XPLAINVERSE: A MILLION-SCALE BENCHMARK FOR EXPLAINABLE DEEPFAKE DETECTION

Anonymous authors
Paper under double-blind review

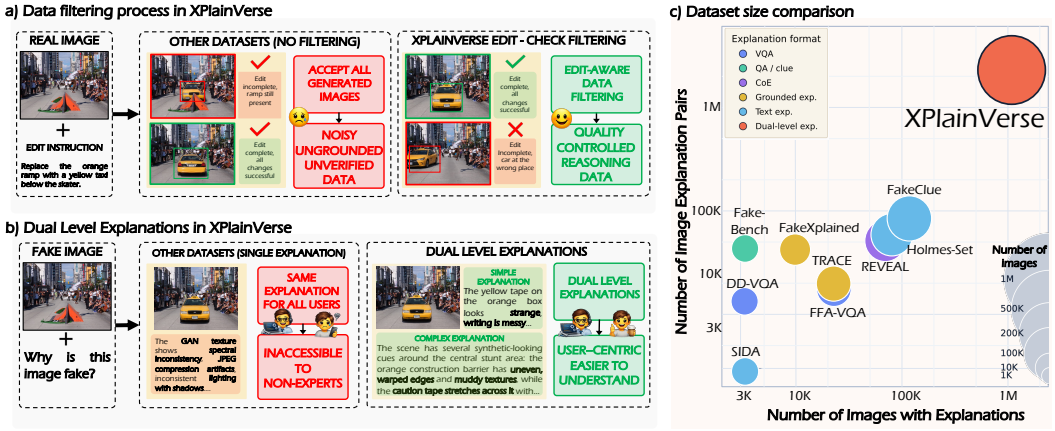


Figure 1: The proposed large-scale deepfake explainability dataset XPlainVerse created with (a) Edit-Aware data filtering process containing (b) Dual-level (complex and simple) deepfake explanations in XPlainVerse. (c) Size comparison of XPlainVerse with other reasoning-centric deepfake benchmarks.

ABSTRACT

As deepfake detection models increasingly produce natural-language explanations, their reasoning often remains weakly grounded in visual artifacts, limiting reliability and user trust. Existing benchmarks mainly evaluate classification accuracy, overlooking whether explanations reflect the actual manipulations. This gap hinders progress toward deployable, explainable deepfake detection systems. To this end, we introduce *XPlainVerse*, a large-scale benchmark designed for joint deepfake detection and human-centered explanation. XPlainVerse comprises *one million* real and manipulated images, pairing authentic images from five established sources with forgeries generated by twelve off-the-shelf image editing and synthesis models. We further propose a multi-stage filtering pipeline, *Edit-Check*, to verify whether manipulations satisfy their intended edits, yielding high-quality, edit-consistent examples for reasoning supervision at scale. Beyond dataset scale, XPlainVerse provides two complementary explanation styles: technical explanations for expert analysis and simplified explanations designed for non-technical users. To evaluate explanation quality beyond surface similarity, we propose novel metrics, *EntityScore* and *EvidenceScore*, that assess whether explanations capture the manipulated entities and supporting evidence reflected in the reference explanations. Human annotations on 2,000 manipulated images, together with complementary human evaluation, validate explanation and edit quality. We believe XPlainVerse will make grounded explanation quality an explicit evaluation target in deepfake detection and support scalable research on trustworthy, interpretable models.

1 INTRODUCTION

Manipulated images no longer rely on obvious visual artifacts to deceive. Off-the-shelf generative models can seamlessly alter faces, bodies, or scenes while preserving visual realism, making human inspection increasingly unreliable (Hertz et al., 2022; Brooks et al., 2023; Wu et al., 2025). As a result, automated deepfake detectors have become essential across journalism, forensics, and content moderation where detection accuracy alone is no longer sufficient (Rössler et al., 2019; Dolhansky et al., 2020; Huang et al., 2025b). When a system flags an image as manipulated, users inevitably ask *why*. Although many detectors now offer natural-language explanations, these often resemble

Dataset: *XPlainVerse Dataset* **Code:** *XPlainVerse Codebase*

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

Table 1: Comparison of deepfake explanation-oriented datasets. QA: Image-conditioned question answering, ME: Model Ensemble, QC: Quality Check, HF: Human Feedback, Human Ann.: Human Annotation

