GIFTBENCH: DIAGNOSING GENERALIZATION IN IMAGE FORGERY LOCALIZATION AND INFORMING MODEL DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable evaluation of image forgery localization (IFL) requires assessing models under diverse distribution changes, yet existing benchmarks often cover limited manipulation conditions or entangle multiple factors in cross-dataset evaluation. Consequently, aggregate performance provides an incomplete view of localization generalization. We introduce **GIFTBench**, a multi-axis benchmark of 115,013 manipulated images with pixel-level annotations spanning manipulation source, semantic target, editing operation, and composition complexity. GIFTBench supports axis-specific transfer analysis and evaluation on twelve external datasets. Its diagnostic studies reveal asymmetric cross-source transfer, recall-dominated failures, and heterogeneous degradation across semantic, operational, and compositional changes. Beyond diagnosis, the scale and diversity of GIFTBench provide a substantially broader training distribution than conventional IFL datasets. Training representative localizers on GIFTBench consistently improves their aggregate transfer to external datasets, showing that the benchmark serves not only as an evaluation tool but also as an effective training resource for cross-domain localization. Guided by the diagnostic findings, we further develop FORENSCOPE, a detection and localization framework combining classification-adapted representations with multi-depth, multi-scale spatial features, learned layer fusion, and selective coarse-scale conditioning. Experiments show improved cross-dataset localization while retaining image-level detection capability. The GIFTBench dataset resources and ForenScope code are available at `https://anonymous.4open.science/r/GIFTBench-ForenScope`.

1 INTRODUCTION

Image manipulation tools have made it increasingly easy to alter image content while preserving a visually plausible appearance. A complete forensic system should therefore answer two complementary questions: *whether* an image is real or manipulated and *where* the manipulation occurs. The former corresponds to image-level manipulation detection, while the latter is typically formulated as image forgery localization (IFL), which predicts a pixel-level manipulation mask. Although recent methods achieve strong performance on established benchmarks, existing evaluation does not fully characterize how they generalize under diverse distribution changes beyond their training data. That distribution shift reduces performance is unsurprising; the more informative question is *which change makes a trained localizer fail*.

Existing datasets vary in image content, manipulation source, semantic target, editing operation, and composition (Dong et al., 2013; Hsu et al., 2006; Wen et al., 2016; Guan et al., 2019; Novozamsky et al., 2020; Tralic et al., 2013), but these factors are commonly entangled across datasets or only partially annotated within them. As Table 1 shows, many benchmarks already contain large collections of classical or AI-based edits and substantial manipulation diversity. The limitation is therefore not simply dataset scale, but how this diversity is represented and used. Without a shared factorized annotation space, cross-dataset evaluation can reveal that performance drops but provides limited evidence about which variation is associated with the failure. At the same time, a useful benchmark should provide sufficiently diverse and consistently annotated data to serve as a reusable training resource rather than only an analysis tool. We therefore argue for a benchmark that jointly repre-

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

Table 1: Comparison of representative image forgery localization benchmarks. ✓ denotes an explicitly represented factor and ○ a factor present but not separately evaluated. GIFTBench scale reports manipulated images; other scales follow the original benchmarks.

Benchmark	Scale	Classical	AI-based	Semantic Target	Operation	Composition
CASIA v2 (Dong et al., 2013)	12.6K	✓	×	○	○	×
DEFACTO (Mahfoudi et al., 2019)	159K	✓	×	○	○	×
IMD2020 (Novozamsky et al., 2020)	40K	✓	×	○	○	×
CocoGlide (Guillaro et al., 2023)	20K	×	✓	○	○	×
AutoSplice (Jia et al., 2023)	110K	×	✓	○	○	×
TGIF (Mareen et al., 2024)	60K	×	✓	○	○	○
GIM (Chen et al., 2025)	2.28M	×	✓	○	○	×
BR-Gen (Cai et al., 2026)	150K	×	✓	✓	○	×
DiffSeg30k (Ci et al., 2025)	30K	×	✓	○	✓	✓
ManipBench (Xu et al., 2025b)	450K+	×	✓	○	○	×
LocateEdit-Bench (Wu et al., 2026)	231K	×	✓	○	✓	×
OpenSDI (Wang et al., 2025b)	300K	×	✓	○	○	×
GIFTBench	**115,013**	✓	✓	✓	✓	✓

sents multiple manipulation factors, supports axis-specific evaluation, and provides a broad training distribution for studying and improving cross-domain localization.

To support both structured evaluation and diverse model training, we introduce **GIFTBench**, a multi-axis benchmark built from ADE20K with 115,013 manipulated images and pixel-level masks. Each sample is annotated by manipulation source, semantic target, editing operation, and composition complexity, covering diverse scenes together with both classical and AI-based manipulations. Rather than constructing separate datasets for individual factors, GIFTBench organizes these variations within a unified generation and annotation framework, yielding a coherent training distribution with diversity across multiple levels. The structured annotations enable axis-specific transfer analysis, while the complete training set can be used directly to train localization models. Generalization beyond the benchmark is further evaluated on twelve external datasets.

The axis-specific protocols reveal asymmetric cross-source transfer, recall-dominated failures, and heterogeneous degradation across semantic, operational, and compositional shifts. Beyond diagnosis, training representative localizers on the complete GIFTBench set improves aggregate transfer to external datasets over their reference training distributions, showing that its diversity benefits both generalization evaluation and transferable model learning.

We develop ForenScope, a joint detection–localization framework built on a classification-adapted DINOv2 representation (Oquab et al., 2024). Its localization branch combines multi-depth patch features, learnable cross-layer fusion, bounded global-to-local modulation, and coarse-to-fine decoding; the design rationale and empirical tests are presented later.

Our contributions are threefold:

- We introduce **GIFTBench**, a large-scale, multi-axis benchmark with diverse scenes and manipulations organized within a unified construction framework, supporting both structured generalization evaluation and model training.
- We show that GIFTBench enables systematic diagnosis of localization generalization, revealing asymmetric cross-source transfer, recall-dominated failures, and heterogeneous semantic, operational, and compositional degradation. Training on the complete GIFTBench distribution also improves aggregate transfer across external datasets.
- We develop and evaluate **ForenScope**, a detection and localization model that retains image-level detection while improving cross-dataset localization.

2 Related Work

Image manipulation localization benchmarks. Early IFL benchmarks focused on splicing, copy-move, and limited post-processing artifacts (Dong et al., 2013; Hsu et al., 2006; Wen et al., 2016; Guan et al., 2019). Later datasets broadened manipulation sources and included more realistic in-the-wild images (Novozamsky et al., 2020; Tralic et al., 2013). Subsequent benchmarks further

increased scale and manipulation diversity (Mahfoudi et al., 2019; Kadam et al., 2021), while recent work has increasingly addressed generative and instruction-driven editing (Jia et al., 2023; Mareen et al., 2024; Chen et al., 2025; Cai et al., 2026).

IFL models. IFL systems use constrained convolutions, noise or frequency cues, patch reasoning, transformer encoders, and segmentation decoders (Bayar & Stamm, 2018; Wu et al., 2019; Chen et al., 2021; Hu et al., 2020; Wang et al., 2022; Ma et al., 2023). More recent approaches use foundation-model representations and stronger multimodal or generative priors (Guillaro et al., 2023; Su et al., 2025; Zhu et al., 2025; Ma et al., 2024).

Foundation models for dense prediction. Large self-supervised vision encoders provide semantically rich features for dense prediction. General segmentation systems have shown that transformer representations can support accurate masks when combined with task-specific decoders (Kirillov et al., 2023; Ravi et al., 2025). In image forensics, however, semantically invariant representations may suppress weak manipulation-specific evidence.

3 GIFTBench: Multi-Axis Benchmark for Generalization Diagnosis

3.1 Benchmark Design

GIFTBench studies image forgery localization under benchmark-defined distribution shifts by annotating manipulation variation along four explicit dimensions and evaluating generalization separately with respect to each. Its design follows three principles: major sources of shift should be explicitly labeled rather than hidden within a random split; manipulation masks should have deterministic provenance; and the benchmark should support both axis-specific analysis and transfer to established external datasets.

Accordingly, each manipulated image is characterized along four complementary axes:

- **Manipulation source:** splicing, copy–move, removal, and AI editing;
- **Semantic target:** object-, component-, and background-level manipulation;
- **Editing operation:** adding, removing, and replacing content;
- **Composition complexity:** single manipulation, fusion of multiple edits, and multi-source composition.

Manipulation source describes the editing pipeline or generation family used to produce a sample, whereas editing operation describes the intended transformation applied to the image. The two annotations are distinct: *Removal* denotes a source-family pipeline, while *Remove* denotes an editing objective that may also be realized by other source families. Each sample retains both labels and can therefore be evaluated from either diagnostic perspective. The first three axes characterize the origin, spatial semantics, and intent of an edit, while the composition axis captures the interaction of multiple manipulation processes or regions within the same image. Together, these factors define the benchmark's multi-axis evaluation space.

Construction summary. GIFTBench uses ADE20K images and semantic annotations (Zhou et al., 2019) to generate manipulations through vision–language-guided planning and multiple editing pipelines, while retaining explicit masks and provenance verified by automatic and manual quality checks. Before generation, all primary and donor sources are assigned to disjoint train or test pools, which are inherited by every generated sample and its source material. The benchmark contains 115,013 manipulated images, including 103,319 training and 11,694 test images; Fig. 2 summarizes the construction process, with full quality-control and release details provided in Appendix A.

3.2 Evaluation Protocols

GIFTBench defines four axis-specific diagnostic protocols. Each organizes training and evaluation along one benchmark axis under common evaluation rules, without requiring the remaining annota-

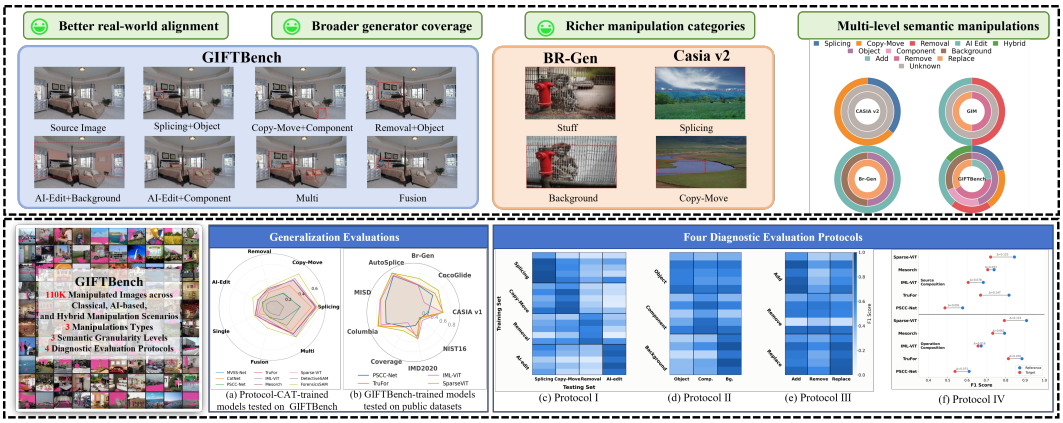


Figure 1: Overview of GIFTBench, summarizing benchmark positioning, data construction, and diagnostic analysis across four generalization axes.

