Beyond Real or Fake: A Dual-Channel Authenticity and Reasoning Protocol for Photographic Assessment

Xiaoxiao Li[1,3,4] **Ruinan Jin**[1,3,4] **Lili Meng**[1]
Miaosen Wang[2] **Athula Balachandran**[1] **Pei Cao**[1]
[1]Google [2]Google DeepMind [3]University of British Columbia [4]Vector Institute

Abstract

Forensic verification has converged on a binary "real vs. fake" label that conflates fully synthetic, tampered, and AI-retouched images despite their very different consequences. We argue that GenAI manipulations decompose along two complementary channels: a *camera channel* of sensor fingerprints (destroyed by neural processing), and a *semantic channel* of scene-level coherence (disrupted by content edits). Each manipulation type leaves a distinctive two-channel signature. We instantiate this taxonomy in **2CAP** (**2**-**C**hannel **A**uthenticity **P**rotocol), pairing a contrastively-trained camera encoder with a frozen semantic encoder to serve two applications on a shared backbone: **(i)** four-class authenticity classification via cross-attention fusion with learnable reliability weights; and **(ii)** evidence generation, where an authentic image serves as a reference for a query and the two encoders produce per-channel alignment scores together with a patch-level saliency map. These signals are supplied as privileged context to a frozen Vision-Language Model (VLM) through an Observe-Generate-Refine loop, yielding explanations with localization without forensic fine-tuning. On a multi-source benchmark, 2CAP attains the best classification performance, including strong retouching detection where existing methods fail, and improves explanation quality.

1 Introduction

When generative AI (GenAI) tools modify a photorealistic image, they do not all break it the same way. A portrait restyled by an enhancement model [1, 2] may still depict the same person while altering the original capture traces; a news photograph with an inserted object [3] may retain camera-related cues in unedited regions while changing local semantic content; a fully synthesized scene [4] has no camera capture of the depicted scene, yet may remain semantically coherent. These scenarios can demand different responses, which a binary authenticity label alone does not express.

Cryptographic provenance (C2PA [5]) addresses authenticity at capture time but cannot reach legacy or stripped content. Computational forensics fills this gap, but the field has organized around binary classification, across both forgery localization [6–8] and AI-generated image detection [9–13], grouping AI-generated, edited, and retouched images under a single "fake" label regardless of how or why they were modified. This loses the structured evidence that journalism, legal proceedings, identity verification, and content moderation actually need: *what was changed, and does it matter under a given policy?* For example, camera-parameter retouching may restyle photographic appearance while preserving the people, objects, and scene relationships relevant to a factual claim. Such an image is modified, but the modification alone does not establish that its depicted information is false; a blanket "fake" label therefore conflates processing history with factual fidelity. Whether the edit is acceptable remains dependent on the claim, context, and applicable policy. We argue that useful forensic systems must not only detect modifications, but characterize GenAI modification type

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

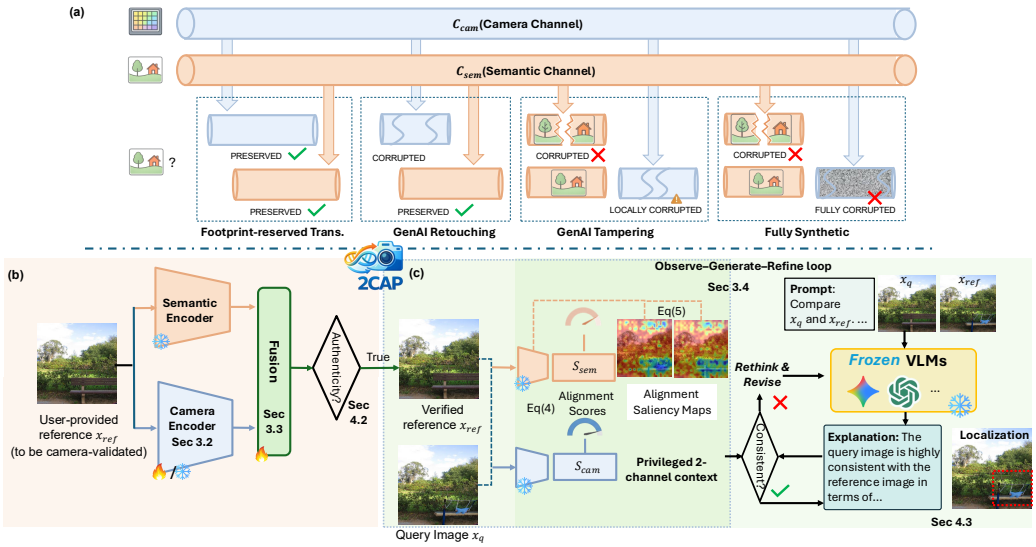

Figure 1: **2CAP overview.** (a) *Two-channel taxonomy.* Authenticity cues are decoupled into a camera channel C_{cam} (sensor traces) and a semantic channel C_{sem} (scene coherence); each manipulation type leaves a distinct signature: footprint-preserved transformations preserve both channels, GenAI retouching corrupts only C_{cam}, GenAI tampering corrupts C_{cam} globally and C_{sem} locally, and fully synthetic images corrupt both. (b) *Authenticity classification* (Q1). A camera encoder (Sec. 3.2) and a frozen semantic encoder feed a cross-attention fusion module with learnable reliability weights (Sec. 3.3) that produces a four-class verdict (Sec. 4.2). (c) *Reference-based evidence generation* (Q2). Given a query $\mathbf{x}_q$ and a camera-verified reference $\mathbf{x}_{\text{ref}}$, the two encoders yield channel alignment scores $s_{\text{cam}}, s_{\text{sem}}$ (Eq. 4) and a patch-level saliency map (Eq. 5); these serve as privileged two-channel context to a frozen VLM in an Observe–Generate–Refine loop (Sec. 3.4) that returns an explanation and localization, revising whenever the VLM's initial assessment is inconsistent with the channel evidence (Sec. 4.3).

by distinguishing (i) *retouching* (camera-parameter restyling intended to preserve scene content), (ii) *tampering* (localized semantic edits), and (iii) *full synthesis* (no camera capture).

We organize photographic assessment around two complementary evidence channels: a **camera channel** sensitive to capture and processing cues, and a **semantic channel** encoding scene content and relationships. We adopt camera-captured, GenAI-retouched, locally tampered, and fully synthetic images as four prototypical image-formation and editing regimes (Fig. 1(a)). They distinguish questions that binary real/fake detection conflates: whether an image originates from camera capture, whether its photographic appearance has been restyled, and whether its scene content has been modified or generated. These distinctions define an operational benchmark for evaluating the information contributed by the two channels across forms of image creation and editing.

Practical workflows also compose operations: local content editing followed by denoising and recompression can affect both semantic correspondence and camera-related traces. We therefore use the four regimes as reference cases for studying complementary evidence, rather than as an exhaustive partition of processing histories. Channel responses can overlap: benign processing can weaken capture cues, and synthesis can remain semantically coherent. Continuous query–reference alignment measurements describe such responses relative to a corresponding reference, while four-class prediction remains a dataset-defined task on joint image features.

We propose **2CAP** (**2**-Channel **A**uthenticity **P**rotocol), which implements this evidence-based formulation with two encoders feeding two complementary applications. The shared backbone consists of a contrastively-trained camera encoder that maps images into a learned space of camera-related processing cues, paired with a frozen SigLIP2 [14] semantic encoder. **(1) Reference-free authenticity classification** (Fig. 1(b)): a cross-attention fusion module with learnable reliability weights adaptively combines the two channels per single image to produce a four-class verdict over camera, retouched, tampered, and fully synthetic. **(2) Reference-based evidence generation** (Fig. 1(c)): given a query image and a trusted corresponding reference, we treat per-channel alignment scores and a patch-level

saliency map as *privileged context* that a frozen VLM cannot compute on its own, and feed them through a structured Observe-Generate-Refine (OGR) loop based on the iterative self-refinement paradigm of Madaan et al. [15] to obtain an explanation and a localized bounding box.

Contributions. (1) A new problem formulation. We move image forensics beyond binary real/fake classification by organizing assessment around camera-related and semantic evidence across four prototypical regimes. We evaluate four-class discrimination using the joint features and inspect pairwise channel scores for camera, retouched, and tampered images. **(2) A new detection capability.** We instantiate this formulation in 2CAP, a cross-attention fusion framework with learnable reliability weights that achieves the highest overall AUC and F1 among the compared methods on our benchmark. **(3) Forensic explanation without VLM fine-tuning.** We show that *frozen* VLMs can produce structured forensic evidence without any forensic SFT data: channel similarity scores and saliency maps, provided as privileged context through an OGR loop, improve change-type accuracy over the single-pass baseline across four evaluated VLMs, with model- and metric-dependent effects in the component ablations.

2 Related Work

Image forgery detection and localization. Classical forensics exploit sensor fingerprints [16, 17] and compression traces [18, 19]. Learning-based methods fuse noise and RGB cues for end-to-end detection and localization [6–8, 20]. In parallel, AI-generated image detection improves cross-generator generalization via pretrained representations [9–13, 21, 22]. These methods largely target binary detection or localization; our evaluation additionally distinguishes GenAI retouching as a separate category. Extended discussion is in Appendix B.

Training data and representation controls. B-Free [23] reduces content bias through aligned real/synthetic training data, while DRCT [24] uses diffusion-reconstructed hard examples with contrastive training. SSAFE [25] studies detection with frozen multimodal encoders. These methods motivate separating the effects of training data, representation choice, and fusion architecture when interpreting our results.

VLM-based forensic reasoning. Recent methods train VLMs for forensic detection, localization, and explanation, including FakeVLM [26], FakeShield [27], ForgeryGPT [28], and SIDA [29]. SIDA predicts tamper masks as well as explanations, while its 300K-image dataset should not be equated with 300K explanation-instruction pairs. More recent work explores expert-grounded reasoning through trained models, including REVEAL [30]. Our distinct evaluation concerns whether channel evidence improves a frozen VLM's reference-based assessment. We demonstrate this across four proprietary configurations across two model families (GPT-4o and Gemini 2.5), without forensic instruction tuning of the VLM; this does not imply that the encoders are training-free or that OGR outperforms fine-tuned forensic VLMs.

3 Method

3.1 Verification Protocol and Problem Formulation

We consider two practical forensic scenarios with complementary input requirements:

1. **Q1: Is a given image authentic?** Given a single image $\mathbf{x}$, predict its dataset-defined label: camera-captured, GenAI-retouched, locally tampered, or fully synthetic. This is a *reference-free* (single-image) task.
2. **Q2: What changed between two images?** Given a query $\mathbf{x}_q$ and a reference $\mathbf{x}_{\mathrm{ref}}$ (e.g., a verified camera image), identify the nature, type, and location of modifications.

The reference-free classifier predicts a benchmark label from the joint encoder features; it does not infer an exhaustive processing history. For reference-based assessment, the continuous camera and semantic alignment scores $(s_{\mathrm{cam}}, s_{\mathrm{sem}})$ provide a two-dimensional description of evidence relative to a corresponding reference. This separates benchmark classification from the interpretation of combined processing effects: overlapping scores need not imply identical histories, and a composite workflow need not correspond uniquely to one of the four reference regimes.

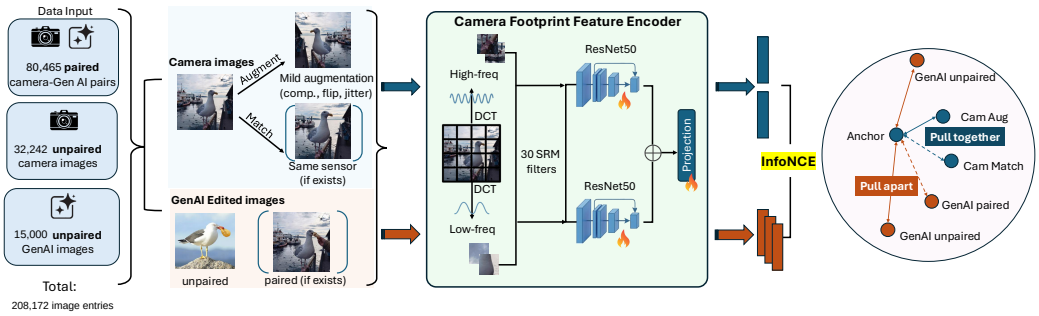

Figure 2: **Camera encoder pretraining.** *Left:* Training data composition: 80,465 pairs, 32,242 unpaired camera entries, and 15,000 unpaired edited entries, totaling 208,172 image entries (Table 3); unique images have not been deduplicated. *Center:* DCT-domain encoder with frequency-based patch selection and dual-branch feature extraction. *Right:* InfoNCE training where camera image embeddings are pulled together while edited image embeddings are pushed apart.

Two applications on a shared dual-channel foundation. Both questions are answered by combining two complementary encoders: a camera encoder $\mathcal{E}_{\text{cam}}$ (Sec. 3.2) that learns camera-related processing cues, and a frozen semantic encoder $\mathcal{E}_{\text{sem}}$ (SigLIP2 [14]) that captures scene-level coherence. On top of these shared dual-channel features we build two applications, evaluated independently in Sec. 4. **(1) Authenticity classification** (Sec. 3.3; evaluated in Sec. 4.2): a cross-attention fusion module with learnable reliability weights adaptively combines the two channels per single input to produce a four-class verdict. **(2) VLM evidence generation** (Sec. 3.4; evaluated in Sec. 4.3): per-channel alignment scores between $\mathbf{x}_q$ and $\mathbf{x}_{\text{ref}}$, together with a patch-level saliency map, are supplied as privileged context to a frozen VLM through an Observe-Generate-Refine loop, producing interpretable evidence with localization.

3.2 Camera Fingerprint Encoder

Limitation of existing pretrained encoders. Pretrained semantic encoders provide useful forensic features [12, 22], but their training objective does not explicitly target capture and processing traces. We therefore complement frozen semantic features with a camera-related encoder trained on camera/edit examples, and evaluate their contributions separately.

Camera encoder architecture. We adopt AIDE's DCT+SRM dual-branch backbone [11] and reprogram its training objective from cross-entropy classification to InfoNCE contrastive metric learning (Figure 2). The low-level feature extraction pipeline (DCT-based patch ranking and 30 SRM steganalysis filters) is inherited; our contribution is the training scheme, the cross-channel fusion, and the frozen-VLM evidence generation (a full architectural comparison is in Appendix C.3). Specifically, (i) we train with InfoNCE contrastive learning over camera-camera positives and paired or unpaired camera-edit negatives, producing a 128-d unit-norm embedding space rather than class logits; (ii) we use cosine similarity in this space as a continuous camera-channel score s_{cam} rather than a binary decision; and (iii) we fuse camera features with features from a frozen semantic encoder (Sec. 3.3).