Dataset	Year	Annotation Source	Annotation Style	Images with Explanation	Total Image-Explanation Pairs	Simple Explanation
DD-VQA (Zhang et al.)	2024	Human Ann.	QA	2.97K	14.8K	✗
FFA-VQA (Huang et al.)	2024	Model + Human QC	QA	20.1K	20.1K	✗
FakeBench (Li et al.)	2024	Model + Human QC	QA + Text Explanations	3K	48K	✗
REVEAL-Bench (Cao et al.)	2025	ME	Chain-of-Evidence	60K	60K	✗
FakeXplained (Ji et al.)	2025	Human Ann.	Grounded Explanations	8.77K	≈47.5K	✗
SIDA (Huang et al.)	2025	Model + Human QC	Text Explanations	3K	3K	✗
Holmes-Set (Zhou et al.)	2025	ME + HF	Text Explanations	69K	69K	✗
FakeClue (Wen et al.)	2025	ME	Text Explanations	100K	100K	✗
TRACE (Ji et al.)	2026	ME + QC	Grounded Explanations	20K	20K	✗
XPlainVerse	**2026**	**ME + Human Ann. + Human QC**	**Dual-Level Explanations**	**1.0M**	**1.53M**	**530K**

plausible narratives rather than visual evidence, limiting trust and obscuring failure modes. This growing gap between confident explanations and the visual cues that truly support them underscores a key challenge: developing deepfake detection systems that not only decide correctly, but also explain their decisions in grounded, interpretable, and user-meaningful ways.

Existing benchmarks only partially address this need. Most focus primarily on binary real versus fake classification and provide little or no support for explanation generation (Rössler et al., 2019; Zhu et al., 2023; Wang et al., 2023). Even when explanations are considered, current generation models are typically trained on small or synthetic datasets that lack visual diversity and grounding, leading to explanations that are fluent but weakly tied to actual image artifacts. A core reason explanation-oriented datasets remain limited in scale (Zhang et al., 2024; Huang et al., 2024; Li et al., 2024; Huang et al., 2025b) is that scaling reasoning annotation automatically introduces substantial label noise, as naively prompting generative models to explain manipulations yields inconsistent and often hallucinated rationales. Our pipeline directly addresses this by combining edit-aware quality filtering with structured generation, producing high-quality explanation supervision at scale. In addition, prevailing evaluation metrics largely emphasize textual similarity or classification accuracy, overlooking whether explanations correctly identify the manipulated entities or supporting visual evidence. Current benchmarks also under-emphasize robustness under realistic post-processing, distribution shift, and adversarial perturbations (Huang et al., 2025a; Jia et al., 2025).

User needs further compound the problem. Research in human-centered XAI shows that different users require qualitatively different explanations (Miller, 2019; Ribera Turró & Lapedriza, 2019; Liao et al., 2020; Ehsan & Riedl, 2020; Linder et al., 2021): a technical analyst wants a precise enumeration of visual artifacts. At the same time, a non-expert moderator needs a single legible reason to act on. LayLens (Narang et al., 2025) demonstrates this in the deepfake domain specifically, showing that simplified explanations reduce cognitive load and improve user confidence. Robustness is also largely absent from evaluation: real deployments involve recompressed, blurred, or adversarially perturbed images, yet most benchmarks test on pristine data (Huang et al., 2025a; Jia et al., 2025).

We introduce **XPlainVerse**, a **1M-image benchmark** for joint deepfake detection and reasoning. XPlainVerse addresses each of the gaps as discussed above. Our contributions are as follows.

Quality-controlled fake image filtering. We introduce *Edit-Check*, a novel data filtering pipeline to generate reasoning data at scale. Every generated image pair passes through a multi-stage filtering pipeline that verifies the visible changes match the intended edit and discards samples with prominent off-target modifications. Further, to evaluate real-world performance, we add perturbations spanning adversarial attacks and natural corruptions such as recompression, denoising, etc.

Multi-level explanations for diverse users. XPlainVerse provides complex explanations targeting technical users, which describe multiple concrete visual artifacts, and simple explanations targeting non-expert users, which communicate a single accessible reason. For real images, authenticity explanations highlight natural details supporting their perceived authenticity.

Entity and evidence-aware explanation metrics. We introduce novel evaluation metrics, *EntityScore* and *EvidenceScore*, that test whether a predicted explanation recovers the manipulated entities and supporting evidence described in the reference explanation. For simple explanations, separate metrics explicitly measure fidelity and readability, operationalizing the user-oriented goals.