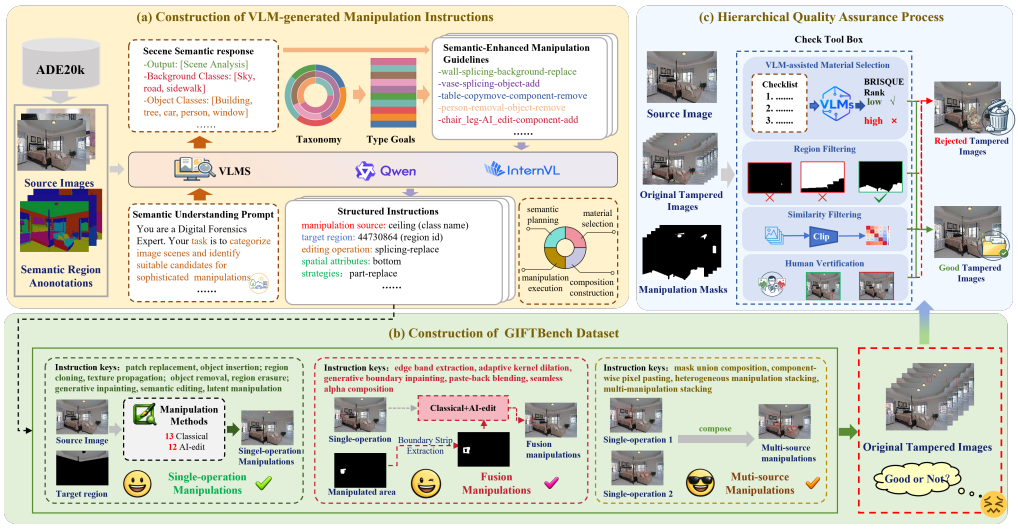


Figure 2: Construction pipeline of GIFTBench.

tions to be statistically matched. Results should therefore be interpreted as axis-associated transfer patterns rather than strict one-variable-at-a-time interventions.

- **Protocol I: Manipulation-source transfer.** Models are trained on one source and evaluated on four manipulation sources, yielding a 4×4 transfer matrix.
- **Protocol II: Semantic Target transfer.** Models are trained and evaluated across three semantic target levels, yielding a 3×3 transfer matrix.
- **Protocol III: Editing-operation transfer.** Models are trained and tested across three operations, yielding a 3×3 transfer matrix.
- **Protocol IV: Composition generalization.** Models are trained on single-operation examples and evaluated on single, fusion, and multi-source manipulations to assess generalization to composed settings.

In addition to the axis-specific protocols, we evaluate two forms of external transfer: released checkpoints are applied directly to GIFTBench without adaptation, while GIFTBench-trained models are evaluated on external datasets without target-domain fine-tuning. The former diagnoses the OOD transfer of existing systems, whereas the latter assesses generalization beyond the GIFTBench construction domain. Pixel-level F1 and IoU are averaged over manipulated images, and cross-

dataset results are macro-averaged over datasets. Real images are excluded from localization metrics; image-level detection is evaluated separately on datasets containing both real and manipulated images.

3.3 Benchmark Diagnosis

Existing models exhibit a substantial transfer gap. We analyze axis-specific failures by retraining representative CNN-, Transformer-, and foundation-model-based localizers with identical source-image splits, thresholds, and metrics. For released-checkpoint transfer, we use Protocol-MVSS and Protocol-CAT checkpoints from IMDL-BenCo (Ma et al., 2024) and the MMTD-Set checkpoint from FakeShield (Xu et al., 2025a). Transfer is markedly stronger on Splicing and Copy–Move than on Removal and AI–Edit. The same architecture behaves differently across training protocols, while model rankings vary among source and composition groups. Thus, OOD performance depends on both the training distribution and target condition and cannot be adequately characterized by a single aggregate score. Complete results are provided in Appendix Table 9.

Across the source, semantic-target, and operation protocols, transfer varies with the train–test pairing; Appendix Fig. 6 provides a compact visualization of these cross-domain asymmetries. Figure 3 analyzes the failure mechanisms beyond aggregate scores. In panel (a), off-diagonal source transfers are dominated by false negatives while false positives remain low, and recall drops sharply relative to matched evaluation. Unfamiliar source traces are therefore chiefly missed rather than producing indiscriminate foreground predictions. Panel (b) shows that semantic-target groups also differ in their forensic statistics: background edits tend to occupy larger regions and exhibit weaker boundary gradients and lower high-frequency energy than object- and component-level edits. The semantic transfer gap thus coincides with changes in spatial extent, texture, and boundary strength rather than with category labels alone.

Panel (c) further reveals operation-dependent spatial failures. Add-trained models show relatively balanced boundary and core recall degradation, Remove-trained models lose more core recall, and Replace-trained models lose more boundary recall, indicating different reliance on interior and boundary evidence. Controlled single-to-composition evaluation also degrades across models (Appendix Table 13); panel (d) illustrates how co-occurring manipulations can redirect SparseViT attention from the original target toward competing visual regions, producing incomplete or expanded predictions. Together, these analyses show that generalization fails through distinct spatial mechanisms: missed unfamiliar cues, altered region statistics, boundary/interior sensitivity, and attention competition under composition.

3.4 External Generalization of GIFTBench-trained Models

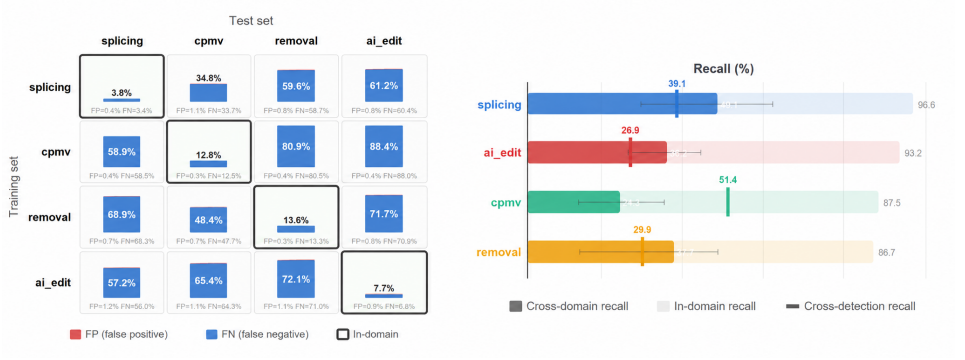
To assess generalization beyond GIFTBench, we evaluate representative localizers trained under GIFTBench and two established protocols on twelve external datasets. Table 2 reports pixel-level F1 across nine conventional IFL datasets and three related manipulation-localization datasets. GIFTBench training improves the aggregate external performance of all evaluated architectures relative to their corresponding reference training distributions. In particular, SparseViT trained on GIFTBench achieves the strongest Avg.-All F1 of 0.6029 among the evaluated benchmark-trained baselines. The gains are nevertheless target dependent, indicating that no single training distribution uniformly dominates across all external datasets.

Protocol-CAT includes IMD2020 during training; its IMD2020 result is therefore excluded from external evaluation. These results complement the axis-specific diagnosis above: GIFTBench is not only useful for diagnosing generalization failures, but also provides a diverse training distribution that transfers effectively to established external benchmarks.

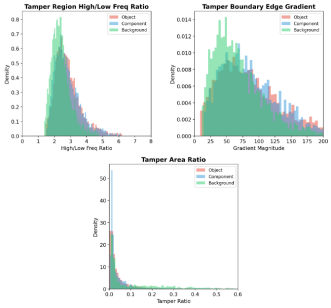
4 ForenScope: Benchmark-Informed Detection and Localization

4.1 Overview

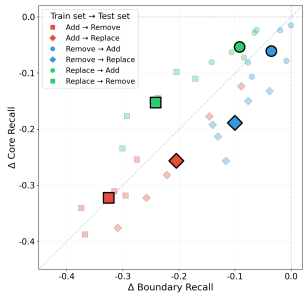
The GIFTBench diagnosis further indicates that manipulation evidence can be source sensitive, spatially heterogeneous under composition, and easily missed when local cues become ambiguous.



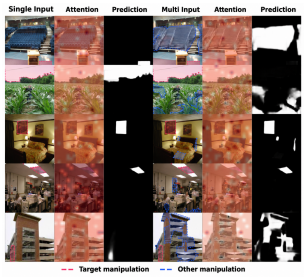
(a) Source error and recall decomposition



(b) Semantic-target region statistics



(c) Boundary/core recall degradation



(d) Attention under composition

Figure 3: Failure analyses for source, semantic-target, operation, and composition shifts in GIFT-Bench.

Table 2: Cross-dataset localization performance under different training protocols.

Training Protocol	Model	IFL Dataset									Related Dataset			Average	
		CASIA v1	NIST16	IMD2020	Coverage	Columbia	MISD	AutoSplice	CocoGlide	Br-Gen	DDL	T-IC13	OSTF	Avg. (All)	Avg. (Excl. IMD)
Protocol-MVSS	PSCC-Net	.375	.184	.259	.239	.605	.690	.659	.285	.267	.358	.254	.190	.3638	–
	TruFor	.721	.321	.323	.424	.865	.756	.320	.204	.085	.486	.364	.230	.4249	–
	IML-ViT	.718	.291	.322	.438	.747	.710	.221	.211	.079	.312	.370	.239	.3882	–
Protocol-CAT	PSCC-Net	.569	.344	–	.386	.856	.763	.564	.509	.133	.348	.173	.240	–	.4441
	TruFor	.821	.301	–	.483	.876	.772	.327	.283	.078	.596	.259	.218	–	.4558
	IML-ViT	.801	.488	–	.659	.945	.710	.252	.315	.096	.475	.415	.286	–	.4947
	SparseViT	.820	.324	–	.543	.957	.767	.385	.387	.096	.570	.291	.214	–	.4867
GIFTBench	PSCC-Net	.320	.189	.271	.339	.649	.641	.856	.567	.563	.415	.461	.285	.4630(+27%)	.4805(+8%)
	TruFor	.642	.164	.490	.344	.801	.763	.917	.603	.574	.533	.527	.368	.5605(+32%)	.5669(+24%)
	IML-ViT	.628	.251	.552	.241	.742	.749	.868	.306	.568	.468	.481	.316	.5145(+33%)	.5107(+3%)
	SparseViT	.633	.200	.543	.430	.813	.763	.923	.698	.643	.710	.521	.358	.6029	.6084(+25%)

These observations do not uniquely determine an architecture; instead, they suggest three design requirements: avoiding dependence on a single representation, modeling evidence across spatial extents, and allowing uncertain local predictions to exploit complementary image-level context. ForenScope instantiates these requirements as a joint detection–localization model with multi-depth layer–scale fusion, bounded CLS conditioning, and coarse-to-fine decoding.

As illustrated in Fig. 4, ForenScope builds on a classification-adapted DINOv2 encoder. Its CLS representation supports image-level detection, while intermediate patch representations retain spatial evidence for localization. The two tasks remain separately predicted, that allows the model to exploit task complementarity without making localization depend on a hard image-level decision.

The inherited classification head predicts the image-level manipulation probability as

$$p_{\text{det}}(I) = \sigma(\text{ClsHead}(z_{\text{cls}})) \,. \tag{1}$$

For localization, patch features from four encoder depths are projected to three spatial scales and fused by independent scale-wise LayerMixers. The class representation modulates the coarser features through bounded residual Feature-wise Linear Modulation (FiLM), after which a coarse-to-fine decoder produces the direct manipulation mask. This learned network output is the default ForenScope prediction; an optional prototype-guided refinement is studied separately at inference time.

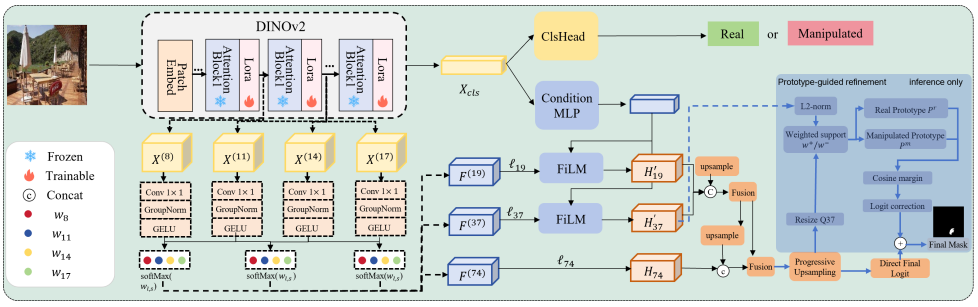


Figure 4: Overview of ForenScope. The model jointly performs image-level detection and pixel-level localization through a frozen classification-adapted encoder and a multi-scale localization decoder.

4.2 Multi-Depth Layer–Scale Fusion

Manipulation cues need not be represented most strongly at the same encoder depth or spatial resolution. We therefore extract spatial patch features $X^{(l)}$ from four depths, $\mathcal{L} = \{8, 11, 14, 17\}$, and project every depth to three localization scales, $\mathcal{S} = \{74, 37, 19\}$. For each depth–scale pair, an independent lateral node maps the feature from 1024 to 128 channels and to the target resolution using a 1×1 convolution, GroupNorm, GELU, and scale-specific resizing, producing $U_{l,s}$.

