while the patch tokens produce a patch-aware logit z_{patch}. They are combined through a learnable residual:

$$z_{\text{final}} = z_{\text{base}} + \tanh(\alpha)\, z_{\text{patch}}, \tag{1}$$

where α is a learnable scalar and $\tanh(\alpha) \in (-1, 1)$ bounds the contribution of patch evidence.

1) Patch Evidence Scoring. For each patch token $\mathbf{p}_i$, we normalize its feature $\widetilde{\mathbf{p}}_i = \mathrm{LN}_p(\mathbf{p}_i)$ and compute its editing response:

$$s_i = g_{\text{score}}\left(\widetilde{\mathbf{p}}_i\right), \qquad i = 1, \ldots, N, \tag{2}$$

where g_{score} is a lightweight two-layer GELU MLP, and s_i denotes the local editing evidence for patch i.

2) Patch-Aware Fusion. A single aggregation operator may not fully capture both sparse local artifacts and global patch statistics, so we use three complementary aggregations: *score-weighted pooling* (weights patches by forensic scores), *Top-K pooling* (keeps the strongest local evidence), and *mean pooling* (captures overall patch statistics).

Score-weighted coefficients are

$$\omega_i = \frac{\exp(s_i/\tau)}{\sum_{j=1}^{N} \exp(s_j/\tau)}, \tag{3}$$

where τ is the temperature. For Top-K pooling, we average the normalized features of the K highest-scoring patches. Let $s_{(i)}$ and $\widetilde{\mathbf{p}}_{(i)}$ be the i-th largest score and its corresponding normalized patch feature. The pooled features are

$$\mathbf{h}_s = \sum_{i=1}^{N} \omega_i \widetilde{\mathbf{p}}_i, \qquad \mathbf{h}_t = \frac{1}{K} \sum_{i=1}^{K} \widetilde{\mathbf{p}}_{(i)}, \qquad \mathbf{h}_m = \frac{1}{N} \sum_{i=1}^{N} \widetilde{\mathbf{p}}_i. \tag{4}$$

To retain the strength of the strongest local evidence, we additionally use the mean Top-K scores $\bar{s}_K = \frac{1}{K} \sum_{i=1}^{K} s_{(i)}$. We fuse these local representations with the global CLS feature as

$$\mathbf{u} = [\phi_c(\mathrm{LN}_c(\mathbf{C})); \phi_p(\mathbf{h}_s); \phi_p(\mathbf{h}_t); \phi_p(\mathbf{h}_m); \bar{s}_K], \qquad z_{\mathrm{patch}} = g_{\mathrm{fusion}}(\mathbf{u}), \tag{5}$$

where ϕ_c and ϕ_p map global and local features to a shared space, and g_{fusion} is a LayerNorm-MLP fusion network.

3) Learnable Residual Scaling. In Eq. equation 1, the learnable scalar α scales the residual contribution of patch evidence. With binary cross-entropy, its gradient is

$$\frac{\partial \mathcal{L}_{\mathrm{cls}}}{\partial \alpha} = [\sigma(z_{\mathrm{final}}) - y]\, z_{\mathrm{patch}} \left[1 - \tanh^2(\alpha)\right], \tag{6}$$

where $y \in \{0, 1\}$ is the image-level label. The gradient depends on the classification error, patch response, and residual state. α is optimized with other parameters using AdamW to learn how much patch evidence refines the CLS prediction. The detection probability is $\widehat{y} = \sigma(z_{\mathrm{final}})$.

4.3 Localization Prompt Generator

DINOv3 offers complementary global-local features, while SAM needs effective localization prompts. We introduce a *Localization Prompt Generator* that maps edit-related DINOv3 representations directly to SAM-compatible forensic prompts, avoiding coarse masks or spatial prompts. It comprises CLS-conditioned query construction, local evidence retrieval, and prompt-space projection.

1) CLS-Conditioned Query Construction. To retrieve evidence of edited regions from dense patch tokens, we use K_q learnable forensic queries $\mathbf{Q}_0 \in \mathbb{R}^{K_q \times d}$. Because fixed queries lack image-specific context, we condition them on the global CLS token:

$$\mathbf{Q} = \mathbf{Q}_0 + \mathbf{1}_{K_q} (W_c \mathbf{C})^\top, \tag{7}$$

where W_c is a learnable linear projection and $\mathbf{1}_{K_q}$ is an all-ones vector of length K_q. Thus, $\mathbf{Q}$ combines a learnable query prior with global image context.

2) Cross-Attention. Conditioned queries extract local forensic evidence from DINOv3 patch tokens via multi-head cross-attention (MHA):

$$\mathbf{Z} = \mathrm{LN}\left(\mathrm{MHA}(\mathbf{Q}, \mathbf{P}, \mathbf{P})\right), \tag{8}$$

where the conditioned queries attend to diverse patch responses to aggregate complementary editing cues. We then refine $\mathbf{Z}$ through a residual Linear–GELU transformation followed by Layer Normalization: $\mathbf{Z}' = \mathrm{LN}(\mathbf{Z} + \mathrm{GELU}(\mathbf{Z} W_f + \mathbf{b}_f))$, where W_f and $\mathbf{b}_f$ are learnable parameters.

3) Prompt Space Projection. The retrieved representation $\mathbf{Z}'$ is then linearly projected via $\psi(\cdot)$ into SAM's prompt space to guide mask prediction.

$$\mathbf{\Pi} = \psi(\mathbf{Z}'), \qquad \widehat{M} = G_{\mathrm{SAM}}(x, \mathbf{\Pi}), \tag{9}$$

where $\mathbf{\Pi} \in \mathbb{R}^{K_q \times d_{\mathrm{SAM}}}$ denotes the generated forensic prompts (d_{SAM} is the SAM prompt-embedding dimension), and $\widehat{M}$ is the predicted edited-region mask.

Table 2: Image-level Accuracy across eight domains. Qwen is in-domain; the other seven are OOD.

Method	Qwen	Flux	FireRed	Nano	GPT	Seedream	Hunyuan	Grok	OOD Avg.
MVSS-Net (21)	0.7397	0.6258	0.7175	0.6128	0.6123	0.5760	0.6500	0.6388	0.6333
PSCC-Net (22)	0.9978	0.6092	0.9043	0.7238	0.6570	0.5770	0.9663	0.5605	0.7140
TruFor (23)	0.9847	0.6645	0.9853	0.7168	0.6888	0.5848	0.9790	0.5705	0.7414
NFA-ViT (26)	**0.9988**	0.5385	0.9828	0.5898	0.5255	0.5012	0.8510	0.5325	0.6459
Ours	0.9963	**0.7515**	**0.9935**	**0.7373**	**0.7508**	**0.6567**	**0.9960**	**0.8908**	**0.8252**

4.4 Joint Loss for Edit Detection and Localization

EditScan is jointly supervised for image-level detection and pixel-level localization. Pixel-level supervision is applied to all training samples: edited images use their ground-truth masks, while authentic images use all-zero masks to suppress false localization responses.

The image-level loss $\mathcal{L}_{\text{cls}}$ uses binary cross-entropy (BCE), while the localization loss $\mathcal{L}_{\text{mask}}$ combines pixel-wise BCE and Dice loss. The overall objective is

$$\mathcal{L} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}}, \tag{10}$$

where λ_{cls} and λ_{mask} weight the two loss terms and are both set to 1.0.