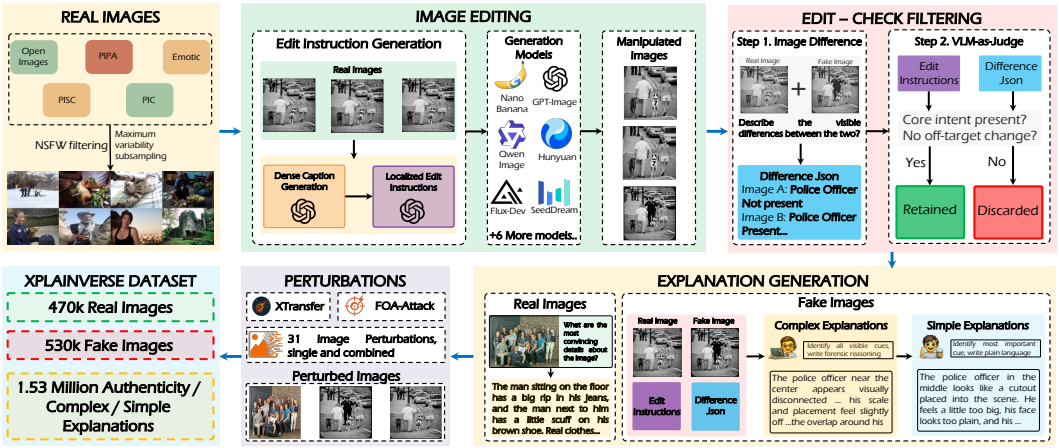


Figure 2: Overview of the XPlainVerse data generation pipeline. For each real image, we generate multiple edit instructions and use them to create synthetic images. The resulting images are filtered through our multi-stage pipeline and augmented with low-level perturbations and adversarial attacks.

2 Related Work

Detection datasets and explanation benchmarks. Deepfake and synthetic-image detection have been shaped by a sequence of progressively larger benchmarks. Early datasets, including FaceForensics++ (Rössler et al., 2019) and DFDC (Dolhansky et al., 2020), focused on facial manipulations, while more recent resources such as MultiFakeVerse (Gupta et al., 2025) and SemiTruths (Pal et al., 2024) extend to object- and scene-level edits. These datasets are designed for classification, localization, or robustness evaluation, not for natural-language explanation supervision. A separate line of explanation-oriented benchmarks, including DD-VQA (Zhang et al., 2024), FFA-VQA (Huang et al., 2024), FakeBench (Li et al., 2024), and SIDA (Huang et al., 2025b), introduces QA-style and textual explanations for manipulated images. However, they remain small relative to modern detection benchmarks and primarily target technical forensic analysis rather than non-expert communication.

Detection methods and explainability. Detection methods have evolved from handcrafted frequency and artifact cues (Liu et al., 2021a; Li et al., 2020; 2018b) to learning-based approaches using convolutional, recurrent, and transformer architectures (Afchar et al., 2018; Güera & Delp, 2018; Zhao et al., 2021; Zhang et al., 2022; Wang et al., 2022). Some recent work introduces localization supervision to identify manipulated regions (Dang et al., 2020; Xu et al., 2025), and vision-language approaches such as SIDA (Huang et al., 2025b), BusterX++ (Wen et al., 2026), and FakeShield (Xu et al., 2025) augment classifiers with textual explanations. Yet surface-level fluency does not guarantee that an explanation correctly identifies the manipulated entity or cites evidence present in the image. Further, some recent attempts have tried to evaluate deepfake reasoning through images directly (Kuckreja et al., 2026). Human-centered XAI research (Rong et al., 2024; Liao & Varshney, 2022) further argues that explanations should be tailored to user expertise, a distinction that existing forensic benchmarks rarely make. Concurrently, advances in diffusion-based image editing (Hertz et al., 2022; Brooks et al., 2023; Wu et al., 2025) have made manipulations harder to detect via low-level artifacts alone, reinforcing the need for edit-aware supervision and grounded natural-language explanations. As such, these gaps motivate XPlainVerse.

3 XPlainVerse Dataset

We start our data generation pipeline by first collecting publicly available real-image datasets spanning diverse real-world scenes, objects, people, activities, and social interactions, including OpenImages (Kuznetsova et al., 2020), EMOTIC (Kosti et al., 2019), PIPA (Oh et al., 2015), PIC 2.0 (Liu et al., 2021b), and PISC (Li et al., 2018a). This is followed by a comprehensive data filtering and explanation generation process, as detailed in Figure 2. Figure 3 summarizes source and generator distributions by split; detailed dataset accounting is provided in Appendix C. We use an ensemble of multiple non-overlapping LLM / VLM families during each step of data generation to avoid potential biases, with higher sampling for stronger models. Our data generation process is as follows.[1]

[1]Full prompts for all data filtering and generation steps are provided in Appendix K.