At each scale, an independent LayerMixer combines the four projected features. Let $a_{l,s}$ denote the learnable logit of depth l at scale s. The normalized coefficient and fused feature are

$$\bar{\alpha}_{l,s} = \frac{\exp(a_{l,s})}{\sum_{l' \in \mathcal{L}} \exp(a_{l',s})}, \qquad F_s = \sum_{l \in \mathcal{L}} \bar{\alpha}_{l,s} U_{l,s}. \tag{2}$$

The coefficients are learned during localization training and normalized separately at each scale, allowing the decoder to use a different depth mixture for coarse and fine spatial evidence.

4.3 Global-Conditioned Coarse-to-Fine Localization

Image-level detection and pixel-level localization provide complementary views of manipulation evidence: patch features indicate where an edit may occur, whereas the CLS representation summarizes whether the image contains manipulation cues globally. When local evidence is weak, this global representation can provide a useful prior; however, using it as a hard mask gate would propagate every detection error into localization. We therefore inject CLS context through a bounded residual modulation that can adjust, but not replace, the spatial representation.

The image-level class token provides global context to the localization branch. A two-layer MLP maps z_{cls} to a conditioning vector c. For the two coarser scales $s \in \{19, 37\}$, bounded residual FiLM is applied as

$$\widetilde{F}_s = F_s + \rho_s \left(g_s \odot F_s + r_s \right), \qquad \begin{cases} g_s = \tanh(W_{\gamma,s} c + b_{\gamma,s}), \\ r_s = \tanh(W_{\beta,s} c + b_{\beta,s}), \end{cases} \tag{3}$$

where $\rho_{19} = 1$ and $\rho_{37} = 0.25$. The FiLM projections are zero-initialized. We restrict conditioning to H19 and H37 because these coarser features provide broader context, while leaving H74 unconditioned preserves a high-resolution pathway for fine-grained spatial evidence.

Manipulated regions may span broad structures while retaining fine boundaries, making direct prediction from one resolution inadequate for both. The decoder therefore integrates the three scales in a coarse-to-fine manner. $\widetilde{F}_{19}$ is upsampled and fused with $\widetilde{F}_{37}$, and the resulting feature is further upsampled and fused with F_{74}. Each fusion stage reduces the concatenated representation to 128 channels. Progressive upsampling then recovers resolutions 148, 296, 592, and 1036, followed by a one-channel head that predicts the manipulation logit map.

4.4 Training and Inference

For each training protocol, the DINOv2 encoder is first adapted for image-level manipulation classification on the corresponding training data using LoRA. The adapted encoder and inherited classification head are then frozen, and the localization-specific modules are trained on the same data with pixel-level supervision. This preserves the image-level representation while allowing the localization branch to learn how to reuse it.

Auxiliary heads supervise H19, H37, and H74 so that every spatial pathway remains directly predictive instead of relying only on the final fusion output. Let ℓ_s denote the weighted sum of binary cross-entropy and soft Dice loss at output scale (s), and let λ_s denote the corresponding auxiliary-loss coefficient. The localization objective is

$$\mathcal{L} = \ell_{\text{final}} + \lambda_{19}\ell_{19} + \lambda_{37}\ell_{37} + \lambda_{74}\ell_{74}. \tag{4}$$

The auxiliary and per-image class weights are fixed training hyperparameters specified in Appendix D. No classification loss is optimized during localization training; the frozen classification branch remains available for image-level detection, while its class representation provides global context to the localization decoder.

Optional prototype-guided refinement. The direct mask can still contain locally uncertain predictions even when its confident regions are reliable. We therefore study an inference-only prototype refinement applied to the direct ForenScope prediction. Using the H37 features and resized localization scores, manipulated and real prototypes are estimated from confident regions, and their cosine-similarity margin is used to adjust the final logits. The refinement is applied with the same fixed rule across datasets, requires no target-domain labels or parameter updates, and is reported as an inference-only extension of the direct prediction. Full formulation and hyperparameters are provided in Appendix D.7.

5 Experiments

5.1 Experimental Setup

We evaluate ForenScope for image-level manipulation detection and pixel-level localization under two independently trained settings. The GIFTBench setting performs both training stages on the GIFTBench training split, whereas Protocol-MVSS repeats them using only CASIAv2; the settings share neither training data nor adapted checkpoints. All OOD datasets are reserved for evaluation, and checkpoints are selected using validation data from the corresponding training domain. Unless otherwise stated, ablations follow the GIFTBench setting.

At inference, detection and localization are predicted separately; the detection decision does not gate the mask, although CLS features condition the localization branch through FiLM. Unless otherwise specified, ForenScope denotes the direct output of the selected EMA checkpoint at a fixed threshold of 0.5. Localization F1 and IoU are computed on manipulated images, while image-level ROC AUC is reported on datasets containing both real and manipulated samples. ForenScope + Proto. denotes the fixed inference-only refinement. Full implementation details are provided in Appendix D.

5.2 Overall Performance

Table 3 reports pixel-level F1 on three held-out GIFTBench composition subsets and nine OOD datasets; complete in-domain and twelve-dataset F1/IoU results are provided in Appendix E. On the source-disjoint GIFTBench split, ForenScope remains strong across source, semantic-target, operation, and composition views. Across the nine OOD datasets, the direct ForenScope output achieves the highest average F1 among the compared direct-output models and outperforms Sparse-ViT on most datasets, despite dataset-specific variation.

The lower panel reports cross-dataset results under Protocol-MVSS training. ForenScope achieves the highest average F1 among the compared methods, although its relative advantage varies across individual OOD datasets. The optional prototype refinement further improves the direct prediction and produces the strongest overall performance in this setting.

Table 3: Pixel-level F1 under GIFTBench and Protocol-MVSS training.

GIFTBench training													
	In-domain GIFTBench			Out-of-domain datasets									
Method	Single	Fusion	Multi	CASIAv1	NIST16	IMD2020	COVERAGE	Columbia	MISD	AutoSplice	CocoGlide	Br-Gen	Avg
PSCC-Net	.610	.538	.658	.320	.189	.271	.339	.649	.641	.856	.567	.563	.4883
TruFor	.837	.816	.819	.642	.164	.490	.344	.801	**.763**	.917	.603	.574	.5887
IML-ViT	.730	.656	.793	.628	.251	.552	.241	.742	.749	.868	.306	.568	.5450
SparseViT	.862	.796	.850	.633	.200	.543	.430	**.813**	**.763**	**.923**	.698	.643	.6273
FORENSCOPE	**.891**	**.871**	.867	.679	.548	.579	.578	.546	.696	.871	.799	.746	.6712
FORENSCOPE + Proto.	.885	**.896**	**.867**	**.695**	**.566**	**.580**	**.592**	.564	.726	.873	**.820**	**.753**	**.6855**

Protocol-MVSS training: external-only comparison										
Method	CASIAv1	NIST16	IMD2020	COVERAGE	Columbia	MISD	AutoSplice	CocoGlide	Br-Gen	Avg
PSCC-Net	.375	.184	.259	.239	.605	.690	**.659**	.285	**.267**	.3959
TruFor	**.721**	.321	.323	.424	**.865**	**.756**	.320	.204	.085	.4466
IML-ViT	.718	.291	.322	.438	.747	.710	.221	.211	.079	.4152
FORENSCOPE	.681	.524	.563	.518	.454	.632	.276	.467	.191	.4782
FORENSCOPE + Proto.	.691	**.539**	**.571**	**.543**	.487	.658	.290	**.499**	.210	**.4986**

Table 4: Ablation of ForenScope components on the nine-dataset external average.

Variant	ADE Adapt.	Multi-depth	Multi-scale	CLS-FiLM	F1	IoU
Retrained structural variants (direct output)						
w/o ADE adaptation	×	✓	✓	✓	.632	.535
F17 only	✓	×	✓	✓	.619	.523
H37 only	✓	✓	×	✓	.641	.544
H74 only	✓	✓	×	N/A	.644	.540
w/o CLS-FiLM	✓	✓	✓	×	.659	.567
FORENSCOPE	✓	✓	✓	✓	**.671**	**.577**
Fixed-checkpoint inference refinement						
FORENSCOPE + Proto.	✓	✓	✓	✓	.685	.592

Image-level detection. Although our architectural contributions focus on localization, ForenScope retains the image-level classifier inherited from the classification-adapted encoder. On the eight external datasets containing both real and manipulated images, the GIFTBench-adapted detector achieves an average ROC AUC of 0.935 and F1 of 0.871. The full per-dataset comparison, including the controlled CASIAv2-trained setting, is provided in Appendix E.

5.3 ABLATION STUDY

Table 4 evaluates the principal components of ForenScope using the nine-dataset external average. The upper block contains retrained structural variants; the lower block reports the optional fixed inference-time refinement, while the equal-layer control is provided in Appendix 20.

Using only F17 causes the largest structural drop, supporting the aggregation of complementary evidence across encoder depths. The H37-only and H74-only variants perform comparably to each other but both underperform the full model, supporting the combination of spatial evidence across resolutions rather than reliance on a particular single scale. The w/o-ADE result supports classification-adapted representations, while removing CLS-FiLM has a smaller effect, suggesting a modest rather than uniform benefit from explicit conditioning. Prototype refinement improves the average without retraining; the equal-layer control and additional FiLM-off, shuffled-CLS, and joint-control diagnostics are reported in Appendix E.4.

6 CONCLUSION

We presented **GIFTBench**, a multi-axis benchmark for diagnosing image forgery localization generalization, together with FORENSCOPE, a detection–localization model using learned depth–scale fusion and bounded global conditioning. **GIFTBench** exposes distinct transfer behavior across manipulation source, semantic target, operation, and composition while complementing controlled diagnosis with twelve-dataset external evaluation. FORENSCOPE retains image-level detection and improves cross-dataset localization, demonstrating how structured benchmark diagnosis can provide testable guidance for more transferable forensic models.

AI USE STATEMENT

Generative AI tools were used to assist with polishing the writing of the manuscript, retrieving and identifying relevant literature, and drafting portions of the paper. All AI-assisted content was reviewed, revised, and verified by the authors, including factual claims and references. The authors take full responsibility for the final content of this work.

REFERENCES

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space, 2025.

Belhassen Bayar and Matthew C Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018.

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

Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 14185–14193, 2021.

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

Hai Ci, Ziheng Peng, Pei Yang, Yingxin Xuan, and Mike Zheng Shou. Diffseg30k: A multi-turn diffusion editing benchmark for localized aigc detection. *arXiv preprint arXiv:2511.19111*, 2025.

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

Haiying Guan, Mark Kozak, Eric Robertson, Yooyoung Lee, Amy N Yates, Andrew Delgado, Daniel Zhou, Timothee Kheyrkhah, Jeff Smith, and Jonathan Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *IEEE Winter Applications of Computer Vision Workshops*, pp. 63–72. IEEE, 2019.

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

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

Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. Span: Spatial pyramid attention network for image manipulation localization. In *European conference on computer vision*, pp. 312–328, 2020.

Jitesh Jain, Yuqian Zhou, Ning Yu, and Humphrey Shi. Keys to better image inpainting: Structure and texture go hand in hand. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 208–217, 2023.

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

Kalyani Dhananjay Kadam et al. Multiple image splicing dataset (misd): a dataset for multiple splicing. *Data*, 6(10):102, 2021.

Alexander Kirillov, Nikhila Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Ruibin Li, Tao Yang, Song Guo, and Lei Zhang. Rorem: Training a robust object remover with human-in-the-loop. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14024–14035, 2025.

Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10758–10768, 2022.

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

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

Gaël Mahfoudi, Badr Tajini, Florent Retraint, Frederic Morain-Nicolier, Jean Luc Dugelay, and Marc Pic. Defacto: Image and face manipulation dataset. In *27Th european signal processing conference*, pp. 1–5, 2019.

Hayk Manukyan, Andranik Sargsyan, Barsegh Atanyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Hd-painter: high-resolution and prompt-faithful text-guided image inpainting with diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 96301–96330, 2025.