Contrastive pretraining for generalizability. A key design principle is that our camera encoder should learn *camera-specific traces*, not semantic shortcuts that fail to generalize across generators or editing tools. We encourage this through contrastive learning (Figure 2).

We construct two types of positives for each camera anchor: 1) *Camera Aug*: A mild augmentation of the same image (horizontal flip, mild color jitter, and mild JPEG compression). 2) *Camera Match*: Another image from the same camera sensor, when available. This encourages the encoder to learn sensor-consistent representations. The negative pool consists of all edited images in the batch, including 1) *GenAI paired*: The GenAI-edited version of the anchor's source image. This provides hard negatives that share scene content but lack authentic camera traces; and 2) *GenAI unpaired*: Other edited images in the batch, providing diverse negative examples across different editing styles and generators. The total loss combines paired and unpaired streams:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{paired}} + \lambda\, \mathcal{L}_{\text{unpaired}}, \tag{1}$$

where $\lambda = 0.5$. Each term is an InfoNCE loss with temperatures $\tau_p = 0.07$ (paired) and $\tau_u = 0.1$ (unpaired).

3.3 Cross-Channel Fusion for Authenticity Classification

The camera and semantic encoders provide features with different training objectives. We combine them using bidirectional cross-attention and input-dependent gating (Appendix C.2). The complementarity of the learned representations is evaluated through channel and fusion ablations, rather than assumed from a deterministic mapping between manipulation type and channel response.

Cross-attention fusion. We distinguish the token features used for classification from the global embeddings used for pairwise cosine scores. Let $Z_c \in \mathbb{R}^{B\times N_c\times d_c}$, for $c \in \{\text{cam}, \text{sem}\}$, denote the channel patch-token sequences, where N_{cam} and N_{sem} are the numbers of camera and semantic patches (tensor shapes are summarized in Table 5). Learned projections map them to $\widetilde{Z}_c \in \mathbb{R}^{B\times N_c\times 256}$. For each attention head,

$$\text{Attn}(Q, K, V) = \text{softmax}_{\text{keys}}\Big(QK^{\mathsf{T}}/\sqrt{d_h}\Big)\, V, \tag{2}$$

with camera-to-semantic attention weights of shape $B \times N_{\text{cam}} \times N_{\text{sem}}$ and the reverse shape in the other direction. Residual cross-attention yields Z'_{cam} and Z'_{sem}; channel-wise pooling produces the 256-d vectors $\mathbf{z}'_c$ used below. Attention is normalized over the key patches for each query patch, producing a patch-to-patch interaction matrix rather than a single attention coefficient.

Reliability-weighted combination. Not all channels are equally informative for every image. A two-layer gating MLP takes the concatenated channel vectors and produces two logits,

$$\boldsymbol{\ell} = g_{\text{gate}}([\mathbf{z}'_{\text{cam}}; \mathbf{z}'_{\text{sem}}]) \in \mathbb{R}^2, \qquad \alpha_c = \frac{\exp(\ell_c)}{\exp(\ell_{\text{cam}}) + \exp(\ell_{\text{sem}})}, \quad c \in \{\text{cam}, \text{sem}\}.$$

Here, softmax is applied over the two channels for each image, so $\alpha_{\text{cam}} + \alpha_{\text{sem}} = 1$. This normalization is separate from the key-patch softmax inside cross-attention. We fuse the channel vectors using these input-dependent weights:

$$\mathbf{z}_{\text{fused}} = \alpha_{\text{cam}} \cdot \mathbf{z}'_{\text{cam}} + \alpha_{\text{sem}} \cdot \mathbf{z}'_{\text{sem}}, \quad \mathbf{p} = \text{softmax}(h_{\text{cls}}(\mathbf{z}_{\text{fused}})) \tag{3}$$

The classifier h_{cls} has dimensions $256 \to 128 \to 4$ and is trained with four-class cross-entropy; its output categories are camera-captured, retouched, tampered, and fully synthetic.

3.4 VLM Evidence Synthesis with Privileged-Context Feedback

The dual-channel classifier (Sections 3.2-3.3) answers Question 1 (camera authenticity). To answer Question 2 (what changed) with interpretable evidence, we employ a structured Observe-Generate-Refine (OGR) loop with frozen VLMs, following the iterative self-refinement paradigm [15] and taking the *image pair* as input for direct comparison.

Channel alignment scores and saliency. We compute alignment scores via cosine similarity $\text{sim}(\cdot, \cdot)$:

$$s_{\text{cam}} = \text{sim}(\mathcal{E}^g_{\text{cam}}(\mathbf{x}_q),\, \mathcal{E}^g_{\text{cam}}(\mathbf{x}_{\text{ref}}))\,, \quad s_{\text{sem}} = \text{sim}(\mathcal{E}^g_{\text{sem}}(\mathbf{x}_q),\, \mathcal{E}^g_{\text{sem}}(\mathbf{x}_{\text{ref}})) \tag{4}$$

where $\mathcal{E}^g_c$ denotes the global image embedding of channel c, distinct from its patch-token output $Z_c = \mathcal{E}^{\text{tok}}_c(\mathbf{x})$ used for cross-attention. The camera global embedding is 128-d and unit-norm. These are learned feature similarities, not probabilities of authenticity or direct measurements of retained sensor information. Low s_{cam} can reflect processing beyond GenAI, while high s_{sem} indicates similar global content without excluding small local edits. For spatial localization, we compute a saliency map using patch-level features $\{p_i(x)\}$ from SigLIP2 [14]:

$$\mathcal{M}_{\text{sem},i} = 1 - \max_j \text{sim}(p_i(\mathbf{x}_q),\, p_j(\mathbf{x}_{\text{ref}}))\,, \tag{5}$$

where high $\mathcal{M}_{\text{sem},i}$ indicates query patches lacking close matches in the reference. The resulting map $\mathcal{M}_{\text{sem}} = \{\mathcal{M}_{\text{sem},i}\}$ highlights query patches with weak reference matches. This directed map can miss removals or repeated-object changes when remaining query patches still find close matches.

Motivation. A frozen VLM comparing two images tends to describe visually salient differences first (lighting, color) and may miss two important cases: (i) retouching that changes camera traces

Algorithm 1 OGR Loop: Observe-Generate-Refine

Require: Image pair $(\mathbf{x}_q, \mathbf{x}_{\text{ref}})$, scores $(s_{\text{cam}}, s_{\text{sem}})$, saliency $\mathcal{M}_{\text{sem}}$
1: **Observe:** VLM receives $(\mathbf{x}_q, \mathbf{x}_{\text{ref}})$ and performs visual inventory
2: **Generate:** VLM outputs initial assessment: change type, description, bounding box
3: **Refine:** VLM receives Pass-1 output with $(s_{\text{cam}}, s_{\text{sem}}, \mathcal{M}_{\text{sem}})$; revises if evidence conflicts
4: **return** Structured evidence: $\{\hat{y}, s_{\text{cam}}, s_{\text{sem}}, \mathcal{M}_{\text{sem}}, \text{explanation}\}$

without visible content differences, and (ii) localized edits in low-saliency regions. Channel scores quantify evidence the VLM cannot directly compute (e.g., sensor-trace similarity), so providing them as feedback after an initial assessment lets the VLM correct both failure modes.

OGR protocol. Unlike prior methods that train VLMs for forensic tasks [26, 28, 29], we keep the VLM frozen and supply pre-computed signals (s_{cam}, s_{sem}, $\mathcal{M}_{\text{sem}}$) through a two-pass Observe-Generate-Refine loop (Algorithm 1). *Pass 1 (Generate):* the VLM receives the image pair and outputs an initial assessment including change type (addition/removal/replacement), description, and bounding box. *Pass 2 (Refine):* the VLM receives its own Pass-1 output together with the channel scores and saliency map, and revises if the evidence conflicts with its initial assessment. The VLM is not forced to agree with the scores; it performs independent reasoning and can confirm, refine, or override. Revisions can correct prior errors or introduce new ones; accuracy contrasts alone do not identify the two transition counts. We use a single prompt template across the tested VLM configurations; Appendix E.1 shows examples of the structured outputs. Evaluation across four frozen VLMs (Table 2) shows that the evidence contribution depends on both model and metric.

4 Experiments

Our experiments evaluate the two applications of 2CAP's dual-channel features, mirroring the questions posed in Sec. 1. **(i) Dual-channel classifier** (Sec. 4.2): a reference-free four-class classifier with cross-attention fusion and learnable reliability weights, addressing Q1 (*is the image authentic?*). **(ii) OGR feedback loop** (Sec. 4.3): a privileged-context protocol that supplies similarity scores and saliency map to frozen VLMs for evidence generation, addressing Q2 (*what changed?*).

4.1 Dataset Construction and Implementation Details

We construct a multi-source benchmark for distinguishing *camera-captured* images from *edited* images across three modification categories (see Appendix A.3 for full details):

- **Fully synthetic**: Images generated entirely by GenAI from text prompts, lacking camera fingerprints, including Defactify Image (MS COCOAI) [31] (SD2.1 [1], SDXL [2], SD3 [32], DALL-E 3 [33], Midjourney 6 [1]) and the Flickr8K-derived subset of RU-AI [34] (SDv1.5 [1], SDAbsoluteReality [2], SDepiCRealism [3], SDXL [2], SDv6.0 [35]).
- **Tampered**: Localized GenAI editing (object insertion/removal/inpainting) that corrupts semantic content in specific regions. Sources: SID-Set [29], SAGI-D [36], and MagicBrush [37].
- **GenAI-retouched**: Generative Photography [38] dataset with global camera-parameter restyling (bokeh, color temperature, focal length, motion blur) using AnimateDiff [39] intended to preserve scene content.

Camera images are aggregated from Dresden [40], VISION [41], Defactify Image [31], and SID-Set [29]; paired camera originals from SAGI-D and MagicBrush also contribute to contrastive pretraining (Table 3). This multi-source design broadens coverage, but unequal source and tool distributions across classes can introduce confounding. Shared sources across some classes do not eliminate that risk.

Implementation Details. Our architecture comprises a camera encoder $\mathcal{E}_{\text{cam}}$ (DCT-domain dual-branch ResNet-50 [11, 42] with SRM filters, outputting 128-d embeddings), a frozen semantic

[1] `https://www.midjourney.com/home`
[2] `https://huggingface.co/Lykon/AbsoluteReality`
[3] `https://huggingface.co/emilianJR/epiCRealism`

encoder $\mathcal{E}_{\text{sem}}$ (SigLIP2 [14], 768-d), and a cross-attention fusion module with learnable reliability weights. We pretrain $\mathcal{E}_{\text{cam}}$ on 208,172 image entries using InfoNCE contrastive learning (Table 3).

4.2 Camera Authenticity: Classification on Camera vs. Fully Synthetic vs. Tampered vs. GenAI-retouched Images

This section evaluates the *first* application: the dual-channel classifier with cross-attention fusion and learnable reliability weights (Sec. 3.3), trained to assign each image to one of four classes: *camera-captured*, *fully synthetic*, *tampered*, or *GenAI-retouched*, the categories corresponding to distinct forensic scenarios requiring different responses.

4.2.1 Baselines and Setup

We compare against ten methods: TruFor [8], D3 [22], CNNSpot [10], AIDE [11], VIBAIGCD [13], FreqNet [21], FatFormer [12], UnivFD [9], SAFE [43], and GAPL [44].

Training protocol. The adapted baseline rows use the historical 12,800-image downstream training pool. For the controlled 2CAP ablation, we exclude 468 training rows whose source-reference groups overlap the historical test split, deduplicate four repeated training paths, and then partition the remainder into 9,877 training and 2,451 validation images by source-reference group. All six 2CAP variants use these same examples, frozen encoder embeddings, minibatch order, optimizer settings, and 20-epoch budget; validation macro-F1 selects each checkpoint before one evaluation on the common 3,200-image test split (800 images per class). The full-model row uses a separate rerun in which all weights shared with the no-gate control start identically and the gate starts at equal weights; the no-gate control is the previously completed run under the same seed and recipe. Thus the method and baseline rows share the test population but *not* the exact downstream training population or adaptation budget. 2CAP also uses separate task-specific contrastive pretraining, so Table 1 should not be read as a fully matched baseline comparison. The historical test split was inspected during prior development and is not a pristine blind holdout.

Evaluation protocol. VISION and the Flickr8K-derived RU-AI subset are held out from both of our training stages. This tests dataset and camera-domain transfer. Dataset holdout does not itself establish generator-family holdout: for example, SDXL appears in both the listed training-source and held-out-source generator sets. We distinguish these settings from the stricter evaluation in which an entire generator or editing-tool family is excluded from all training and model selection.

4.2.2 Results

Quantitative classification results. A key contribution of 2CAP is distinguishing *among* modification types. All evaluations in this section are *reference-free* (single-image). Table 1 shows that the paired-initialization 2CAP run achieves overall AUC .982 and F1 .906 on the common four-class test split. These exceed the reported baseline values, subject to the downstream-training differences described above. For retouching on GP, 2CAP reaches OvR AUC .998 and F1 .964, compared with F1 .868 for D3, the strongest compared baseline. The semantic-only variant reaches retouching F1 .914; the full model is 5.0 points higher and exceeds semantic-only overall F1 by 12.5 points. These differences support combining both channels on this test population, but do not isolate a benefit from reliability gating. Generalization is setting-dependent: Flickr8K AUC is .859 for the full model, below FatFormer (.938) and GAPL (.996).

Ablation. Table 1 (bottom) compares the six variants under the common training and selection protocol. *Channels.* Semantic-only obtains AUC .938 / F1 .781 and camera-only .939 / .775, below the full model (.982 / .906). *Fusion strategy.* Concat-MLP reaches .977 / .890 and addition-MLP .978 / .888. Cross-attention with fixed equal channel weights reaches .983 / .908, slightly *above* the reliability-gated model (.982 / .906). The full and no-gate variants have matched initial shared weights; in this single-seed comparison, the learned gate has no demonstrated performance benefit.