3.1 Data Generation Pipeline

Step 1: Caption-guided edit-instruction generation. We begin by leveraging the extensive visual understanding capabilities of foundational vision–language models (VLMs) (particularly GPT-4o-mini (OpenAI, 2024)) to obtain concise, information-dense captions to describe primary and secondary subjects, appearance details, spatial layout, lighting, and object interactions without speculation. Further, GPT-4o-mini uses these captions to produce structured edit instructions comprising 2-4 concrete, semantically targeted manipulations. The edits span *person-centric*, *object-centric*, *geometry-centric*, and *scene-centric* transformations, including attribute and emotion changes, object insertion or removal, copy-move duplication, local planar warping, and shape deformation.

Step 2: Candidate manipulated-image construction. We pair the instructions with the corresponding real images and generate manipulated images using a diverse set of off-the-shelf image-editing models such as FLUX.2-dev (Black Forest Labs, 2026), HunyuanImage (Cao et al., 2026b), LongCat-Image-Edit (Meituan LongCat Team et al., 2025), Qwen-Image-Edit-2511 (Wu et al., 2025), GPT-Image-1.5 (OpenAI, 2025b), Seedream-4.5 (ByteDance Seed Team, 2026), Wan 2.6 (Alibaba Cloud, 2026), Nano Banana 2 (Google, 2026), and Nano Banana Pro (Google DeepMind, 2025). To further broaden generator coverage, we also use manipulated candidates from MultiFakeVerse (Gupta et al., 2025), which includes images generated using three additional editing models: Gemini 2.0 Flash (Google, 2025), GPT-Image-1 (OpenAI, 2025a), and ICEdit (Zhang et al., 2025). Together, these sources yield an initial pool of approximately 609K manipulated candidates spanning twelve generation models.

Step 3: Edit-Check Pipeline. To retain only high-quality image-instruction pairs, we apply a two-stage filtering pipeline. First, given a real image I_{real} and its generated counterpart I_{fake}, we randomly sample one VLM from a pool of four models (Gemini-3-Flash (Google, 2025b), GPT-5-mini (OpenAI, 2026), GLM-4.6V (Z.ai, 2025), and Qwen3.5-27B (Qwen Team, 2026)) to extract a JSON object enumerating all verifiable visual differences between the two images. Each identified change is annotated with a salience score ranging from 1 to 5, along with a precise description of the changed region or entity. In the second stage, a text-only judge sampled from either DeepSeek-V3.2 (DeepSeek-AI et al., 2025) or GPT-OSS-120B (OpenAI, 2025c) determines whether the extracted differences align with the original edit instruction. Each judge outputs a `KEEP`, `BORDERLINE`, or `DISCARD` decision based on two criteria: whether the core edit intent is present, and whether prominent unintended changes were introduced. Only `KEEP` samples are retained. Samples where off-target modifications dominate are hard-discarded regardless of partial edit-intent compliance.

The filtering pipeline reduces the initial pool of approximately 609K manipulated candidates to about 265K retained clean manipulated images, corresponding to a discard rate of approximately 56.5%. This high discard rate highlights the failure rate of generators: without filtering, many explanations would be generated for images whose intended edits are absent or dominated by unintended scene changes, resulting in noisy explanation supervision. Qualitative examples of accepted and discarded candidates are shown in Figure 1, with additional filtering examples in Appendix E.

Step 4: Introducing Perturbation. To assess robustness under real-world conditions such as post-processing, recompression, and adversarial manipulation, each image is paired with a perturbed counterpart. Perturbations are applied cumulatively across splits, i.e., validation includes all training perturbation families, and test includes all validation perturbation families while adding harder corruptions. Training perturbations include an xTransfer-based adversarial variant (Huang et al., 2025a) and standard corruptions spanning noise, blur, compression, and photometric changes. Validation additionally introduces a targeted FOA-style attack (Jia et al., 2025) and additional perturbations, while the test set further adds harder geometric and frequency-domain corruptions, including affine transformations and ringing halo effect. Across all splits, most perturbed counterparts contain a single perturbation, while smaller proportions combine two or multiple perturbations. The exact split-specific mixture schedules and parameter ranges are provided in Appendix F.

Step 5: Explanation Generation For each image, we generate natural-language explanations that justify either its manipulation (for fake images) or its authenticity (for real images), with different levels of detail depending on the explanation type.