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

Adam Novozamsky et al. 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*, pp. 71–80, 2020.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaa El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

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

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. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Andranik Sargsyan, Shant Navasardyan, Xingqian Xu, and Humphrey Shi. Mi-gan: A simple baseline for image inpainting on mobile devices. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7335–7345, 2023.

Lei Su, Xiaochen Ma, Xuekang Zhu, Chaoqun Niu, Zeyu Lei, and Ji-Zhe Zhou. Can we get rid of handcrafted feature extractors? sparsevit: Nonsemantics-centered, parameter-efficient image manipulation localization through spare-coding transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 7024–7032, 2025.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2149–2159, 2022.

Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004.

Dijana Tralic, Ivan Zupancic, Sonja Grgic, and Mislav Grgic. Comofod—new database for copy-move forgery detection. In *Proceedings ELMAR-2013*, pp. 49–54, 2013.

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

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025a.

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*, pp. 4291–4301, 2025b.

Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. Coverage—a novel database for copy-move forgery detection. In *IEEE international conference on image processing*, pp. 161–165, 2016.

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

Yue Wu et al. 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*, pp. 9543–9552, 2019.

Zhipei Xu, Xuanyu Zhang, Runyi Li, Zecheng Tang, Qing Huang, and Jian Zhang. Fakeshield: Explainable image forgery detection and localization via multi-modal large language models. In *International Conference on Learning Representations*, volume 2025, pp. 31186–31216, 2025a.

Zitong Xu, Huiyu Duan, Xiaoyu Wang, Zhaolin Cai, Kaiwei Zhang, Qiang Hu, Jing Liu, Xiongkuo Min, and Guangtao Zhai. Manipshield: A unified framework for image manipulation detection, localization and explanation, 2025b.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International journal of computer vision*, 127(3):302–321, 2019.

Xuekang Zhu, Xiaochen Ma, Lei Su, Zhuohang Jiang, Bo Du, Xiwen Wang, Zeyu Lei, Wentao Feng, Chi-Man Pun, and Ji-Zhe Zhou. Mesoscopic insights: orchestrating multi-scale & hybrid architecture for image manipulation localization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 11022–11030, 2025.

Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *European Conference on Computer Vision*, pp. 195–211, 2024.

A BENCHMARK CONSTRUCTION AND DATA SPECIFICATION

This appendix section specifies what GIFTBench contains, how each manipulated sample is constructed, and how split membership, masks, and provenance are recorded. It supplements the benchmark overview in the main paper with the details needed to reconstruct protocol memberships without repeating the main motivation.

A.1 SCOPE, TAXONOMY, AND TASK DEFINITION

GIFTBench is derived from ADE20K training images and their semantic annotations (Zhou et al., 2019). Its purpose is to keep the real-image universe and localization target controlled while exposing several sources of distribution shift that are usually entangled in existing IFL datasets. Each manipulated image has a split-assigned primary source image and, where applicable, split-assigned donor or component sources; it is also assigned metadata for manipulation source, semantic target, editing operation, composition level, split, construction configuration, and quality-control status. Multi-Source samples additionally retain component-level masks and provenance.

The four axes are defined as follows:

Table 5: The four diagnostic axes recorded by **GIFTBench**. A sample may participate in multiple protocol-specific groupings.

Axis	Labels and meaning
Manipulation source	Splicing, Copy–Move, Removal, and AI-Edit. This identifies the family of editing pipeline that produced the sample.
Semantic Target	Object, Component, and Background. This identifies the semantic target category or spatial scale of the manipulated content.
Editing operation	Add, Remove, and Replace. This identifies the intended editing objective independently of the source family.
Composition complexity	Single, Fusion, and Multi-Source. This identifies whether the edit is isolated, boundary-refined, or composed with other manipulation sources.

These labels are intentionally not interchangeable. *Removal* in the source axis denotes a family of removal pipelines, whereas *Remove* in the operation axis denotes an editing objective that can also be instantiated by other source families. Similarly, Fusion and Multi-Source are composition levels derived from the same source-image universe, not independent image collections. This representation allows one sample to be grouped by different axes in different diagnostic protocols while keeping its image, mask, and provenance fixed.

The released image-level label indicates whether an image is real or manipulated, while the pixel-level target is the binary mask of pixels changed by the recorded construction operation. Source-family, semantic-target, operation, and composition labels are metadata used to form controlled evaluation groups; they are not additional prediction targets in the localization experiments. The primary source identifier links an image and its mask, while donor and component provenance is retained when applicable; an image can therefore be assigned to different protocol views without changing its supervision.

A.2 CONSTRUCTION PIPELINE

The construction process begins with a real ADE20K image and its semantic annotations. Human experts define the taxonomy, admissible combinations, and construction constraints. A vision–language model then analyzes the scene and produces a structured manipulation plan, including the target region, source material from the assigned split, intended operation, and spatial attributes. A family-specific editing engine performs the pixel-level change, after which automatic checks, VLM-based verification, and manual review validate the image, mask, and metadata. Figure 2 summarizes this construction pipeline.

The semantic taxonomy and admissible group assignments are fixed before generation. Candidate selection and editing use the recorded random state for the corresponding construction job, whereas mask derivation, normalization, quality checks, and split assignment are deterministic once the construction record is fixed. Geometric transforms are applied jointly to image and mask; photometric transforms are applied only to the image. Masks are resized with nearest-neighbor interpolation

and re-binarized after every geometry-changing operation. These rules prevent augmentation from changing the localization target while preserving the intended appearance variation.

Semantic planning and material selection. VLMs are used as semantic controllers rather than as the sole pixel generators. Qwen-VL and InternVL-family models (Bai et al., 2025b;a; Wang et al., 2025a) identify eligible regions and contextual relations, propose or verify target regions and source materials, and assess placement, scale, occlusion, illumination, and scene compatibility where applicable. Candidate source contents are filtered for geometric completeness and contextual compatibility within the assigned split before execution. This separation lets the benchmark vary the editing engine without making the entire release an artifact of one VLM.

Pixel-level execution and mask generation. Classical compositing, traditional or learned inpainting, and diffusion-based editors perform the actual image modification. The executor is selected according to the source family and the semantic characteristics of the target. A binary mask is generated from the recorded construction operation and identifies pixels changed relative to the original source. For composed samples, masks for the individual components are retained in addition to the final union or refined mask.

A.3 Manipulation source instantiation

Splicing. Splicing includes object insertion or replacement and background replacement. Background edits include partial-region, whole-region, and cross-class substitutions. For object edits, the structured plan specifies the candidate object, placement, support surface, scale, lighting relation, and occlusion constraints; compatible donor material is selected from the ADE20K pool assigned to the same train/test partition as the primary source before classical compositing executes the edit. These cases vary both local blending traces and scene-level consistency requirements.

Copy–Move. Copy–Move includes background-to-background texture copying, background-to-object overwriting, whole-object relocation, superpixel-level cloning, texture extension, and the attribute subset. Candidate regions are selected according to semantic role and geometric completeness, followed by controlled transformations, color alignment, and blending. The attribute subset is categorized as Copy–Move in the release taxonomy and is included in the corresponding source-level counts.

Removal. Removal covers background-texture repair, structural-object removal, and semantic-foreground removal. The construction plan distinguishes locally repairable textures from extended structures and complete foreground instances. Traditional inpainting (Telea, 2004) or learned inpainting (Suvorov et al., 2022; Li et al., 2022; Sargsyan et al., 2023; Jain et al., 2023; Li et al., 2025) is selected according to the target's spatial and semantic characteristics, so the source family is not restricted to one inpainting mechanism.

AI-Edit. AI-Edit samples use mask-conditioned diffusion editing driven by visually grounded, structured instructions. The plans cover subject-level, background/stuff-level, component-level, and spatial-interaction edits; each can implement Add, Remove, or Replace. Multiple diffusion and instruction-guided inpainting backends (Rombach et al., 2022; Podell et al., 2024; Manukyan et al., 2025; Zhuang et al., 2024; Batifol et al., 2025) are used to provide variation in generative mechanisms, while semantic target and operation remain separately recorded.

A.4 Composition construction

Composition is not treated as a fifth manipulation source. It tests whether evidence learned from isolated edits survives two controlled changes in the same source-image setting. **Fusion** begins with a single-operation manipulation and applies latent-diffusion inpainting (Rombach et al., 2022; Podell et al., 2024) only to a narrow morphological band around the existing manipulation boundary; pixels outside the band remain unchanged. The final mask includes the refined boundary band and the retained preceding manipulation, providing a matched single-operation reference.

Multi-Source composition combines two to four distinct single-manipulation components derived from the same real image and its assigned source pool. The visible components are placed back

onto the original background with controlled spatial layouts, and their masks are merged into a final union mask. Component-level masks and provenance support analyses beyond the union-mask localization target. The composition training partitions are released for future composition-aware methods, whereas the Protocol IV baselines in this paper train only on GIFTBench-Core and therefore measure zero-composition generalization.

A.5 Quality assurance, masks, and provenance

Before large-scale generation, each manipulation family and its mask-extraction procedure is checked on pilot examples for visual plausibility and mask correctness. During generation, samples whose manipulated region is below 0.5% of the image area are discarded. Candidate samples are further filtered by geometry, manipulated-area, semantic consistency, visual quality, saliency change, and near-duplicate content. AI-Edit candidates additionally undergo BRISQUE no-reference quality assessment, used for relative ranking within each batch rather than with a fixed threshold, and CLIP-based semantic similarity between the edited region and the target instruction.

All remaining samples receive VLM-based quality assessment. Classical manipulations use three-level scoring (1: superior, 2: acceptable, 3: discard), while AI-Edit samples receive type-specific binary verification of physical plausibility and image–instruction consistency. Samples with low saliency relative to the source, low visual quality, or duplicated scene content are removed. Finally, retained samples are manually inspected for incorrect edits, implausible results, and inaccurate masks. These checks prioritize reliable localization targets rather than only maximizing the number of generated images.

The released mask is a single-channel binary image with values 0 and 255. Construction-time grayscale masks are normalized at release time using threshold 128: values ≥ 128 are encoded as 255 and all other values as 0. Positive pixels denote changes relative to the original source under the recorded construction procedure. The construction pipeline initially produced 115,025 candidates; twelve Copy–Move training candidates with no positive pixels after normalization were excluded as invalid localization targets, yielding the final release of 115,013 samples. The excluded records are documented in the release audit and are not treated as a separate benchmark category.

Each release record stores the manipulated-image identifier, original source identifier, mask, four-axis labels, split, construction engine and configuration identifier, and quality-control decision. For Multi-Source samples, component masks and component-level provenance are also retained. This record structure allows protocol memberships and dataset statistics to be regenerated from a common manifest.

A.6 Dataset composition and split integrity

Before manipulation generation, all ADE20K primary and donor source images are assigned to one of two disjoint pools, train or test. Every source image and all donor/component material used to construct a sample come from the same pool; the generated manipulation and all of its descendants inherit that assignment. Thus, no primary source, donor source, source material, or manipulated descendant is shared across the train/test boundary. The complete release composition is:

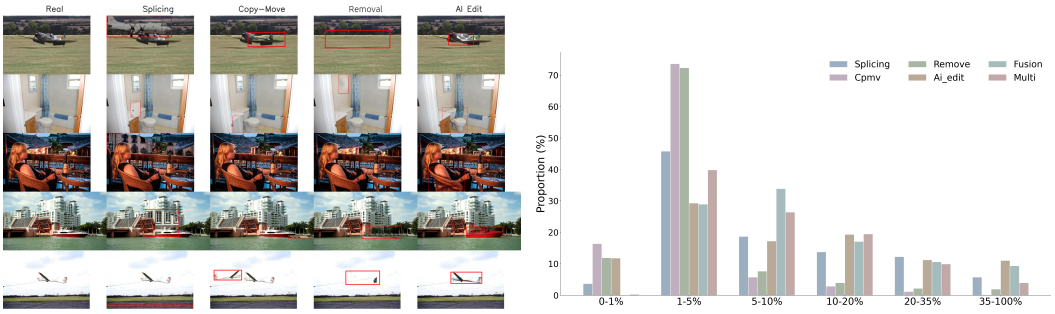
Table 6: Release composition of **GIFTBench**. GIFTBench-Core is the single-operation benchmark used for baseline training; composition training partitions are released for future composition-aware methods. The attribute subset is included under Copy–Move.