Learned reliability weights. Fig. 3 (left) shows category-dependent gate distributions: the camera-side representation has higher weight for tampered images, whereas the semantic-side representation is favored for retouching and synthesis. These are descriptive diagnostics, not calibrated uncertainty or causal feature attributions. Each representation has already received information from the other

Table 1: **Four-class authenticity classification.** Overall and per-class performance on the common 3,200-image test set, plus held-out datasets (*). Per-class metrics report one-vs-rest AUC (OvR) and F1. Best in **bold**, second underlined (ranking computed over baselines and 2CAP only). The bottom block uses a common group-filtered training/validation protocol for all six variants and is excluded from the ranking; the full row uses paired, neutral gate initialization and baseline training populations differ.

Method	Overall				Camera		Tampered		Synthetic		Retouched		Flickr8K*	VISION*
	Prec.	Rec.	F1	AUC	OvR	F1	OvR	F1	OvR	F1	OvR	F1	AUC	Brier ↓
TruFor [8]	.100	.254	.123	.559	.618	.410	.724	.000	.497	.000	.396	.082	.637	.123
D3 [22]	.755	.754	.750	.928	.864	.591	.899	.680	.973	.863	.975	.868	.671	.179
CNNSpot [10]	.370	.361	.359	.631	.601	.349	.565	.316	.685	.430	.672	.344	.583	.237
AIDE [11]	.767	.770	.767	.934	.872	.638	.946	.777	.963	.835	.952	.816	.870	.228
VIBAIGCD [13]	.752	.740	.730	.925	.863	.523	.890	.673	.975	.871	.974	.851	.879	.253
FreqNet [21]	.530	.518	.484	.768	.663	.258	.791	.569	.886	.688	.731	.422	.840	.329
FatFormer [12]	.539	.547	.537	.783	.695	.422	.727	.452	.851	.621	.858	.653	.938	.159
UnivFD [9]	.380	.323	.269	.615	.576	.415	.559	.061	.600	.252	.727	.347	.818	.165
SAFE [43]	.591	.594	.583	.838	.776	.464	.891	.669	.888	.675	.797	.523	.937	.136
GAPL [44]	.455	.445	.387	.695	.796	.548	.637	.396	.727	.524	.620	.081	**.996**	**.095**
2CAP	**.908**	**.907**	**.906**	**.982**	**.957**	**.836**	**.987**	**.899**	**.987**	**.926**	**.998**	**.964**	.859	.394
Ablation (ours): channels														
Semantic only ($\mathcal{E}_{\text{sem}}$)	.779	.784	.781	.938	.875	.628	.915	.707	.973	.874	.990	.914	.850	.634
Camera only ($\mathcal{E}_{\text{cam}}$)	.782	.776	.775	.939	.905	.706	.960	.811	.960	.815	.933	.767	.944	.270
Ablation (ours): fusion strategy														
Concat + MLP	.890	.890	.890	.977	.947	.821	.982	.886	.985	.914	.994	.938	.914	.464
Addition + MLP	.888	.888	.888	.978	.948	.818	.983	.881	.986	.915	.995	.939	.928	.390
Cross-Attn (no rel. wts)	.908	.908	.908	.983	.958	.834	.987	.899	.989	.931	.998	.968	.914	.281

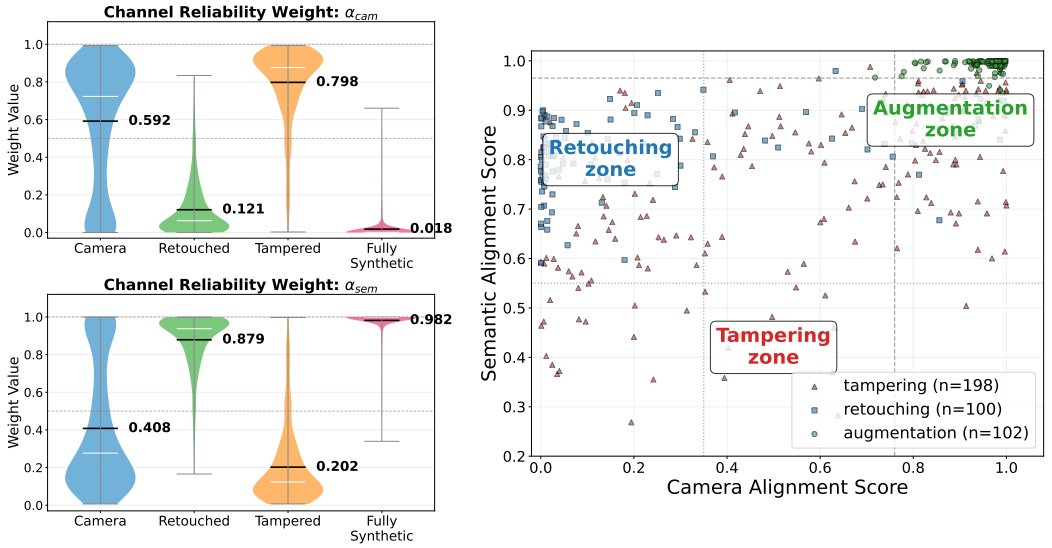

Figure 3: **Left: learned reliability weights** α_{cam} and α_{sem} shown by category as descriptive post-attention diagnostics. **Right: score distribution.** Retouching: low s_{cam}, high s_{sem}; camera: high/high; tampering: higher variance.

channel through cross-attention; gate magnitudes alone therefore do not establish which original forensic cue caused a prediction.

4.3 Evaluation on VLM Evidence Generation

This section evaluates the *second* application: *what is the forensic nature of the change?* We provide three qualitative examples of 2CAP (left–right) in Fig. 4. The examples illustrate how alignment

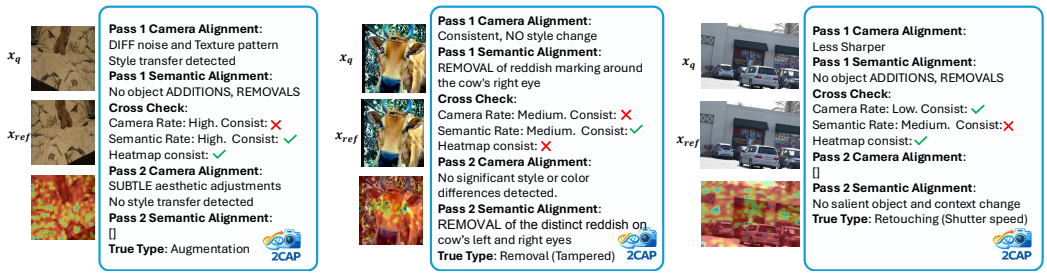

Figure 4: **2CAP evidence generation case study.** For each query-reference pair, 2CAP performs camera and semantic alignment (Pass 1), applies a reliability cross-check (camera-, semantic-, and heatmap consistency), and optionally uses Pass 2 to finalize the edit type, separating augmentation, tampered removal, and retouching.

cues and saliency inform the cross-check and subsequent revision. More qualitative examples can be found in Appendix E.1.

Channel score space as VLM context. Fig. 3 (right) visualizes the (s_{cam}, s_{sem}) score distribution that we supply as privileged context to the VLM. These scores are raw cosine similarities computed from the pretrained encoders alone, with no fusion module or learned weights involved, and so are independent of the classifier evaluated in Sec. 4.2. Retouching clusters tightly in the low-s_{cam}, high-s_{sem} region (camera traces destroyed, semantics preserved), while camera-captured images (including fingerprint-preserved augmentations) cluster at high values on both axes. Tampered images show higher within-class variance because localized edits range from minor inpainting (preserving most camera traces) to large-region replacement (disrupting both channels). The two-dimensional space therefore provides meaningful separation that single-score methods cannot achieve, motivating its use as a feedback signal in the OGR loop. Notably, this category-wise channel separation mirrors the pattern revealed by the trained reliability weights in Fig. 3 (left).

We evaluate whether dual-channel scores and saliency heatmaps improve VLM-generated explanations for answering "what changed." We compare VLM baselines with and without our privileged-context feedback. Recent VLM-based approaches (FakeVLM [26], SIDA [29], ForgeryGPT [28]) address a related but distinct task by fine-tuning VLMs on extensively annotated images with localization and explanations. Direct numerical comparison is therefore not straightforward. Instead, our method is plug-and-play and can be combined with these VLM models. To make a reliable comparison, we choose to leverage frozen frontier VLMs with and without our privileged-context feedback.

Setup and metrics. We evaluate 400 query–reference pairs: 102 benign camera-derived augmentations, 100 retouched pairs, and 198 tampered pairs. For each of four frozen VLMs—GPT-4o-mini, GPT-4o [45], Gemini 2.5 Flash, and Gemini 2.5 Pro [46]—the baseline receives the image pair without forensic feedback; +OGR adds dual-channel similarity scores and a saliency heatmap through the Observe–Generate–Refine loop. Both conditions use the same structured output schema. Table 2 reports change-type accuracy (ChgType), changed-object Recall, action-compatible object Hit, localization proxy IoU, policy-review accuracy/precision/recall, and revision frequency (Rev). Object metrics are evaluated on tampered pairs, and localization uses coarse annotation-derived targets rather than pixel-level masks. Rev measures how often the assessment changes, not the fraction of errors corrected. Metric definitions and policy rules are in Appendices C.5 and C.6.

Overall results. OGR improves ChgType, object Recall and Hit, and localization proxy IoU for all four models in Table 2. The largest ChgType gains occur for GPT-4o-mini (+6.0 percentage points) and Gemini 2.5 Flash (+7.7 points), showing that the feedback benefits the smaller Mini and Flash variants. GPT-4o and Gemini 2.5 Pro show smaller gains (+5.4 and +4.7 points) and lower revision rates (15.2% and 16.8%, versus 76.4% for Mini and 36.8% for Flash). This association is consistent with different refinement behavior across models. Given that the grounding annotation for IoU is a proxy, we performed a separate mask-derived-box diagnostic in Appendix D.6. Appendix D.9 reports the component-ablation study to evaluate the contribution of each component.

Policy review. We pass the structured VLM evidence to the same rule-based checker in both conditions. Under a predefined mock policy, the checker rejects biometric modifications or additions,

Table 2: **VLM evidence generation and downstream policy review.** Baseline versus +OGR on 400 pairs. All values are percentages. Forensic evidence and Rev report mean±standard deviation over three runs; policy columns retain separate point estimates. Bold marks the better result within each model; ↑ indicates higher is better. IoU is a localization proxy. Policy metrics use reject as the positive class. Rev is revision frequency and has no preferred direction.

		Forensic evidence				**Policy review**			
Model	**Config.**	ChgType ↑	Recall ↑	Hit ↑	IoU ↑	Acc. ↑	Prec. ↑	Recall ↑	Rev (%)
GPT-4o-mini	Baseline	66.1 ±0.5	21.3 ±0.9	32.5 ±1.1	25.6 ±0.6	95.5	**94.7**	51.4	–
	+OGR	**72.1** ±0.5	**29.3** ±0.6	**39.7** ±1.6	**36.1** ±1.5	**96.2**	83.3	**71.4**	76.4 ±0.3
GPT-4o	Baseline	84.4 ±0.3	35.6 ±0.7	59.4 ±1.2	60.9 ±3.7	97.8	86.1	88.6	–
	+OGR	**89.8** ±0.2	**36.4** ±0.2	**60.3** ±0.3	**61.3** ±0.3	**98.2**	**86.8**	**94.3**	15.2 ±0.8
Gemini 2.5 Flash	Baseline	71.0 ±1.0	41.8 ±2.1	64.0 ±2.3	57.4 ±1.4	92.8	56.8	71.4	–
	+OGR	**78.7** ±2.6	**46.8** ±0.5	**72.4** ±0.3	**70.0** ±1.7	**95.2**	**70.0**	**80.0**	36.8 ±1.1
Gemini 2.5 Pro	Baseline	73.1 ±0.8	54.0 ±0.2	77.6 ±1.2	73.5 ±0.8	93.5	58.5	88.6	–
	+OGR	**77.8** ±0.6	**55.3** ±0.3	**79.0** ±0.3	**76.1** ±0.0	**94.5**	**62.7**	**91.4**	16.8 ±1.7

removals, and replacements involving people or sensitive entities; other edits are accepted when neither rule triggers. It uses regular-expression and lexicon matching, with safe-phrase handling in the biometric branch, rather than assigning verdicts directly from channel scores. We compare pair-level decisions with 400 human-annotated verdicts (365 accept, 35 reject). Because an always-accept rule reaches 91.25% accuracy, Table 2 also reports precision and recall with reject as the positive class. OGR improves accuracy and rejection recall for every model. GPT-4o has the highest +OGR accuracy (98.2%) and rejection recall (94.3%). For GPT-4o-mini, rejection recall increases from 51.4% to 71.4%, while precision decreases from 94.7% to 83.3%; the other models improve both. These results concern a fixed mock policy and the combined evidence-and-checker pipeline, rather than general policy compliance. Appendix C.6 details the rules, input fields, and verdict aggregation.

5 Conclusion

We presented 2CAP, combining camera-related and semantic representations for reference-free authenticity classification and reference-based evidence generation. Channel ablations support combining both representations on the evaluated benchmark. The learned gate distributions vary by category, but do not establish causal attribution to the original channels; the matched single-seed comparison does not demonstrate a performance benefit from reliability gating. For reference-based assessment, dual-channel similarity scores and saliency maps guide frozen VLMs through OGR without forensic instruction tuning. Across four evaluated VLM configurations, full OGR improves change-type accuracy over the single-pass baseline, with model-dependent trade-offs in the component ablations. The mock-policy evaluation illustrates how structured evidence can support a downstream decision under explicit rules. These results motivate distinguishing *what changed* from *whether it matters under a given policy*.

References

[1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CVPR*, 2022.

[2] 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. *arXiv preprint arXiv:2307.01952*, 2023.

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. *CVPR*, 2023.

[4] Black Forest Labs. FLUX.1: State-of-the-art text-to-image generation. `https://blackforestlabs.ai`, 2024.

[5] Coalition for Content Provenance and Authenticity. C2PA Technical Specification. https://c2pa.org, 2024.

[6] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *CVPR*, 2019.

[7] Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao, and Xirong Li. MVSS-Net: Multi-view multi-scale supervised networks for image manipulation detection. In *IEEE TPAMI*, 2022.

[8] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva. TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *CVPR*, 2023.

[9] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, 2023.

[10] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. CNN-generated images are surprisingly easy to spot... for now. In *CVPR*, 2020.

[11] Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for AI-generated image detection. *arXiv preprint arXiv:2406.19435*, 2024.

[12] Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10770–10780, 2024.