For manipulated images, we generate two types of explanations. A complex explanation describes the visual evidence of manipulation in enough detail for technical or forensic analysis. To produce it, one model randomly sampled from a pool of four (Gemini-3-Flash (Google, 2025b), GPT-5-mini

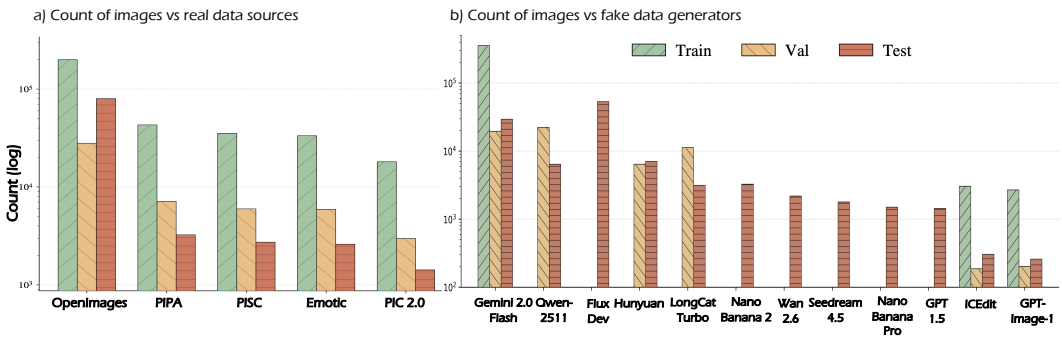


Figure 3: a) Distribution for the real-image sources across the Train, Validation, and Test splits. b) Distribution of the fake-image generators across the Train, Validation, and Test splits.

(OpenAI, 2026), GLM-4.6V (Z.ai, 2025), and Qwen3.5-27B (Qwen Team, 2026)) receives the fake image alongside three auxiliary references: the paired real image, the edit-instruction, and the visual-difference produced during filtering. These references help the model identify informative regions and manipulation-relevant cues. The resulting explanations describe cues such as geometric inconsistencies, implausible text or boundaries, inconsistent lighting or reflections, and localized detail failures. A simple explanation is then generated for each fake image by conditioning on the corresponding complex explanation and the same image-level context. Written in plain, layman's terms and focused on a single salient reason why the image appears manipulated, these explanations are designed to be accessible to users of diverse backgrounds.

For real images, each sample is paired with an authenticity explanation that highlights visible properties supporting its perceived authenticity, such as natural imperfections, physically plausible interactions, or realistic material and texture details. These explanations are generated using the same four models above, with the addition of Gemini-2.5-Pro (Google, 2025a).

3.2 Human Annotations

To obtain a *gold-standard reference*, we collect human-written explanations from 12 annotators on a subset of 2,000 manipulated images. Annotators are shown each fake image together with its paired real image for context and are instructed to describe all visible cues that indicate manipulation, focusing on evidence observable in the fake image itself and using the real image only as contextual support. To assess consistency, we collect 400 independently written explanations across 50 shared images, with eight annotators per image. Mean agreement scores using *EntityScore* and *EvidenceScore* (defined in Section 4) are 0.519 and 0.491, respectively. This overlap study provides a consistency check on whether annotators identify similar manipulation cues despite natural variation in phrasing. All annotations follow strict university IRB guidelines (see Appendix B). These human annotations, written independently of our explanation pipeline, help validate the grounding, completeness, and usefulness of our generated explanations against human-written references (Table 4).

3.3 Dataset Quality Analysis and Validation

We analyze XPlainVerse along four complementary dimensions: image fidelity and edit preservation, semantic and explanation diversity, distribution coverage, and human-validated explanation quality. These analyses are intended to verify that the benchmark is not only large but also visually realistic, semantically diverse, and suitable for high-quality explanation supervision.

Table 2: Image-quality comparisons.

Pair type	Subset	PSNR	SSIM	LPIPS	CLIP-I
Real-fake	ID discarded	17.82	0.563	0.272	0.900
	ID kept	**18.80**	**0.613**	**0.204**	**0.920**
	OOD discarded	16.83	0.572	0.321	0.928
	OOD kept	**17.87**	**0.667**	**0.210**	**0.930**
Clean-pert.	Train	37.83	0.945	0.053	0.985
	Val	36.15	0.920	0.090	0.981
	Test	35.58	0.890	0.098	0.976

1. Image Fidelity and Explanation Diversity. To assess curation quality in real-fake pairs, we examine whether retained generated images more faithfully reproduce their paired real images compared to discarded candidates. We compute PSNR, SSIM, LPIPS, and CLIP-I across 20K candidate pairs (10K retained and 10K discarded), balanced across ID and OOD samples (Table 2).