Release partition	Train	Test
GIFTBench-Core (single-operation)		
Splicing	20,673	2,327
Copy–Move (including attribute)	20,887	2,386
Removal	21,074	2,380
AI-Edit	25,758	2,851
Core subtotal	88,392	9,944
GIFTBench-Composition (challenge set)		
Fusion	5,925	750
Multi-Source	9,002	1,000
Total	103,319	11,694

The final release contains 115,013 manipulated images: 103,319 training and 11,694 test samples. The split definitions and protocol memberships in the machine-readable manifest are the source of truth for these counts. The release distributes the manipulated images and masks but not the original ADE20K source images; users obtain the latter independently under their original terms. This separation preserves the applicable source-data conditions while allowing users with access to ADE20K to reconstruct paired provenance for auditing.

A.7 Dataset-level visual summaries

The release statistics are complemented by two visual summaries. Figure 5(a) gives representative instances of the four manipulation-source families, while Fig. 5(b) shows manipulated-area distributions across the source and composition groups. These views describe the released benchmark rather than additional evaluation results.



(a) Representative construction examples (b) Manipulated-area distributions

Figure 5: Dataset-level summaries of GIFTBench: representative manipulation-source examples and manipulated-area distributions across source and composition groups.

B Benchmark Protocols and Diagnosis-to-Method Bridge

This section fixes the interpretation of every benchmark result and makes explicit how the diagnosis motivates the ForenScope design. It distinguishes public-checkpoint transfer, controlled retraining, and GIFTBench-to-external transfer before giving the metric conventions and the supplemental protocol profiles.

B.1 Protocol definitions

Table 7 summarizes the four controlled protocols and the two transfer settings that appear throughout the paper. A training source in this table denotes the data or checkpoint used to obtain the model; it does not imply that the test partition is used for adaptation.

B.2 Metric and aggregation definitions

For each manipulated image, the predicted probability map is thresholded at 0.5 and compared with the binary target mask. Pixel-level F1 and IoU are computed independently for each manipulated image and then averaged within each dataset or benchmark group. Real images are not assigned a manipulated-mask F1/IoU when no positive mask exists; their predicted-positive area is retained only for image-level false-positive inspection. Cross-dataset summaries macro-average the dataset-level scores, so All-12 is the equal-weight mean over the twelve external datasets and the Avg columns in the main tables are equal-weight means over the displayed datasets.

Image-level detection uses the frozen CLS head score and ROC AUC. AUC is defined only when both real and manipulated images are present in the evaluated split; the four pure-manipulated external splits therefore receive "–" rather than a fabricated AUC. The image-level score and localization mask are retained as separate outputs at inference; no post-hoc detector gate is applied to the mask. Unless a table states otherwise, localization uses the EMA checkpoint, the fixed threshold above,

Setting	Training source	Evaluation partition	Diagnostic question
Axis-specific GIFTBench protocols			
Protocol I	One source group	All four source groups	Source transfer: does evidence transfer across manipulation families?
Protocol II	One semantic target group	All three semantic target groups	Semantic-target transfer: does behavior survive changes in target category?
Protocol III	One operation group	All three operation groups	Operation transfer: does evidence transfer across editing intent?
Protocol IV	Single-operation training	Single, Fusion, and Multi-Source groups	Composition: does isolated-edit evidence survive composition?
Cross-domain transfer settings			
Public checkpoint transfer	Released checkpoint, no adaptation	GIFTBench evaluation groups	What is the training-distribution gap of existing models?
External transfer	GIFTBench-trained model	Twelve external datasets	Does benchmark training transfer beyond its construction domain?

Table 7: Benchmark evaluation settings. The first block organizes training and testing according to one focal GIFTBench axis at a time; the second block distinguishes public-checkpoint transfer to GIFTBench from GIFTBench-trained transfer to external datasets.

Benchmark observation	Evidence in benchmark analysis	Design requirement / hypothesis	Architectural realization
Source-transfer asymmetry	Protocol I matrices show strong dependence on the manipulation source	Avoid a single source-sensitive representation; test whether complementary evidence reduces this specialization	Multi-depth patch features
Transfer profiles vary across inputs and groups	Aggregate profiles are not uniform across examples or conditions	Combine complementary depth evidence with scale-specific learnable aggregation	Learnable scale-wise LayerMixer
Composition creates heterogeneous spatial evidence	Fusion/Multi-Source results and attention analysis show changes in extent, boundaries, and evidence distribution	Represent coarse extent and fine boundaries at multiple resolutions	H19/H37/H74 coarse-to-fine pathway
Recall-dominated errors under source shift	False-negative decomposition shows that unfamiliar local evidence is often missed	Test whether image-level manipulation context can complement ambiguous local evidence	Bounded CLS-conditioned FiLM on H19/H37
Resolution-specific architectural choice	The benchmark does not identify a unique conditioning location	Preserve a high-resolution local pathway while conditioning broader-context features	No FiLM on H74

Table 8: Diagnosis-to-design mapping used to formulate ForenScope. The table separates benchmark observations from intermediate design requirements and their architectural realization; the mapping expresses testable motivation rather than a causal conclusion.

and the direct ForenScope output. Optional ForenScope + Proto. results apply the fixed prototype refinement described in Appendix D.7.

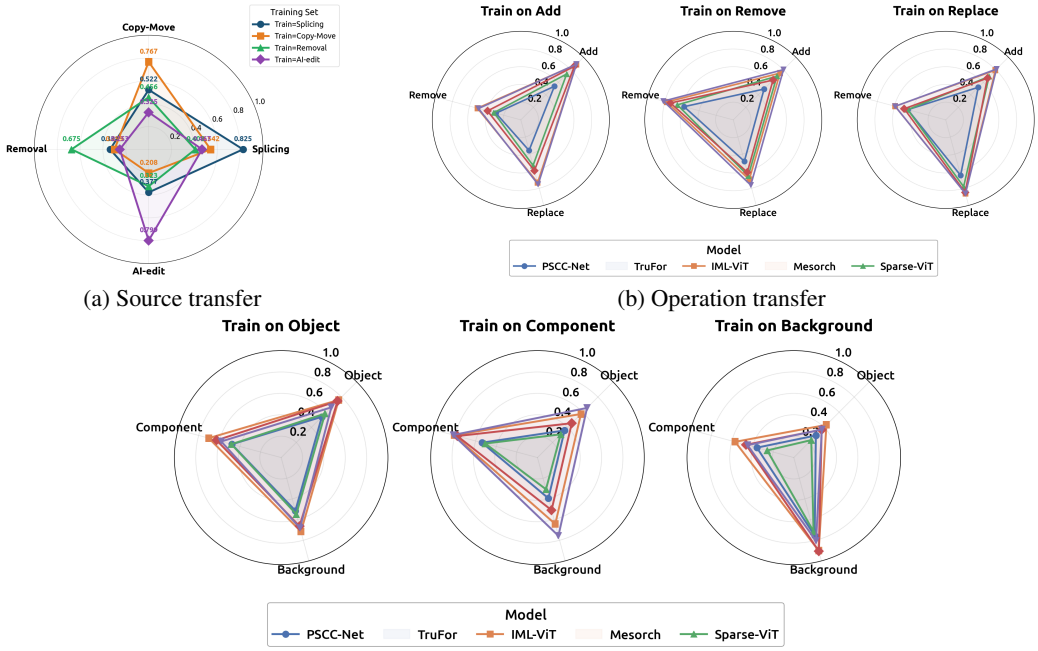
B.3 Diagnosis-to-design mapping

The benchmark analysis is descriptive: it exposes recurring transfer patterns, from which we formulate architectural hypotheses. Table 8 records the intended mapping without treating the observations as causal proof.

The table intentionally separates benchmark-derived requirements from implementation choices. In particular, the absence of FiLM at H74 is a resolution-specific architectural decision evaluated by ablation, not a direct conclusion from the benchmark. The ablations in Appendix E test whether the corresponding realizations improve transfer, while the benchmark analysis alone is not presented as causal proof.

B.4 Supplemental diagnostic visualizations

The main text presents detailed source, semantic-target, operation, and composition diagnostics. Figure 6 provides complementary aggregate radar profiles for Protocols I–III. Panels (a)–(c) summarize cross-source, cross-operation, and cross-semantic-target localization behavior; they are descriptive comparisons and do not establish causal explanations for the observed gaps.



(a) Source transfer

(b) Operation transfer

(c) Cross-semantic-target transfer profile

Figure 6: Aggregate transfer profiles for the benchmark protocols. Panels (a)–(c) summarize cross-source, cross-operation, and cross-semantic-target localization performance, respectively. These radar plots complement the fine-grained diagnostic visualizations in the main text and are intended as descriptive summaries rather than causal evidence.

C Complete Benchmark Results and Robustness

This section provides the complete controlled benchmark matrices and the available stress-test evidence. Each table states its training and test convention explicitly so that diagonal performance, cross-condition transfer, and composition references are not conflated.

C.1 Released-checkpoint transfer to GIFTBench

Table 9 reports the complete evaluation of existing released or unified checkpoints on GIFTBench without additional training or fine-tuning. The training-protocol column identifies the source protocol of each checkpoint. Because these models were trained on different data, the table measures transfer to GIFTBench rather than a matched architectural comparison. ForenScope is reported separately in Table 3, since its main configuration is trained on GIFTBench and is therefore not comparable to these zero-adaptation rows.

Table 9: Released-checkpoint transfer to GIFTBench without adaptation. Entries are pixel-level F1/IoU.

Training Protocol	Model	Manipulation Source								Composition					
		Splicing		Copy–Move		Removal		AI–Edit		Single		Fusion		Multi–Source	
		F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU
Protocol-MVSS	CAT-Net	.376	.293	.239	.171	.128	.084	.151	.106	.128	.084	.151	.106	.219	.160
	ObjectFormer	.170	.106	.077	.043	.070	.041	.168	.105	.070	.041	.168	.105	.174	.103
	PSCC-Net	.295	.209	.161	.103	.074	.045	.143	.094	.074	.045	.143	.094	.233	.149
	MVSS-Net	.307	.224	.219	.152	.105	.067	.142	.095	.105	.067	.142	.095	.286	.194
	TruFor	.591	.533	.490	.426	.235	.195	.199	.162	.235	.195	.199	.162	.450	.353
	IML-ViT	.548	.482	.448	.373	.191	.150	.180	.139	.191	.150	.180	.139	.452	.358
Protocol-CAT	MVSS-Net	.402	.318	.246	.177	.134	.090	.152	.104	.134	.090	.152	.104	.230	.170
	CAT-Net	.556	.494	.392	.330	.210	.167	.133	.107	.210	.167	.133	.107	.430	.340
	ObjectFormer	.207	.155	.124	.087	.051	.033	.104	.073	.121	.081	.112	.073	.175	.118
	PSCC-Net	.394	.303	.254	.182	.133	.091	.156	.106	.133	.091	.156	.106	.332	.234
	TruFor	.607	.557	.479	.422	.265	.228	.219	.186	.265	.228	.219	.186	.479	.390
	IML-ViT	.622	.566	.514	.441	.236	.197	.194	.155	.236	.197	.194	.155	.534	.446
	Mesorch	.620	.566	.480	.419	.267	.227	.229	.191	.267	.227	.229	.191	.483	.389
	SparseViT	.640	.583	.488	.418	.249	.205	.232	.188	.249	.205	.232	.188	.506	.410
	SAFIRE	.329	.266	.177	.134	.125	.091	.190	.143	.203	.157	.248	.198	.293	.217
	DetectiveSAM	.164	.118	.109	.069	.084	.051	.093	.063	.084	.051	.093	.063	.158	.100
	ForensicsSAM	**.671**	**.619**	**.598**	**.537**	**.281**	**.240**	**.304**	**.259**	.281	.240	.304	.259	**.544**	**.444**
MMTD-Set	FakeShield	.597	.519	.215	.170	.161	.118	.302	.246	**.330**	**.274**	**.414**	**.341**	.371	.282

C.2 PROTOCOL I COMPLETE MATRIX

The rows in Table 10 fix the training source/condition and the columns enumerate all four test sources. Each cell reports F1/IoU/AUC; the AUC is retained because these protocol splits include the corresponding image-level classification labels. The Copy–Move block includes the attribute-containing subset under the same source label.