[13] Haifeng Zhang, Qinghui He, Xiuli Bi, Weisheng Li, Bo Liu, and Bin Xiao. Towards universal AI-generated image detection by variational information bottleneck network. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23828–23837, 2025.

[14] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv preprint arXiv:2502.14786*, 2025.

[15] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-Refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2023.

[16] Jan Lukas, Jessica Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. *IEEE TIFS*, 1(2):205–214, 2006.

[17] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukáš. Determining image origin and integrity using sensor noise. *IEEE TIFS*, 3(1):74–90, 2008.

[18] Tiziano Bianchi and Alessandro Piva. Image forgery localization via block-grained analysis of JPEG artifacts. In *IEEE TIFS*, 2012.

[19] Irene Amerini, Rudy Becarelli, Roberto Caldelli, and Andrea Del Mastio. Splicing forgeries localization through the use of first digit features. In *IEEE WIFS*, 2014.

[20] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A CNN-based camera model fingerprint. *IEEE TIFS*, 2020.

[21] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5052–5060, 2024.

[22] Yongqi Yang, Zhihao Qian, Ye Zhu, Olga Russakovsky, and Yu Wu. D^3: scaling up deepfake detection by learning from discrepancy. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23850–23859, 2025.

[23] Fabrizio Guillaro, Giada Zingarini, Ben Usman, Avneesh Sud, Davide Cozzolino, and Luisa Verdoliva. A bias-free training paradigm for more general AI-generated image detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18685–18694, 2025.

[24] Baoying Chen, Jishen Zeng, Jianquan Yang, and Rui Yang. DRCT: Diffusion Reconstruction Contrastive Training towards Universal Detection of Diffusion Generated Images. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 7621–7639, 2024. URL `https://proceedings.mlr.press/v235/chen24ay.html`.

[25] Seunghyun Lee, Byoungkwon Kim, Jaehyun Nam, Kyungmin Lee, and Jinwoo Shin. SSAFE: Simple and Strong AI-Generated Image Detection via Frozen Vision Encoders. *arXiv preprint arXiv:2606.08634*, 2026. URL `https://arxiv.org/abs/2606.08634`.

[26] Siwei Wen, Junyan Ye, Peilin Feng, Hengrui Kang, Zichen Wen, Yize Chen, Jiang Wu, Wenjun Wu, Conghui He, and Weijia Li. Spot the fake: Large multimodal model-based synthetic image detection with artifact explanation. *arXiv preprint arXiv:2503.14905*, 2025.

[27] 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. *arXiv preprint arXiv:2410.02761*, 2024.

[28] Jiawei Liu, Fanrui Zhang, Jiaying Zhu, Esther Sun, Qiang Zhang, and Zheng-Jun Zha. ForgeryGPT: Multimodal large language model for explainable image forgery detection and localization. *arXiv preprint arXiv:2410.10238*, 2024.

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

[30] Huangsen Cao, Qin Mei, Zhiheng Li, Yuxi Li, Zhan Meng, Ying Zhang, Chen Li, Zhimeng Zhang, Xin Ding, Yongwei Wang, Jing Lyu, and Fei Wu. REVEAL: Reasoning-Enhanced Forensic Evidence Analysis for Explainable AI-Generated Image Detection. *arXiv preprint arXiv:2511.23158*, 2025. URL `https://arxiv.org/abs/2511.23158`.

[31] Rajarshi Roy, Nasrin Imanpour, Ashhar Aziz, Shashwat Bajpai, Gurpreet Singh, Shwetangshu Biswas, Kapil Wanaskar, Parth Patwa, Subhankar Ghosh, Shreyas Dixit, et al. A Comprehensive Dataset for Human vs. AI Generated Image Detection. *arXiv preprint arXiv:2601.00553*, 2026.

[32] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[33] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. PMLR, 2021.

[34] Liting Huang, Zhihao Zhang, Yiran Zhang, Xiyue Zhou, and Shoujin Wang. RU-AI: A large multimodal dataset for machine-generated content detection. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 733–736, 2025.

[35] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-SD: Trajectory segmented consistency model for efficient image synthesis. *Advances in neural information processing systems*, 37:117340–117362, 2024.

[36] Paschalis Giakoumoglou, Dimitrios Karageorgiou, Symeon Papadopoulos, and Panagiotis C Petrantonakis. SAGI: Semantically Aligned and Uncertainty Guided AI Image Inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16090–16101, 2025.

[37] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. MagicBrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.

[38] Yu Yuan, Xijun Wang, Yichen Sheng, Prateek Chennuri, Xingguang Zhang, and Stanley Chan. Generative Photography: Scene-Consistent Camera Control for Realistic Text-to-Image Synthesis. *CVPR*, 2025.

[39] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

[40] Thomas Gloe and Rainer Böhme. The Dresden image database for benchmarking digital image forensics. In *ACM SAC*, 2010.

[41] Dasara Shullani, Marco Fontani, Massimo Iuliani, Omar Al Shaya, and Alessandro Piva. VISION: A video and image dataset for source identification. *EURASIP Journal on Information Security*, 2017.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[43] Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Fuli Feng. Improving synthetic image detection towards generalization: An image transformation perspective. In *Proceedings of the 31st ACM SIGKDD conference on knowledge discovery and data mining v. 1*, pages 2405–2414, 2025.

[44] Ziheng Qin, Yuheng Ji, Renshuai Tao, Yuxuan Tian, Yuyang Liu, Yipu Wang, and Xiaolong Zheng. Scaling up AI-generated image detection with generator-aware prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 43008–43017, 2026.

[45] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[46] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[47] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. ImgEdit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025.

[48] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in GAN fake images. In *IEEE WIFS*, 2019.

[49] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *ICML*, 2020.

[50] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do GANs leave artificial fingerprints? In *IEEE MIPR*, 2019.

[51] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to GANs: Learning and analyzing GAN fingerprints. In *ICCV*, 2019.

[52] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. DIRE for diffusion-generated image detection. *ICCV*, 2023.

[53] Aparna Bharati, Richa Singh, Mayank Vatsa, and Kevin W Bowyer. Detecting facial retouching using supervised deep learning. *IEEE TIFS*, 11(9):1903–1913, 2016.

[54] Aparna Bharati, Mayank Vatsa, Richa Singh, Kevin W Bowyer, and Xin Tong. Demography-based facial retouching detection using subclass supervised sparse autoencoder. In *IEEE IJCB*, 2017.

[55] Qichao Ying, Jiaxin Liu, Sheng Li, Haisheng Xu, Zhenxing Qian, and Xinpeng Zhang. RetouchingFFHQ: A large-scale dataset for fine-grained face retouching detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 737–746, 2023.

[56] Fengchuang Xing, Xiaowen Shi, Yuan-Gen Wang, and Chunsheng Yang. FRRffusion: Unveiling Authenticity with Diffusion-Based Face Retouching Reversal. *arXiv preprint arXiv:2405.07582*, 2024.

[57] Nasir Ahmed, T. Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974.

[58] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[59] Christoph Zauner. Implementation and benchmarking of perceptual image hash functions. Master's thesis, Upper Austria University of Applied Sciences, Hagenberg Campus, 2010.

[60] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[61] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020.

Appendix

A Dataset Details

We curate a multi-source dataset comprising four categories: camera-captured images, fully synthetic images, tampered images, and GenAI-retouched images. This section provides full details on data composition, construction methodology, and the measures taken to ensure evaluation integrity.

To prevent data leakage, we enforce strict image-level disjointness: no image (camera-captured or edited) appearing in any evaluation split is used during either Stage A (contrastive pretraining) or Stage B (fusion training). For datasets that contribute to both training and evaluation (e.g., SID-Set [29], Defactify [31]), we use the official train/test splits. For held-out datasets (VISION [41], Flickr8K [34]), zero training images are drawn from these sources. We check exact-image duplication using image hashes. This check does not establish that an evaluation image's original photograph, crop, or other edited version is absent from training; the parent-image audit described in Appendix A.5 is an additional requirement.

A.1 Training Data for Contrastive Learning

We focus on datasets whose real images are verifiably camera-captured photographs, excluding common image-editing benchmarks (*e.g.*, ImgEdit [47]) that contain specialized content such as remote sensing or document images. Such non-photographic imagery would weaken the camera-fingerprint signal our encoder aims to learn.

Table 3 details the training data composition for contrastive learning.

Table 3: **Training data for contrastive learning.** Paired sources provide camera-edit correspondences; unpaired sources contribute additional camera and edited images. The total counts image entries ($2 \times 80{,}465 + 17{,}242 + 15{,}000 + 15{,}000$), not deduplicated unique images. The unique-image count requires a per-image manifest.

Source	**Type**	**Size**
MagicBrush [37] + SAGI-D [36]	Paired (camera-edit)	80,465 pairs
Dresden [40]	Unpaired camera	17,242
SID-Set [29]	Unpaired camera	15,000
SID-Set [29]	Unpaired edited	15,000
Total image entries		**208,172**

A.2 Evaluation Data Statistics

Table 4 summarizes the historical downstream training pool and source-level evaluation inventory; the controlled four-class split is described in Sec. 4.2.

A.3 Dataset Construction Details

Camera images. We aggregate authentic photographs from complementary sources. To capture classic device-forensics conditions, we include images from the Dresden Image Database (Dresden) [40], which is widely used for camera model identification, and VISION [41], which includes native captures as well as versions processed by online sharing pipelines. To better reflect modern content distributions encountered in real-vs-synthetic detection, we additionally use the authentic splits of Defactify [31] and SID-Set [29]. Paired camera originals from SAGI-D and MagicBrush additionally contribute to contrastive pretraining (Table 3); the source examples in Appendix C are illustrative rather than an exhaustive depiction of any one split.

Fully synthetic images. We define *full synthesis* as images generated entirely by GenAI from text prompts, with no camera capture involved. These images lack any authentic camera fingerprints and may exhibit subtle semantic inconsistencies (lighting, geometry, physics). We compile images from

Table 4: **Historical downstream training pool and source-level evaluation inventory.** The historical training pool contains 12,800 images (3,200 per class); the controlled fusion experiments instead use the filtered 9,877/2,451 training/validation split described in Sec. 4.2. The common four-class test split in Table 1 samples 800 images per class from the evaluation pools, for 3,200 images total; the inventory counts below describe the larger pools rather than the size of that test split. * indicates not seen in contrastive learning.

Dataset	Historical training pool	Evaluation inventory
SAGI-D [36]	2,240	3,838
Defactify* [31]	3,840	22,000
SID-Set [29]	640	16,000
MagicBrush [37]	2,240	0
Dresden [40]	640	0
Generative Photography* [38]	3,200	800
Flickr8K* (*held-out*) [34]	0	3,156
VISION* (*held-out*) [41]	0	6,000

diverse text-to-image generators: Defactify [31] (multiple popular models including SD2.1, SDXL, SD3, DALL-E 3, Midjourney 6) and Flickr8K-derived generations [34] (controlled text-conditioned synthesis). The absence of camera capture does not determine either learned channel response: synthesis can be semantically coherent, and learned camera-related features can respond to processing artifacts. In particular, reference-based cosine scores depend on the chosen query–reference pair and are not fixed by the synthesis label.

Tampered images. We define *tampering* as localized GenAI editing that modifies semantic content in specific image regions while potentially preserving camera traces elsewhere. This includes object insertion, removal, replacement, and inpainting operations. We include three complementary families: SID-Set [29] (social-media motivated edits), SAGI-D [36] (AI-based inpainting and region-level modifications), and MagicBrush [37] (instruction-guided editing triplets). Tampering can change semantic content locally while leaving camera-related cues elsewhere; small or coherent edits need not produce low global semantic similarity. Channel responses can overlap with those of other modification types.

GenAI-retouched images. We define *retouching* here as GenAI-enabled camera-parameter restyling intended to preserve scene content. We use the Generative Photography dataset [38], covering bokeh, color temperature, focal length, and shutter speed. The content-preservation criteria and limits of this dataset are discussed in Appendix A.4.

A.4 Retouching Dataset Scope

The retouching evaluation uses GP examples covering bokeh, color temperature, focal length, and shutter speed (motion blur). The intended operation changes photographic appearance while preserving scene content; this intention does not guarantee that every generated output meets the definition. Object identity, object presence, and scene layout must be checked independently of detector predictions. Photometric changes affecting foreground or background can be retouching; replacing foreground or background content cannot. Focal-length changes that reveal or remove content at crop boundaries require an explicit annotation policy.

The current experiments do not establish transfer to an independent GenAI retouching generator. Different effects or checkpoints within GP are not independent generator families. The classical OpenCV diagnostic is informative about processing sensitivity but does not close this gap. Broader portrait, computational-photography, and mixed-operation settings also remain outside the validated scope. Suitable external data would require genuine camera originals, documented generative editing provenance, and verified content preservation. Availability of a general image-editing model or a physically simulated retouching target set is not sufficient by itself.

A.5 Proposed Controls for Source and Processing Confounds

Status. The following controls are proposed additional analyses; no results from them are reported in this paper. They are designed to use existing images where provenance and original-edit links can be recovered. The within-source, OpenCV, and mild-processing results in Appendices D.1–D.3 are separate, already reported diagnostics.

Parent-image separation. Create a joint manifest for Stage A, Stage B, validation, and evaluation. For every image, record the source collection, original-photo identifier, derivative links, operation, tool/checkpoint, and split. Group each original with all crops, recompressions, and edited descendants before splitting. Check both provenance identifiers and exact/near-duplicate image matches across datasets; related source collections can contain the same original under different identifiers. All evaluation groups must be absent from training and model selection. If overlap is found, retrain from the earliest affected stage and rerun the affected comparisons. Report this task-specific audit separately from unknown foundation-model pretraining exposure.

Matched-original comparison. Where genuine camera originals for GP outputs are available, evaluate the original and its retouched version using the same frozen checkpoint, restricted to original-photo groups excluded from both training stages. Report camera recall, retouched recall, the four-way confusion matrix, and the paired change in retouched probability. Preserve all four output classes; do not silently discard predictions assigned to tampered or synthetic. This comparison reduces content/source differences between the two classes, but remains a within-GP evaluation. If original-camera provenance cannot be recovered, this control cannot be claimed.

Processing controls. Evaluate native files and a separately reported common-export condition. Apply the same specified resizing and encoding policy to originals and derivatives, and record original dimensions, codec, and export settings. Report the change in performance rather than treating normalization as guaranteed to preserve forensic traces. Use the existing mild-processing and OpenCV examples only after resolving parent overlap and documenting operation parameters and labels. Under a GenAI-specific definition, conventional processing is not positive evidence of GenAI provenance; examine retouched false alarms separately from all non-camera predictions. A source/format-only diagnostic can additionally test how predictive these nuisance variables are of the labels.

Leave-one-effect-out evaluation. If the existing GP samples have reliable effect labels, construct four folds: exclude bokeh, color temperature, focal length, or shutter-speed examples in turn from all task-specific training and validation, while maintaining parent-image separation. Retrain the affected components and test the excluded effect on unseen original-photo groups. Ensure that another derivative of a test original does not appear among the three training effects. Keep the remaining-class evaluation protocol fixed and report each fold separately, alongside a control trained with all four effects using a matched training budget. This tests generalization to a held-out photographic effect within GP; it is not a test of an unseen generator.

Comparison and uncertainty. Use the same evaluation groups for full 2CAP, semantic-only and camera-only variants, and applicable baselines. Report effect- and source-specific results, sample counts in original-photo groups as well as derivative images, and confidence intervals resampled at the original-photo-group level. Select thresholds and processing settings on validation data. Freeze these choices before testing, and distinguish training-seed variability from uncertainty over photographs.

Remaining external-data requirement. None of these controls removes the need for an independent GenAI retouching pipeline if claiming cross-generator transfer. Such a test requires camera-original/generated-retouch pairs with documented provenance and content preservation. An editor that replaces scene content, a set of conventional simulated targets, or a new checkpoint within the same family does not automatically meet that requirement. Until suitable data become available, the retouching claim remains limited to GP and the reported processing diagnostics.

A.6 Metric Choice Rationale

We select metrics to match the evaluation goal of each experiment:

Camera authenticity detection. We report AUC, precision, recall, and F1 for the four-class classification task in Table 1. Each image receives a single predicted label using four-way argmax, $\hat{y} = \arg\max_{k \in \{1,2,3,4\}} p_k$. Per-class AUC is computed one-vs-rest from the class scores; per-class precision, recall, and F1 use the argmax predictions. Overall AUC, precision, recall, and F1 are macro averages: the unweighted arithmetic mean of the corresponding four per-class values.

Cross-generator generalization. We report **AUC only**, as the goal is to assess ranking performance across diverse generators rather than to fix a specific operating point. Per-generator AUC isolates how well each method generalizes to individual generator families, while the **Average** column summarizes overall robustness.

Held-out camera-only datasets (VISION). For datasets containing only authentic camera images (a single class), we report the **Brier score**:

$$\text{Brier Score} = \frac{1}{N} \sum_{i=1}^{N} (p_i - y_i)^2,$$

where p_i is the predicted camera probability and $y_i = 1$ for every VISION image. This camera-only score measures squared error on authentic images, but cannot by itself establish calibration across authentic and manipulated classes. AUC requires both positive and negative classes and is therefore not applicable to single-class datasets.

VLM evidence generation (Sec. 4.3). We evaluate 400 query–reference pairs (198 tampered, 100 retouched, 102 augmented), comparing the visual-only baseline with full OGR and its component ablations. Table 2 reports the main evidence-generation and policy-review results. Table 14 extends the baseline/+OGR comparison with component ablations and reports change-type accuracy, object Recall and Hit, localization proxy IoU, and revision rate as means with run standard deviations over three runs for the component-ablation study. Detailed metric definitions appear in Appendix C.5.

B Extended Related Work

Several concurrent works address related problems. We discuss key differences and clarify 2CAP's positioning.

Classical forensic cues. Earlier image forensics focused on hand-crafted traces left by image formation and post-processing pipelines. Sensor-pattern-noise methods exploit device-specific PRNU fingerprints for source attribution and integrity analysis [16, 17], while JPEG-based analysis detects recompression inconsistencies and local block artifacts introduced by splicing or resaving [18]. Related statistical approaches also localize tampered regions from distributional irregularities in transform-domain statistics, such as first-digit features [19]. These approaches are physically grounded and often interpretable, but their effectiveness can degrade under resizing, heavy post-processing, or when manipulations do not leave stable low-level traces. In contrast, 2CAP retains access to low-level forensic evidence while embedding it into a learned reference-based comparison framework.

Learning-based image forgery detection and localization. Deep IFDL methods replace hand-crafted cues with end-to-end feature learning for tamper detection and localization. ManTra-Net [6] models local anomaly patterns across a wide range of manipulations, MVSS-Net [7] combines multiview and multiscale supervision for manipulation segmentation, and TruFor [8] integrates complementary clues including RGB content and forensic residuals to improve trustworthy localization, building on learned camera fingerprints such as Noiseprint [20]. These methods have substantially improved robustness over classical pipelines, but they remain primarily single-image predictors trained for fixed output spaces (labels or masks). 2CAP is complementary: rather than predicting authenticity from a single image alone, it explicitly compares a query against a reference and returns evidence in a metric space that supports verification, retrieval, and downstream reasoning.

Early AI-generated image detection. A parallel line of work studies artifacts specific to synthetic image generation. Early methods analyzed spectral and checkerboard artifacts in GAN outputs [48, 49], while others showed that generative models leave characteristic fingerprints that can support

both detection and attribution [50, 51]. CNNSpot [10] demonstrated that CNN-generated images were surprisingly easy to distinguish under in-domain conditions, helping establish cross-model generalization as a central evaluation criterion. This literature highlights the importance of low-level generator traces, but most methods still formulate the task as direct classification. By contrast, 2CAP uses low-level forensic cues as one component of a reference-based embedding, which better supports fine-grained comparison across images and manipulation settings.

Generalizable detection across generators. More recent work emphasizes robustness to unseen synthesis models and post-processing. UnivFD [9] shows that pretrained representations can provide strong transfer across diverse generators with minimal task-specific adaptation, while DIRE [52] exploits reconstruction discrepancies induced by diffusion inversion for diffusion-image detection. Frequency-oriented methods continue this trend by targeting generator-dependent spectral artifacts, including AIDE [11]. D3 [22] further scales multi-generator discrepancy learning with shallow detection heads over pretrained features. Relative to these approaches, 2CAP is less tied to a fixed discriminative decision rule: its contrastive objective and reference-conditioned inference allow the same representation to support verification, nearest-neighbor evidence retrieval, and compatibility with frozen VLM-based reasoning modules.

Specialized manipulation domains. Some related work focuses on narrower but practically important manipulation families, especially facial retouching. Earlier methods study supervised retouching detection from facial appearance changes [53, 54], and more recent datasets such as RetouchingFFHQ [55] enable fine-grained benchmarking at larger scale. Diffusion-based restoration or reversal has also been explored for authenticity analysis in this setting [56]. These methods are valuable for domain-specific benchmarks, but they are typically tailored to a particular manipulation family. 2CAP instead targets a broader verification setting spanning AI-generated imagery and image-level forensic comparison without assuming a single manipulation type.

Model provenance and metadata-based authenticity. Content provenance standards such as C2PA [5] provide a complementary route to authenticity by attaching cryptographically verifiable provenance metadata at creation or editing time. Such approaches are important for deployment ecosystems, but they depend on adoption across capture and editing pipelines and may be unavailable for legacy or redistributed content. 2CAP is orthogonal to provenance standards: it operates directly on image evidence and remains applicable when trusted metadata is absent, stripped, or incomplete.

Positioning of 2CAP. The literature spans (i) classical low-level forensic analysis, (ii) end-to-end image forgery detection and localization, (iii) generalizable AI-generated image detection, and (iv) fine-tuned VLM-based explainable detectors. 2CAP occupies a different point in this design space. We decompose GenAI manipulations along two complementary information channels: a *camera channel* that encodes sensor-level traces, and a *semantic channel* that encodes scene-level coherence. On top of this shared dual-channel foundation we build two applications. **(1) Authenticity classification:** a four-class classifier that distinguishes camera-captured, fully synthetic, tampered, and GenAI-retouched images via cross-attention fusion with learnable reliability weights, moving beyond the binary real/fake formulation that dominates prior work in (i)-(iii). **(2) VLM evidence generation:** dual-channel alignment scores and patch-level saliency are supplied to a *frozen* VLM through an Observe-Generate-Refine loop, eliminating forensic instruction tuning used by FakeVLM [26], SIDA [29], and ForgeryGPT [28] in (iv). The dual-channel decomposition is the unifying primitive that lets a single framework address both authenticity classification and interpretable evidence generation, while the frozen-VLM design avoids the annotation and retraining costs of supervised forensic VLM pipelines.

C Experimental Details

C.1 Visualization of Data

We visualize source-dataset examples from the four modification categories. These illustrations show the available source material and do not define the composition of the common test split.

Camera images. Figure 5 shows authentic camera-captured photographs from diverse sources (Dresden, VISION, Defactify, SAGI-D, MagicBrush). These span a wide range of scenes, lighting conditions, and camera models.



Figure 5: **Visualization of camera-captured images** from the source collections shown. For Dresden and VISION, we show images from diverse camera models (Agfa, Canon, Nikon, Samsung, Huawei, iPhone, Sony, etc). The source names identify the illustrated collections; split membership is specified separately in the evaluation protocol.

Fully synthetic images. Figure 6 shows text-to-image generations from multiple generators (SD2.1, SDXL, SD3, DALL-E 3, Midjourney 6). These images lack camera fingerprints entirely and may exhibit subtle semantic artifacts.



Figure 6: **Visualization of fully synthetic images** generated by different text-to-image models. These images contain no camera traces and may show subtle semantic inconsistencies in lighting, geometry, or physics.

Tampered images. Figure 7 shows localized GenAI edits (object insertion, removal, replacement, inpainting) from SAGI-D and MagicBrush. Each example is shown alongside its camera-captured reference to highlight the semantic modifications.



Figure 7: **Visualization of tampered images** with their camera-captured references. Tampered images contain localized semantic edits; their effects on semantic similarity and camera-related cues depend on the edit and can overlap with other categories.

GenAI-retouched images. Figure 8 shows global camera-parameter restyling from the Generative Photography dataset, covering bokeh adjustment, color temperature shift, focal length change, and shutter speed modification. Each example is paired with its original camera capture.

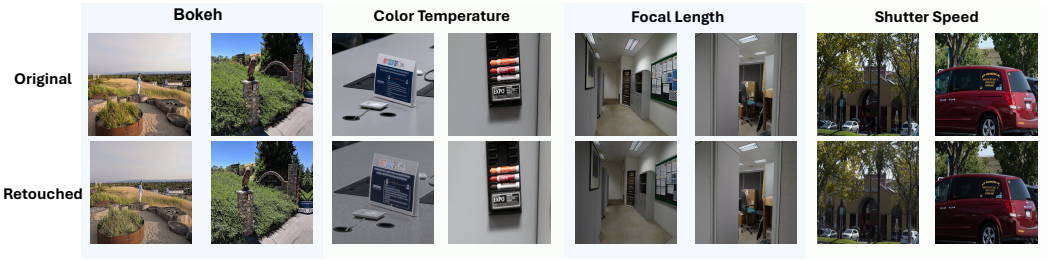

Figure 8: **Visualization of GenAI-retouched images** with their original camera captures. These examples show retouching with low camera-feature similarity and high semantic similarity; actual responses may overlap with tampering.

C.2 Model Architecture and Training Details

We provide complete implementation details for reproducibility.

Camera encoder $\mathcal{E}_{\text{cam}}$. We use a DCT-domain dual-branch pipeline with patch-level frequency analysis [57]. Images are split into 32×32 patches, ranked by DCT frequency energy, and selected patches are processed with 30 SRM steganalysis filters [11] to extract noise residuals. Two ResNet-50 backbones [42] separately encode high-frequency (textured) and low-frequency (smooth) patches; branch features are fused by element-wise addition and projected to 128-d unit-norm embeddings.

Semantic encoder $\mathcal{E}_{\text{sem}}$. We use frozen SigLIP2 [14] (ViT-base) to extract 768-d global semantic embeddings. For the saliency map, we extract 27×27 patch-level features and compute pairwise cosine similarities between query and reference patches.

Cross-attention fusion. The camera and semantic patch-token sequences retain their respective patch counts when projected to 256-d via learned linear layers (Sec. 3.3). Bidirectional 4-head cross-attention (2 layers) normalizes over key patches in each direction. After channel-wise pooling, a 2-layer gating MLP takes the concatenated vectors and produces two logits; a softmax over channels converts these into input-dependent weights that sum to one. A 2-layer classification head ($256\to128\to4$) outputs probabilities over camera-captured, retouched, tampered, and fully synthetic images. Table 5 lists the tensor shapes and normalization axes. The two patch counts need not be equal: each head forms an $N_{\text{cam}} \times N_{\text{sem}}$ attention matrix in one direction and its reverse shape in the other. Pooling is applied only after cross-attention, rather than reducing the inputs to singleton global tokens before attention.

Contrastive pretraining for $\mathcal{E}_{\text{cam}}$. We pretrain $\mathcal{E}_{\text{cam}}$ on 208,172 image entries (not a verified unique-image count) with InfoNCE loss (data composition in Table 3). Positives include mild augmentations (horizontal flip, mild color jitter ±0.1, JPEG quality ≥90) and, when available, same-camera images; negatives include paired edits and batch negatives. Training uses AdamW [58] optimizer (lr 3×10^{-4}, weight decay 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.999$) for 20 epochs on $4\times$ NVIDIA A100 80GB GPUs with batch size 512. Temperature-weighted sampling balances dataset sizes. Augmentations avoid blur and heavy JPEG recompression to limit processing changes; this does not guarantee preservation of sensor fingerprints.

Supervised fusion training. With both encoders frozen, we train only the fusion module (cross-attention layers + reliability MLP + classification head, $\sim$2M parameters) on the supervised four-class authenticity task. For the controlled comparison in Sec. 4.2, all six variants use the same 9,877 training and 2,451 validation images after source-reference overlap removal, deduplication, and group-based partitioning. Training uses AdamW (lr 10^{-3}) with a cosine annealing schedule, batch size 2048, and a 20-epoch budget. Validation macro-F1 selects each checkpoint before a single evaluation on the common 3,200-image test split (800 images per class). Cross-entropy loss with label smoothing (0.1) is used.

Table 5: **Cross-channel fusion architecture and tensor shapes.** B is batch size; N_{cam} and N_{sem} are channel-specific patch counts. Shapes include the batch axis. Attention uses four heads and two layers; the attention-weight shapes below are per head. The key-patch, channel, and class softmax operations normalize over different axes.

Stage	Tensor / shape	Operation and normalization
Patch tokens	$Z_{\text{cam}} : B \times N_{\text{cam}} \times 128$ $Z_{\text{sem}} : B \times N_{\text{sem}} \times 768$	Retain the patch sequence in each channel; no global pooling before attention.
Linear projections	$\widetilde{Z}_c : B \times N_c \times 256$	Project camera and semantic features independently to 256 dimensions.
Camera-query attention	$A_{\text{cam}} : B \times N_{\text{cam}} \times N_{\text{sem}}$ $Z'_{\text{cam}} : B \times N_{\text{cam}} \times 256$	Camera queries; semantic keys and values. Softmax over the N_{sem} keys for each query and head.
Semantic-query attention	$A_{\text{sem}} : B \times N_{\text{sem}} \times N_{\text{cam}}$ $Z'_{\text{sem}} : B \times N_{\text{sem}} \times 256$	Semantic queries; camera keys and values. Softmax over the N_{cam} keys for each query and head.
Channel pooling	$\mathbf{z}'_c : B \times 256$	Pool over each channel's patch axis after residual cross-attention.
Reliability gate	$[\mathbf{z}'_{\text{cam}}; \mathbf{z}'_{\text{sem}}] : B \times 512$ $\boldsymbol{\ell}, \boldsymbol{\alpha} : B \times 2$	Two-layer MLP produces two logits. Channel-axis softmax gives per-image weights with $\alpha_{\text{cam}} + \alpha_{\text{sem}} = 1$.
Weighted fusion	$\mathbf{z}_{\text{fused}} : B \times 256$	$\alpha_{\text{cam}}\mathbf{z}'_{\text{cam}} + \alpha_{\text{sem}}\mathbf{z}'_{\text{sem}}$.
Classification	$\mathbf{p} : B \times 4$	MLP $256 \to 128 \to 4$, followed by softmax over four classes; cross-entropy training.

C.3 Architectural Comparison with AIDE

Table 6 explicitly delineates the architectural differences between 2CAP's camera encoder and AIDE [11], which inspired the frequency-aware patch selection design.

Table 6: **Architectural comparison: 2CAP camera encoder vs. AIDE.**

Component	AIDE	2CAP
Patch selection	DCT energy ranking	DCT energy ranking (same)
Filter bank	30 SRM filters	30 SRM filters (same)
Backbone	Dual-branch ResNet-50	Dual-branch ResNet-50 (same)
Training objective	Cross-entropy (binary)	InfoNCE contrastive
Training data	Single-image labels	Paired camera-edit + unpaired
Embedding space	Class logits	128-d unit-norm (metric space)
Downstream use	Binary classifier	Cosine similarity s_{cam}
Semantic channel	CLIP + linear proj.	Frozen SigLIP2 + cross-attn
VLM integration	None	OGR privileged-context loop

Summary. The low-level feature extraction pipeline is shared with AIDE; 2CAP's novelty lies in (1) contrastive pretraining for a metric embedding space enabling reference-based verification, (2) dual-channel decomposition with cross-attention fusion, and (3) VLM evidence synthesis via OGR.

C.4 Inference Pipeline

Stage A denotes camera-encoder contrastive pretraining and Stage B denotes supervised fusion training; OGR is the subsequent frozen-VLM inference procedure for Q2. Algorithm 2 summarizes the inference procedure of 2CAP. For each channel $c \in \{\text{cam}, \text{sem}\}$, $z_c^v = \mathcal{E}_c^g(\mathbf{x}_v)$ denotes a global embedding, whereas $Z_c^v = \mathcal{E}_c^{\text{tok}}(\mathbf{x}_v)$ is a patch-token sequence, for $v \in \{q, r\}$. The camera global embedding is the 128-d unit-norm contrastive representation. The superscripts g and tok denote distinct outputs of the same encoder, not additional encoders. Q1 uses only query tokens: cross-

attention precedes pooling, gating, and four-class classification. Q2 compares global query/reference embeddings for cosine scores and semantic patch tokens for saliency; these signals guide the frozen VLM. Q1's four-class prediction and Q2's three-class change assessment are distinct outputs; the reference is required only for Q2.

Algorithm 2 2CAP Inference Pipeline

Require: Query image $\mathbf{x}_q$, reference image $\mathbf{x}_{\text{ref}}$
Ensure: Q1 four-class prediction, Q2 forensic explanation, saliency map

$z_c^v \leftarrow \mathcal{E}_c^g(\mathbf{x}_v)$ for $c \in \{\text{cam}, \text{sem}\}$, $v \in \{q, r\}$ *// Global embeddings for Q2*
$Z_c^q \leftarrow \mathcal{E}_c^{\text{tok}}(\mathbf{x}_q)$ for $c \in \{\text{cam}, \text{sem}\}$ *// Query patch tokens for Q1*
$Z_{\text{sem}}^r \leftarrow \mathcal{E}_{\text{sem}}^{\text{tok}}(\mathbf{x}_{\text{ref}})$ *// Reference patch tokens for Q2*
$s_{\text{cam}} \leftarrow \cos(z_{\text{cam}}^q, z_{\text{cam}}^r)$; $s_{\text{sem}} \leftarrow \cos(z_{\text{sem}}^q, z_{\text{sem}}^r)$
$\mathcal{M}_{\text{sem},i} \leftarrow 1 - \max_j \cos(Z_{\text{sem},i}^q, Z_{\text{sem},j}^r)$ *// Eq. 5*
$\mathbf{z}_{\text{fused}} \leftarrow \text{CrossAttnFusion}(Z_{\text{cam}}^q, Z_{\text{sem}}^q)$ *// Attention, pooling, gating; Q1 only*
$\mathbf{p} \leftarrow \text{softmax}(h_{\text{cls}}(z_{\text{fused}}))$; $\hat{y} \leftarrow \arg\max_k p_k$ *// Classification*
// OGR Loop:
obs $\leftarrow$ VLM.Observe$(\mathbf{x}_q, \mathbf{x}_{\text{ref}})$
draft $\leftarrow$ VLM.Generate(obs) *// Initial explanation*
output $\leftarrow$ VLM.Refine(draft, $s_{\text{cam}}, s_{\text{sem}}, \mathcal{M}_{\text{sem}}$) *// Score-guided refinement*
return $\hat{y}$, output, $\mathcal{M}_{\text{sem}}$

C.5 VLM Evaluation Setup Details

This section describes the VLM evaluation metrics used in Section 4.3. The VLM output examples and formats can refer to Appendix E.1.

C.5.1 Change Type Classification

We measure three-class accuracy over tampered, retouched, and augmented. Ground-truth addition/removal/replacement/multiple labels map to tampered, retouching to retouched, and augmentation to augmented.

C.5.2 Object Recall

Object recall measures how completely the VLM identifies changed objects described in the ground truth. We evaluate this metric on tampered pairs only. Specifically, our computation follows the following steps:

1. Extract GT object nouns from the annotation by tokenizing, stemming (plurals, -ing forms), and filtering stopwords, adjectives, colors, and positional words.
2. Concatenate all predicted `object_changes[].description` strings.
3. Match GT nouns to predictions using substring matching, stemming, and a synonym dictionary (e.g., man/woman $\rightarrow$ person, sofa $\rightarrow$ couch).
4. Compute: ObjRecall $=$ (# GT nouns found)/(# total GT nouns).

We report the averaged Recall across all tampered pairs.

C.5.3 Object Hit

It is a stricter object detection metric requiring both *sufficient noun overlap* and *action type agreement*. Evaluated on tampered pairs only.

Requirements for a hit:

1. *Noun threshold*: At least $\min(2, |\text{GT nouns}|)$ GT nouns must appear in predictions.
2. *Action match*: The predicted action type must be compatible with the GT action type.

Action compatibility rules:

- Same action (addition/removal/replacement) → match.
- Replacement is compatible with both addition and removal.
- If either action is undetectable → match (no penalty).
- Addition vs. removal (without replacement) → no match.

C.5.4 Localization Proxy

Localization is evaluated against proxy GT boxes parsed from free-text annotations on tampered pairs with parseable locations. These coarse spatial targets are separate from mask-derived boxes and pixel-level masks.

Location parsing. Free-text location descriptions (e.g., "bottom-right", "left side") are mapped to approximate bboxes on a 0-1000 coordinate system:

- Horizontal: `left` → [0, 500], `right` → [500, 1000], `center` → [250, 750].
- Vertical: `top/upper` → [0, 500], `bottom/lower/foreground` → [500, 1000].
- Corners are tightened to one-third regions (e.g., "top-left corner" → [0, 0, 333, 333]).

Object-relative descriptions (e.g., "on the dog's face") and "entire image" are skipped.

C.6 Policy Review under a Mock Policy

Task and evaluation unit. Policy review converts structured forensic evidence into an accept/reject decision under a predefined mock content policy. The evaluation unit is a query–reference image pair, not an individual object or bounding box. We evaluate 400 pairs with human-annotated policy verdicts: 365 accept and 35 reject. These verdicts are distinct from the three modification-type labels: an edit can be correctly identified as tampering yet remain acceptable under the mock policy if it does not violate a protected-content rule. The checker operates on the VLM's structured description of the change, rather than directly assigning a policy verdict from the channel scores.

Mock policy. The policy specifies two rejection criteria and permits other edits when neither criterion is triggered:

- **Biometric modification (reject):** Alterations to physical identity markers, including skin tone, eye color, hair texture, facial features, or facial geometry.
- **Sensitive entity modification (reject):** Addition, removal, or replacement of people or sensitive entities, including weapons, illegal substances, and hate symbols.
- **Permitted edits (accept):** Aesthetic adjustments (such as retouching, compression, and camera-parameter restyling), changes to animals or ordinary inanimate objects, and background modifications, provided they do not satisfy either rejection criterion.

These categories define a task-specific mock policy, rather than a universal definition of acceptable photographic editing.

Baseline and OGR inputs to the checker. For each VLM, the baseline supplies the structured output from its image-pair assessment. The +OGR condition supplies the assessment after the Observe–Generate–Refine procedure. The policy checker uses the same rule set for both conditions. It reads the `identity_biometric` field and a list of `object_changes[]` entries. Each object entry contains a `change_type` and a textual `description`. For +OGR, the checker uses the revised object-change list in `revised_semantic_alignment` when present; otherwise it retains the original list. The biometric field is unchanged between passes. Thus, differences between the two conditions arise through the evidence supplied to the checker, particularly revisions to the described object changes, rather than a different policy.

Rule application and verdict aggregation. The checker applies regular-expression matching in two branches:

1. **Biometric branch.** Search `identity_biometric` for change phrases such as "facial geometry changed", "identity swap", and "face morph". Safe phrases such as "no identity manipulation

detected" and "facial geometry is consistent" override the biometric trigger. These are examples of the documented patterns, not an exhaustive pattern list.

2. **Sensitive-entity branch.** Examine object entries whose `change_type` is addition, removal, or replacement. Match their descriptions against a person lexicon (29 terms, including person, man, woman, child, and pedestrian) and a sensitive-entity lexicon (14 terms, including weapon, gun, knife, drug, and hate symbol). A matching eligible entry triggers rejection. Mere mention of an entity in an entry with an ineligible change type does not trigger this branch.
3. **Pair-level verdict.** Reject the pair if either branch triggers; otherwise accept it. Multiple flagged objects still produce one pair-level rejection. A safe phrase in the biometric branch does not cancel a sensitive-entity violation.

Writing b_i for a biometric trigger after safe-phrase handling and s_{ij} for an eligible sensitive-entity trigger in object entry j, the decision is

$$\widehat{y}_i = \mathbb{1}\left[b_i \vee \bigvee_j s_{ij} \right], \qquad 1 = \text{reject}, \quad 0 = \text{accept}.$$

For illustration, an entry reporting the removal of a person triggers the sensitive-entity branch, whereas a background color adjustment without a protected-content change is accepted. A reported face morph triggers the biometric branch even if no object is added or removed. These examples illustrate the decision rules and are not additional experimental observations.

Scoring against human verdicts. Each predicted verdict is compared with the human accept/reject label, with reject as the positive class. A true positive is a correctly rejected pair; a false positive is an acceptable pair rejected by the checker; a false negative is a policy-violating pair accepted by the checker; and a true negative is a correctly accepted pair. The pair-level metrics are

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}, \quad \text{Precision} = \frac{TP}{TP+FP},$$
$$\text{Recall} = \frac{TP}{TP+FN}, \qquad \text{FPR} = \frac{FP}{FP+TN}.$$

Table 2 reports accuracy, precision, and recall, multiplied by 100. Precision measures the reliability of a rejection, while recall measures coverage of policy-violating pairs. This evaluation assesses the combined VLM-description and rule-checker pipeline: an incorrect final verdict may arise from a missing or hallucinated change description, or from the keyword rules themselves.

Results and scope. Policy results and their interpretation appear in Table 2 and Sec. 4.3. The policy columns retain the reported point estimates from the policy evaluation, whereas the forensic-evidence and revision-frequency columns use the three-run aggregates also reported in Appendix D.9. Run-level uncertainty estimates are not available for the policy results. Missed or hallucinated entities in the VLM output and limitations of the rule-based checker can both affect the final verdict.

D Additional Experiments

D.1 Within-Source Classification Diagnostics

Table 7 reports two comparisons restricted to a named source collection. This restriction removes variation in the collection name within each comparison, but does not necessarily match underlying original-photo sources, compression, resolution, or generator signatures across classes. The SID-Set AUC indicates ranking discrimination within that collection. The Defactify accuracy additionally depends on class proportions and the decision rule. These metrics therefore cannot be averaged or interpreted as proof that source confounding is absent. Neither comparison includes the GP retouching class.

D.2 Cross-Pipeline Retouching Diagnostic

The OpenCV data are used only for evaluation and are mapped exclusively to the GP evaluation subset; they are not used in either training stage. Table 8 reports the predicted-class distributions for

Table 7: **Within-source diagnostics.** Values are retained in the metric originally reported for each comparison. These are not matched-original or independent-generator evaluations.

Collection	Comparison	Metric	Result
SID-Set	Camera vs. tampered	AUC	95.5%
Defactify	Camera vs. fully synthetic	Accuracy	90.4%

GP effects and corresponding effects recreated using classical OpenCV operations on the associated evaluation source images. This establishes the intended evaluation split; auditing underlying parent identities across other source collections is a separate check. The OpenCV outputs are assigned the retouched label in 81.2-92.7% of cases. This is evidence of sensitivity to related visual processing beyond the GP implementation; it is not a test of attribution to GenAI, because the OpenCV operations are non-neural. It also does not replace evaluation on an entirely unseen GenAI retouching tool. Under the GenAI-specific retouching definition, a conventional operation alone does not establish membership in the GenAI-retouched class, regardless of its visual strength. The predictions therefore expose a potential provenance ambiguity. Operation severity and the policy for conventional edits must be specified before this diagnostic can be scored as a classification task.