Retained pairs consistently exhibit higher PSNR and SSIM, lower LPIPS, and stronger CLIP-I similarity in both settings, supporting that curation successfully filters for structural and semantic fidelity. Because these metrics operate over entire images rather than localized edits, we treat them as a preservation evaluation. For clean-perturbed pairs, we compute the same metrics on 10K pairs per split. The train split remains closest to the clean images, while validation and test splits reflect progressively stronger degradation, consistent with the intended perturbations.

We also evaluate explanation diversity using 3K subsets from XPlainVerse, SIDA, and FakeClue. Self-BLEU measures repetition, Distinct-2 measures lexical diversity, and entity mention rate measures the proportion of explanations that reference at least one concrete region, object, or subject (Table 3). XPlainVerse complex achieves the lowest Self-BLEU and highest Distinct-2, indicating less repetitive wording than SIDA and FakeClue. XPlainVerse simple remains more lexically diverse than both datasets, and over 99% of explanations in both XPlainVerse types mention at least one entity.

Table 3: Explanation-diversity comparison.

Dataset / subset	n	SB↓	D-2↑	Ent.↑
SIDA	3K	0.497	0.153	93.2%
FakeClue	3K	0.689	0.088	99.2%
XPLAINVERSE complex	3K	**0.355**	**0.243**	**99.6%**
XPLAINVERSE simple	3K	0.394	0.219	99.5%

2. Human Evaluation.[2] We conduct a seven-stage evaluation with 30 adult participants covering image realism, edit correctness, explanation quality (grounding, specificity, completeness, unsupported claims), pairwise style comparisons, and readability. Figures 4 and 5 summarize the findings.

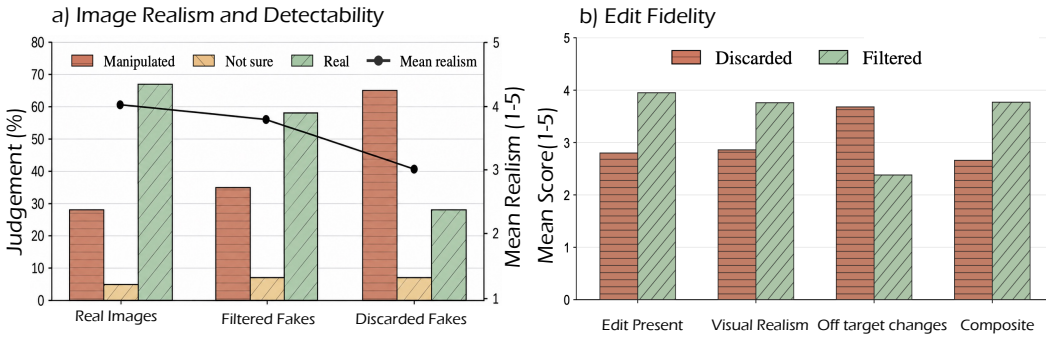


Figure 4: (a) Realism comparison between real, filtered, and discarded fakes. (b) Edit-fidelity scores. Filtered fakes are rated higher in both image realism and edit fidelity than discarded fakes.

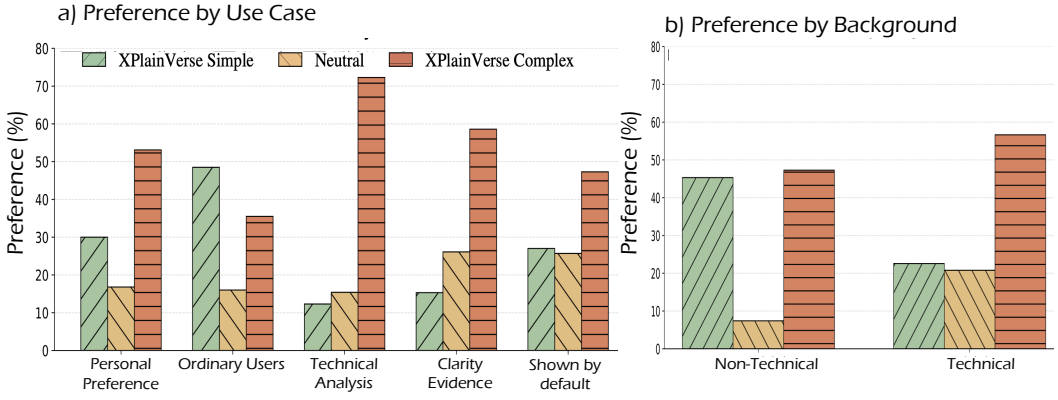


Figure 5: (a) Use-case preference comparison between complex and simple explanations. (b) Preference by background comparison between complex and simple explanations.