Training Set	**Model**	**Splicing**			**Copy-Move**			**Removal**			**AI-Edit**		
		F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC
Splicing	PSCC-Net	0.5143	0.4149	0.9022	0.2333	0.1590	0.8254	0.1723	0.1184	0.7189	0.2258	0.1686	0.6948
	TruFor	0.9024	0.8524	0.9951	0.5732	0.4961	0.9440	0.4178	0.3565	0.8537	0.4870	0.4183	0.8925
	IML-ViT	0.8753	0.8443	0.9262	0.5931	0.5275	0.8454	0.3456	0.2959	0.7506	0.3500	0.2962	0.7655
	Mesorch	0.9155	0.8733	0.9935	0.5380	0.4795	0.9293	0.3130	0.2683	0.8054	0.2269	0.1896	0.7679
	SparseViT	**0.9187**	0.8703	**0.9972**	**0.6730**	**0.5996**	**0.9717**	**0.4392**	**0.3791**	**0.8663**	**0.5964**	**0.5308**	**0.9247**
Copy-Move	PSCC-Net	0.4657	0.3602	0.8710	0.5583	0.4465	0.9625	0.2604	0.1897	0.7732	0.2153	0.1513	0.7246
	TruFor	0.5869	0.5089	0.9394	0.8625	**0.7981**	0.9942	0.3385	0.2958	0.7922	0.2261	0.1791	0.8021
	IML-ViT	0.4324	0.3592	0.8239	0.7067	0.6510	0.8655	0.2306	0.1915	0.6724	0.1401	0.1052	0.6754
	Mesorch	0.5011	0.4331	0.8994	0.8406	0.7788	0.9865	0.2919	0.2530	0.7581	0.1201	0.0939	0.7105
	SparseViT	**0.7224**	**0.6495**	**0.9680**	**0.8641**	0.7965	**0.9958**	**0.3738**	**0.3247**	**0.8209**	**0.3373**	**0.2799**	**0.8509**
Removal	PSCC-Net	0.3500	0.2596	0.8066	0.3043	0.2190	0.8609	0.5780	0.4821	0.9406	0.3153	0.2356	0.8048
	TruFor	0.4533	0.3831	0.8849	0.5258	0.4506	0.9374	0.7586	0.6896	0.9813	0.4063	0.3314	0.8943
	IML-ViT	0.4971	0.4249	0.8313	0.5570	0.4807	0.8415	0.6120	0.5562	0.8567	0.2543	0.2035	0.7849
	Mesorch	0.2009	0.1579	0.7758	0.3303	0.2829	0.8528	0.6412	0.5872	0.9280	0.1054	0.0763	0.7164
	SparseViT	**0.5367**	**0.4624**	**0.9226**	**0.5644**	**0.4814**	**0.9547**	**0.7829**	**0.7118**	**0.9863**	**0.5359**	**0.4514**	**0.9316**
AI-Edit	PSCC-Net	**0.6130**	**0.5190**	**0.8901**	**0.4392**	**0.3473**	0.8609	**0.3485**	0.2700	0.8039	0.7930	0.7071	0.9790
	TruFor	0.5203	0.4579	0.8807	0.3862	0.3223	**0.8950**	0.3380	**0.2825**	**0.8354**	0.8507	0.7827	0.9914
	IML-ViT	0.3041	0.2583	0.7309	0.2025	0.1629	0.7250	0.1305	0.1011	0.6649	0.7344	0.6683	0.9147
	Mesorch	0.4075	0.3530	0.8402	0.3493	0.2956	0.8802	0.2485	0.1995	0.7988	**0.8737**	**0.8116**	**0.9928**
	SparseViT	0.4912	0.4235	0.8532	0.2483	0.1984	0.8272	0.1978	0.1535	0.7634	0.7434	0.6666	0.9710

Table 10: Complete Protocol I cross-source evaluation results (F1/IoU/AUC). Each block groups models by training source and evaluates all four test sources; all Copy–Move variants are included in the Copy–Move view.

The diagonal entries in this matrix measure within-source performance, whereas the off-diagonal entries measure transfer to a different manipulation source. The diagonal-to-off-diagonal variation is the reason the paper reports the matrix rather than only a single benchmark average.

C.3 PROTOCOL II COMPLETE MATRIX

Table 11 groups models by the semantic target used for training and evaluates object-, component-, and background-level test target groups. The source and operation distributions remain part of the benchmark construction; this protocol organizes the Semantic Target view without claiming a one-variable-at-a-time intervention.

The matrix reports transfer patterns associated with semantic-target group changes under a shared construction and evaluation procedure. It does not claim to statistically remove source-related or

Training Set	Model	Object			Component			Background		
		F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC
Object	PSCC-Net	0.4452	0.3530	0.9078	0.3887	0.2895	0.8970	0.4244	0.3330	0.8357
	TruFor	0.7161	0.6466	0.9723	0.6683	0.5794	0.9638	0.7125	0.6440	0.9484
	IML-ViT	0.6497	0.5948	0.8676	0.5595	0.4882	0.8452	0.6110	0.5481	0.8491
	Mesorch	0.6478	0.5807	0.9575	0.5266	0.4465	0.9280	0.5345	0.4677	0.9009
	SparseViT	**0.7793**	**0.7077**	**0.9861**	**0.7019**	**0.6119**	**0.9733**	**0.7551**	**0.6885**	**0.9665**
Component	PSCC-Net	0.3595	0.2697	0.8668	0.5382	0.4286	0.9407	0.3942	0.2992	0.8206
	TruFor	0.5760	0.5035	0.9364	0.8080	0.7275	0.9877	0.6435	0.5765	0.9220
	IML-ViT	0.3069	0.2591	0.7770	0.5067	0.4410	0.8481	0.3066	0.2574	0.7180
	Mesorch	0.4542	0.3909	0.8922	0.7767	0.7011	0.9774	0.5093	0.4455	0.8809
	SparseViT	**0.6563**	**0.5837**	**0.9572**	**0.8159**	**0.7371**	**0.9895**	**0.7569**	**0.6911**	**0.9685**
Background	PSCC-Net	0.2921	0.2354	0.7738	0.3583	0.2824	0.8328	0.7550	0.6783	0.9658
	TruFor	**0.4318**	**0.3762**	**0.8595**	**0.5693**	**0.4929**	**0.9179**	0.9009	0.8534	**0.9947**
	IML-ViT	0.2312	0.1931	0.6992	0.2580	0.2115	0.7247	0.7154	0.6617	0.8905
	Mesorch	0.3690	0.3213	0.8371	0.4634	0.3973	0.8892	**0.9052**	**0.8634**	0.9920
	SparseViT	0.3713	0.3161	0.8494	0.4481	0.3692	0.9100	0.8070	0.7407	0.9859

Table 11: Complete Protocol II cross-semantic-target evaluation results (F1/IoU/AUC). Each block groups models by training semantic target and evaluates all three test target groups.

Training Set	Model	Add			Remove			Replace		
		F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC
Add	PSCC-Net	0.5362	0.4318	0.9407	0.2908	0.2139	0.8105	0.3581	0.2709	0.8263
	TruFor	0.8861	**0.8316**	0.9951	**0.5083**	**0.4422**	**0.8842**	0.7327	0.6630	0.9616
	IML-ViT	0.7332	0.6820	0.8741	0.3151	0.2651	0.7519	0.5335	0.4695	0.8559
	Mesorch	0.8551	0.7998	0.9883	0.3901	0.3309	0.8408	0.5927	0.5215	0.9282
	SparseViT	**0.8878**	0.8309	**0.9956**	0.4947	0.4301	0.8826	**0.7550**	**0.6874**	**0.9683**
Remove	PSCC-Net	0.4398	0.3465	0.8991	0.5328	0.4380	0.9187	0.4482	0.3549	0.8724
	TruFor	0.6794	0.6119	0.9618	0.6980	0.6223	0.9695	0.6337	0.5585	0.9365
	IML-ViT	0.6881	0.6246	0.8662	0.6555	0.5976	0.8700	0.6196	0.5537	0.8720
	Mesorch	0.1918	0.1529	0.8414	0.3171	0.2552	0.8706	0.2096	0.1647	0.7764
	SparseViT	**0.7847**	**0.7178**	**0.9827**	**0.7871**	**0.7131**	**0.9866**	**0.7308**	**0.6582**	**0.9664**
Replace	PSCC-Net	0.5196	0.4316	0.9252	0.4442	0.3594	0.8724	0.6490	0.5644	0.9466
	TruFor	0.7930	0.7234	0.9852	**0.5923**	0.5190	0.9273	0.8609	**0.8010**	**0.9915**
	IML-ViT	0.6933	0.6303	0.8700	0.4453	0.3867	0.8096	0.7811	0.7282	0.9207
	Mesorch	0.6656	0.6052	0.9568	0.4854	0.4227	0.8854	0.8484	0.7913	0.9861
	SparseViT	**0.8115**	**0.7485**	**0.9879**	0.5909	**0.5216**	**0.9313**	**0.8618**	0.8022	0.9909

Table 12: Complete Protocol III cross-operation evaluation results (F1/IoU/AUC). Each block groups models by training operation and evaluates all three test operations.

operation-related variation, and should not be read as fully separating semantic-target shift from source shift.

C.4 Protocol III complete matrix

Table 12 groups models by the editing operation used for training and evaluates Add, Remove, and Replace. It complements Protocols I and II by examining operation-associated transfer under the same source-image universe and pixel-level task definition, without assuming that the remaining empirical distributions are statistically matched.

C.5 Controlled composition/reference results

Protocol IV compares single-operation samples with the corresponding Fusion and Multi-Source settings. Table 13 reports the existing reference results for models trained only on GIFTBench-Core; each composition score is paired with its single-operation reference rather than with an independently sampled test distribution.

The comparison shows the effect of composition relative to a matched single-operation reference. It is a reference evaluation for the benchmark protocols, not a ForenScope ablation and not an external-dataset transfer result.

Table 13: Protocol IV reference results for models trained only on GIFTBench-Core. Each composition score is paired with its corresponding single-operation reference. Entries are F1/IoU; $\Delta_{\mathrm{F-S}}$ and $\Delta_{\mathrm{M-S}}$ are the Fusion-minus-Single and Multi-Source-minus-Single differences, respectively.

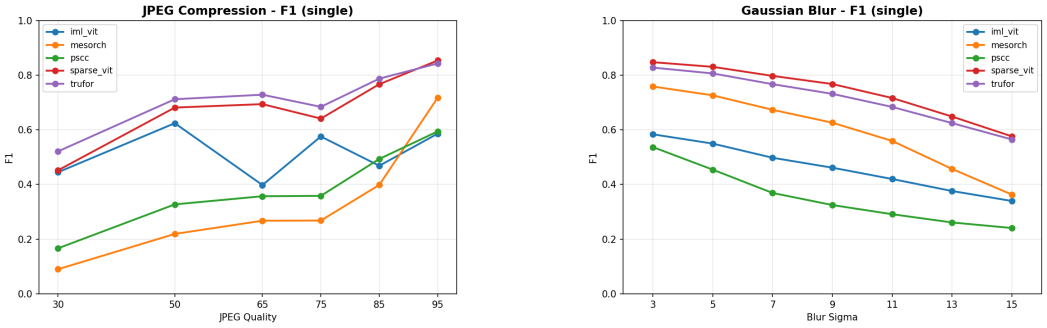
Model	$\mathrm{Single}_{\mathrm{Fusion}}$	Fusion	$\mathrm{Single}_{\mathrm{Multi}}$	Multi-Source	$\Delta_{\mathrm{F-S}}$	$\Delta_{\mathrm{M-S}}$
PSCC-Net	.611/.511	.538/.442	.577/.480	.486/.394	-.073/-.069	-.091/-.086
TruFor	.884/.835	.814/.729	.817/.755	.670/.609	-.070/-.106	-.147/-.146
IML-ViT	.673/.649	.657/.613	.684/.634	.606/.558	-.016/-.036	-.078/-.076
Mesorch	.794/.719	.733/.631	.740/.657	.707/.613	-.061/-.088	-.033/-.044
SparseViT	.908/.862	.793/.713	.845/.781	.722/.653	-.115/-.149	-.123/-.128

C.6 Robustness under image degradation

The existing benchmark post-processing stress tests evaluate single-operation images after JPEG compression, Gaussian blur, resize–JPEG, and double-JPEG processing. The JPEG and blur curves in Fig. 7 provide benchmark-level stress evidence; they are not new exp021 Full-Late measurements and are not folded into the four controlled protocol scores.

Table 14: Post-processing conditions used in the existing benchmark stress tests. The table defines the perturbations; numerical curves for JPEG and Gaussian blur are shown in Fig. 7.

Condition	Perturbation
JPEG	Re-encode the image at the evaluated JPEG quality levels
Gaussian blur	Apply the evaluated blur-strength levels to the image
Resize–JPEG	Resize before JPEG re-encoding
Double-JPEG	Apply two successive JPEG compression stages



(a) JPEG compression (b) Gaussian blur

Figure 7: Post-processing robustness on single-operation samples under JPEG compression and Gaussian blur.

D ForenScope Architecture, Training, and Inference Details

D.1 Classification adaptation with LoRA

Before localization training, DINOv2 is adapted together with a lightweight image-level classification head. The adaptation stage starts from the official DINOv2 ViT-L/14-reg pretrained weights and inserts LoRA adapters into the attention projections of blocks 8–23. The original encoder weights remain frozen; only the LoRA parameters and the CLS head are optimized with a binary classification loss. The head consumes the CLS token and predicts the real/manipulated image label. The LoRA adapters use a low-rank residual parameterization with a small update scale and dropout, while the classification head uses a larger learning rate than the adapters. Classification checkpoints are selected using the available classification validation protocol.

We use two data configurations for this stage. The standard ForenScope configuration performs the adaptation on the ADE-domain classification data and provides the ADE-adapted encoder and CLS head used by the main localization model. The controlled Protocol-MVSS configuration repeats the same adaptation recipe from the official pretrained encoder on CASIAv2, using its real

and manipulated images, and then initializes the subsequent localization stage from the resulting CASIAv2-adapted checkpoint. CASIAv1 is not used to optimize the CASIAv2 classification adaptation. These configurations change the source of the adapted representation, while the downstream localization architecture and the freezing rule remain the same.

After adaptation, the encoder, its LoRA parameters, and the CLS head are frozen during localization training. The localization decoder consumes patch features from the adapted encoder, while the frozen CLS representation is retained for image-level detection and for CLS-conditioned modulation of the localization branch. Thus, the classification adaptation is part of the initialization pipeline rather than an additional loss optimized during localization.

D.2 Initialization and trainable modules

ForenScope initializes its encoder and image-level branch from the ADE-domain classification-adapted DINOv2 model. During localization training, this encoder and the inherited classification branch remain frozen; only the localization-specific modules are optimized. The relevant structural variants and inference settings are summarized below without tying the paper to experiment-directory names or checkpoint hashes.

Variant	Encoder / image-level branch	Trainable localization modules	FiLM	Prototype
ForenScope	ADE-adapted DINOv2; frozen image-level branch	Multi-depth projections, LayerMixers, decoder, auxiliary/output heads	On	Disabled in default direct output
ForenScope + Proto.	Same trained network; frozen image-level branch	Same trained localization branch	On	Enabled at inference only
Existing No-CLS	ADE-adapted encoder; frozen image-level branch	Retrained multi-depth branch without CLS-FiLM	Off	Disabled
H37-only	ADE-adapted encoder; frozen image-level branch	Retrained H37-only localization variant	Off in effective path	Disabled
w/o CLS-FiLM & ADE adaptation	Original DINOv2; no ADE adaptation	Retrained localization branch without CLS-FiLM	Off	Disabled

Table 15: Initialization, trainable modules, and inference variants. ForenScope is the default direct output; ForenScope + Proto. uses the same trained network and adds only inference-time prototype refinement. Structural rows are independently retrained unless stated otherwise.

The no-CLS/no-ADE-adaptation row changes both encoder initialization and the CLS-conditioned branch, so it is a joint control rather than a pure classification ablation.

D.3 Encoder and token extraction

The encoder is DINOv2 ViT-L/14 with four register tokens. At input resolution 1036×1036, the token sequence is ordered as

$$\texttt{CLS},\ \texttt{register}_1,\ldots,\texttt{register}_4,\ \texttt{patch}_1,\ldots,\texttt{patch}_N.$$

The classifier consumes the CLS token. The localization branch removes the first $1 + 4$ tokens and reshapes only the remaining patch tokens into spatial maps. Feature taps are the zero-based DINOv2 block indices $F8, F11, F14, F17$; register tokens are never treated as spatial patches.

D.4 Multi-scale projection and learnable fusion

Each of the four tapped feature maps is projected from 1024 to 128 channels by an independent 1×1 projection followed by GroupNorm and GELU. The four projected features are mapped to the three localization resolutions H19, H37, and H74, producing twelve layer–scale lateral paths. Scale-specific downsampling or upsampling preserves the spatial correspondence of each path before fusion.

For each scale, the LayerMixer maintains four learnable logits and normalizes them with a softmax. The resulting output is a four-way convex mixture $F_s = \sum_l \bar{\alpha}_{l,s} U_{l,s}$, with one learned coefficient

vector for each spatial scale. These coefficients are shared across images at inference and are optimized jointly with the localization branch. The fixed-equal intervention in Appendix E.4 replaces these learned coefficients by 0.25 for each depth at inference without retraining.

D.5 CLS-CONDITIONED LOCALIZATION AND DECODER

The CLS token is mapped by a two-layer MLP to a conditioning vector. At H19 and H37, FiLM uses a bounded residual modulation

$$\widetilde{F}_s = F_s + \rho_s\big(g_s \odot F_s + r_s\big), \qquad g_s = \tanh(W_{\gamma,s}c + b_{\gamma,s}), \quad r_s = \tanh(W_{\beta,s}c + b_{\beta,s}),$$

with $\rho_{19} = 1$, $\rho_{37} = 0.25$, and no FiLM at H74. The FiLM projections are zero-initialized, so the conditioned path starts as a residual perturbation of the unconditioned representation. This placement leaves the highest-resolution H74 path available for fine spatial evidence.

The decoder fuses the conditioned H19 and H37 features from coarse to fine, then combines the result with the unconditioned H74 feature. Progressive upsampling produces intermediate resolutions 148, 296, 592, and the final 1036 image resolution. Auxiliary heads supervise H19, H37, and H74; the final one-channel output head produces the direct localization logit map.

D.6 OBJECTIVE, SAMPLING, AUGMENTATION, OPTIMIZATION, AND SELECTION

The localization objective is binary cross-entropy plus soft Dice at the final output and auxiliary scales:

$$\mathcal{L} = \ell_{\text{final}} + 0.1\ell_{19} + 0.2\ell_{37} + 0.3\ell_{74}.$$

Manipulated images have weight 1.0 and real images have weight 0.3. The main training protocol samples 17,000 examples from each of Splicing, Copy–Move, Removal, and AI-Edit, includes 532 attribute examples, and samples 20,000 real images, for a planned 88,532 samples per epoch (88,512 effective samples under the three-process DDP run). Validation contains 493 manipulated and 493 real images.

The formal configuration uses three-process DDP with batch size 8 per process (global batch size 24), two data-loader workers per process, five epochs, learning rate 3×10^{-4}, and seed 42. Only the 3,210,256 localization-branch parameters are optimized. We use AdamW with weight decay 0.05, clip the decoder gradient norm at 1.0, linearly warm up for 500 optimizer steps, and then apply cosine decay to a learning-rate floor of 5×10^{-7}. EMA with decay 0.999 is updated after every optimizer step. Geometry-changing augmentation is applied jointly to image and mask; photometric augmentation is image-only; masks use nearest-neighbor resizing and are re-binarized. The best checkpoint is selected from held-out in-domain pixel F1 only, with no OOD labels used for checkpoint selection. Both live and EMA weights are saved, but the reported standard localization tables use the selected EMA weights unless a diagnostic row states otherwise.

All twelve external datasets are excluded from optimization, validation, and checkpoint selection; they are used only by the evaluation scripts after model selection.

D.7 PROTOTYPE-GUIDED INFERENCE

Prototype refinement is an optional inference-only strategy evaluated in addition to the default direct ForenScope output. It does not update the encoder, decoder, optimizer state, or checkpoint. For each test image, the procedure is:

1. obtain the direct localization logits and the normalized H37 patch features;
2. use the direct H37 prediction to form manipulated and real support features with the fixed 70/30 trimmed support rule;
3. compute the cosine-similarity margin between each H37 patch and the two support prototypes;
4. if a support set is empty, use the direct H37 probability q_i for the manipulated support or $1 - q_i$ for the real support as its soft fallback; add the calibrated prototype correction with the fixed $+8$ logit scale, and upsample to the output mask;

5. threshold the refined mask at 0.5.

More formally, let h_i denote the ℓ_2-normalized H37 feature at spatial position i, and let $q_i \in [0,1]$ denote the direct localization probability resized to the H37 grid. For the reported `final_trim_70_30_raw` configuration, confident manipulated and real regions are assigned the weights

$$w_i^+ = \max(q_i - 0.7, 0), \qquad w_i^- = \max(0.3 - q_i, 0). \tag{5}$$

The corresponding prototypes are the normalized weighted means

$$\mu^+ = \frac{\sum_i w_i^+ h_i}{\left\|\sum_i w_i^+ h_i\right\|_2}, \qquad \mu^- = \frac{\sum_i w_i^- h_i}{\left\|\sum_i w_i^- h_i\right\|_2}. \tag{6}$$

If either confident support set is empty, its weights are replaced by the soft fallback q_i for the manipulated prototype or $1 - q_i$ for the real prototype. The cosine-similarity margin at each H37 position is

$$m_i = \langle h_i, \mu^+ \rangle - \langle h_i, \mu^- \rangle. \tag{7}$$

If L denotes the direct full-resolution localization logit, the refined logit is

$$L^{\mathrm{ref}} = L + 8\ \mathrm{Up}_{37 \to H}(m), \tag{8}$$

where $\mathrm{Up}_{37 \to H}$ denotes bilinear upsampling to the output resolution. The refined mask is obtained by thresholding $\sigma(L^{\mathrm{ref}})$ at 0.5.

The fixed `final_trim_70_30_raw` configuration is applied unchanged across the twelve external datasets. Prototype computation uses the model's own predictions and features, not target-domain labels, and the refinement does not update any model parameters. "ForenScope" denotes the direct network output in all primary tables, whereas "ForenScope + Proto." denotes the same checkpoint followed by this fixed inference correction.

E Complete ForenScope Evaluation and Ablations

This section gathers the ForenScope evaluation that is summarized in the main paper. The default method is the learned localization network and its direct detection and localization outputs. An optional fixed prototype-guided inference strategy is reported separately as an auxiliary variant. Results distinguish in-domain localization, external localization, image-level detection, retrained structural variants, and fixed-checkpoint interventions.

E.1 Evaluation protocol for ForenScope

In-domain evaluation uses `cross/in_domain/test_all.json`, whose 13 masked groups are organized into source (Splicing, Copy–Move, Removal, AI-Edit), semantic target (Object, Component, Background), operation (Add, Remove, Replace), and composition (Single, Fusion, Multi-Source) views. External localization uses the primary-6 and extra-6 groups listed in Appendix B.2; All-12 is their equal-weight macro average. The reported ForenScope row is the direct output, while ForenScope + Proto. uses the same checkpoint with the optional inference correction.

E.2 Complete external localization

Table 16 reports all twelve external datasets and the optional inference comparison. The ForenScope row is the direct network output, whereas ForenScope + Proto. applies the same fixed checkpoint followed by prototype refinement. The table therefore preserves the distinction between learned-network output and inference-time refinement rather than hiding the per-dataset trade-off.

E.3 Image-level detection results

The detector is the frozen classification head inherited from the ADE adaptation stage; localization training does not optimize its loss. Table 17 reports the complete paired AUC/F1 comparison on the eight external datasets containing both real and manipulated images, including the controlled CASIAv2-trained setting. AUC is undefined for BRGen, SceneText, OSTF, and DDL because these

Table 16: External localization results with methods as rows. Entries are F1/IoU; ForenScope is the direct output and ForenScope + Proto. adds fixed inference-only refinement. Primary-6, Extra-6, and All-12 are macro averages over the corresponding columns.

	Primary-6						Extra-6						Summary		
Method	CASIA	CocoGlide	Columbia	IMD2020	NIST16	COVERAGE	MISD	AutoSplice	BRGen	SceneText	OSTF	DDL	Primary-6	Extra-6	All-12
ForenScope	.679/.598	.799/.697	.546/.452	.579/.488	.548/.469	.578/.472	.696/.560	.871/.789	.746/.665	.703/.596	.595/.476	.520/.412	.621/.529	.688/.583	.655/.556
Existing No-CLS	.660/.582	.762/.657	.534/.442	.572/.482	.516/.438	.581/.467	.694/.562	.897/.827	.759/.679	.724/.617	.615/.497	.486/.386	.604/.511	.696/.595	.650/.553
H37-only	.658/.577	.676/.564	.495/.401	.570/.477	.547/.461	.494/.384	.688/.553	.889/.816	.748/.666	.691/.592	.578/.465	.527/.416	.573/.477	.687/.584	.630/.531
Joint control	.567/.490	.593/.481	.601/.499	.550/.456	.500/.420	.424/.313	.686/.550	.897/.831	.748/.661	.685/.580	.574/.459	.463/.363	.539/.443	.676/.574	.607/.509
ForenScope + Proto.	.695/.615	.820/.724	.564/.472	.580/.487	.566/.485	.592/.487	.726/.597	.873/.793	.753/.671	.705/.598	.600/.478	.532/.425	.636/.545	.698/.594	.667/.569

Table 17: Image-level detection AUC and F1 on eight external datasets containing both classes.

Training Protocol	Model	CASIA v1		CocoGlide		Columbia		IMD2020		NIST16		COVERAGE		MISD		AutoSplice		Avg.	
		AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
Protocol-MVSS	MVSS-Net	.639	.697	.507	.667	.468	.663	.564	.907	.500	.943	.519	.667	.704	.489	.543	.761	.556	.724
	PSCC-Net	.743	.521	.440	.639	.806	.732	.629	.898	.547	.887	.495	.662	**.994**	.869	.827	.822	.685	.754
	TruFor	.148	.392	.446	.625	.049	.100	.392	.796	.425	.819	.352	.592	.034	.075	.459	.646	.288	.506
	FORENSCOPE	.875	.732	.702	.675	.819	.731	.727	.815	.563	.688	.551	.418	.892	.779	.691	.400	.727	.655
GIFTBench	**FORENSCOPE**	**.902**	**.844**	**.989**	**.963**	**.994**	**.967**	**.899**	.834	**.837**	.705	**.891**	**.809**	.993	**.911**	**.977**	**.934**	**.935**	**.871**

splits contain only manipulated images. These results document retained detection capability, not a standalone detection leaderboard.

The same frozen ADE-adapted encoder and classification head are shared by the Full-Late, Existing No-CLS, and H37-only localization variants. Removing CLS-FiLM or changing the localization decoder therefore does not create a new detector, so the detection table is not duplicated for those localization variants.

E.4 ABLATIONS AND MECHANISM ANALYSIS

The following tables separate independently retrained structural variants from fixed-checkpoint inference interventions. This distinction is essential: the structural table evaluates the final retrained Static LayerMixer model, whereas the diagnostic table changes only the prediction path of separately identified reference checkpoints. Thus, the diagnostic rows are not a second evaluation of the final ForenScope checkpoint. All structural rows use direct network outputs; the final inference row adds the prototype without retraining.

Table 18: Definitions of the principal structural variants and inference interventions used in the complete evaluation.

Variant	Definition
ForenScope	Full multi-depth network with learned LayerMixer and CLS-FiLM; direct output only.
Existing No-CLS	ADE-adapted encoder retained, but the localization branch is retrained without CLS-FiLM.
H37-only	H37 is the only effective spatial scale; the remaining fusion topology is retained.
w/o ADE adaptation	Original DINOv2 encoder with the full multi-scale, CLS-conditioned localization architecture.
Joint control	Original DINOv2 initialization, no ADE adaptation, and no CLS-FiLM; retrained jointly.
Equal fixed mixer	Fixed-checkpoint intervention replacing learned four-depth weights by equal weights.
FiLM-off / CLS-shuffle	Fixed original Full-Late checkpoint interventions disabling FiLM or shuffling the CLS conditioning.
ForenScope + Proto.	Fixed inference-time refinement applied after the direct network output.

Table 19: Retrained structural ablations. ForenScope denotes the final Static LayerMixer checkpoint; entries are direct-output F1/IoU, with in-domain as the thirteen-group macro average and OOD as All-12.

Variant	In-domain	OOD
ForenScope	.886/.823	.655/.556
Existing No-CLS	.881/.816	.650/.553
H37-only	.871/.801	.630/.531
w/o ADE adaptation	–	.618/.519
Joint control	.873/.805	.607/.509

The pure no-adaptation variant was evaluated on All-12 but not on the thirteen-group in-domain test, so its in-domain entry is left undefined. The structural rows show that ADE adaptation and multi-scale evidence improve transfer, whereas removing CLS-FiLM has a smaller effect after retraining. The joint control changes both encoder adaptation and conditioning and therefore does not isolate either factor.

Table 20: Fixed-checkpoint mechanism diagnostics on All-12. The equal-layer control uses the final Static LayerMixer checkpoint; the No-CLS and Full-Late rows use their respective EMA checkpoints. None of these rows retrains the network.

Reference checkpoint (EMA)	Inference condition	F1/IoU
ForenScope (Static)	Equal layer weights	.653/.557
Existing No-CLS	Learned layer fusion	.650/.553
Existing No-CLS	Equal layer weights	.553/.450
Original Full-Late	Normal CLS-FiLM	.643/.548
Original Full-Late	FiLM off	.622/.518
Original Full-Late	CLS shuffled	.633/.536

These fixed-checkpoint interventions test whether the corresponding trained prediction paths use learned fusion and CLS conditioning, rather than comparing independently retrained architectures. In particular, the *Original Full-Late* normal row (.643/.548) is a separate diagnostic reference and should not be compared as the Full baseline of Table 19, whose final Static LayerMixer ForenScope row is .655/.556. A matched FiLM intervention on the final Static LayerMixer checkpoint would require a separate inference run. The optional prototype refinement is parameter-free and improves the external All-12 average, but its in-domain evaluation shows that the gain is not uniform across benchmark groups.

E.5 Reproducibility checklist

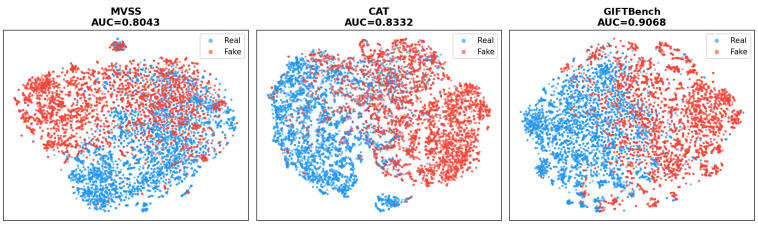
The released experiment records specify: DINOv2 ViT-L/14 reg4; input resolution 1036×1036; feature taps $F8/F11/F14/F17$; H19/H37/H74 spatial scales; threshold 0.5; EMA decay 0.999; seed 42; live/EMA checkpoint distinction; the fixed prototype rule; the single-seed policy; no target-domain fine-tuning; in-domain-only checkpoint selection; and the in-domain and external evaluation manifests. Prototype inference uses the model's own predictions and features with shared fixed hyperparameters, and requires neither target-domain labels nor parameter updates.

F Qualitative Analysis

This section complements the numerical results with visual evidence of transfer behavior and failure cases.

F.1 Dataset-level transfer and representation views

Figure 8 provides a descriptive two-dimensional projection of TruFor decoder features under different training distributions.



(a) t-SNE view of representative decoder features

Figure 8: Supplementary representation visualization of TruFor decoder features under different benchmark training conditions.

F.2 External qualitative comparisons

Figure 9 qualitatively compares representative models trained under Protocol-MVSS, Protocol-CAT, and GIFTBench; these examples are not used for model selection.

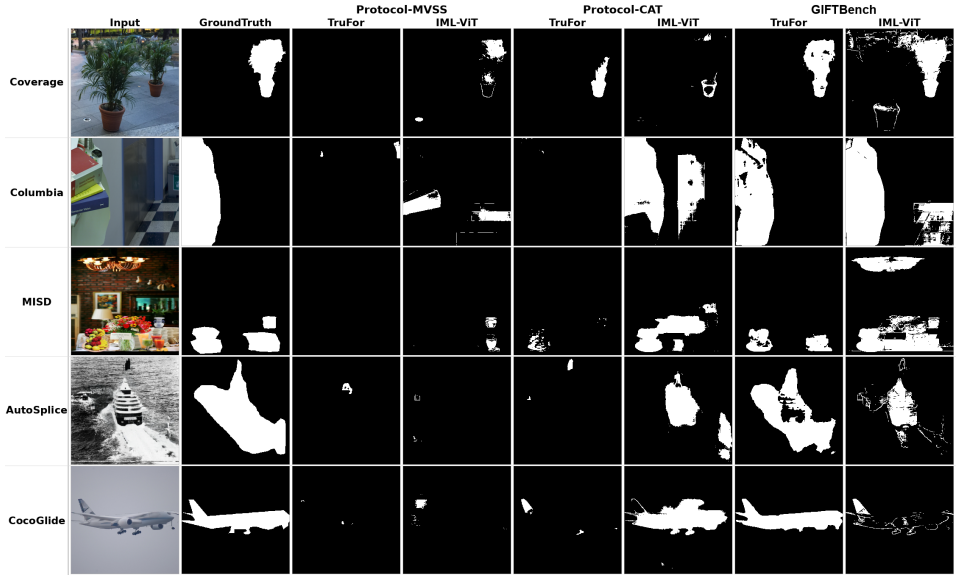
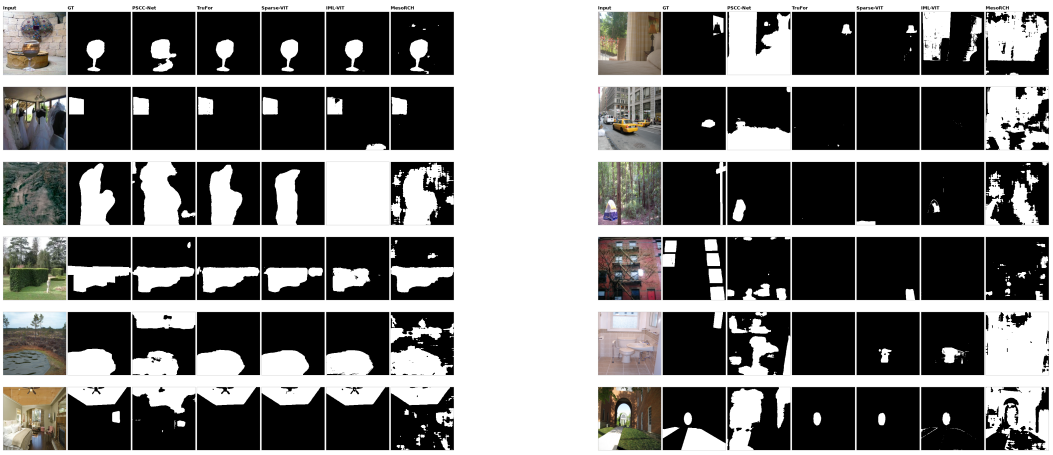


Figure 9: Qualitative external-transfer comparison under Protocol-MVSS, Protocol-CAT, and GIFT-Bench training.

F.3 Good and bad cases on GIFTBench

Figure 10 contrasts high-agreement benchmark examples with cases for which all displayed baseline models obtain F1 below 0.15.



(a) Representative high-agreement cases

(b) Representative all-model failure cases

Figure 10: Representative high-agreement and all-model failure cases on GIFTBench.