Table 8: **Cross-pipeline diagnostic.** Row-normalized predicted-class percentages. "Retouched" is a model prediction, not proof of GenAI processing or an established correctness label for OpenCV operations.

Pipeline	Effect	Camera	Retouched	Tampered	Synthetic
GP	Color temperature	1.0	97.0	2.0	0.0
GP	Bokeh	1.5	94.5	4.0	0.0
GP	Focal length	5.5	84.5	10.0	0.0
GP	Shutter speed	1.0	97.0	1.5	0.5
OpenCV	Color temperature	3.2	90.6	5.4	0.8
OpenCV	Bokeh	2.7	81.2	15.3	0.8
OpenCV	Focal length	4.4	90.0	5.4	0.2
OpenCV	Shutter speed	0.3	92.7	6.7	0.3

D.3 Mild Benign-Processing Diagnostic

Table 9 reports a limited diagnostic under mild settings. Camera recall decreases by at most 4.0 points relative to the unprocessed images. However, the unprocessed camera recall is only 62.5%, corresponding to a 37.5% non-camera prediction rate before any transformation; denoising raises that rate to 41.5%. Small relative changes therefore do not establish a low false-positive rate. This diagnostic does not cover operation severity, real smartphone capture pipelines, or compound processing.

Table 9: **Mild benign-processing diagnostic.** Predicted-class percentages on camera-derived images. All rows should retain the camera label under this diagnostic's policy.

Transformation	Camera	Tampered	Synthetic	Retouched
No processing	62.5	26.5	9.5	1.5
Resize	60.5	29.0	9.0	1.5
Denoise	58.5	29.5	10.0	2.0
HDR tonemap	59.5	27.5	11.0	2.0
Social recompress	60.5	27.0	11.0	1.5
Computational photo	62.0	26.0	10.5	1.5

D.4 Per-Generator Classification Results

Table 10 provides a per-generator breakdown of camera-vs-edited AUC across the Defactify (in-distribution) and Flickr8K* (held-out) datasets, complementing the dataset-level summary reported

elsewhere. 2CAP achieves the best AUC on every Defactify generator (.892-.923), showing strong within-source performance across the listed diffusion families (SD2.1/SDXL/SD3), DALL-E 3, and Midjourney rather than overfitting to any one generator. On the held-out Flickr8K* set, FatFormer leads on all five generators while 2CAP remains competitive (.772-.929), and the two methods are essentially tied on the ten-column average (2CAP: .893; FatFormer: .894).

Table 10: **Camera authenticity across generators (AUC).** Defactify and Flickr8K*, each with five generators. This breakdown covers eight baselines; SAFE and GAPL are included only in the main four-class comparison. *Flickr8K** is held-out. Best in **bold**, second underlined.

Method / AUC	**Defactify Images**					**Flickr8K***					**Avg.**
	SD2.1	**SDXL**	**SD3**	**DALL-E 3**	**Midjourney 6**	**SDAbso.**	**SDEpic.**	**SDv1.5**	**SDv6.0**	**SDXL**	
TruFor [8]	.455	.407	.382	.452	.516	.656	.668	.660	.550	.689	.544
D3 [22]	.613	.715	.618	.778	.762	.723	.693	.668	.643	.635	.685
CNNSpot [10]	.561	.547	.542	.606	.560	.625	.574	.598	.550	.533	.570
AIDE [11]	.626	.701	.633	.567	.862	.929	.905	.895	.835	.796	.775
VIBAIGCD [13]	.757	.808	.740	.844	.820	.932	.924	.922	.903	.698	.835
FreqNet [21]	.541	.584	.522	.544	.579	.851	.868	.837	.843	.823	.699
FatFormer [12]	.836	.860	.812	.895	.839	**.967**	**.962**	**.961**	**.958**	**.850**	**.894**
UnivFD [9]	.751	.781	.752	.852	.805	.876	.861	.852	.845	.676	.805
2CAP	**.895**	**.899**	**.892**	**.923**	**.911**	.929	.905	.910	.893	.772	.893

D.5 Matched-source Four-class Diagnostic

We additionally test whether four-way predictions remain reliable when all four labels are derived from the same source scene. We select 100 original-retouched pairs from the existing Generative Photography test-source pool, with 25 pairs per camera setting (bokeh, color temperature, focal length, and shutter speed). The camera and retouched images are the existing paired files. For each camera original, Big-LaMa removes a masked object to form a tampered candidate, with pixels outside the mask copied from the original. Separately, FLUX.1-dev generates a fully synthetic image from the original's caption alone, without image conditioning. This produces 100 source groups and 400 images (100 per class).

We evaluate frozen Stage A and Stage B checkpoints without retraining or tuning on these images. Camera and SigLIP2 patch-token embeddings are extracted once and supplied to the full model and its camera-only and semantic-only ablations. Each model receives one image at a time, without access to the group identity. Table 11 reports four-way argmax accuracy, macro-F1, macro one-vs-rest AUC, and class recalls. The full model attains 46.25% accuracy, compared with 44.50% for semantic-only. Its 1.75-point accuracy difference has a paired 95% bootstrap interval of $[-0.75, 4.50]$ points when resampling the 100 original-image groups, and therefore does not establish a reliable gain over the semantic channel alone. In particular, the full model identifies only 4% of camera images and 8% of tampered candidates.

Table 11: **Matched-source four-class diagnostic.** All values are percentages.

Frozen model	**Acc.**	**Macro-F1**	**Macro-AUC**
Full 2CAP	46.25	39.14	71.99
Camera only	28.75	23.88	64.33
Semantic only	44.50	36.51	66.42

The full model's recalls are 4%, 99%, 8%, and 74% for camera, retouched, tampered, and fully synthetic images, respectively; the corresponding per-class one-vs-rest AUCs are 0.450, 0.848, 0.582, and 0.999. These ranking scores should not obscure the poor camera and tampered argmax recalls. The cohort reuses the paper's existing test-source pool, rather than providing a new independent model-selection holdout. Exact source-path and pixel-change checks passed, but visual review of all Big-LaMa removals and scene consistency is incomplete; some spot-checked removals are small or visibly blurred. We therefore treat this as a failure-oriented diagnostic, not evidence that matched-source confounding has been resolved. The group manifest, model-specific generation records, predictions, and inference logs are retained with the dataset for auditing.

D.6 Validation of VLM Evaluation with Annotated Bounding Box

In Sec. 4 of the main paper, the VLM evidence evaluation uses location text as a proxy ground truth for localization quality (IoU) as not all the datasets have bounding-box annotation. To validate this proxy, we manually annotate bounding boxes for a subset of tampered image pairs from SAGI-D (those for which the VLM actually produces a bounding box), using the ground-truth masks provided by the dataset. We then compare the VLM-generated bounding boxes against both the human-annotated boxes and the saliency-derived proxy boxes.

Table 12: **Proxy GT validation.** Agreement between VLM-derived bounding boxes and human-annotated bounding boxes on 50 tampered pairs.

	Baseline		Ours	
Metric	Mean	Std	Mean	Std
IoU	.500	.241	.597	.228
Precision	.722	.244	.823	.203
Recall	.638	.297	.690	.254

Table 12 reports higher agreement with mask-derived human boxes for OGR on the selected 50-pair subset (IoU .500 to .597). This supports localization on that subset, but does not directly validate the text-derived proxy: doing so requires comparing proxy boxes themselves with mask-derived boxes. Selecting pairs on which a VLM outputs a box can also exclude localization failures. The subset selection and missing-box handling therefore limit the conclusion.

D.7 Design-Space Comparison with VLM-Based Forensic Methods

Table 13 provides a systematic design-space comparison with VLM-based forensic methods, clarifying fundamental architectural and formulation differences.

Scope of a direct comparison. Forensic VLMs differ in reference access, training data, and output format. These differences require explicit reporting; they do not make practical comparison infeasible. Our current experiments measure improvements to the same frozen VLM under pairwise inputs, rather than competitiveness against fine-tuned forensic systems. A complementary comparison could evaluate a released forensic VLM on a shared camera/tampered subset with mask-derived boxes, showing its native single-image results separately from any pair-input adaptation and reporting the reference advantage of OGR.

Table 13: **Design-space comparison with VLM-based forensic methods.** 2CAP is the only method that (i) uses reference-based verification, (ii) decomposes camera and semantic channels, (iii) distinguishes retouching from tampering, and (iv) uses the tested VLMs without forensic instruction tuning.

Property	**FakeVLM**	**FakeShield**	**ForgeryGPT**	**SIDA**	**2CAP**
Input type	Single	Single	Single	Single	**Pair**
VLM training	SFT	SFT	Staged tuning	SFT	**Frozen**
Channel decomp.	×	×	×	×	✓
Retouch vs. tamp.	×	×	×	×	✓
Forgery localization	×	Mask	Mask	Mask	**Saliency**
Output type	Binary+text	Binary+mask	Desc.+mask	Label+mask+desc.	**Scores+sal.+OGR**
VLM portability	Fixed	Fixed	Fixed	Fixed	**Tested VLMs**

Key advantage. The OGR interface uses frozen VLMs without forensic instruction tuning, while the camera encoder and fusion classifier are trained. The experiments demonstrate improvements across four tested proprietary configurations from the GPT-4o and Gemini 2.5 families. Extension to other backends is plausible but requires evaluation, and the current results do not establish a performance ranking against fine-tuned forensic VLMs.

D.8 Discussion of Reference-Based Baselines

We consider the following reference-based similarity alternatives for change assessment (Sec. 4.3):

- **C2PA/Content Credentials:** Cryptographic provenance, not learned forensic analysis. Requires adoption at capture time and does not work for legacy content.
- **Perceptual hashing (pHash) [59]:** Detects near-duplicates but cannot characterize *what* changed or distinguish retouching from tampering.
- **Image-pair similarity (LPIPS [60], DISTS [61]):** Measures perceptual distance but provides no forensic decomposition into change type, affected objects, or spatial localization.

A quantitative comparison with these reference-based alternatives requires the same evaluation pairs, a specified multi-class decision rule, and thresholds selected on a separate validation split. We leave this matched comparison to further evaluation.

D.9 OGR Component Ablation

Table 14 evaluates the contribution of each forensic input over three runs on the 400-pair evaluation set. Each ablation retains OGR while removing the semantic score, camera score, or saliency heatmap. The baseline and full-OGR means and standard deviations are the same as those in the forensic-evidence and revision-frequency columns of Table 2. The ablation rows remove one input at a time under the same reporting protocol. Policy-review metrics are not included in this ablation.

Table 14: **VLM evidence generation and component ablation.** All values are percentages; smaller gray values denote ±standard deviation over three runs. Shaded rows: full OGR; indented rows: removal of one OGR input. Bold marks the highest mean per model (including ties). ↑: higher is better. IoU is a localization proxy; revision frequency (Rev) has no preferred direction.

Configuration	ChgType ↑	Recall ↑	Hit ↑	IoU ↑	Rev (%)
GPT-4o-mini					
Baseline	66.1 ±0.5	21.3 ±0.9	32.5 ±1.1	25.6 ±0.6	–
Ours (+OGR)	72.1 ±0.5	**29.3** ±0.6	**39.7** ±1.6	**36.1** ±1.5	76.4 ±0.3
w/o semantic	67.2 ±1.6	27.7 ±0.5	38.2 ±1.1	32.8 ±0.0	71.3 ±2.8
w/o camera	69.8 ±0.6	28.4 ±0.6	38.0 ±1.6	35.4 ±0.3	71.5 ±0.0
w/o heatmap	**72.7** ±0.6	25.5 ±0.6	35.5 ±0.8	34.3 ±1.2	69.2 ±0.3
GPT-4o					
Baseline	84.4 ±0.3	35.6 ±0.7	59.4 ±1.2	60.9 ±3.7	–
Ours (+OGR)	**89.8** ±0.2	36.4 ±0.2	60.3 ±0.3	61.3 ±0.3	15.2 ±0.8
w/o semantic	85.7 ±0.1	**36.5** ±0.1	**60.6** ±0.5	**61.5** ±0.3	11.7 ±0.3
w/o camera	87.0 ±0.7	36.0 ±0.1	60.1 ±0.0	61.1 ±0.0	5.1 ±0.9
w/o heatmap	87.4 ±0.1	36.0 ±0.1	60.3 ±0.3	61.1 ±0.0	22.8 ±0.4
Gemini 2.5 Flash					
Baseline	71.0 ±1.0	41.8 ±2.1	64.0 ±2.3	57.4 ±1.4	–
Ours (+OGR)	**78.7** ±2.6	**46.8** ±0.5	**72.4** ±0.3	**70.0** ±1.7	36.8 ±1.1
w/o semantic	77.6 ±0.8	46.4 ±0.3	72.1 ±0.3	69.3 ±1.5	39.2 ±2.0
w/o camera	74.4 ±0.8	45.0 ±0.7	70.6 ±0.3	64.1 ±0.8	13.6 ±0.6
w/o heatmap	75.2 ±1.0	44.9 ±0.2	70.5 ±0.3	62.0 ±0.3	29.2 ±1.1
Gemini 2.5 Pro					
Baseline	73.1 ±0.8	54.0 ±0.2	77.6 ±1.2	73.5 ±0.8	–
Ours (+OGR)	**77.8** ±0.6	55.3 ±0.3	**79.0** ±0.3	**76.1** ±0.0	16.8 ±1.7
w/o semantic	77.4 ±0.8	55.3 ±0.5	78.3 ±1.0	75.4 ±0.8	20.5 ±0.5
w/o camera	76.2 ±0.2	**55.5** ±0.4	78.1 ±0.3	75.2 ±0.6	10.9 ±0.8
w/o heatmap	76.2 ±0.2	54.8 ±0.1	78.8 ±0.5	75.0 ±0.0	12.0 ±0.2

Contributions of the forensic inputs. Removing either alignment score lowers mean ChgType accuracy for every model. The semantic score has the larger effect for GPT-4o-mini and GPT-4o, whereas Gemini-Flash and -Pro show the opposite ordering. Thus, the relative utility of the two signals depends on the VLM. The heatmap's contribution is most visible in spatial and object-level metrics for Gemini-Flash and GPT-4o-mini. Full OGR is not uniformly best across all ablations, and the differences vary by model and metric.

E Additional Qualitative Results

E.1 Extended Qualitative Examples

Figures 9-13 present additional qualitative results beyond the main paper, covering benign augmentation, retouching, and localized tampering. Each example shows the query-reference pair, the two-pass OGR pipeline output (Pass 1 visual-only assessment, cross-check, and Pass 2 revision when triggered), and the final forensic conclusion. These examples illustrate the outputs of the Observe–Generate–Refine loop, describing *what* changed, *where*, and *how*. We include both clear-cut cases where the dual-channel signatures are distinctive and borderline cases where scores are ambiguous, illustrating the range of 2CAP's behavior across diverse editing scenarios.

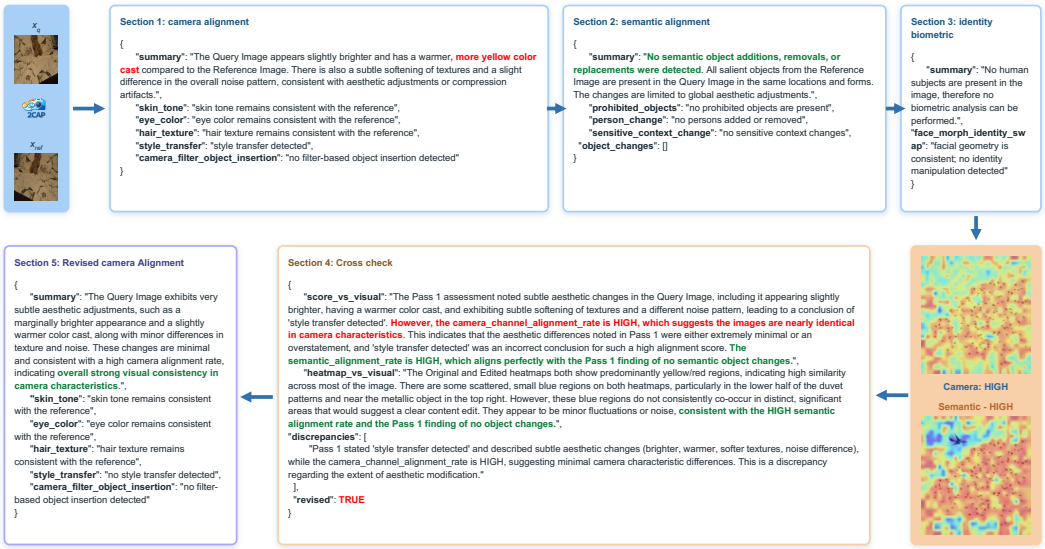

Figure 9: **OGR example: benign augmentation.** The refinement retracts the initial style-transfer assessment and reports no object changes. The pair illustrates a camera-derived augmentation rather than localized tampering.

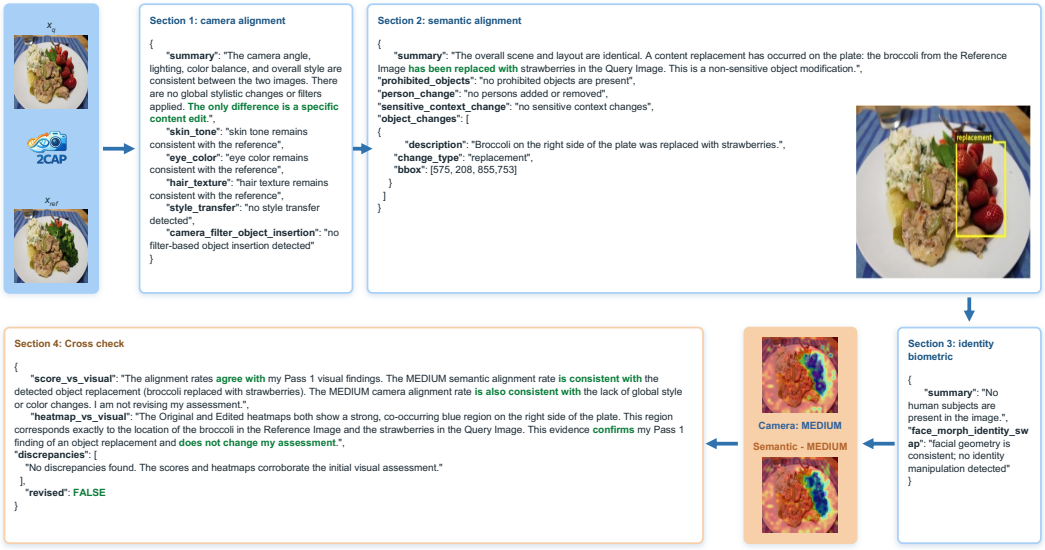

Figure 10: **OGR example: localized tampering by replacement.** Broccoli on the right side of the plate is replaced with strawberries. The object-change description identifies a replacement rather than a global retouch.



Figure 11: **OGR example: retouching by style transfer.** A global watercolor/sketch-like aesthetic filter changes appearance while the displayed output reports no object changes. This is a paired retouching example, not text-to-image full synthesis.

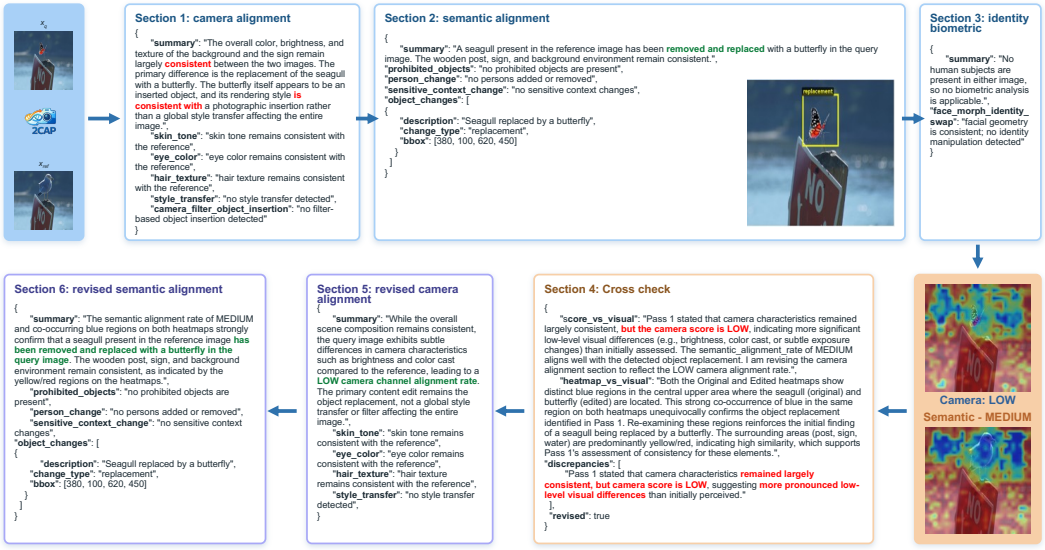

Figure 12: **OGR example: localized tampering by replacement.** A seagull is replaced with a butterfly. The paired reference makes the changed object and operation explicit.



Figure 13: **OGR example: localized tampering by addition.** A boat is added to the water. The displayed cross-check reports high camera alignment, illustrating that localized tampering can coexist with preserved camera-related cues.

E.2 Failure Case Analysis

Understanding failure modes is critical for assessing deployment readiness. We present three representative failure cases in Figure 14, each selected because it represents a genuinely difficult sample that challenges not only 2CAP but also human annotators. These edge cases reveal the boundaries of the current dual-channel framework and inform directions for future improvement.

Case 1: Retouching with hallucinated object change (left). The query image is a focal-length retouching of the reference: a global camera-parameter edit that preserves all semantic content. This

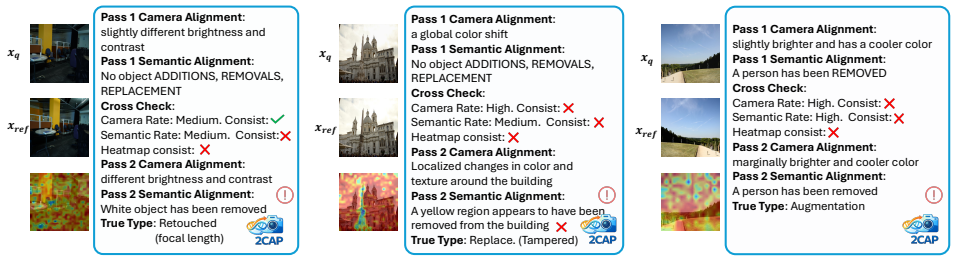

Figure 14: **Failure cases of 2CAP.** Three challenging examples where the two-pass pipeline produces incorrect conclusions. Left: retouching misclassified due to VLM hallucination under ambiguous saliency. Middle: a building-region replacement missed in Pass 1, with Pass 2 incorrectly describing the edit as a removal. Right: augmentation misclassified as tampering due to VLM hallucinating a person removal.

is an inherently difficult sample because the focal-length change introduces perspective cropping that creates apparent object disappearances at image borders. Pass 1 correctly identifies brightness and contrast differences with no object changes. However, the cross-check flags heatmap inconsistency, and Pass 2 hallucinates that a "white object has been removed," likely triggered by border-region saliency artifacts from the perspective shift. The result is a misclassification from retouched to tampered.

Case 2: Subtle building-region replacement with an imprecise description (middle). The query contains a localized replacement affecting a region of the building. Pass 1 describes a global color shift without identifying the localized edit. After the cross-check, Pass 2 detects a change around the building but describes it as a removal rather than a replacement.

Case 3: Augmentation with hallucinated person removal (right). The query is an authentic augmentation (camera-parameter restyling) with no semantic changes, yet Pass 1 confidently reports "a person has been REMOVED." The cross-check flags inconsistencies, but Pass 2 reinforces the hallucination rather than correcting it. This failure arises because the VLM's instruction prompt implicitly biases toward finding edits; when saliency maps show diffuse activity (common in augmentations with global brightness/color shifts), the VLM may anchor on spurious visual differences and fabricate a plausible edit narrative.

Discussion and mitigations. These failure cases share a common theme: they are *challenging* samples that lie near the decision boundaries between modification categories. Two key observations emerge:

1. **VLM hallucination under weak saliency** (Cases 1 and 3): When both channels produce ambiguous scores, saliency maps are diffuse, providing the VLM with insufficient spatial grounding. The +OGR cross-check partially mitigates this–it correctly detects inconsistencies in all three cases–but the current Pass 2 revision cannot fully suppress hallucination when the VLM has high confidence in its fabricated description. One potential mitigation is training a lightweight hallucination detector that cross-references VLM claims against the quantitative channel scores. We leave these extensions to future work.
2. **Edit-operation ambiguity** (Case 2): The revised explanation identifies the building region but confuses replacement with removal. A possible mitigation is an explicit comparison of what occupies the region in the reference and query before assigning an operation label; this remains to be evaluated.

These examples characterize possible failure mechanisms. Aggregate improvements on the 400-pair evaluation do not establish their prevalence in deployment or the frequency of hallucination; that requires explicit failure annotation and measurement.

F Broad Impact and Limitations

Policy-dependent decisions. 2CAP provides channel similarity scores, localized evidence, and natural-language explanations to support policy-dependent human review. Different stakeholders, such as newsroom editors, scientific reviewers, legal professionals, platform moderators, may apply different thresholds and rubrics. 2CAP is designed to produce structured evidence without assuming a universal definition of disallowed manipulation; this policy-agnosticism is a deliberate design choice.

Adversarial robustness. Sophisticated adversaries may craft modifications that fool both channels simultaneously. While we observe robustness to standard perturbations and common post-processing pipelines, dedicated adversarial attacks that preserve apparent camera traces while introducing semantic modifications remain an open challenge for all forensic systems.

Score context influence. While the VLM can in principle disagree with the channel scores, we observe that score context exerts a strong anchoring effect: when scores are misleading (*e.g.*, a sophisticated edit that preserves camera traces), the VLM may under-weight its own visual observations. Presenting scores as context rather than hard labels mitigates but does not fully eliminate this anchoring bias. Developing score-aware calibration strategies that balance VLM autonomy with quantitative guidance is an important direction.

Semantic encoder scope. Our evidence-generation experiments use frozen SigLIP2 for both the semantic score and patch-level saliency map. The component ablations do not measure sensitivity to the choice of semantic encoder, including cases where highly realistic synthetic or edited images confuse the encoder. Comparing multiple frozen semantic encoders and adversarially stress-testing their semantic matches are important next steps.

VLM reliability. Generated explanations may occasionally be inaccurate or hallucinate forensic details not present in the image. The structured evidence trail (dual-channel scores, saliency maps) provides independent verification beyond the text, and human-in-the-loop verification remains essential for high-stakes decisions.

G Declaration of LLM Usage

In addition to incorporating LLMs into the proposed method, where they are part of the pipeline and evaluation, we also used general-purpose LLM-based assistants to help prepare this paper.

Coding assistance. LLMs were occasionally consulted to help with auxiliary coding tasks such as debugging error messages and improving code readability. All code that produced the experiments, datasets, models, and numerical results reported in this manuscript was subsequently inspected, tested, and validated by the authors. The architecture, training procedure, evaluation protocol, and analysis scripts were designed by the authors.

Language polishing. LLMs were used as a writing aid to suggest minor revisions for clarity, grammar, and word choice on text that the authors had already drafted. For every LLM-suggested revision that we adopted, the authors carefully read the rephrased text, confirmed that it preserves the intended technical meaning of the original sentence, and edited or rejected the suggestion whenever it introduced any ambiguity, imprecision, or change in meaning.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: All claims match the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Justified in the appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: No theory in this paper.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Experimental details are provided in the appendix. The code is not currently public and will be open-sourced after the release review is complete.

   Guidelines:

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

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

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

Answer: [No]

Justification: We use publicly available source datasets. The code is not currently public and will be open-sourced after the release review is complete.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Provided in the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Run standard deviations are provided for forensic-evidence and revision-frequency results and component ablations (Tables 2 and 14); a paired bootstrap confidence interval accompanies Table 11, and Table 12 reports variability across evaluated pairs. Table 1 and the policy columns of Table 2 report point estimates.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Provided in the appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We aligned with the code of ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Provided in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This is not applicable to our paper.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have cited them.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: We haven't provided the new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Not applicable for this paper.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Not applicable for this study.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: Added in the appendix.

Guidelines:

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