5 Experiments

5.1 Experimental Setup

Baselines and protocol. We compare with MVSS-Net Chen et al. (2021), PSCC-Net Liu et al. (2022), TruFor Guillaro et al. (2023), IML-ViT Ma et al. (2023), APSC-Net Qu et al. (2024), Sparse-ViT Su et al. (2025), Mesorch Zhu et al. (2025), NFA-ViT Cai et al. (2026), Co-Transformers Zhang et al. (2026), and RelayFormer Huang et al. (2026), all with publicly available code. We evaluate fixed checkpoints on the same eight test domains, without domain-specific adaptation. Following IMDL-BenCo Ma et al. (2024), "DH" (Detection Head) denotes whether a method provides image-level predictions; image-level metrics are reported only for methods with "DH".

Implementation details. EditScan is optimized with AdamW using a per-GPU batch size of 16, an initial learning rate of 10^{-4}, weight decay of 0.01, and a 2-epoch warm-up. The input resolutions of DINOv3 and SAM are 512×512 and 1024×1024, respectively. The Localization Prompt Generator uses $K_q = 4$ query tokens, and rank-32 LoRA is applied to both DINOv3 and the SAM mask decoder. All experiments use seed 42 for reproducibility.

Metrics. Pixel F1 and Pixel IoU are evaluated on edited images with valid masks; Image F1 and Accuracy are evaluated on both authentic and edited images. Qwen is the in-domain test set, and OOD Avg. is the macro average across seven unseen generators. Due to space constraints, we mainly report Accuracy and Pixel F1; full Image F1 and Pixel IoU results are in Appendix B. All metrics use a 0.5 threshold.

5.2 Performance Comparison

Image-level detection comparison. Table 2 shows that our method achieves the best image-level accuracy across domains, especially out of domain. It reaches 0.9963 on the in-domain Qwen dataset and attains the highest average OOD accuracy of 0.8252, outperforming TruFor (0.7414) and PSCC-Net (0.7140). This advantage is consistent across OOD domains (Flux, FireRed, Nano, GPT, Seedream, Hunyuan, and Grok), where it often achieves the top score, indicating strong generalization to unseen images. Full Image F1 results are reported in Appendix B.

Pixel-level localization comparison. Table 3 shows that our method achieves the best Pixel F1 in both in-domain and out-of-domain (OOD) settings. In-domain on Qwen, it reaches 0.8120, outperforming APSC-Net (0.7123) and Co-Transformers (0.6942), indicating strong performance on data

Table 3: Pixel F1 across eight domains; DH indicates image-level detection. Qwen is in-domain; the other seven are OOD.

Method	DH	Qwen	Flux	FireRed	Nano	GPT	Seedream	Hunyuan	Grok	OOD Avg.
MVSS-Net (21)	✓	0.4651	0.3088	0.4845	0.2499	0.2051	0.2261	0.2168	0.2908	0.2831
PSCC-Net (22)	✓	0.4616	0.1826	0.4245	0.2065	0.1281	0.0995	0.1786	0.1640	0.1977
TruFor (23)	✓	0.6927	0.2085	0.7108	0.1779	0.1211	0.0812	0.3412	0.0798	0.2458
IML-ViT (23)	×	0.6150	0.2537	0.6263	0.2039	0.2021	0.0825	0.3211	0.3034	0.2847
APSC-Net (24)	×	0.7123	0.2844	0.7115	0.2324	0.0937	0.0803	0.3712	0.2731	0.2924
SparseViT (25)	×	0.6119	0.1781	0.5899	0.1375	0.0472	0.0301	0.1936	0.1457	0.1889
Mesorch (25)	×	0.6736	0.1568	0.6689	0.1286	0.0407	0.0320	0.2467	0.1760	0.2071
NFA-ViT (26)	✓	0.5541	0.0335	0.5272	0.0481	0.0097	0.0021	0.1054	0.0290	0.1079
Co-Transformers (26)	×	0.6942	0.2142	0.7015	0.1348	0.0521	0.0392	0.2973	0.2393	0.2398
RelayFormer (26)	×	0.6757	0.1103	0.6949	0.1097	0.1430	0.0141	0.3171	0.2832	0.2389
Ours	✓	**0.8120**	**0.5416**	**0.8315**	**0.4293**	**0.3529**	**0.3175**	**0.5440**	**0.6152**	**0.5189**

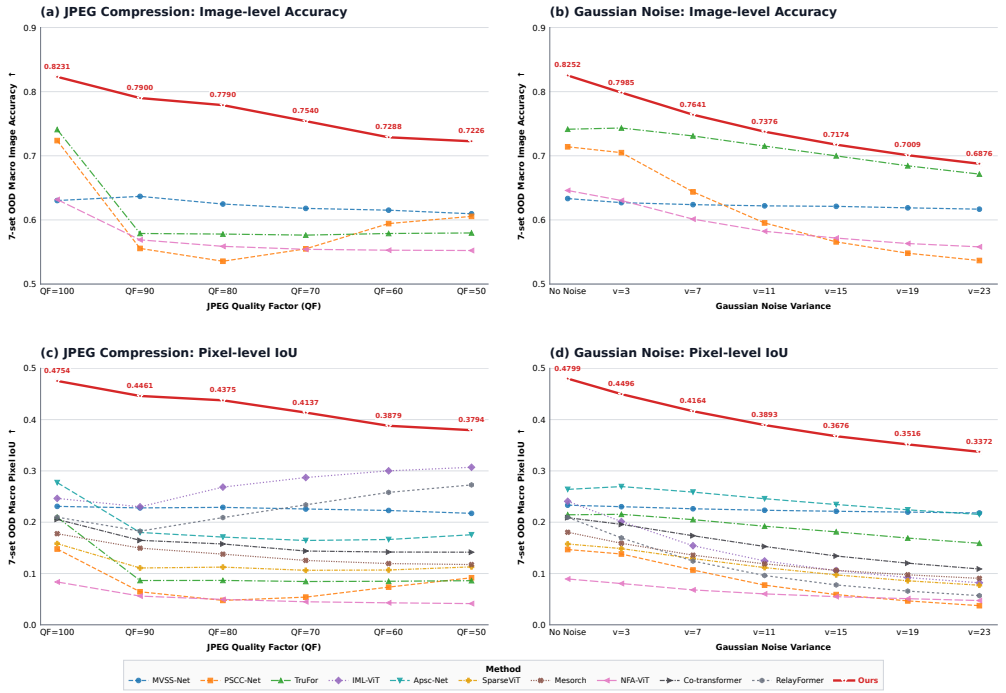


Figure 4: Cross-generator robustness to JPEG compression and Gaussian noise. (a,b) OOD image accuracy; (c,d) OOD pixel IoU under the same degradations.

close to the training distribution. For OOD generalization, it leads across all seven domains with an average F1 of 0.5189, well above APSC-Net (0.2924), IML-ViT (0.2847), and MVSS-Net (0.2831). These consistent gains demonstrate strong generalizability on unseen data. Full Pixel IoU results are provided in Appendix B.

Robustness against Image Degradation We evaluate robustness to JPEG compression and Gaussian noise. Figure 4 reports OOD Accuracy and Pixel IoU; per-domain results are in Appendix C.

As shown in Figure 4, EditScan consistently delivers the best detection and localization robustness across all degradation levels. Under the strongest degradations, it achieves Accuracy/Pixel IoU of $0.7226/0.3794$ at $QF = 50$ and $0.6876/0.3372$ at $v = 23$, indicating stable cross-generator robustness to common image degradations.

Table 4: Ablation of patch-artifact-aware detection enhancement. "Qwen" is in-domain; the other seven are OOD.

Metric	Variant	Qwen	Flux	FireRed	Nano	GPT	Seedream	Hunyuan	Grok	OOD Avg.
Pixel F1	w/o Patch-aware	0.8117	0.4999	0.8313	0.3784	0.2868	0.2720	0.5399	0.5675	0.4823
	w/ Patch-aware	**0.8120**	**0.5416**	**0.8315**	**0.4293**	**0.3529**	**0.3175**	**0.5440**	**0.6152**	**0.5189**
Pixel IoU	w/o Patch-aware	**0.7542**	0.4614	**0.7770**	0.3488	0.2635	0.2462	0.5001	0.5213	0.4455
	w/ Patch-aware	0.7527	**0.5046**	0.7749	**0.3976**	**0.3244**	**0.2887**	**0.5017**	**0.5673**	**0.4799**
Image F1	w/o Patch-aware	0.9963	0.6485	0.9917	0.5224	0.4604	0.4409	0.9894	0.8122	0.6951
	w/ Patch-aware	0.9963	**0.6713**	**0.9935**	**0.6479**	**0.6708**	**0.4821**	**0.9960**	**0.8782**	**0.7628**
Accuracy	w/o Patch-aware	0.9963	0.7393	0.9918	0.6750	0.6478	0.6405	0.9895	0.8415	0.7893
	w/ Patch-aware	0.9963	**0.7515**	**0.9935**	**0.7373**	**0.7508**	**0.6567**	**0.9960**	**0.8908**	**0.8252**

Table 5: Ablation of the Localization Prompt Generator. "Qwen" is in-domain; the other seven are OOD.

Metric	Variant	Qwen	Flux	FireRed	Nano	GPT	Seedream	Hunyuan	Grok	OOD Avg.
Pixel F1	SAM-only	0.3652	0.4132	0.4348	0.3948	0.3469	**0.3221**	0.2638	0.3594	0.3621
	EditScan	**0.8120**	**0.5416**	**0.8315**	**0.4293**	**0.3529**	0.3175	**0.5440**	**0.6152**	**0.5189**
Pixel IoU	SAM-only	0.3088	0.3545	0.3706	0.3343	0.2933	0.2723	0.2171	0.3082	0.3072
	EditScan	**0.7527**	**0.5046**	**0.7749**	**0.3976**	**0.3244**	**0.2887**	**0.5017**	**0.5673**	**0.4799**

5.3 Ablation Study

Patch-Artifact-Aware Detection Enhancement. We build a *w/o Patch-aware* variant by removing Patch Evidence Scoring, Patch-Aware Fusion, and Learnable Residual Scaling, leaving only the DINOv3 CLS classifier; all other settings remain unchanged. Table 4 shows that the patch-aware branch raises OOD Image F1 from 0.6951 to 0.7628 and Accuracy from 0.7893 to 0.8252, suggesting patch-level forensic evidence improves cross-generator image detection without degrading pixel-level localization.

Localization Prompt Generator. We compare against a *SAM-only* baseline with the same SAM ViT-H, pixel-level supervision, and LoRA-adapted mask decoder, but without DINOv3 or the Localization Prompt Generator. Table 5 shows that DINOv3-guided prompts improve OOD Pixel F1 from 0.3621 to 0.5189 and Pixel IoU from 0.3072 to 0.4799, confirming the benefit of mapping DINOv3 global-local representations into SAM-compatible forensic prompts.

6 Conclusion

We present EditScan-Bench and EditScan to detect and localize instruction-driven local image edits. EditScan-Bench contains 44,000 annotated image pairs covering 8 generative editors and 4 edit types, with image-level labels and pixel-level masks. EditScan uses DINOv3 global-local features with two modules: Patch-Artifact-Aware Detection Enhancement for stronger image-level detection from patch evidence, and a Localization Prompt Generator that converts edit features into SAM-compatible prompts for precise localization. Results show strong generalization to unseen generators and robustness to degradations such as JPEG compression and Gaussian noise.

AI USE STATEMENT

Generative AI tools were used in this work for two purposes. First, generative image-editing models were used to create the edited images in EditScan-Bench, as detailed in the dataset creation process and the experimental configuration. Second, AI tools supported writing and editorial refinement of the manuscript. All AI-assisted text was examined and validated by the authors. The authors take full responsibility for the final content of the paper and for the artifacts generated with the assistance of generative AI.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Lvpan Cai, Haowei Wang, Jiayi Ji, Yanshu Zhoumen, Shen Chen, Taiping Yao, and Xiaoshuai Sun. Zooming in on fakes: A novel dataset for localized ai-generated image detection with forgery amplification approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2534–2542, 2026.

Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14165–14173. IEEE, 2021.

Yirui Chen, Xudong Huang, Quan Zhang, Wei Li, Mingjian Zhu, Qiangyu Yan, Simiao Li, Hanting Chen, Hailin Hu, Jie Yang, et al. Gim: A million-scale benchmark for generative image manipulation detection and localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 2311–2319, 2025.

Jing Dong, Wei Wang, and Tieniu Tan. Casia image tampering detection evaluation database. In *2013 IEEE China summit and international conference on signal and information processing*, pp. 422–426. IEEE, 2013.

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

Jing Hao, Zhixin Zhang, Shicai Yang, Di Xie, and Shiliang Pu. Transforensics: image forgery localization with dense self-attention. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15035–15044. IEEE, 2021.

Yu-Feng Hsu and Shih-Fu Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *2006 IEEE international conference on multimedia and expo*, pp. 549–552. IEEE, 2006.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Wen Huang, Jiarui Yang, Tao Dai, Jiawei Li, Shaoxiong Zhan, Bin Wang, and Shu-Tao Xia. Relayformer: A unified local-global attention framework for scalable image and video manipulation localization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=e61YQdLIam.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28831–28841. IEEE, 2025a.

Zhenglin Huang, Jason Li, Haiquan Wen, Tianxiao Li, Xi Yang, Lu Qi, Bei Peng, Xiaowei Huang, Ming-Hsuan Yang, and Guangliang Cheng. Rethinking cross-generator image forgery detection through dinov3. *arXiv preprint arXiv:2511.22471*, 2025b.

Shan Jia, Mingzhen Huang, Zhou Zhou, Yan Ju, Jialing Cai, and Siwei Lyu. Autosplice: A text-prompt manipulated image dataset for media forensics. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 893–903. IEEE, 2023.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 4396–4405. IEEE, 2019.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Myung-Joon Kwon, In-Jae Yu, Seung-Hun Nam, and Heung-Kyu Lee. Cat-net: Compression artifact tracing network for detection and localization of image splicing. In *2021 IEEE winter conference on applications of computer vision (WACV)*, pp. 375–384. IEEE, 2021.

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.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9579–9589. IEEE, 2024.

Yingxin Lai, Zhiming Luo, and Zitong Yu. Detect any deepfakes: Segment anything meets face forgery detection and localization. In *Chinese conference on biometric recognition*, pp. 180–190. Springer, 2023.

Gert Lek, Nicolas Van Schaik, Chaoyi Zhu, Pin-Yu Chen, Robert Birke, and Lydia Chen. Detective sam: Adaptive ai-image forgery localization. In *International Conference on Learning Representations*, volume 2026, pp. 22463–22490, 2026.

Dong Li, Jiaying Zhu, Xueyang Fu, Xun Guo, Yidi Liu, Gang Yang, Jiawei Liu, and Zheng-Jun Zha. Noise-assisted prompt learning for image forgery detection and localization. In *European conference on computer vision*, pp. 18–36. Springer, 2024.

Zhenliang Li, Yutao Hu, Qixiong Wang, Wenpeng Du, Hongxiang Jiang, Jiasong Wu, Xiaolong Jiang, and Jungong Han. Impostor: An agent-curated benchmark for realistic aigc manipulation localization. *arXiv preprint arXiv:2606.04545*, 2026.

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.

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.

Xiaochen Ma, Xuekang Zhu, Lei Su, Bo Du, Zhuohang Jiang, Bingkui Tong, Zeyu Lei, Xinyu Yang, Chi-Man Pun, Jiancheng Lv, et al. Imdl-benco: A comprehensive benchmark and codebase for image manipulation detection & localization. *Advances in Neural Information Processing Systems*, 37:134591–134613, 2024.

Hannes Mareen, Dimitrios Karageorgiou, Glenn Van Wallendael, Peter Lambert, and Symeon Papadopoulos. Tgif: Text-guided inpainting forgery dataset. In *2024 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6. IEEE, 2024.

Dong-Hyun Moon, Ju-Hyeon Nam, and Sang-Chul Lee. Sarif: Segment anything for robust image forensics. *arXiv preprint arXiv:2606.21108*, 2026.

Ju-Hyeon Nam, Dong-Hyun Moon, and Sang-Chul Lee. M2sformer: Multi-spectral and multi-scale attention with edge-aware difficulty guidance for image forgery localization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15927–15938. IEEE, 2025.

Adam Novozámskỳ, Babak Mahdian, and Stanislav Saic. Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 71–80. IEEE, 2020.

Chenfan Qu, Yiwu Zhong, Chongyu Liu, Guitao Xu, Dezhi Peng, Fengjun Guo, and Lianwen Jin. Towards modern image manipulation localization: A large-scale dataset and novel methods. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10781–10790. IEEE, 2024.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

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 spare-coding transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 7024–7032, 2025.

Yang Su, Shunquan Tan, and Jiwu Huang. A novel universal image forensics localization model based on image noise and segment anything model. In *Proceedings of the 2024 ACM Workshop on Information Hiding and Multimedia Security*, pp. 149–158, 2024.

Zhihao Sun, Haipeng Fang, Juan Cao, Xinying Zhao, and Danding Wang. Rethinking image editing detection in the era of generative ai revolution. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 3538–3547, 2024.

Zhihao Sun, Haoran Jiang, Haoran Chen, Yixin Cao, Xipeng Qiu, Zuxuan Wu, and Yu-Gang Jiang. Forgerysleuth: Empowering multimodal large language models for image manipulation detection. 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. 153136–153175. Curran Associates, Inc., 2025. doi: 10.52202/085713-5121. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/e0e25d425450b6fc8e34380de71b3aee-Paper-Conference.pdf.

Junke Wang, Zuxuan Wu, Jingjing Chen, Xintong Han, Abhinav Shrivastava, Ser-Nam Lim, and Yu-Gang Jiang. Objectformer for image manipulation detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2364–2373, 2022.

Xinghao Wang, Tao Gong, Qi Chu, Bin Liu, and Nenghai Yu. Context-aware weakly supervised image manipulation localization with sam refinement. *IEEE Signal Processing Letters*, 32:3047–3051, 2025a.

Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. Opensdi: Spotting diffusion-generated images in the open world. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4291–4301. IEEE, 2025b.

Shiyu Wu, Shuyan Li, Jing Li, Jing Liu, and Yequan Wang. Locateedit-bench: A benchmark for instruction-based editing localization. *arXiv preprint arXiv:2602.05577*, 2026.

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*, volume 2025, pp. 31186–31216, 2025.

Jieming Yu, Qiuxiao Feng, Zhuohan Wang, and Xiaochen Ma. Dinov3 beats specialized detectors: A simple foundation model baseline for image forensics. *arXiv preprint arXiv:2604.16083*, 2026.

Jiwei Zhang, Wenbo Feng, Siwei Wang, Feifei Kou, Haoyang Yu, and Shaozhang Niu. Collaborative transformers with multi-level forensic attention for image manipulation localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12556–12563, 2026.

Quan Zhang, Yuxin Qi, Xi Tang, Jinwei Fang, Xi Lin, Ke Zhang, and Chun Yuan. Imdprompter: Adapting sam to image manipulation detection by cross-view automated prompt learning. In *International Conference on Learning Representations*, volume 2025, pp. 76218–76242, 2025a.

Rui Zhang, Hongxia Wang, Hangqing Liu, Yang Zhou, and Qiang Zeng. Deal-300k: Diffusion-based editing area localization with a 300k-scale dataset and frequency-prompted baseline. *arXiv preprint arXiv:2511.23377*, 2025b.

Yue Zhou, Xinan He, Kaiqing Lin, Bing Fan, Feng Ding, and Bin Li. Simplicity prevails: The emergence of generalizable aigi detection in visual foundation models. *arXiv preprint arXiv:2602.01738*, 2026.

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.

Table 6: Image F1 over eight test domains for methods with image-level detection capability. Qwen is the in-domain test set; the seven domains to the right of the dashed separator are OOD test sets. OOD Avg. is their macro average.

Method	Qwen	Flux	FireRed	Nano	GPT	Seedream	Hunyuan	Grok	OOD Avg.
MVSS-Net (21)	0.7637	0.6319	0.7524	0.6138	0.6258	**0.5814**	0.6717	0.6562	0.6476
PSCC-Net (22)	0.9978	0.3633	0.8945	0.6188	0.4807	0.2682	0.9652	0.2221	0.5447
TruFor (23)	0.9846	0.5066	0.9853	0.6126	0.5612	0.3088	0.9790	0.2733	0.6038
NFA-ViT (26)	**0.9988**	0.1470	0.9825	0.3085	0.1056	0.0089	0.8255	0.1302	0.3583
Ours	0.9963	**0.6713**	**0.9935**	**0.6479**	**0.6708**	0.4821	**0.9960**	**0.8782**	**0.7628**

Table 7: Pixel IoU over eight test domains. DH, domain partitioning, and averaging follow Table 3.

Method	DH	Qwen	Flux	FireRed	Nano	GPT	Seedream	Hunyuan	Grok	OOD Avg.
MVSS-Net (21)	✓	0.3881	0.2544	0.4053	0.2049	0.1667	0.1845	0.1769	0.2397	0.2332
PSCC-Net (22)	✓	0.3557	0.1401	0.3231	0.1518	0.0925	0.0730	0.1219	0.1249	0.1468
TruFor (23)	✓	0.6128	0.1841	0.6297	0.1536	0.1030	0.0706	0.2921	0.0683	0.2145
IML-ViT (23)	×	0.5393	0.2201	0.5411	0.1690	0.1651	0.0658	0.2654	0.2589	0.2408
APSC-Net (24)	×	0.6493	0.2592	0.6486	0.2090	0.0831	0.0723	0.3303	0.2455	0.2640
SparseViT (25)	×	0.5273	0.1494	0.5037	0.1104	0.0383	0.0241	0.1567	0.1204	0.1576
Mesorch (25)	×	0.5968	0.1377	0.5917	0.1099	0.0350	0.0273	0.2102	0.1514	0.1805
NFA-ViT (26)	✓	0.4707	0.0265	0.4443	0.0384	0.0077	0.0018	0.0835	0.0234	0.0894
Co-Transformers (26)	×	0.6133	0.1889	0.6204	0.1134	0.0441	0.0329	0.2534	0.2088	0.2088
RelayFormer (26)	×	0.6026	0.0960	0.6191	0.0950	0.1223	0.0122	0.2740	0.2483	0.2096
Ours	✓	**0.7527**	**0.5046**	**0.7749**	**0.3976**	**0.3244**	**0.2887**	**0.5017**	**0.5673**	**0.4799**

A Qualitative Localization Results

Figure 5 visualizes pixel-level localization results for multiple local-editing examples. Each column corresponds to one test sample. From top to bottom, the rows show the original image, edited image, predictions from the compared methods, our prediction, and the ground-truth mask. The examples cover four edit types—*Add*, *Replace*, *Remove*, and *Attribute Change*—with two columns per type. The visualization highlights differences in edited-region coverage, boundary completeness, and false-positive regions.

B Additional Cross-Generator Results

B.1 Image-Level Detection: Image F1

Table 6 reports the complete Image F1 results across the eight test domains. EditScan achieves an OOD Image F1 of 0.7628, outperforming the strongest baseline, MVSS-Net, by 0.1152. Across individual domains, EditScan obtains the highest Image F1 on six of the eight test domains. Its performance is slightly below NFA-ViT on Qwen and below MVSS-Net on Seedream, while showing consistently stronger generalization across the remaining unseen generators.

B.2 Pixel-Level Localization: Pixel IoU

Table 7 reports the complete Pixel IoU results across the eight test domains. EditScan achieves an OOD Pixel IoU of 0.4799, outperforming APSC-Net, the strongest baseline in OOD average performance, by 0.2159. Moreover, EditScan obtains the highest Pixel IoU on all eight test domains, further confirming its strong localization generalization across unseen generators.

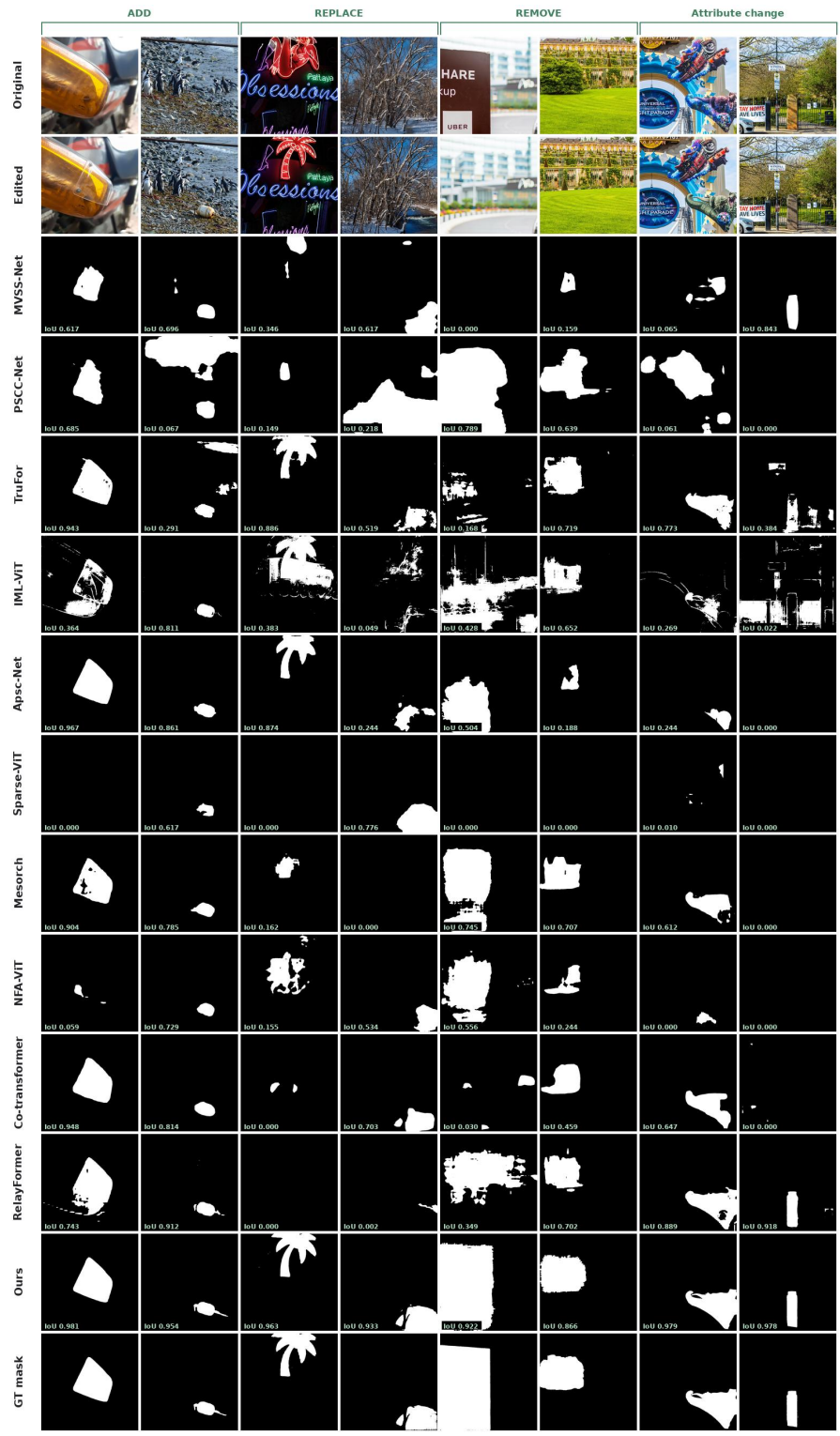


Figure 5: Qualitative comparison of pixel-level edited-region localization. From top to bottom: Original, Edited, compared methods, Ours, and GT mask. White regions denote predicted edited areas.

C FULL ROBUSTNESS RESULTS

This section provides the complete numerical results underlying the robustness plots in the main paper, including Image Accuracy for image-level detection and Pixel IoU for pixel-level localization. As in the main experiments, Qwen is the in-domain test set, the other seven generator families are unseen OOD domains, and OOD Avg. is the macro average over these seven external domains.

C.1 JPEG COMPRESSION ROBUSTNESS

The JPEG experiments use $QF \in \{100, 90, 80, 70, 60, 50\}$. Each setting applies one JPEG encode–decode operation, so $QF = 100$ is not identical to the uncompressed input. Complete image-level detection and pixel-level localization results under JPEG compression are reported in Tables 8 and 9.

C.2 GAUSSIAN NOISE ROBUSTNESS

The Gaussian-noise experiments use fixed variance parameters $v \in \{3, 7, 11, 15, 19, 23\}$ with $\mathrm{var_limit} = (v, v)$. The "No Noise" setting corresponds to the main experiment without additional noise perturbation. Complete image-level detection and pixel-level localization results are reported in Tables 10 and 11.

Table 8: Image Accuracy across test domains under JPEG compression. Qwen is the in-domain test set; OOD Avg. is the macro average over the seven unseen generator domains: Flux, FireRed, Nano Banana, GPT Image, Seedream, Hunyuan, and Grok.

Method	QF	Qwen	Flux	FireRed	Nano	GPT	Seedream	Hunyuan	Grok	OOD Avg.
MVSS-Net (21)	100	0.7372	0.6242	0.7133	0.6072	0.6062	0.5777	0.6458	0.6370	0.6302
	90	0.7337	0.6322	0.7278	0.6145	0.6145	0.5843	0.6475	0.6362	0.6367
	80	0.7227	0.6205	0.7080	0.5952	0.6048	0.5770	0.6332	0.6345	0.6247
	70	0.7088	0.6220	0.7043	0.5990	0.5917	0.5630	0.6208	0.6250	0.6180
	60	0.6985	0.6222	0.7035	0.5950	0.5830	0.5700	0.6200	0.6125	0.6152
	50	0.6973	0.6105	0.6997	0.5845	0.5783	0.5667	0.6038	0.6235	0.6096
PSCC-Net (22)	100	0.9973	0.6198	0.8820	0.7400	0.6490	0.6350	0.9655	0.5745	0.7237
	90	0.9405	0.5085	0.6510	0.5663	0.5125	0.5050	0.6360	0.5095	0.5555
	80	0.8667	0.5045	0.5845	0.5483	0.5080	0.5035	0.5938	0.5065	0.5356
	70	0.8593	0.5085	0.6132	0.5725	0.5150	0.5095	0.6422	0.5235	0.5549
	60	0.8648	0.5168	0.6713	0.6005	0.5375	0.5373	0.7262	0.5705	0.5943
	50	0.8358	0.5220	0.6755	0.6048	0.5407	0.5420	0.7310	0.6228	0.6055
TruFor (23)	100	0.9840	0.6650	0.9840	0.7255	0.6875	0.5655	0.9738	0.5850	0.7409
	90	0.8878	0.5350	0.8530	0.5182	0.5120	0.5040	0.5988	0.5310	0.5789
	80	0.8812	0.5433	0.8373	0.5150	0.5148	0.5050	0.5910	0.5393	0.5779
	70	0.8567	0.5420	0.8280	0.5160	0.5185	0.5055	0.5810	0.5430	0.5763
	60	0.8405	0.5475	0.8223	0.5192	0.5182	0.5095	0.5850	0.5500	0.5788
	50	0.8312	0.5490	0.8110	0.5205	0.5175	0.5122	0.5915	0.5557	0.5796
NFA-ViT (26)	100	0.9988	0.5385	0.9795	0.5942	0.5090	0.5012	0.7622	0.5377	0.6318
	90	0.9638	0.5142	0.8750	0.5405	0.5020	0.5008	0.5317	0.5182	0.5689
	80	0.9437	0.5108	0.8502	0.5230	0.5010	0.5005	0.5178	0.5082	0.5588
	70	0.9123	0.5092	0.8285	0.5212	0.5000	0.5012	0.5112	0.5075	0.5541
	60	0.8893	0.5105	0.8227	0.5180	0.5005	0.5008	0.5092	0.5078	0.5528
	50	0.8768	0.5092	0.8145	0.5208	0.5015	0.5010	0.5102	0.5088	0.5523
Ours	100	0.9952	0.7490	0.9928	0.7245	0.7390	0.6657	0.9932	0.8978	0.8231
	90	0.9962	0.7230	0.9878	0.6627	0.7083	0.6278	0.9868	0.8340	0.7900
	80	0.9943	0.7100	0.9810	0.6450	0.7045	0.6252	0.9768	0.8103	0.7790
	70	0.9917	0.6863	0.9645	0.6252	0.6873	0.6102	0.9337	0.7710	0.7540
	60	0.9832	0.6750	0.9470	0.6132	0.6635	0.5968	0.8690	0.7372	0.7288
	50	0.9768	0.6725	0.9345	0.6138	0.6683	0.5923	0.8468	0.7302	0.7226

Table 9: Pixel IoU across test domains under JPEG compression. Qwen is the in-domain test set; OOD Avg. is the macro average over the seven unseen generator domains: Flux, FireRed, Nano Banana, GPT Image, Seedream, Hunyuan, and Grok.

Method	QF	Qwen	Flux	FireRed	Nano	GPT	Seedream	Hunyuan	Grok	OOD Avg.
MVSS-Net (21)	100	0.3855	0.2522	0.4012	0.2015	0.1648	0.1847	0.1734	0.2382	0.2309
	90	0.3801	0.2476	0.3981	0.1995	0.1659	0.1830	0.1700	0.2314	0.2279
	80	0.3743	0.2535	0.3971	0.2007	0.1646	0.1818	0.1696	0.2351	0.2289
	70	0.3655	0.2504	0.3918	0.1971	0.1614	0.1769	0.1705	0.2321	0.2257
	60	0.3517	0.2452	0.3863	0.1954	0.1582	0.1781	0.1671	0.2300	0.2229
	50	0.3406	0.2359	0.3788	0.1885	0.1579	0.1734	0.1622	0.2242	0.2173
PSCC-Net (22)	100	0.3552	0.1428	0.3156	0.1495	0.0843	0.0900	0.1245	0.1276	0.1477
	90	0.3413	0.0332	0.2029	0.0642	0.0170	0.0145	0.0751	0.0440	0.0644
	80	0.3106	0.0242	0.1417	0.0521	0.0164	0.0128	0.0570	0.0291	0.0476
	70	0.2749	0.0315	0.1453	0.0535	0.0259	0.0179	0.0577	0.0461	0.0540
	60	0.2464	0.0565	0.1718	0.0673	0.0418	0.0406	0.0652	0.0713	0.0735
	50	0.2188	0.0867	0.1878	0.0806	0.0542	0.0598	0.0706	0.1025	0.0917
TruFor (23)	100	0.6104	0.1779	0.6240	0.1550	0.0954	0.0580	0.2787	0.0788	0.2097
	90	0.4799	0.0368	0.4347	0.0189	0.0097	0.0046	0.0701	0.0295	0.0863
	80	0.4652	0.0471	0.4189	0.0180	0.0114	0.0063	0.0639	0.0401	0.0865
	70	0.4352	0.0477	0.3976	0.0193	0.0144	0.0082	0.0585	0.0451	0.0844
	60	0.4147	0.0526	0.3865	0.0227	0.0148	0.0117	0.0588	0.0471	0.0849
	50	0.3950	0.0606	0.3728	0.0262	0.0151	0.0154	0.0592	0.0531	0.0861
IML-ViT (23)	100	0.5372	0.2260	0.5436	0.1753	0.1735	0.0814	0.2605	0.2645	0.2464
	90	0.5310	0.2417	0.5348	0.1841	0.1497	0.0767	0.2501	0.1730	0.2300
	80	0.5269	0.2828	0.5341	0.2140	0.1707	0.1365	0.2571	0.2838	0.2684
	70	0.5201	0.3082	0.5286	0.2386	0.1996	0.1621	0.2579	0.3148	0.2871
	60	0.5110	0.3304	0.5211	0.2588	0.2211	0.1877	0.2547	0.3271	0.3001
	50	0.5024	0.3421	0.5156	0.2699	0.2339	0.2052	0.2528	0.3298	0.3070
APSC-Net (24)	100	0.6479	0.2745	0.6547	0.2280	0.0888	0.0875	0.3265	0.2806	0.2772
	90	0.6386	0.1390	0.6273	0.0726	0.0160	0.0274	0.2269	0.1509	0.1800
	80	0.6186	0.1340	0.6047	0.0740	0.0198	0.0309	0.1928	0.1408	0.1710
	70	0.5886	0.1311	0.5818	0.0779	0.0227	0.0358	0.1692	0.1322	0.1644
	60	0.5662	0.1381	0.5660	0.0875	0.0278	0.0444	0.1618	0.1385	0.1663
	50	0.5547	0.1556	0.5587	0.0996	0.0359	0.0566	0.1683	0.1545	0.1756
SparseViT (25)	100	0.5235	0.1521	0.5034	0.1122	0.0321	0.0294	0.1523	0.1268	0.1583
	90	0.4993	0.0910	0.4519	0.0473	0.0095	0.0081	0.1211	0.0468	0.1108
	80	0.4761	0.0944	0.4348	0.0504	0.0148	0.0119	0.0991	0.0820	0.1125
	70	0.4456	0.0892	0.4035	0.0528	0.0178	0.0163	0.0857	0.0786	0.1063
	60	0.4222	0.0962	0.3894	0.0573	0.0226	0.0239	0.0754	0.0831	0.1068
	50	0.4002	0.1089	0.3811	0.0644	0.0289	0.0344	0.0771	0.0957	0.1129
Mesorch (25)	100	0.5915	0.1384	0.5868	0.1073	0.0296	0.0297	0.1946	0.1568	0.1776
	90	0.5796	0.1007	0.5646	0.0741	0.0207	0.0218	0.1607	0.1036	0.1495
	80	0.5637	0.0927	0.5415	0.0641	0.0195	0.0212	0.1347	0.0906	0.1378
	70	0.5368	0.0823	0.5174	0.0597	0.0172	0.0193	0.1058	0.0768	0.1255
	60	0.5113	0.0791	0.4977	0.0576	0.0179	0.0221	0.0925	0.0689	0.1194
	50	0.4938	0.0773	0.4848	0.0584	0.0197	0.0261	0.0857	0.0700	0.1174
NFA-ViT (26)	100	0.4635	0.0266	0.4342	0.0345	0.0050	0.0016	0.0588	0.0241	0.0835
	90	0.4027	0.0100	0.3343	0.0180	0.0027	0.0011	0.0119	0.0140	0.0560
	80	0.3633	0.0072	0.3027	0.0135	0.0022	0.0011	0.0100	0.0100	0.0495
	70	0.3226	0.0076	0.2763	0.0111	0.0020	0.0013	0.0064	0.0098	0.0449
	60	0.2949	0.0079	0.2628	0.0109	0.0022	0.0013	0.0063	0.0089	0.0429
	50	0.2738	0.0080	0.2494	0.0120	0.0031	0.0017	0.0052	0.0101	0.0414
Co-Transformers (26)	100	0.6086	0.1867	0.6146	0.1106	0.0392	0.0312	0.2394	0.2171	0.2056
	90	0.5992	0.1364	0.5898	0.0757	0.0257	0.0203	0.1895	0.1154	0.1647
	80	0.5834	0.1371	0.5685	0.0712	0.0256	0.0186	0.1693	0.1127	0.1576
	70	0.5572	0.1219	0.5410	0.0655	0.0250	0.0197	0.1307	0.1031	0.1438
	60	0.5370	0.1245	0.5241	0.0654	0.0284	0.0229	0.1199	0.1079	0.1419
	50	0.5208	0.1294	0.5097	0.0706	0.0312	0.0268	0.1133	0.1097	0.1415
RelayFormer (26)	100	0.6003	0.1047	0.6170	0.0984	0.1195	0.0160	0.2594	0.2536	0.2098
	90	0.5957	0.0937	0.6134	0.0907	0.1202	0.0145	0.2253	0.1201	0.1826
	80	0.5876	0.1304	0.6012	0.1177	0.1210	0.0338	0.2343	0.2248	0.2090
	70	0.5739	0.1762	0.5901	0.1515	0.1380	0.0525	0.2540	0.2749	0.2339
	60	0.5649	0.2243	0.5845	0.1800	0.1581	0.0747	0.2679	0.3172	0.2581
	50	0.5510	0.2576	0.5710	0.2040	0.1718	0.1000	0.2719	0.3324	0.2727
Ours	100	0.7520	0.4984	0.7733	0.3822	0.3101	0.2940	0.4988	0.5709	0.4754
	90	0.7493	0.4672	0.7638	0.3337	0.2848	0.2570	0.4962	0.5199	0.4461
	80	0.7419	0.4540	0.7549	0.3171	0.2844	0.2543	0.4915	0.5066	0.4375
	70	0.7299	0.4213	0.7417	0.2969	0.2744	0.2353	0.4648	0.4614	0.4137
	60	0.7148	0.4010	0.7254	0.2748	0.2557	0.2114	0.4278	0.4194	0.3879
	50	0.7055	0.3958	0.7146	0.2729	0.2492	0.2039	0.4089	0.4104	0.3794

Table 10: Image Accuracy across test domains under Gaussian noise. Qwen is the in-domain test set; OOD Avg. is the macro average over the seven unseen generator domains: Flux, FireRed, Nano Banana, GPT Image, Seedream, Hunyuan, and Grok.

Method	v	Qwen	Flux	FireRed	Nano	GPT	Seedream	Hunyuan	Grok	OOD Avg.
MVSS-Net (21)	3	0.7297	0.6175	0.7100	0.6082	0.6012	0.5800	0.6348	0.6368	0.6269
	7	0.7272	0.6200	0.7065	0.6020	0.5948	0.5783	0.6338	0.6315	0.6238
	11	0.7183	0.6178	0.7095	0.6002	0.5913	0.5793	0.6240	0.6315	0.6219
	15	0.7167	0.6208	0.7095	0.5965	0.5978	0.5765	0.6208	0.6262	0.6211
	19	0.7168	0.6178	0.7050	0.5960	0.5897	0.5787	0.6162	0.6282	0.6188
	23	0.7177	0.6185	0.7070	0.5940	0.5863	0.5743	0.6102	0.6270	0.6167
PSCC-Net (22)	3	0.9913	0.6100	0.8708	0.7202	0.6255	0.5520	0.9377	0.6178	0.7049
	7	0.9895	0.5657	0.7752	0.6677	0.5735	0.5265	0.8373	0.5600	0.6437
	11	0.9762	0.5410	0.6985	0.6142	0.5473	0.5138	0.7215	0.5308	0.5953
	15	0.9550	0.5245	0.6472	0.5845	0.5315	0.5088	0.6470	0.5170	0.5658
	19	0.9258	0.5145	0.6138	0.5627	0.5220	0.5072	0.6070	0.5092	0.5481
	23	0.8945	0.5115	0.5887	0.5493	0.5178	0.5065	0.5757	0.5072	0.5367
TruFor (23)	3	0.9785	0.6700	0.9800	0.7125	0.6620	0.5795	0.9685	0.6305	0.7433
	7	0.9740	0.6532	0.9760	0.6795	0.6308	0.5693	0.9467	0.6610	0.7309
	11	0.9690	0.6360	0.9700	0.6532	0.6062	0.5593	0.9145	0.6667	0.7151
	15	0.9612	0.6188	0.9640	0.6352	0.5895	0.5525	0.8782	0.6607	0.6999
	19	0.9552	0.6102	0.9530	0.6160	0.5747	0.5460	0.8377	0.6525	0.6843
	23	0.9468	0.6002	0.9435	0.6042	0.5653	0.5410	0.8027	0.6430	0.6714
NFA-ViT (26)	3	0.9988	0.5385	0.9798	0.5743	0.5135	0.5012	0.7812	0.5235	0.6303
	7	0.9958	0.5235	0.9595	0.5507	0.5030	0.5010	0.6550	0.5160	0.6013
	11	0.9880	0.5148	0.9290	0.5343	0.5008	0.5002	0.5837	0.5130	0.5823
	15	0.9758	0.5115	0.9025	0.5268	0.5008	0.5005	0.5487	0.5112	0.5717
	19	0.9667	0.5080	0.8745	0.5198	0.5000	0.5010	0.5315	0.5075	0.5632
	23	0.9545	0.5052	0.8540	0.5162	0.4995	0.5010	0.5235	0.5065	0.5580
Ours	3	0.9972	0.7230	0.9910	0.6750	0.7125	0.6330	0.9892	0.8658	0.7985
	7	0.9932	0.6905	0.9758	0.6315	0.6667	0.6042	0.9573	0.8225	0.7641
	11	0.9863	0.6693	0.9593	0.6110	0.6352	0.5865	0.9147	0.7870	0.7376
	15	0.9763	0.6573	0.9470	0.5972	0.6138	0.5753	0.8732	0.7578	0.7174
	19	0.9633	0.6465	0.9335	0.5890	0.5962	0.5663	0.8347	0.7398	0.7009
	23	0.9512	0.6372	0.9253	0.5785	0.5855	0.5603	0.8033	0.7230	0.6876

Table 11: Pixel IoU across test domains under Gaussian noise. Qwen is the in-domain test set; OOD Avg. is the macro average over the seven unseen generator domains: Flux, FireRed, Nano Banana, GPT Image, Seedream, Hunyuan, and Grok.

Method	v	Qwen	Flux	FireRed	Nano	GPT	Seedream	Hunyuan	Grok	OOD Avg.
MVSS-Net (21)	3	0.3808	0.2494	0.3989	0.2004	0.1669	0.1839	0.1750	0.2370	0.2302
	7	0.3701	0.2450	0.3938	0.1953	0.1639	0.1820	0.1709	0.2325	0.2262
	11	0.3644	0.2426	0.3892	0.1926	0.1618	0.1801	0.1668	0.2298	0.2233
	15	0.3593	0.2412	0.3863	0.1911	0.1591	0.1806	0.1648	0.2261	0.2213
	19	0.3536	0.2379	0.3842	0.1902	0.1569	0.1802	0.1641	0.2244	0.2197
	23	0.3506	0.2347	0.3819	0.1898	0.1568	0.1783	0.1620	0.2227	0.2180
PSCC-Net (22)	3	0.3286	0.1371	0.2994	0.1378	0.0759	0.0593	0.1147	0.1423	0.1381
	7	0.3172	0.1007	0.2501	0.1131	0.0504	0.0364	0.1026	0.0943	0.1068
	11	0.3103	0.0646	0.1922	0.0879	0.0331	0.0247	0.0807	0.0587	0.0774
	15	0.2977	0.0453	0.1546	0.0682	0.0261	0.0181	0.0610	0.0392	0.0589
	19	0.2790	0.0330	0.1311	0.0544	0.0177	0.0139	0.0474	0.0290	0.0466
	23	0.2600	0.0262	0.1082	0.0434	0.0130	0.0118	0.0376	0.0211	0.0373
TruFor (23)	3	0.6035	0.1858	0.6184	0.1509	0.0864	0.0699	0.2736	0.1225	0.2154
	7	0.5907	0.1670	0.6027	0.1320	0.0692	0.0621	0.2549	0.1472	0.2050
	11	0.5790	0.1502	0.5875	0.1146	0.0568	0.0552	0.2315	0.1493	0.1922
	15	0.5651	0.1357	0.5735	0.1010	0.0511	0.0508	0.2120	0.1446	0.1812
	19	0.5519	0.1228	0.5581	0.0905	0.0456	0.0455	0.1884	0.1326	0.1691
	23	0.5362	0.1115	0.5423	0.0812	0.0411	0.0419	0.1690	0.1255	0.1589
IML-ViT (23)	3	0.5079	0.1649	0.5160	0.1307	0.1199	0.0401	0.2338	0.2037	0.2013
	7	0.4268	0.1069	0.4521	0.0930	0.0825	0.0261	0.1726	0.1461	0.1542
	11	0.3462	0.0773	0.4010	0.0687	0.0578	0.0205	0.1340	0.1126	0.1246
	15	0.2857	0.0633	0.3606	0.0542	0.0448	0.0172	0.1071	0.0931	0.1058
	19	0.2458	0.0529	0.3315	0.0445	0.0360	0.0147	0.0872	0.0778	0.0921
	23	0.2162	0.0438	0.3077	0.0380	0.0294	0.0128	0.0745	0.0674	0.0820
APSC-Net (24)	3	0.6428	0.2584	0.6516	0.1942	0.0814	0.0771	0.3193	0.3042	0.2695
	7	0.6361	0.2433	0.6479	0.1766	0.0750	0.0722	0.3044	0.2915	0.2587
	11	0.6288	0.2262	0.6408	0.1603	0.0695	0.0686	0.2843	0.2709	0.2458
	15	0.6213	0.2134	0.6334	0.1490	0.0630	0.0654	0.2637	0.2537	0.2345
	19	0.6138	0.2041	0.6266	0.1388	0.0583	0.0619	0.2437	0.2358	0.2242
	23	0.6056	0.1963	0.6175	0.1303	0.0549	0.0600	0.2270	0.2200	0.2151
SparseViT (25)	3	0.5185	0.1363	0.4961	0.0968	0.0268	0.0184	0.1492	0.1165	0.1486
	7	0.5023	0.1081	0.4773	0.0731	0.0167	0.0101	0.1298	0.0889	0.1291
	11	0.4874	0.0811	0.4541	0.0536	0.0106	0.0062	0.1089	0.0662	0.1115
	15	0.4726	0.0582	0.4332	0.0401	0.0063	0.0045	0.0890	0.0500	0.0973
	19	0.4555	0.0438	0.4098	0.0302	0.0049	0.0035	0.0717	0.0372	0.0859
	23	0.4399	0.0354	0.3863	0.0242	0.0041	0.0029	0.0570	0.0305	0.0772
Mesorch (25)	3	0.5834	0.1067	0.5834	0.0783	0.0230	0.0210	0.1781	0.1222	0.1590
	7	0.5688	0.0786	0.5633	0.0517	0.0132	0.0164	0.1348	0.0926	0.1358
	11	0.5538	0.0597	0.5373	0.0387	0.0090	0.0115	0.1029	0.0724	0.1188
	15	0.5385	0.0471	0.5081	0.0292	0.0070	0.0097	0.0840	0.0584	0.1062
	19	0.5191	0.0399	0.4832	0.0241	0.0056	0.0082	0.0716	0.0501	0.0975
	23	0.4980	0.0344	0.4629	0.0213	0.0048	0.0069	0.0619	0.0418	0.0906
NFA-ViT (26)	3	0.4570	0.0217	0.4276	0.0296	0.0061	0.0016	0.0565	0.0201	0.0804
	7	0.4349	0.0144	0.3892	0.0202	0.0038	0.0011	0.0326	0.0151	0.0681
	11	0.4138	0.0112	0.3580	0.0137	0.0031	0.0008	0.0212	0.0139	0.0603
	15	0.3926	0.0088	0.3331	0.0114	0.0028	0.0008	0.0154	0.0123	0.0549
	19	0.3742	0.0074	0.3135	0.0091	0.0024	0.0007	0.0125	0.0110	0.0509
	23	0.3578	0.0061	0.2953	0.0073	0.0021	0.0007	0.0100	0.0097	0.0473
Co-Transformers (26)	3	0.6019	0.1723	0.6141	0.0940	0.0364	0.0287	0.2340	0.1912	0.1958
	7	0.5893	0.1401	0.5988	0.0710	0.0303	0.0226	0.2027	0.1500	0.1736
	11	0.5758	0.1125	0.5763	0.0565	0.0246	0.0181	0.1616	0.1195	0.1527
	15	0.5594	0.0894	0.5495	0.0475	0.0203	0.0146	0.1237	0.0941	0.1341
	19	0.5415	0.0757	0.5256	0.0399	0.0161	0.0120	0.0954	0.0766	0.1202
	23	0.5226	0.0642	0.5019	0.0329	0.0117	0.0100	0.0774	0.0639	0.1089
RelayFormer (26)	3	0.5794	0.0570	0.5973	0.0606	0.0744	0.0076	0.2129	0.1762	0.1694
	7	0.4910	0.0283	0.5185	0.0341	0.0395	0.0040	0.1331	0.1122	0.1243
	11	0.3835	0.0154	0.4446	0.0228	0.0264	0.0024	0.0870	0.0739	0.0961
	15	0.3043	0.0090	0.3834	0.0181	0.0189	0.0016	0.0582	0.0549	0.0777
	19	0.2418	0.0049	0.3384	0.0145	0.0152	0.0012	0.0424	0.0436	0.0657
	23	0.1994	0.0032	0.3030	0.0119	0.0121	0.0011	0.0322	0.0347	0.0569
Ours	3	0.7482	0.4586	0.7720	0.3368	0.2937	0.2574	0.4901	0.5390	0.4496
	7	0.7382	0.4165	0.7560	0.2880	0.2592	0.2233	0.4731	0.4983	0.4164
	11	0.7258	0.3872	0.7419	0.2599	0.2287	0.1981	0.4442	0.4653	0.3893
	15	0.7118	0.3638	0.7239	0.2418	0.2078	0.1813	0.4172	0.4372	0.3676
	19	0.6976	0.3504	0.7104	0.2275	0.1938	0.1716	0.3944	0.4131	0.3516
	23	0.6834	0.3386	0.6977	0.2152	0.1815	0.1621	0.3715	0.3940	0.3372