Image and edit quality. Filtered fake images were difficult to distinguish from real photographs, with annotators labeling them *Real* 57.4% of the time versus *Manipulated* 35.4%, closely mirroring the

[2]Details about the user study and ethics are available in Appendices H and B.

Table 4: Human preferences (%) for XPlainVerse vs SIDA-style and human-written explanations.

		SIDA				Human explanation			
Section	**Criterion**	**Complex**	**Tie**	**SIDA**	**Δ vs. SIDA**	**Complex**	**Tie**	**Human**	**Δ vs. Human**
Quality	Grounding	64.9	6.2	28.9	+36.0	63.7	13.3	23.0	+40.7
	Specificity	58.7	5.7	35.6	+23.1	61.6	12.7	25.7	+35.9
	Completeness	57.9	6.3	35.8	+22.1	63.2	14.2	22.6	+40.6
	Fewer Unsupported claims	54.9	14.0	31.1	+23.7	58.0	17.3	24.7	+33.4
Overall	Preference	59.2	8.8	32.0	+27.2	59.5	15.2	25.3	+34.2

real-image distribution (Figure 4(a)). Discarded generations were identified as manipulated 64.9% of the time, with lower mean realism scores (2.98 vs. 3.72), confirming the filtering pipeline removes conspicuous artifacts and off-target changes. Filtered edits also outperform discarded edits, with a composite score of 3.77 vs. 2.66 (difference: 1.11, 95% CI: 0.73–1.49; Figure 4(b)), supporting the effectiveness of Edit-Check in retaining higher-quality edits.

Preference and Explanation quality. Preferences split clearly by use case (Figure 5(a)): Simple is favored for ordinary users (48.5% vs. 35.5% for Complex), while Complex dominates for technical analysis (72.3% vs. 12.3%). No single explanation style serves both audiences well, and the gap between the two is large enough that choosing one over the other would leave a significant fraction of users with a suboptimal experience, underscoring the need for user-specific explanations. Further, complex explanations achieved a composite score of 3.86/5 and received a 59.2% preference share over the SIDA-style explanations and 59.5% over human-written explanations (Table 4), with the largest margins over human-written explanations in grounding and completeness. XPlainVerse Simple scored 3.90/5, with ease-of-understanding at 3.99 and memorability at 3.97 (Appendix Table 18).

4 Evaluation Metrics

A central contribution of XPlainVerse is an *intent-aware* evaluation metric for complex explanations. Standard metrics like BERTScore (Zhang et al., 2020) measure semantic similarity but fail to capture whether a prediction identifies the same manipulated regions or recovers the same supporting evidence as the reference. We address this with two tailored metrics: *EntityScore* and *EvidenceScore*.

Given a predicted explanation and a reference explanation, we first use Qwen3.5-4B (Qwen Team, 2026) to extract from each explanation a set of *diagnostic entities* and a set of *evidence claims*. Diagnostic entities correspond to the manipulated regions, subjects, or semantically meaningful image parts mentioned in the predicted and reference explanations. Evidence claims correspond to the concrete visual cues used to justify why the image appears fake.

We then compute coverage in two directional passes using the same Qwen3.5-4B evaluator. In the first pass, Qwen3.5-4B is given the extracted entities and evidence claims from the *predicted* explanation, together with the full *reference* explanation, and judges whether each predicted item is supported by the reference text. This yields the precision-side scores: *Entity-Precision* and *Evidence-Precision*. In the second pass, it is given the extracted entities and evidence claims from the *reference* explanation, together with the full *predicted* explanation, and judges whether each reference item is supported by the prediction. This yields the recall-side scores: *Entity-Recall* and *Evidence-Recall*. We compute *EntityScore* and *EvidenceScore* as harmonic means of their corresponding precision and recall terms. Full metric definitions are provided in Appendix I. Cross-family validation with three judge models shows consistent relative assessments, suggesting that the metric conclusions are not specific to the default Qwen evaluator (Appendix I). Alongside these intent-aware metrics, we report BERTScore-F1 as a complementary measure of overall semantic similarity.

For simple explanations, the goal is to preserve the main reason while remaining accessible to non-experts. We therefore report BERTScore-F1 for semantic fidelity and the SLE simplicity score (Cripwell et al., 2023) for readability, where higher SLE indicates simpler explanations.

5 Benchmarks and Experiments

Evaluation Protocol. We evaluate three tasks: detection, complex explanation generation, and simple explanation generation, across in-domain (ID) and out-of-domain (OOD) manipulated-image subsets. We report detection as *binary F1 score*. Complex explanations are evaluated using EntityScore,

Table 5: In-distribution and out-of-distribution results on XPlainVerse. Best scores across ID and OOD are shown in **bold**. Entity: EntityScore, Evidence: EvidenceScore, BERT: BERTScore

			Complex explanation				Simple explanation		
Split	**Setting**	**Method**	F1 (%)	BERT	Entity	Evidence	F1 (%)	BERT	SLE
ID	*Closed-source*	GPT-5-mini	58.92	0.602	0.310	0.170	54.59	0.584	0.076
		Gemini-3-Flash	75.25	0.630	0.313	0.170	75.08	0.606	0.126
	Open-source zero-shot	Qwen-3.5-4B	5.43	0.593	0.253	0.069	3.96	0.575	0.121
		Qwen3-VL-8B	16.14	0.605	0.266	0.082	15.26	0.594	0.139
		BusterX++	57.59	0.563	0.278	0.102	56.54	0.524	0.220
		InternVL-3.5-14B	41.17	0.614	0.181	0.055	40.37	0.602	0.120
		Qwen-3.5-35B	17.03	0.591	0.312	0.117	11.15	0.583	0.152
	Fine-tuned	BusterX++	91.89	0.691	0.400	0.301	92.45	0.689	**0.576**
		Qwen-3.5-4B	94.13	0.702	**0.464**	**0.371**	94.39	0.700	0.568
		LLaVA-1.6-7B	**96.22**	**0.703**	0.453	0.361	**96.18**	**0.701**	0.569
		Qwen3-VL-8B	94.43	0.702	0.458	0.366	94.84	0.700	0.572
		InternVL-3.5-14B	83.68	0.686	0.408	0.315	84.64	0.688	0.571
OOD	*Closed-source*	GPT-5-mini	50.50	0.599	0.338	0.168	46.78	0.583	0.074
		Gemini-3-Flash	**71.77**	0.618	0.308	0.135	**70.05**	0.601	0.120
	Open-source zero-shot	Qwen-3.5-4B	7.84	0.591	0.274	0.061	10.10	0.574	0.125
		Qwen3-VL-8B	25.34	0.601	0.294	0.085	23.10	0.592	0.145
		BusterX++	49.58	0.558	0.305	0.089	48.94	0.521	0.221
		InternVL-3.5-14B	38.66	0.608	0.204	0.045	36.52	0.598	0.119
		Qwen-3.5-35B	22.31	0.592	0.345	0.108	13.57	0.582	0.150
	Fine-tuned	BusterX++	37.14	0.639	0.312	0.166	37.87	0.653	**0.580**
		Qwen-3.5-4B	33.38	**0.646**	**0.357**	**0.221**	33.34	**0.658**	0.549
		LLaVA-1.6-7B	25.14	0.642	0.333	0.200	27.41	0.655	0.551
		Qwen3-VL-8B	32.67	0.645	0.351	0.218	32.75	0.657	0.554
		InternVL-3.5-14B	42.33	**0.646**	0.330	0.205	44.68	0.656	0.557

EvidenceScore, and BERTScore-F1, while simple explanations are evaluated using BERTScore-F1 and SLE. Further details are available in Appendix J.

Baselines. Table 5 reveals a consistent pattern across all model families. Fine-tuned models perform strongly on ID data, with LLaVA-1.6-7B reaching 96.22% F1 and BERTScore of 0.703, but this does not transfer: the same models fall to 25-42% F1 under OOD conditions, a drop of over 50 points in several cases. Increasing model size does not reliably close this gap, indicating substantial generator-shift sensitivity under the original training distribution. In a broader-generator training study, expanding exposure from three to five generators increases Qwen3.5-4B's mean fake recall on the same seven held-out generators from 15.3% to 49.0%. Nevertheless, a 33.8-percentage-point gap remains between training-exposed and held-out generators, showing that broader generator exposure improves transfer but does not eliminate cross-generator sensitivity (Appendix J, Table 24). Zero-shot models tell a complementary story. Gemini-3-Flash achieves the highest OOD fake-class F1 (71.77%) among the evaluated baselines. For fine-tuned models, BERTScore remains relatively stable under distribution shift (dropping by roughly 0.04 - 0.06) while EntityScore and EvidenceScore degrade more sharply, showing models retain surface fluency but fail to recover manipulation-relevant entities and evidence. Robust explainable detection requires more than fine-tuning on in-distribution data.

Cross-dataset transfer. We additionally evaluate Qwen-3.5-4B and LLaVA-1.6-Mistral-7B on SIDA, GenImage, and Community-Forensics-Eval, comparing released checkpoints with counterparts fine-tuned on XPlainVerse training data. Accuracy improves for both models across all three datasets. