SIGMA: Semantic-Difference Instruction-Grounding Mask Annotator for Text-Driven Image Manipulation Localization

Peiyu Zhuang[1], Jianquan Yang[1]*, Haodong Li[2], Zhuoying Cai[1], Ruitao Xie[3], Jishen Zeng[4], Baoying Chen[4], Jiwu Huang[5], Xiaochun Cao[1]*

[1]Shenzhen Campus of Sun Yat-sen University, China
[2]Shenzhen University, China
[3]Shenzhen University of Advanced Technology, China
[4]Alibaba Group, China [5]Shenzhen MSU-BIT University, China

{zhuangpy, yangjq65, caoxiaochu}@mail.sysu.edu.cn;
lihaodong@szu.edu.cn; caizhy55@mail2.sysu.edu.cn
xieruitao@suat-sz.edu.cn; jishen.zjs@alibaba-inc.com;
1900271059@email.szu.edu.cn;jwhuang@smbu.edu.cn

Abstract

Text-driven image editing has advanced rapidly, but reliably localizing these manipulations requires image manipulation localization (IML) models trained on large pixel-annotated datasets, and there is still no low-cost way to obtain such training data at scale. We observe that these data already exist in disguise: public editing datasets contain millions of structurally identical *(original, edited)* pairs to IML training samples, lacking only pixel-level masks. Recovering these masks automatically is non-trivial: pixel differencing is overwhelmed by diffusion-induced perturbations across all pixels, and instruction-only grounding localizes only what the prompt describes, missing unintended editor side-effects. We propose **SIGMA** (**S**emantic-difference **I**nstruction-**G**rounding **M**ask **A**nnotator), which performs semantic-feature differencing in a vision foundation backbone and injects an instruction-derived spatial prior into this visual stream via bidirectional cross-modal refinement, amplifying the difference signal at intended-edit regions when the editor faithfully realizes user intent. SIGMA is trained in two complementary stages: Stage I supervises on inpainting masks and Stage II closes the diffusion-domain shift via VAE-roundtrip noise calibration, EMA self-training, and an edit-noise disentanglement loss. SIGMA outperforms existing automatic mask generators on five benchmarks (**+12.20%** F1, **+11.16%** IoU). When applied to public editing corpora, it produces a $\sim$1.1M IML training set that improves four diverse detectors by **+20.98%** F1 across five datasets, turning previously unused editing data into a model-agnostic supervisory resource for IML. For reproducibility, our project page is available at `https://zhuangpeiyu.github.io/projects/SIGMA/`.

1 Introduction

Recent advances in text-driven image editing [1, 2, 3, 4, 5, 6] have lowered the barrier to producing semantically coherent forgeries that fuel misinformation and erode public trust. Mitigating this risk demands robust image manipulation localization (IML) models, which in turn require large-scale, pixel-accurately annotated training data, a resource the IML community currently lacks. Nowadays, public editing corpora have reached the million-sample scale (*e.g.*, ByteMorph[7] 6.0M, AnyEdit[8]

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

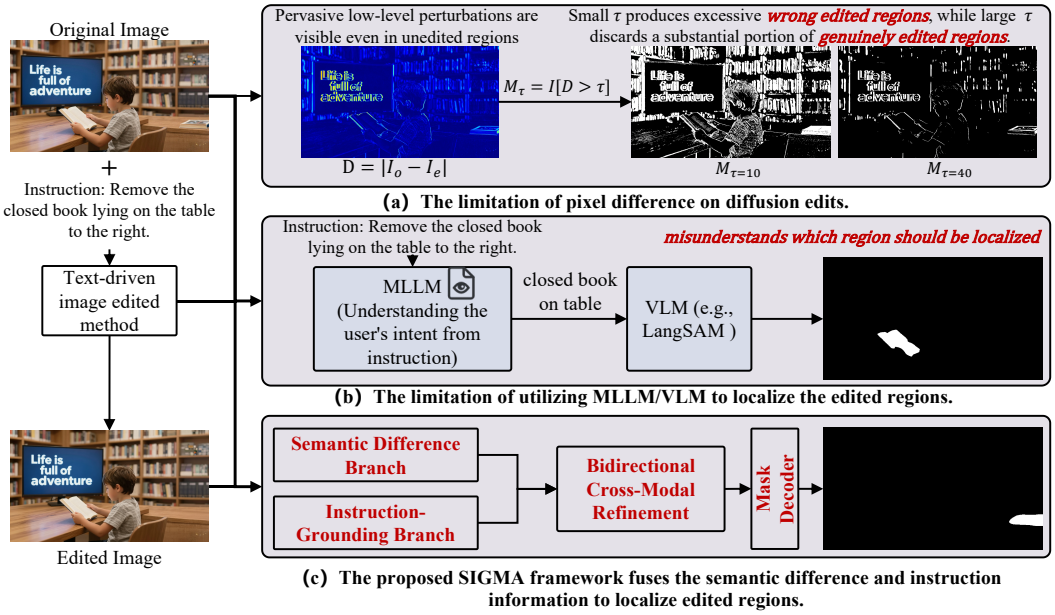


Figure 1: (a) Failure modes of pixel-level differencing under diffusion-based text-driven editing. (b) Failure modes of MLLM/VLM-based mask inference from instructions. (c) Overview of the SIGMA framework.

2.5M), while the largest IML resources with pixel-level masks (SIDA[9] 300K, GRE[10] 228.7K) are dwarfed by even a single editing corpus (see Fig. 5 in Appendix A for additional information). We note that existing editing datasets already comprise *(original, edited)* pairs that are structurally equivalent to the training samples used in IML; however, they lack the corresponding pixel-level annotation masks. This raises a natural question: *can existing editing pairs be automatically turned into pixel-accurate IML supervision, enabling million-scale corpora without rerunning editors or manual annotation?*

Answering this question hinges on an automatic, scalable annotator. Two seemingly natural strategies prove ineffective. **Pixel-level differencing with thresholding** works for inpainting-style edits but breaks down under diffusion-based editing, where every pixel is perturbed by VAE encoding/decoding and iterative denoising. As shown in Fig. 1(a), low thresholds drown the mask in reconstruction noise, while high thresholds discard subtle yet semantically meaningful edits. **Instruction-grounding via MLLMs/VLMs** (Fig. 1(b)) localizes only what the prompt requests and cannot verify whether the editor faithfully realized that intent or introduced unintended side-effects. Hybrid pipelines (e.g., SAM with human refinement) yield accurate masks but trade scalability for accuracy. The two automatic strategies are not coincidentally inadequate: they are *complementary, not alternative*. Pixel differencing observes what changed but cannot separate noise from edits, while instruction grounding encodes what was intended but cannot verify execution.

Building on this view, we propose **SIGMA** (**S**emantic-difference **I**nstruction-**G**rounding **M**ask **A**nnotator), a two-branch framework whose semantic branch performs multi-scale differencing on dense features from a frozen DINOv2 [11] encoder (*where content actually differs*), while its instruction branch grounds a parsed transformation tuple (original concept, edited concept, action) onto each image via a vision-language grounder (*where the change was intended*). A bidirectional cross-modal refinement module then injects the instruction prior into the visual stream to amplify true edit signals while continuously re-anchoring the prior to the most supported regions. To bridge supervision scarcity and diffusion-domain shift, SIGMA is trained in two stages: **Stage I** pre-trains on inpainting pairs with exact masks, and **Stage II** adapts to unlabeled editing pairs via VAE-roundtrip zero-edit calibration, exponential moving average (EMA) self-training, and an edit-noise feature disentanglement loss, jointly suppressing diffusion-induced false positives without any target-domain annotation.

Our contributions can be summarized as follows.

- We reframe scalable IML annotation as **semantic change localization**, showing that pixel differencing and instruction grounding are complementary but individually insufficient for diffusion-based edits.
- We propose **SIGMA**, a two-branch framework fusing semantic-feature differencing with parsed-instruction grounding via bidirectional cross-modal refinement, trained by a two-stage strategy that targets supervision scarcity and domain shift respectively.
- Using SIGMA, we construct a **million-scale IML dataset** from public editing corpora at no regeneration or manual annotation cost, achieving **+12.20%** F1 over prior automatic mask generators and boosting four diverse IML detectors by **+20.98%** F1 in cross-dataset generalization.

2 Related Work

IML and its data bottleneck. IML has progressed from hand-crafted forensic cues to learning-based methods exploiting noise inconsistencies, frequency artifacts, or diffusion priors [9, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22], with recent work integrating multimodal large language models for explainable, pixel-level localization [23, 24, 25]. Yet their performance is bottlenecked by training data: early benchmarks [26, 27, 28, 29] rely on hand-crafted operations whose artifacts no longer reflect modern forgeries, while diffusion-era datasets [13, 30, 31, 32, 33] remain dwarfed by public text-driven editing corpora in scale and edit diversity (Fig. 5 in Appendix A). Our contribution is orthogonal to architectural advances: rather than proposing yet another localization model, we provide a scalable, fully automated annotator that turns previously unused text-driven editing data into supervision for arbitrary downstream IML models.

Change detection. SIGMA's semantic differencing branch shares architectural intuition with change detection (CD) in remote sensing, where co-registered multi-temporal images are compared via Siamese CNNs, Transformers [34], or foundation models such as SAM [35] and pretrained DDPMs [36]. Despite this shared backbone, the two tasks differ at two levels. First, CD nuisances are low-level and semantically independent of content, so Siamese differencing can suppress them. However, in text-driven editing, the entire image is re-synthesized through a VAE and iterative denoising, yielding high-level, content-correlated perturbations at every pixel that feature normalization alone cannot remove. Second, mainstream CD operates on the image pair alone. In our setting, however, the editing instruction provides a strong spatial prior that, when the editor faithfully realizes user intent, amplifies the visual difference signal at the targeted regions.

3 Proposed Method

3.1 Problem Formulation

Given a source image $I^o \in \mathbb{R}^{H\times W\times 3}$, its text-driven edited counterpart $I^e \in \mathbb{R}^{H\times W\times 3}$, and the editing instruction T, we predict a binary mask $M \in \{0,1\}^{H\times W}$ delineating the semantically edited region. A pixel (x, y) is considered edited iff its underlying semantic properties (object identity, material, color, or contextual role) have meaningfully changed:

$$M(x,y) = \mathbf{1}\left[\mathcal{S}(I^o)_{(x,y)} \neq \mathcal{S}(I^e)_{(x,y)}\right], \tag{1}$$

where $\mathcal{S}(\cdot)$ denotes a high-level semantic representation, so that VAE roundtrip noise and stochastic denoising are excluded. This definition imposes two requirements that neither pixel differencing nor instruction grounding alone satisfies (shown in Sec. 1). SIGMA addresses (i) by lifting comparison from pixels to a frozen DINOv2 [11] feature space, and (ii) by parsing the instruction into a structured spatial prior and injecting it into the visual differencing branch via bidirectional cross-modal refinement.

3.2 The Proposed SIGMA Framework

As illustrated in Fig. 2, SIGMA is powered by four key components: a semantic difference branch, an instruction-grounding branch, a bidirectional cross-modal refinement module, and a mask decoder.

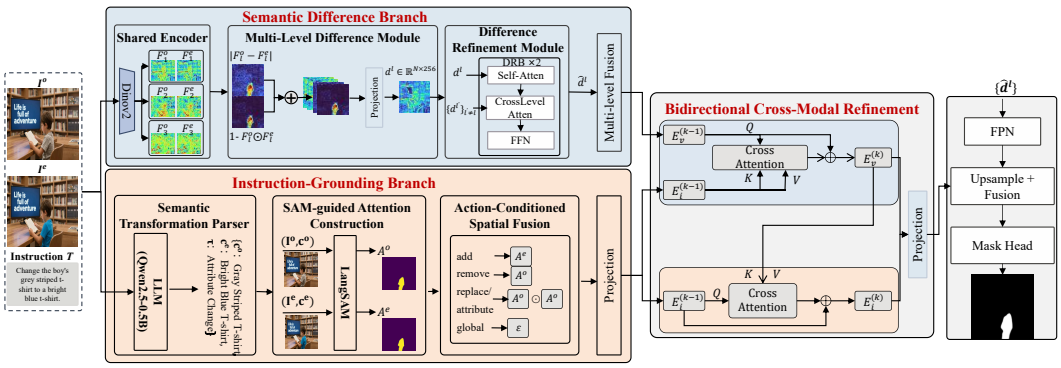


Figure 2: Overview of the SIGMA architecture.

3.2.1 Semantic-Difference Branch

Shared Encoder. We adopt frozen DINOv2-Base as the shared backbone for both images. Both images are resized to 518×518, and patch token features are extracted at three designated layers $\mathcal{L} = \{l_1, l_2, l_3\}$ corresponding to layers 2, 5, and 11 of the ViT-Base architecture:

$$\mathbf{f}_o^l = \mathrm{DINOv2}(I^o), \quad \mathbf{f}_e^l = \mathrm{DINOv2}(I^e), \quad l \in \mathcal{L}, \tag{2}$$

where $\mathbf{f}_o^l, \mathbf{f}_e^l \in \mathbb{R}^{N \times D}$, $N = 37 \times 37 = 1{,}369$, and $D = 768$. Shallow layers encode fine-grained texture and local structure, whereas deeper layers capture object-level semantics and holistic scene understanding.

Multi-Level Differencing Module. For each level $l \in \mathcal{L}$, we construct a difference representation that encodes both magnitude and directional dissimilarity of feature change. Let $\tilde{\mathbf{f}}_o^l$ and $\tilde{\mathbf{f}}_e^l$ denote the L2-normalized features. We compute

$$\mathbf{d}_{\mathrm{abs}}^l = \left|\tilde{\mathbf{f}}_o^l - \tilde{\mathbf{f}}_e^l\right|, \qquad \mathbf{d}_{\cos}^l = \tilde{\mathbf{f}}_o^l \odot \tilde{\mathbf{f}}_e^l, \tag{3}$$

We then fuse these signals with the original features and project into a shared embedding:

$$\mathbf{d}^l = \mathrm{Proj}^l\Big(\big[\mathbf{d}_{\mathrm{abs}}^l \parallel \mathbf{d}_{\cos}^l \parallel \tilde{\mathbf{f}}_o^l \parallel \tilde{\mathbf{f}}_e^l\big]\Big), \quad \mathrm{Proj}^l : \mathbb{R}^{4D} \to \mathbb{R}^C, \; C = 256. \tag{4}$$

Difference Refinement Module. Raw difference features are corrupted by patch-level quantization artifacts. We refine them with transformer-based Difference Refinement Blocks (DRBs):

$$\hat{\mathbf{d}}^l = \mathrm{DRB}^l\Big(\mathbf{d}^l, \; \{\mathbf{d}^{l'}\}_{l' \neq l}\Big). \tag{5}$$

Each DRB consists of a within-level self-attention layer, a cross-level attention layer (queries at l attend to keys and values from all $l' \neq l$), and a feed-forward network with GELU activation. Cross-level attention disambiguates patches whose editing evidence is strong at one semantic scale but weak at another. The refined multi-level difference features are aggregated into a dense per-patch visual evidence representation:

$$\mathbf{E}_v^{(0)} = \mathrm{Agg}\Big(\{\hat{\mathbf{d}}^l\}_{l \in \mathcal{L}}\Big) \in \mathbb{R}^{N \times C'}, \quad C' = 256, \tag{6}$$

where $\mathrm{Agg}(\cdot)$ denotes channel-wise concatenation followed by a 1×1 convolutional projection. This branch answers: *where does the semantic content of the image pair differ?*

3.2.2 Instruction-Grounding Branch

The instruction-grounding branch supplies a structurally independent stream of evidence distilled from the editing instruction itself. The majority of instructions specify a well-defined semantic transformation, characterized by an original concept c^o, an edited concept c^e, and an operation type op (representative examples are provided in Appendix B). We exploit the informative content contained in the instruction by employing a sequence of three consecutive submodules.

Semantic Transformation Parser (STP). STP extracts the tuple (c^o, c^e, op) from T, with $op \in \{add, remove, replace, attribute\ change, global\}$. We use Qwen2.5-0.5B [37] with in-context prompting (full prompt is presented in Appendix C), chosen for its strong instruction-following at minimal parameter cost.

SAM-guided Attention Construction. Each non-empty concept is associated with its corresponding image, c^o with I^o and c^e with I^e, and provided to LangSAM (GroundingDINO[38] + SAM2.1[39]) as a textual input prompt in order to derive a concept-conditioned attention map:

$$\phi(I, c) = \begin{cases} \epsilon_{\text{global}}\mathbf{1}, & c = \varnothing \text{ or } \mathcal{L}(I, c) = \varnothing, \\ \mathcal{R}(\max \mathcal{L}(I, c)), & \text{otherwise}, \end{cases} \tag{7}$$

where $\mathcal{L}(I, c)$ is the LangSAM prediction, $\max \mathcal{L}$ denotes instance-wise max merging, $\mathcal{R}$ is bilinear resizing to the working resolution. We set the global smoothing parameter to $\epsilon_{\text{global}} = 0.001$ in order to maintain numerical stability in the generation process and to avoid obtaining a completely empty output. We obtain the two downstream concept maps as $A^o = \phi(I^o, c^o)$ and $A^e = \phi(I^e, c^e)$.

Action-Conditioned Spatial Fusion (ACSF). Different actions imply structurally different spatial relationships between A^o and A^e, which we encode by an action-conditioned rule:

$$M_{\text{intent}}(p) = \begin{cases} A^e(p) & op = \textit{add}, \\ A^o(p) & op = \textit{remove}, \\ A^o(p) \cdot A^e(p) & op \in \{\textit{replace}, \textit{attribute change}\}, \\ \epsilon_{\text{global}}\mathbf{1} & op = \textit{global}. \end{cases} \tag{8}$$

The product rule for *replace* and *attribute change* reflects the joint requirement that patch p shows the original concept in I^o and the edited concept in I^e. For the *global* case, the near-zero uniform map provides only weak instruction evidence, naturally letting the semantic branch dominate. Representative visualizations of the instruction-grounding branch are provided in the Appendix D. The intent map is then passed through a 1×1 convolution with ReLU to yield the initial instruction evidence $\mathbf{E}_i^{(0)} = \text{Conv}(M_{\text{intent}}) \in \mathbb{R}^{N \times C'}$.

3.2.3 Bidirectional Cross-Modal Refinement

The two branches play distinct but coupled roles: the semantic-difference branch acts as the engine, generating dense cues that drive mask prediction, while the instruction-grounding branch acts as the navigator, providing a spatial prior that predicts where edits should occur. To fuse them, the Bidirectional Cross-Modal Refinement (BCMR) module stacks cross-attention blocks with an asymmetric order: at each layer, the semantic-difference branch first attends to the current prior to amplify edit-relevant signals, then the instruction branch attends to this freshly updated state to re-center the prior on regions with strongest visual support. This ordering enforces the intended asymmetry: the semantic-difference branch carries the primary signal, while the instruction prior continuously self-refines to sharpen where the model should act.

Formally, let $\mathbf{E}_v^{(0)}$ and $\mathbf{E}_i^{(0)}$ be the initial visual and instruction representations. At each iteration $k = 1, \ldots, K$, we perform two sequential cross-attention updates:

$$\mathbf{E}_v^{(k)} = \mathbf{E}_v^{(k-1)} + \text{CrossAttn}\big(\mathbf{E}_v^{(k-1)}, \mathbf{E}_i^{(k-1)}\big), \quad \mathbf{E}_i^{(k)} = \mathbf{E}_i^{(k-1)} + \text{CrossAttn}\big(\mathbf{E}_i^{(k-1)}, \mathbf{E}_v^{(k)}\big). \tag{9}$$

The instruction update queries the freshly updated $\mathbf{E}_v^{(k)}$ rather than the stale $\mathbf{E}_v^{(k-1)}$, so the prior is always re-anchored against the most current visual evidence.

Consensus representation. After K iterations, the two refined representations are fused into a shared consensus space:

$$\mathbf{Z} = \text{Proj}\Big(\big[\mathbf{E}_v^{(K)} \parallel \mathbf{E}_i^{(K)}\big]\Big) \in \mathbb{R}^{N \times C'}, \tag{10}$$

where $\text{Proj}\colon \mathbb{R}^{2C'} \to \mathbb{R}^{C'}$ is a 1×1 convolution. $\mathbf{Z}$ encodes the image patches at which the two evidence sources intersect, corresponding exactly to the regions that underwent semantic modification.

3.2.4 Mask Decoder

The mask decoder aggregates DRB outputs $\{\hat{\mathbf{d}}^\ell\}_{\ell \in \mathcal{L}}$ with a top-down FPN and then predicts dense mask logits. Although these tensors come from different ViT depths, they are all defined on the same patch lattice (H_p, W_p); the decoder therefore builds a spatial pyramid internally by adaptive pooling of $\hat{\mathbf{d}}^3$ and $\hat{\mathbf{d}}^2$ to approximately $(H_p/4, W_p/4)$ and $(H_p/2, W_p/2)$ while keeping $\hat{\mathbf{d}}^1$ at (H_p, W_p),

followed by a 1×1 projections, top-down fusion, and 3×3 convolutional layer. The highest-resolution FPN feature map is upsampled to the full input size (H, W) and concatenated with $\mathbf{Z}$ (also upsampled to (H, W)). A convolutional residual fusion block then distills this combined signal into $\hat{\mathbf{Z}}$. Starting from $\hat{\mathbf{Z}}$, a terminal convolutional residual block diverges into a mask head, which outputs logits corresponding to the edited regions.

3.3 Noise-Calibrated Progressive Domain Adaptation

A central bottleneck in text-driven IML is the absence of pixel-accurate ground truth, which causes models trained only on inpainting data to conflate stochastic diffusion noise with semantic change. We address this with a two-stage strategy. Let $f(I^o, I^e) \in [0,1]^{H\times W}$ denote the mask probabilities produced by SIGMA, obtained by applying a sigmoid to the mask-head logits. For an inpainting pair, $M^{\text{inp}} \in \{0,1\}^{H\times W}$ is the pixel-level ground-truth mask of the edited region.

Stage I pre-trains SIGMA on inpainting pairs with these masks. The segmentation objective adds binary cross-entropy, which scores each pixel independently, to a soft Dice loss [40], which scores region overlap and remains informative when the edited area is small:

$$\begin{aligned} \mathrm{BCE}(\hat{M}, M) &= -\frac{1}{HW}\sum_{x,y}\Big[M_{xy}\log\hat{M}_{xy} + (1-M_{xy})\log(1-\hat{M}_{xy})\Big], \\ \mathrm{Dice}(\hat{M}, M) &= 1 - \frac{2\sum_{x,y}\hat{M}_{xy}M_{xy} + \epsilon}{\sum_{x,y}\hat{M}_{xy} + \sum_{x,y}M_{xy} + \epsilon}, \end{aligned} \tag{11}$$

where $\hat{M} = f(I^o_{\text{inp}}, I^e_{\text{inp}})$, $M = M^{\text{inp}}$, and ϵ is a small constant that keeps the Dice ratio finite. Both terms take a probability map and a binary target and return a scalar loss. We set $\mathcal{L}_{\text{seg}} = \mathrm{BCE}(\hat{M}, M) + \mathrm{Dice}(\hat{M}, M)$. Stage I therefore supplies a strong prior for localized-edit detection, while leaving the model overly sensitive to global pixel perturbations.

Stage II adapts SIGMA to unlabeled text-driven pairs from public corpora (AnyEdit [8], CrispEdit-2M [41], OmniEdit-Filtered-1.2M [42], PromptfixData [43]) with three self-supervised signals on top of the retained $\mathcal{L}_{\text{seg}}$.

(i) Noise calibration. For each I^o, a VAE-roundtrip $\hat{I}^o = \mathcal{V}_{\text{dec}}(\mathcal{V}_{\text{enc}}(I^o) + \epsilon)$ ($\epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, $\sigma \sim \mathcal{U}(0, 0.08)$) yields a semantically identical reference whose ground truth is all-zero, supervised by $\mathcal{L}_{\text{calib}} = \mathrm{BCE}(f(I^o, \hat{I}^o), \mathbf{0})$. The VAE $\mathcal{V}$ is that of Stable Diffusion 1.5 [44] or Stable Diffusion XL [45].

(ii) EMA self-training. After each student update, the teacher parameters follow

$$\theta_{\text{EMA}} \leftarrow \mu\,\theta_{\text{EMA}} + (1-\mu)\,\theta, \tag{12}$$

where θ denotes the trainable student parameters, θ_{EMA} denotes the teacher parameters, and the decay is $\mu = 0.999$. The teacher f^{EMA}_θ supplies confident pseudo-labels $\hat{M}$ on (I^o, I^e), used by $\mathcal{L}_{\text{pl}} = C(\hat{M}) \cdot \mathrm{BCE}(f(I^o, I^e), \mathrm{sg}(\hat{M}))$ with $C(\hat{M}) = \mathbf{1}[\hat{M} > 0.8] + \mathbf{1}[\hat{M} < 0.2]$. Here $\mathrm{sg}(\cdot)$ is the stop-gradient operator, so the teacher is not updated through $\mathcal{L}_{\text{pl}}$.

(iii) Edit-noise disentanglement. On the predicted-edit support P, we contrast the DRB difference vectors of the real edit pair and the noise pair via $\mathcal{L}_{\text{disent}} = \frac{1}{|P|}\sum_{i\in\Omega_p} P_i \max(0, \langle \mathbf{d}^{(i)}_{\text{edit}}, \mathbf{d}^{(i)}_{\text{noise}}\rangle)$, pushing them into separable feature subspaces.

The full objective $\mathcal{L}_{\text{Stage II}} = 10\mathcal{L}_{\text{seg}} + 0.1\mathcal{L}_{\text{calib}} + 0.5\mathcal{L}_{\text{pl}} + 0.5\mathcal{L}_{\text{disent}}$ assigns the dominant weight to the only strongly supervised term while preventing collapse on the self-supervised streams.

4 Experiments and Analysis

4.1 Experimental Setup

Datasets. We use three groups of datasets: (i) For evaluation, we adopt five text-driven forgery localization benchmarks that provide original images, edited images, and pixel-level ground-truth masks: CoCoGlide [13], AutoSplice [30], MagicBrush [31], DEAL-300K [32], and OpenSDI [33], where all benchmarks except OpenSDI provide explicit text instructions describing the edits.They

Table 1: Quantitative comparison across different datasets. F1 and IoU scores (%) are reported.

Methods	Datasets											
	CoCoGlide		AutoSplice		MagicBrush		DEAL-300K		OpenSDI		Average	
	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU
PixDiff-Fixed($\tau = 0$)	35.46	25.22	53.90	40.94	23.48	14.79	16.38	10.33	24.65	16.17	30.77	21.49
PixDiff-Fixed($\tau = 10$)	92.26	86.22	92.18	86.03	74.89	62.93	28.04	19.11	68.11	56.18	71.10	62.09
PixDiff-Fixed($\tau = 20$)	82.01	71.00	86.15	76.57	69.11	55.52	33.44	23.28	63.85	49.65	66.91	55.20
PixDiff-Fixed($\tau = 30$)	72.27	58.62	79.76	67.50	62.10	47.66	35.24	24.41	55.50	40.61	60.97	47.76
PixDiff-Fixed($\tau = 40$)	63.34	48.59	73.46	59.37	55.45	40.83	35.24	24.10	47.23	32.81	54.94	41.14
PixDiff-Fixed-morph	59.92	46.12	77.67	65.22	64.06	50.03	46.33	34.44	51.91	37.45	59.98	46.65
PixDiff-Otsu	62.40	46.54	62.97	47.07	51.88	36.84	34.21	23.17	50.07	34.95	52.31	37.71
Qwen2.5VL-LangSAM	84.54	77.66	86.19	80.30	38.95	29.09	43.13	33.68	43.71	36.19	59.31	51.38
Meta-CD (TGRS 2025)	73.50	62.40	86.90	77.80	76.00	64.10	45.80	33.80	74.28	62.19	71.30	60.06
DDPM-CD (WACV 2025)	**97.80**	**96.00**	96.20	92.80	**89.90**	**83.10**	20.90	13.80	83.35	76.14	77.63	72.37
SIGMA	95.62	91.84	**96.69**	**93.74**	89.03	81.91	**76.43**	**65.31**	**91.37**	**84.82**	**89.83**	**83.52**



Figure 3: Qualitative comparison of SIGMA and other methods.

are used both to compare the mask quality of SIGMA against existing automatic generators and as held-out test sets for evaluating the cross-dataset generalization of IML detectors fine-tuned on SIGMA-annotated data. (ii) To generate large-scale training IML data, we employ SIGMA on four extensive public image-editing corpora that were originally distributed without pixel-level annotation masks: AnyEdit [8], CrispEdit-2M [41], OmniEdit-Filtered-1.2M [42], and PromptfixData [43]. Together, these corpora comprise millions of *(original, edited, instruction)* triplets across a broad range of editors and edit types. (iii) For Stage I training of SIGMA, we utilize the BR-Gen[46] dataset, specifically concentrating on image inpainting operations.

Comparative methodologies for mask generation. We compare SIGMA against two families of baselines: (i) pixel-differencing variants without learning, including **PixDiff-Fixed(τ)** with $\tau \in \{0, 10, 20, 30, 40\}$, **PixDiff-Fixed-morph** (on the PixDiff-Fixed binary mask at τ=25, we apply morphological opening (5×5 kernel) followed by closing), and **PixDiff-Otsu** (automatic thresholding); and (ii) learning-based change detectors adapted from remote sensing, namely **DDPM-CD** [36] and **Meta-CD** [35]. For fair comparison, the two learning-based baselines are trained on the same inpainting dataset used in SIGMA's Stage I.

Implementation details. We employ DINOv2-Base [11] as a frozen feature-extraction backbone and set the number of DRB modules to 2. The number of iterative refinement steps K employed in the Bidirectional Cross-Modal Refinement procedure is fixed to 2. The model is optimized using AdamW (with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay $= 0.05$), an initial learning rate of 10^{-3} with cosine annealing, and a batch size of 8 on a single NVIDIA A100 GPU. Input images are resized to 518×518 pixels. Data augmentation consists of synchronized random horizontal flipping and random cropping, applied identically to both images in each input pair. Flipping does not rewrite side-specific words in the instruction. Appendix G measures how often this misaligns the instruction with the image and compares a variant that withholds the flip for those samples. We trained the SIGMA model for 10 epochs during Stage I and subsequently fine-tuned it for an additional 5 epochs during Stage II.

4.2 Quality Assessment of Generated Masks

Tab. 1 compares mask quality for pixel-level thresholding, morphological heuristics, and learning-based change detection. PixDiff-Fixed is sensitive to the threshold: at τ=10 it reaches 71.10% average F1, and performance drops for the other values of τ. PixDiff-Otsu reaches 52.31% F1. DDPM-CD [36] reaches 97.80% F1 on CoCoGlide but only 20.90% F1 on DEAL-300K, so its 77.63% average does not hold across datasets. After Stage II, SIGMA reaches 89.83% average F1 and 83.52% IoU, which is 12.20 points above DDPM-CD in F1. The largest margin is on DEAL-300K,

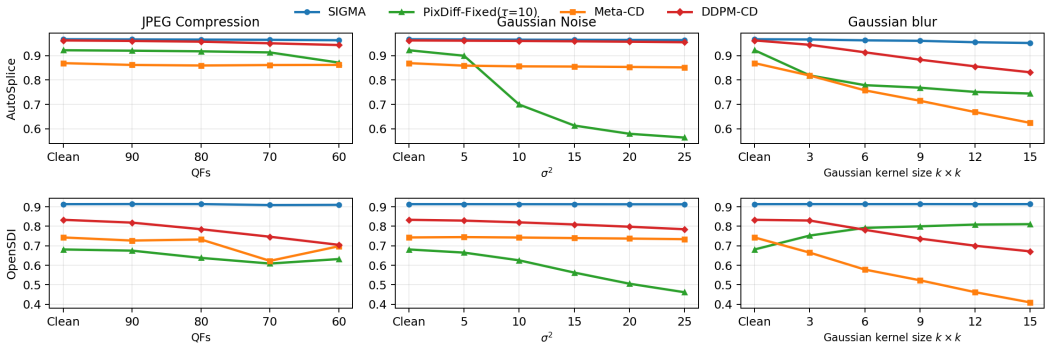


Figure 4: Evaluation of the robustness of SIGMA and comparative baseline methods to JPEG compression, additive white Gaussian noise, and Gaussian blurring on the AutoSplice and OpenSDI datasets.

where SIGMA exceeds DDPM-CD by 55.53 F1 points. This pattern underscores genuine robustness rather than dataset-specific overfitting. Qualitative comparisons (Fig. 3) reveal that pixel-difference baselines fail under dense synthesis noise, while Meta-CD and DDPM-CD show complementary failure modes, namely over-dilated masks and missed fine-grained edits, respectively. SIGMA, in contrast, closely matches the ground truth across object-level, identity, and largely edits, visually corroborating its quantitative superiority and reinforcing the necessity of semantic-level, instruction-grounded reasoning. A stratified human audit of these masks on AnyEdit and CrispEdit-2M is reported in Appendix F.

4.3 Robustness to Post-Processing Operations

To assess the robustness of SIGMA to typical post-processing operations, we subject the test images to three widely used perturbations: JPEG compression (quality factor QF $\in \{90, \ldots, 60\}$), additive white Gaussian noise with variance $\sigma^2 \in \{5, \ldots, 25\}$, and Gaussian blurring with kernel sizes $\in \{3, \ldots, 15\}$. We then quantitatively compare the performance of SIGMA against PixDiff-Fixed (τ=10), Meta-CD [35], and DDPM-CD [36] on the AutoSplice and OpenSDI benchmarks. As shown in Fig. 4, SIGMA delivers near-flat curves across all three perturbation types and severity levels on both benchmarks, holding $\sim$0.95 F1 on AutoSplice and $\sim$0.91 F1 on OpenSDI throughout. In contrast, PixDiff-Fixed collapses under Gaussian noise (e.g., from $\sim$0.92 to $\sim$0.56 on AutoSplice at σ^2=25), confirming that pixel-difference signals are intrinsically vulnerable to additive perturbations; Meta-CD and DDPM-CD degrade visibly under blur, with Meta-CD dropping by over 0.24 F1 from clean to a 15×15 kernel on AutoSplice and from $\sim$0.74 to $\sim$0.41 on OpenSDI, while DDPM-CD declines by $\sim$0.13 on AutoSplice and from $\sim$0.83 to $\sim$0.67 on OpenSDI under the same kernels. These results confirm that the semantic-feature differencing at the heart of SIGMA confers substantially greater forensic robustness than pixel-level differencing or remote-sensing-style change detection, both of which inherit a strong dependence on the low-level statistics most affected by post-processing.

4.4 Enhancing the Generalization Ability of Existing IML Models

The principal contribution of SIGMA does not reside in its intrinsic localization accuracy, but rather in its capacity to transform large-scale, unlabeled, text-driven image editing corpora into pixel-level supervisory signals that can be effectively leveraged by arbitrary downstream IML models. We apply SIGMA to large-scale public image-editing datasets to generate corresponding mask annotations (see Appendix E for more details about the SIGMA-annotated dataset), and use the resulting data to fine-tune four representative IML models spanning diverse architectural paradigms: CAT-Net [47], MVSS-Net [48], PSCC-Net [49], and TruFor [13]. These models are finetuned using the implementation from[50]. The fine-tuned models are then evaluated on five disjoint benchmarks: AutoSplice, CoCoGlide, DEAL-300K, MagicBrush, and OpenSDI, to assess their generalization performance. As reported in Tab. 2, fine-tuning on SIGMA-annotated data raises the average F1 of all four detectors over their public weights, by +23.43% for CAT-Net (21.45 to 44.88), +30.79% for

Table 2: Cross-dataset localization after fine-tuning on automatically annotated editing data. For each detector we report publicly released weights, fine-tuning on the same image pairs annotated by DDPM-CD, and fine-tuning on those pairs annotated by SIGMA (Finetune w/ SIGMA). F1 and IoU scores (%) are reported; the best result within each method is in **bold**.

Method	Weights	AutoSplice		CocoGlide		DEAL300K		MagicBrush		OpenSDI		Average	
		F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU
CAT-Net	Public Weight	41.49	31.79	40.97	33.69	12.14	8.75	5.33	3.29	7.32	5.27	21.45	16.56
	Finetune w/ DDPM-CD	44.27	30.48	32.83	21.99	13.83	8.40	59.19	42.99	20.56	12.68	34.14	23.31
	Finetune w/ SIGMA	**60.82**	**47.23**	**45.53**	**34.42**	**29.99**	**20.98**	**61.02**	**50.05**	**27.02**	**18.14**	**44.88**	**34.16**
MVSSNet	Public Weight	18.04	12.22	32.80	25.30	5.43	3.61	1.79	1.01	1.46	0.91	11.90	8.61
	Finetune w/ DDPM-CD	53.98	41.03	35.51	25.27	16.40	10.34	**91.20**	**85.10**	**25.06**	**16.46**	**44.43**	**35.64**
	Finetune w/ SIGMA	**54.14**	**41.43**	**35.63**	**25.38**	**20.05**	**13.29**	81.95	73.72	21.69	14.51	42.69	33.67
PSCC-Net	Public Weight	56.36	43.83	**50.85**	**40.60**	20.19	13.69	14.87	9.37	14.50	9.64	31.35	23.43
	Finetune w/ DDPM-CD	54.19	41.33	36.16	27.46	16.22	10.77	**41.15**	**32.71**	18.54	12.10	33.25	24.87
	Finetune w/ SIGMA	**57.06**	**45.46**	42.30	31.45	**21.16**	**14.20**	36.69	26.67	**22.44**	**15.25**	**35.93**	**26.61**
Trufor	Public Weight	32.69	25.52	28.33	24.33	9.26	6.77	2.51	1.48	6.93	5.22	15.94	12.66
	Finetune w/ DDPM-CD	57.06	41.97	55.33	42.31	22.49	14.36	20.63	12.36	20.13	12.78	35.13	24.76
	Finetune w/ SIGMA	**63.42**	**50.63**	**58.72**	**47.34**	**27.85**	**18.92**	**28.35**	**20.06**	**27.02**	**19.20**	**41.07**	**31.23**

Table 3: Ablation study of SIGMA's architectural components and training strategy across five benchmarks. The first six rows progressively add the proposed modules (MSDB, DRBs at varying depths, and IGB), while the last row adds Stage II adaptation on top of the full architecture. F1 and IoU scores (%) are reported; best results in each column are in **bold**.

Model Variants	Datasets											
	CoCoGlide		AutoSplice		MagicBrush		DEAL-300K		OpenSDI		Average	
	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU
baseline	93.80	88.90	95.49	91.79	78.34	68.00	57.89	46.22	85.54	77.13	82.21	74.41
+msdb	94.82	90.52	96.32	93.09	89.87	82.94	55.16	43.29	89.66	82.64	85.17	78.50
+msdb+2DRBs	95.60	91.81	**96.82**	**93.98**	89.81	82.82	67.20	55.92	90.75	84.03	88.04	81.71
+msdb+4DRBs	95.14	90.99	96.33	93.10	**90.04**	**83.13**	65.43	54.40	90.07	83.03	87.40	80.93
+msdb+8DRBs	95.32	91.32	96.33	93.08	89.23	82.09	68.70	57.31	90.15	83.13	87.95	81.39
+msdb+2DRBs+IGB	94.91	90.62	96.27	92.99	87.89	80.18	75.02	63.48	90.97	84.29	89.01	82.31
+msdb+2DRBs+IGB+Stage II training	**95.62**	**91.84**	96.69	93.74	89.03	81.91	**76.43**	**65.31**	**91.37**	**84.82**	**89.83**	**83.52**

MVSS-Net (11.90 to 42.69), +4.58% for PSCC-Net (31.35 to 35.93), and +25.13% for TruFor (15.94 to 41.07). On the same image pairs annotated by DDPM-CD, SIGMA gives a higher average F1 for CAT-Net (44.88 versus 34.14), PSCC-Net (35.93 versus 33.25), and TruFor (41.07 versus 35.13). These results show that SIGMA-annotated data is a model-agnostic supervisory resource that lets existing IML detectors use large-scale text-driven editing corpora.

4.5 Ablation Studies

Architectural components. Tab. 3 reports the ablation. Replacing single-scale differencing with the **Multi-Scale Difference Block (MSDB)** yields +2.96% F1 and +4.09% IoU on average, with the largest gains on MagicBrush (+11.53% F1) and OpenSDI (+4.12% F1), confirming that edits span heterogeneous scales and benefit from cross-level aggregation. Stacking **Difference Refinement Blocks (DRBs)** further refines the representation, but the gain is non-monotonic: 2 DRBs reach the best average (88.04% F1, 81.71% IoU), while 4 and 8 DRBs slightly degrade easier datasets (e.g., 0.46% F1 drop on CoCoGlide and 0.49% on AutoSplice from 2 to 4 DRBs). We trace this to two reinforcing effects: residual stacking steadily smooths the difference map and washes out edit boundaries, while the inpainting-bounded Stage I supervision starves deeper stacks of useful gradients. Extra parameters thus inject optimization noise instead of real capacity. We therefore adopt 2 DRBs by default. Adding the **Instruction-Grounding Branch (IGB)** delivers another +0.97% F1 on average and, most strikingly, raises DEAL-300K F1 from 67.20% to 75.02% (+7.82%), confirming that visual differencing alone cannot disambiguate which synthesis-induced perturbations correspond to user intent and that instruction conditioning is critical for global or semantically nuanced edits.

Training strategy. Stage I alone attains 89.01% F1 and 82.31% IoU, indicating that supervised pre-training on inpainting masks already instills a strong semantic-change prior. However, never having seen diffusion-induced reconstruction noise, it remains partially sensitive to the pervasive pixel perturbations of text-driven edits. Stage II addresses this by adapting the model to unlabeled editing pairs under three complementary objectives, noise calibration, EMA self-training, and edit-noise

feature disentanglement, without requiring any pixel-level annotation in the target domain. The adaptation adds +0.82% F1 and +1.21% IoU, with the largest gains on the hardest distributions (+1.41% F1 on DEAL-300K), and additionally reverses the small regressions that Stage I alone incurs on CoCoGlide and AutoSplice.

5 Conclusion and Limitations

Conclusion. We presented **SIGMA**, a framework that turns publicly available image-editing datasets into pixel-accurate supervision for image manipulation localization (IML). Building on the observation that pixel differencing and instruction-only grounding are complementary but individually insufficient under diffusion-based editing, SIGMA performs semantic-feature differencing in a frozen foundation backbone and fuses it with a parsed-instruction prior via bidirectional cross-modal refinement, while a two-stage training strategy bridges supervision scarcity and the diffusion-domain shift. SIGMA outperforms existing automatic mask generators by **+12.20%** F1 on five benchmarks and, when applied to public editing corpora, yields a million-scale IML training set that boosts four diverse detectors by **+20.98%** F1 in cross-dataset generalization, establishing SIGMA-annotated data as a model-agnostic supervisory resource for the IML community.

Limitations. SIGMA relies on off-the-shelf parsing and grounding models, whose precision limits can occasionally contaminate the instruction prior (mitigated, but not eliminated, by cross-modal refinement). The parser also returns a single operation tuple per instruction. The 1.1M corpus and the benchmarks used here are almost entirely single-round, single-operation edits, and explicitly multi-operation samples account for approximately 0% of that corpus. Multi-round instructions that apply several operations in sequence are therefore outside the present scope: a later operation can be missed by the instruction branch, and the predicted mask is not split into one region per operation. SIGMA targets semantic edits and is therefore less precise on purely global stylistic changes, where the prompt provides only a weak spatial signal. Finally, our evaluation focuses on still images; extending SIGMA to text-driven video forgeries is an urgent open direction.

Broader Impact. SIGMA aims to strengthen detection of text-driven forgeries and thereby support media verification. The same automated mask-generation capability could in principle refine adversarial editing pipelines. However, SIGMA improves only the detector side rather than the editor, and we believe the net benefit to media integrity is positive.

Acknowledgments and Disclosure of Funding

This work was supported in part by the New Generation Artificial Intelligence-National Science and Technology Major Project under Grant 2025ZD0123402, in part by the National Natural Science Foundation of China under Grant 62602719, Grant 62625602, Grant 62441619, Grant 62372489, Grant 62572325, and Grant 6260072840, in part by the Guangdong S&T Program under Grant 2026B0101100003, in part by the Shenzhen Science and Technology Program under Grant SYSRD20250529113401002 and Grant JCYJ20230807111207015, and in part by Guangdong Provincial Key Laboratory of Information Security Technology.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning To Follow Image Editing Instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[2] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*, 2023.

[3] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

[4] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[5] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

[6] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[7] Di Chang, Mingdeng Cao, Yichun Shi, Bo Liu, Shengqu Cai, Shijie Zhou, Weilin Huang, Gordon Wetzstein, Mohammad Soleymani, and Peng Wang. Bytemorph: Benchmarking instruction-guided image editing with non-rigid motions. *arXiv preprint arXiv:2506.03107*, 2025.

[8] 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*, pages 26125–26135, 2025.

[9] 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*, pages 28831–28841, 2025.

[10] 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*, pages 3538–3547, 2024.

[11] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 2024.

[12] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries With Anomalous Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9543–9552, 2019.

[13] 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*, pages 20606–20615, 2023.

[14] 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*, pages 3155–3165, 2023.

[15] Zeqin Yu, Jiangqun Ni, Yuzhen Lin, Haoyi Deng, and Bin Li. DiffForensics: Leveraging Diffusion Prior to Image Forgery Detection and Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12765–12774, 2024.

[16] 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(7):7024–7032, 2025.

[17] Peiyu Zhuang, Haodong Li, Shunquan Tan, Bin Li, and Jiwu Huang. Image tampering localization using a dense fully convolutional network. *IEEE Transactions on Information Forensics and Security*, 16:2986–2999, 2021.

[18] Peiyu Zhuang, Haodong Li, Rui Yang, and Jiwu Huang. Reloc: A restoration-assisted framework for robust image tampering localization. *IEEE Transactions on Information Forensics and Security*, 18:5243–5257, 2023.

[19] Haodong Li, Peiyu Zhuang, Yang Su, and Jiwu Huang. Towards generalizable and robust image tampering localization with multi-task learning and contrastive learning. *Expert Systems with Applications*, 270:126492, 2025.

[20] Zeqin Yu, Haotao Xie, Jian Zhang, Jiangqun Ni, Wenkang Su, and Jiwu Huang. Toward real-world text image forgery localization: Structured and interpretable data synthesis. *Advances in Neural Information Processing Systems*, 38, 2026.

[21] Zeqin Yu, Jiangqun Ni, Jian Zhang, Haoyi Deng, and Yuzhen Lin. Reinforced multi-teacher knowledge distillation for efficient general image forgery detection and localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 995–1003, 2025.

[22] Zeqin Yu, Bin Li, Yuzhen Lin, Jinhua Zeng, and Jishen Zeng. Learning to locate the text forgery in smartphone screenshots. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[23] 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 *The Thirteenth International Conference on Learning Representations*, 2025.

[24] 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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[25] Hengrui Kang, Siwei Wen, Zichen Wen, Junyan Ye, Weijia Li, Peilin Feng, Baichuan Zhou, Bin Wang, Dahua Lin, Linfeng Zhang, et al. Legion: Learning to ground and explain for synthetic image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18937–18947, 2025.

[26] Y. Hsu and S. Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *IEEE Inter. Conf. Multim. Expo*, pages 549–552. IEEE, 2006.

[27] J. Dong, W. Wang, and T. Tan. Casia image tampering detection evaluation database. In *IEEE China Summit Inter. Conf. Signal Info. Proc.*, pages 422–426, 2013.

[28] Adam Novozamsky, Babak Mahdian, and Stanislav Saic. IMD2020: A large-scale annotated dataset tailored for detecting manipulated images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 71–80, 2020.

[29] Gaël Mahfoudi, Badr Tajini, Florent Retraint, Frederic Morain-Nicolier, Jean Luc Dugelay, and PIC Marc. DEFACTO: Image and face manipulation dataset. In *Proceedings of the European Signal Processing Conference*, pages 1–5, 2019.

[30] Shan Jia, Mingzhen Huang, Zhou Zhou, Yan Ju, Jialing Cai, and Siwei Lyu. AutoSplice: A Text-Prompt Manipulated Image Dataset for Media Forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 893–903, 2023.

[31] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.

[32] 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*, 2025.

[33] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. OpenSDI: Spotting Diffusion-Generated Images in the Open World. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4291–4301, 2025.

[34] Wele Gedara Chaminda Bandara and Vishal M. Patel. A Transformer-Based Siamese Network for Change Detection. In *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 207–210, 2022.

[35] Junyu Gao, Da Zhang, Feiyu Wang, Lichen Ning, Zhiyuan Zhao, and Xuelong Li. Combining SAM With Limited Data for Change Detection in Remote Sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–11, 2025.

[36] Wele Gedara Chaminda Bandara, Nithin Gopalakrishnan Nair, and Vishal M. Patel. DDPM-CD: Denoising Diffusion Probabilistic Models as Feature Extractors for Remote Sensing Change Detection. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5250–5262, 2025.

[37] Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[38] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[39] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[40] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016.

[41] Wei Chow, Linfeng Li, Lingdong Kong, Zefeng Li, Qi Xu, Hang Song, Tian Ye, Xian Wang, Jinbin Bai, Shilin Xu, et al. Editmgt: Unleashing potentials of masked generative transformers in image editing. *arXiv preprint arXiv:2512.11715*, 2025.

[42] Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. Omniedit: Building image editing generalist models through specialist supervision. In *The Thirteenth International Conference on Learning Representations*, 2025.

[43] Yongsheng Yu, Ziyun Zeng, Hang Hua, Jianlong Fu, and Jiebo Luo. Promptfix: You prompt and we fix the photo. *Advances in Neural Information Processing Systems*, 37:40000–40031, 2024.

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 10674–10685, 2022.

[45] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

[46] 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, pages 2534–2542, 2026.

[47] 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)*, pages 375–384, 2021.

[48] Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao, and Xirong Li. MVSS-Net: Multi-View Multi-Scale Supervised Networks for Image Manipulation Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3539–3553, March 2023.

[49] 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.

[50] 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, 2025.

A Scale Disparity between Image Editing and IML Datasets

A core motivating premise of this work is the substantial scale asymmetry between publicly available image-editing datasets and image manipulation localization (IML) datasets that supply pixel-accurate masks. We document this asymmetry quantitatively in Fig. 5, plotting both groups on a shared linear axis to make the gap directly visible. Editing datasets already contain *(original, edited, instruction)* triplets that, by construction, share the manipulation distribution targeted by modern IML; the only missing ingredient is the pixel-level mask. Conversely, IML datasets carry the masks but are limited in scale, in the diversity of editors covered, and in their coverage of the latest text-driven editing paradigms (e.g., FLUX Kontext, OmniGen2, Qwen-Image-Edit).

This observation grounds the central question of the paper: rather than collecting another IML benchmark from scratch, can we automatically annotate the editing corpora that already exist? SIGMA is the affirmative answer. By generating pixel-level accurate masks for arbitrary text-driven editing pairs, our approach converts four large-scale datasets (AnyEdit[8], CrispEdit-2M[41], OmniEdit-Filtered-1.2M[42], and PromptfixData[43]) to SIGMA-annotated dataset (see Appendix E for more details) into rich supervisory signals that can be leveraged by any downstream IML detector. In doing so, it converts a vast, long-neglected data reservoir into a powerful, model-agnostic IML training pool.

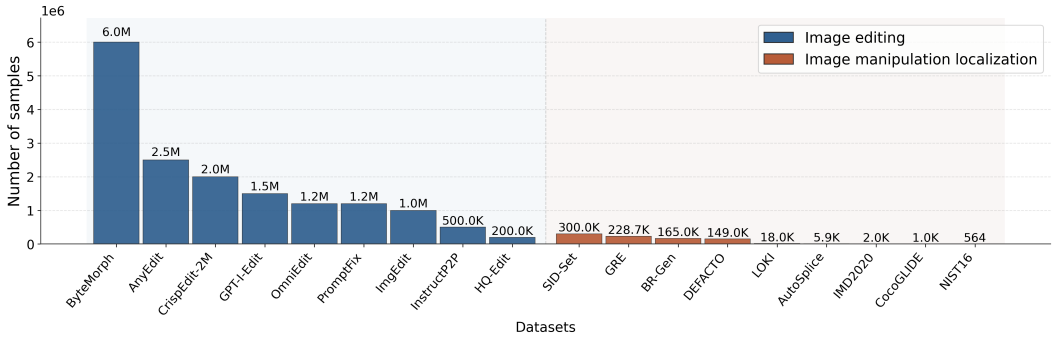


Figure 5: Sample counts of public image-editing datasets (blue) and IML datasets (orange) on a shared linear scale.

B Parsing Editing Instructions into Transformation Tuples

Table 4 primarily reports the original and edited concepts embodied in several representative editing instructions, together with their corresponding editing operations. We anticipate that our STP module will efficiently process these instructions, accurately identify the principal conceptual representations, and extract the precise editing operations they specify.

Table 4: Representative editing instructions and the semantic transformation tuples (c^o, c^e, op) extracted by our STP module. $\varnothing$ denotes an empty concept slot, used when an instruction introduces a new concept (*add*), removes an existing one (*remove*), or alters the image globally without a specific subject (*global*).

Instruction	c^o	c^e	op
"remove the cat"	"a cat"	$\varnothing$	remove
"add a hat on the person"	$\varnothing$	"a hat"	add
"turn the wood into marble"	"wood texture"	"marble texture"	replace
"Change the boy's grey striped t-shirt to a bright blue t-shirt"	"grey striped t-shirt"	"blue t-shirt"	attribute change
"make it more dramatic"	$\varnothing$	$\varnothing$	global

```
SYSTEM_PROMPT = """You are an expert at understanding image editing instructions.
Convert each instruction into EXACTLY one JSON object with these keys:
- "original concept"
- "edited concept"
- "action type"

Allowed values for "action type" are only:
1) "add"
2) "remove"
3) "attribute change"
4) "replace"
5) "global"

Definitions:
- add: a new object/concept is introduced.
- remove: an existing object/concept is deleted.
- attribute change: same object identity, only attribute/state/color/pose/
  expression/action/style detail changes.
- replace: one object/concept is swapped with another object/concept.
- global: whole-image change (e.g., weather, season, style, lighting,
  time of day).

Rules:
- Return strict JSON only, no markdown, no extra text.
- If information is implicit, infer concise concepts.
- Keep concepts short noun phrases.

Examples:
Instruction: "add a bird"
{"original concept":"null","edited concept":"a bird","action type":"add"}

Instruction: "remove the umbrella"
{"original concept":"umbrella present","edited concept":"null",
 "action type":"remove"}

Instruction: "change the red flowers to white flowers"
{"original concept":"red flowers","edited concept":"white flowers",
 "action type":"attribute change"}

Instruction: "replace the cat with a dog"
{"original concept":"cat","edited concept":"dog","action type":"replace"}

Instruction: "make it snowy"
{"original concept":"null","edited concept":"null","action type":"global"}
"""
```

Figure 6: System prompt used by our Semantic Transformation Parser (STP).

C The Prompt used in Semantic Transformation Parser

Fig. 6 presents the prompt we developed for the Semantic Transformation Parser module. Its primary components comprise task configuration, output format specifications, task definition, rule-based constraints, and contextualized examples. By employing this prompt, we can systematically elicit the large language model's inference of the tuple (c^o, c^e, op) from a given instruction.

D Visualization of the Instruction-Grounding Branch

Fig. 7 visualizes the intermediate outputs of the instruction-grounding branch on four representative editing types, illustrating how the parsed transformation tuple (c^o, c^e, op) propagates through SAM-guided attention construction and action-conditioned spatial fusion. For *attribute change* (row 1, recoloring the armchairs), STP extracts nearly identical concepts that differ only in the color attribute, so A^o and A^e both highlight the armchair regions, and the product rule yields a tightly localized M_{intent} that pinpoints the affected objects. For *replace* (row 2, in which the wood-and-stone interior is transformed into a white living room), the original and edited concepts, each grounded in their respective images, span broad and partially overlapping spatial regions. ACSF's product rule preserves only those patches that express the original concept in I^o and the edited concept in I^e, specifically the walls and fireplace. This process excludes patches corresponding to unedited furniture. For *remove*

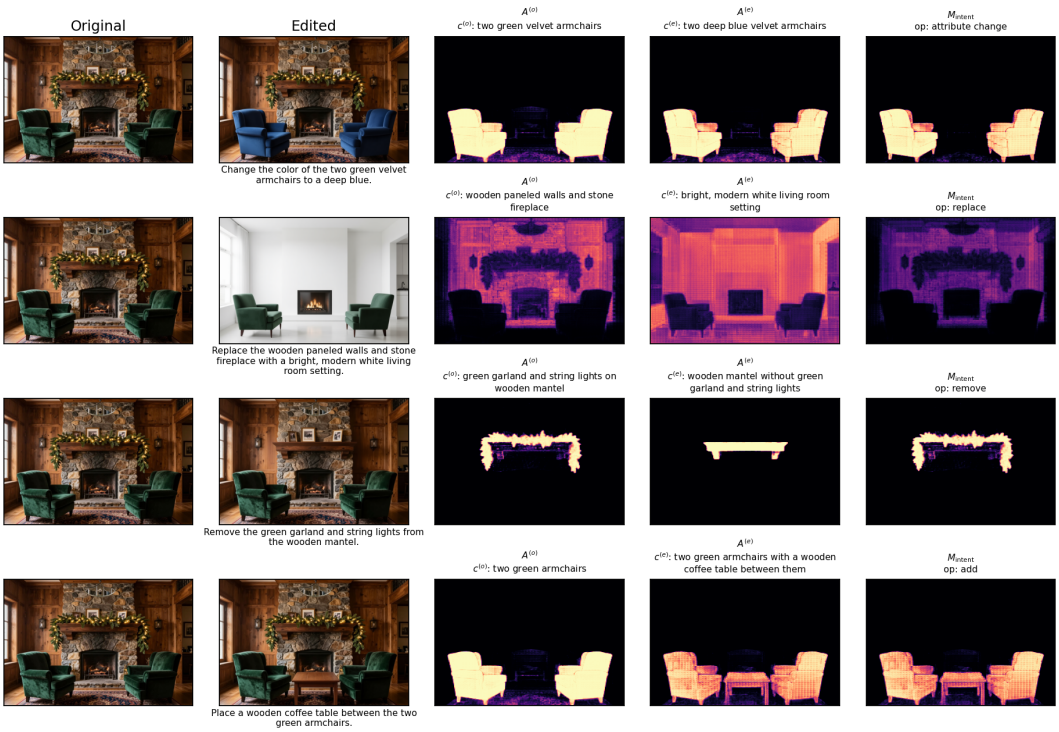


Figure 7: The Visualization of the intermediate image outputs of the text-grounding branch.

(row 3, removing the garland), c^e describes a residual "mantel without garland"; A^o cleanly localizes the garland on I^o, ACSF outputs A^o as the intent map, and the removed region is faithfully recovered.

Row 4 (*add*) is informative as a *failure case* of instruction grounding alone: STP over-extracts the contextual phrase "two green armchairs with a wooden coffee table between them" as c^e, causing A^e to highlight not only the newly added table but also the surrounding armchairs that were never edited. Since the *add* rule directly forwards A^e as M_{intent}, the resulting prior inherits this contamination, illustrating that the instruction-grounding branch is intrinsically vulnerable to the precision limits of off-the-shelf parsers and grounders. This specific failure mode illustrates why the instruction branch is not designated as authoritative. Within the complete SIGMA pipeline, the bidirectional cross-modal refinement module validates M_{intent} against the available semantic evidence. The unaltered armchairs, which exhibit no discernible semantic modifications, are subsequently attenuated or filtered out by the semantic branch during the multimodal fusion stage. The example thus concretely supports our central design decision: semantic differencing and instruction grounding must serve as *mutually verifiable* sources rather than either being trusted alone.

E Details of the SIGMA-annotated Dataset

We build a large-scale, annotation-rich IML dataset by systematically mining four public generative image editing corpora: AnyEdit[8], CrispEdit-2M[41], OmniEdit-Filtered-1.2M[42], and PromptfixData[43]. Each corpus comprises paired original and edited images, accompanied by structured editing metadata, which we transform using SIGMA into a precise, fully automated localization signal, thereby obviating any requirement for manual annotation.

As shown in Fig. 8, the final dataset contains $\sim$1.1M image pairs covering 24 distinct categories of editing operations, with substantial diversity in both editing semantics and visual appearance. The four sources offer complementary coverage across manipulation regimes. AnyEdit[8] provides the broadest semantic coverage to date, comprising 12 carefully curated and specialized operation types. The most frequent categories include large-scale background modification (103K instances), object addition (64K instances), and color transformation (51K instances). The benchmark further incorporates highly localized, brush-level editing primitives (visual scribble, visual segment, and

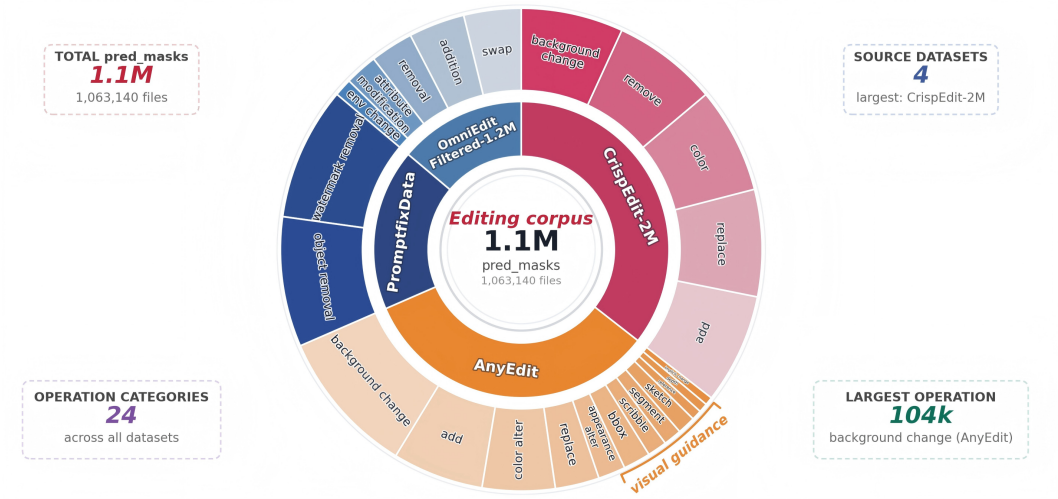


Figure 8: Overview of SIGMA-converted IML training data from four generative image editing corpora and their editing operation taxonomy.

visual sketch), which collectively impose stringent requirements on ultra-fine spatial localization and thus serve as rigorous stress tests for high-precision editing capabilities. CrispEdit-2M[41] provides the largest single-source corpus (377K samples), characterized by a well-balanced distribution over five object-centric transformation types: addition, background modification, color alteration, object removal, and object replacement, thereby furnishing robust coverage of high-frequency editing intents observed in real-world image manipulation scenarios. OmniEdit-Filtered-1.2M[42] introduces a set of structurally distinct edit categories, including attribute modification, attribute swapping, and environment transformation. These categories are designed to encompass complex, composite multi-attribute manipulations that present qualitatively different challenges compared to conventional single-operation forgeries. PromptfixData[43] places a distinctive emphasis on object removal and watermark removal (187K instances), thereby directly addressing the practically important yet historically underexplored problem of content-erasure forgeries. Collectively, these dataset design choices mitigate the dominance of any single editing modality and extend the localization problem to cover a continuum of manipulations, ranging from coarse regional edits to fine-grained, pixel-level erasures.

Ground-truth localization masks are derived automatically from SIGMA as high-fidelity, binary, pixel-level annotations that precisely delineate manipulated regions. Since all source images are generated using contemporary text-guided, diffusion-based editing pipelines, the resulting forgeries exhibit generator-consistent texture statistics, realistic illumination harmonization, and strong semantic coherence. These characteristics distinguish them from traditional cut-and-paste splicing operations and render them representative of the forensic challenges associated with modern AI-generated and AI-edited imagery.

F Human Audit on Editing Corpora

The five public benchmarks in Tab. 1 measure mask quality only where pixel-level ground truth is already available. They leave open whether automatic masks agree with humans on the unlabeled editing corpora that SIGMA is used to annotate. We therefore built a stratified audit from the edit-type distribution in Fig. 8, drawing 1,500 image-instruction pairs from AnyEdit and 1,500 from CrispEdit-2M (3,000 pairs in total). Two annotators produced a pixel-level change mask for every pair. SIGMA and the competing generators were then scored by pixel-level F1 against these human masks. In addition to the baselines in Sec. 4.1, the comparison includes the instruction-grounding branch alone (IGB) and Qwen2.5VL-LangSAM, an instruction-only MLLM-plus-grounding pipeline.

SIGMA attains the highest average F1 on both corpora: **77.35%** on AnyEdit and **72.05%** on CrispEdit-2M (Tab. 5 and Tab. 6). These scores exceed the strongest baseline, PixDiff-Fixed-morph (70.09% and 60.55%), by 7.26 and 11.50 points. On AnyEdit, SIGMA is best on Appearance alter

Table 5: Pixel-level agreement with human masks on a stratified AnyEdit sample (1,500 pairs). F1 scores (%) are reported. Bold marks the best score in each column.

Methods	Add	Appearance alter	Background change	Color alter	Replace	Visual bbox	Visual depth	Visual reference	Visual scribble	Visual segment	Visual sketch	Avg
PixDiff-Fixed($\tau = 0$)	8.89	16.25	56.88	23.85	33.22	17.10	14.91	18.76	28.21	17.38	22.93	23.49
PixDiff-Fixed($\tau = 10$)	69.47	45.35	78.55	52.26	65.43	55.32	55.10	41.97	57.27	58.13	55.27	57.65
PixDiff-Fixed($\tau = 20$)	**74.05**	46.86	75.46	55.27	73.08	71.90	72.67	54.79	72.90	73.95	68.93	67.26
PixDiff-Fixed($\tau = 30$)	73.23	42.19	71.99	48.75	73.74	75.46	75.90	61.91	74.54	75.63	70.80	67.65
PixDiff-Fixed($\tau = 40$)	71.08	36.55	68.38	40.54	71.85	73.07	73.20	65.08	71.99	72.49	68.89	64.83
PixDiff-Fixed-morph	73.42	37.10	73.16	49.53	78.64	**76.57**	76.91	**75.42**	76.71	78.16	75.41	70.09
PixDiff-Otsu	68.98	42.53	61.28	54.54	65.33	67.38	68.39	65.22	66.40	67.91	63.99	62.90
Meta-CD	59.60	43.85	72.31	40.80	71.70	58.13	56.16	55.28	68.35	65.77	66.93	59.90
DDPM-CD	57.98	39.07	80.63	62.17	53.88	65.54	71.33	41.75	55.78	69.34	57.92	59.58
IGB	41.35	54.63	46.21	67.92	53.99	55.81	63.62	42.92	46.42	52.42	51.49	52.43
Qwen2.5VL-LangSAM	53.54	51.54	73.75	61.89	77.36	51.29	40.55	65.96	63.54	59.26	65.20	58.68
SIGMA	65.52	**64.59**	**88.64**	**76.48**	**83.61**	75.36	**77.57**	68.44	**84.59**	**81.98**	**84.09**	**77.35**

Table 6: Pixel-level agreement with human masks on a stratified CrispEdit-2M sample (1,500 pairs). F1 scores (%) are reported. Bold marks the best score in each column.

Methods	Add	Background change	Color alter	Remove	Replace	Avg
PixDiff-Fixed($\tau = 0$)	25.50	70.40	22.14	29.76	30.92	35.75
PixDiff-Fixed($\tau = 10$)	46.03	78.15	46.26	42.54	41.31	50.86
PixDiff-Fixed($\tau = 20$)	52.97	79.47	55.19	47.47	43.27	55.67
PixDiff-Fixed($\tau = 30$)	55.12	78.20	58.54	48.37	42.98	56.64
PixDiff-Fixed($\tau = 40$)	55.02	75.49	59.02	47.50	41.74	55.76
PixDiff-Fixed-morph	61.51	80.01	64.30	52.54	44.40	60.55
PixDiff-Otsu	52.63	64.87	59.15	44.69	38.50	51.97
Meta-CD	42.06	68.65	49.91	46.76	44.60	50.40
DDPM-CD	45.93	77.32	48.35	45.07	39.80	51.29
IGB	54.91	44.69	58.57	49.80	42.98	50.19
Qwen2.5VL-LangSAM	55.31	73.60	54.23	46.35	50.51	59.37
SIGMA	**74.99**	**92.22**	**78.80**	**59.55**	**54.68**	**72.05**

(64.59%), Background change (88.64%), Color alter (76.48%), Replace (83.61%), and four visual-prompt edits (depth, scribble, segment, and sketch), 8 of 11 categories. PixDiff-Fixed(τ=20) remains higher on Add (74.05% versus 65.52%), and PixDiff-Fixed-morph remains higher on Visual bbox (76.57% versus 75.36%) and Visual reference (75.42% versus 68.44%). On CrispEdit-2M, SIGMA leads every reported category. Its margins over PixDiff-Fixed-morph are largest on Add (+13.48), Background change (+12.21), and Color alter (+14.50). Agreement with humans is therefore strongest on semantic edits whose pixel residual is easily confused with diffusion noise.

The same tables locate the contribution of semantic differencing. IGB alone averages 52.43% F1 on AnyEdit and 50.19% on CrispEdit-2M. Its lowest scores are Add (41.35%), Visual reference (42.92%), Background change (46.21%), and Visual scribble (46.42%) on AnyEdit, and Replace (42.98%) and Background change (44.69%) on CrispEdit-2M. Full SIGMA raises the two averages by 24.92 and 21.86 points. Background change rises from 46.21% to 88.64% and from 44.69% to 92.22%. IGB localizes only the parsed instruction, so these gains are the regions recovered by comparing the image pair directly when parsing or grounding is inaccurate. Pixel F1 summarizes that combined outcome. It does not separate failures of object granularity, prompt ambiguity, parsing, and grounding.

G Spatial Alignment under Horizontal Flip

The training augmentation in Sec. 4.1 applies the same horizontal flip to both images in a pair and to the mask. The instruction is left unchanged. Words such as "left" and "right" then point to the opposite side of the flipped image, so the instruction and the visual target can disagree. We checked the four corpora used to build the SIGMA-annotated set, namely AnyEdit, CrispEdit-2M, OmniEdit-Filtered-1.2M, and PromptfixData. Flip-sensitive expressions occur in 5.728% of the

Table 7: Effect of horizontal flip on human-mask agreement for the AnyEdit audit (1,500 pairs). F1 scores (%) are reported. "Flip all pairs" applies the original augmentation and keeps the instruction unchanged. "Skip flip-sensitive" does not flip samples whose instructions contain side-specific expressions. Bold marks the better score in each column.

Policy	Add	Appearance alter	Background change	Color alter	Replace	Visual bbox	Visual depth	Visual reference	Visual scribble	Visual segment	Visual sketch	Avg
Flip all pairs	65.52	64.59	**88.64**	76.48	83.61	75.36	77.57	68.44	84.59	81.98	84.09	77.35
Skip flip-sensitive	**67.66**	**66.59**	87.85	**76.73**	**84.77**	**77.11**	**78.85**	**71.29**	**85.65**	**83.52**	**85.27**	**78.66**

Table 8: Effect of horizontal flip on human-mask agreement for the CrispEdit-2M audit (1,500 pairs). F1 scores (%) are reported. The two policies are defined as in Tab. 7. Bold marks the better score in each column.

Policy	Add	Background change	Color alter	Remove	Replace	Avg
Flip all pairs	74.99	**92.22**	78.80	59.55	54.68	72.05
Skip flip-sensitive	**76.75**	91.78	**79.42**	**59.88**	**55.25**	**72.62**

samples. With the 50% flip probability used in training, about 2.864% of training samples can receive a flipped image and an instruction that still names the original side.

The corrected policy keeps every other setting fixed and skips horizontal flipping whenever the instruction contains a flip-sensitive expression. We retrain SIGMA under this policy and score it with the same human masks and edit-type split as Appendix F. Tab. 7 and Tab. 8 compare it with the original policy, which flips every pair and keeps the instruction as written. The original policy is the SIGMA row of Tab. 5 and Tab. 6.

Average F1 rises from 77.35% to 78.66% on AnyEdit and from 72.05% to 72.62% on CrispEdit-2M. The corrected policy is higher in 10 of the 11 AnyEdit categories and in 4 of the 5 CrispEdit-2M categories. The only decreases are on Background change, by 0.79 points on AnyEdit (88.64% to 87.85%) and by 0.44 points on CrispEdit-2M (92.22% to 91.78%). The gains are modest because the correction changes only the small subset identified above. They are in the same direction on both corpora, so withholding the flip for side-specific instructions removes a genuine mismatch between the image and the instruction without altering the rest of the pipeline.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The motivation, contributions, and scope of our work are explicitly detailed in the Abstract and Introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of our work in the Conclusion and Limitations section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: Our work does not focus on theoretical contributions; therefore, this item is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper fully discloses the algorithmic steps, model architectures, and all necessary hyperparameters to ensure that our experimental results can be independently reproduced.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All models and datasets used in our work are publicly accessible in link: `https://github.com/ZhuangPeiyu/SIGMA`, and we have provided comprehensive implementation details and algorithmic descriptions in the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have specified all essential experimental configurations in Section 4.1

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report formal error bars in the main results, in line with common practice in large-scale IML and dense prediction benchmarks where the per-experiment training cost makes multi-seed runs prohibitively expensive, and the downstream IML fine-tuning experiments span four detectors $\times$ five datasets. Instead, we provide several alternative forms of evidence that the reported gains are statistically meaningful rather than due to run-to-run variance: (1) consistency of improvement across five independent text-driven editing benchmarks (CoCoGlide, AutoSplice, MagicBrush, DEAL-300K, OpenSDI), each with thousands of evaluation samples, gives effective sample-level error bars within each benchmark; (2) fine-tuning on SIGMA-annotated data raises the average F1 of all

four detectors (CAT-Net, MVSS-Net, PSCC-Net, and TruFor) in Table 2, with mean F1 gains from +4.58% (PSCC-Net) to +30.79% (MVSS-Net), and improves F1 in 19 of the 20 method×benchmark cells relative to the public weights (the remaining cell is PSCC-Net on CoCoGlide, from 50.85% to 42.30%); and (3) the magnitude of the headline gains (**+12.20%** F1 on mask quality and **+20.98%** F1 on cross-dataset generalization) substantially exceeds the typical run-to-run variance reported in this literature (<1% F1). We will release training code and configurations to enable independent replication.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details regarding the hardware specifications in Section4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This study does not involve any ethical issues.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have briefly discussed the potential impacts of our work in the Broader Impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Our work focuses on foundational algorithmic improvements and architectural design. It does not involve the release of high-risk pre-trained models or sensitive datasets that require specific safeguards against misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All existing assets used in this work, including models, datasets, and codebases, are properly cited in the references. We have explicitly mentioned their original sources and confirmed that our use of these assets strictly adheres to their respective open-source licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We introduce two new assets that are documented in detail. (1) The **SIGMA framework**, including model architecture, training pipeline, and pretrained checkpoints in Sec. 3. (2) A **million-scale IML training dataset** automatically annotated by SIGMA from four public image-editing corpora (AnyEdit, CrispEdit-2M, OmniEdit-Filtered-1.2M, PromptfixData), described in Appendix E with statistics on size, edit-type distribution, and source-corpus composition. The source images and editing instructions in this dataset originate entirely from the four public editing corpora cited above and are used in accordance with their original licenses; we redistribute only the SIGMA-annotated masks together with index files referencing the original source-image identifiers, so no source imagery is re-released. The dataset card, licensing terms (we plan to release the masks and code under a permissive research license, such as CC BY-NC 4.0 or MIT for code, subject to compatibility with each upstream corpus's license), known limitations (e.g., reduced precision on global stylistic edits, dependence on off-the-shelf parser/grounder), and intended use cases will accompany the release. As all source images come from public datasets without per-subject consent collected by us, the released artifacts inherit the consent and ethical scope of the upstream corpora. Anonymized code, configuration files, and a representative subset of the annotated dataset are provided at the time of submission.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: SIGMA's text-grounding branch employs a small pretrained LLM (Qwen2.5-0.5B) as the Semantic Transformation Parser (STP). The LLM is used in a frozen, in-context prompting setting to convert each free-form editing instruction T into a structured transformation tuple (c^o, c^e, op), where c^o and c^e are the original and edited concepts and $op \in \{\textit{add}, \textit{remove}, \textit{replace}, \textit{attribute change}, \textit{global}\}$. The full system prompt and representative parsed examples are provided in Appendix C and Appendix B. We do not fine-tune the LLM, do not use any other LLM in the SIGMA pipeline, and use no LLM-generated text in the training data of any IML detector evaluated in this work. Beyond this methodological role, LLMs were also used for routine writing assistance (grammar polishing, paraphrasing) on the manuscript itself, with no impact on technical content, experiments, or results.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.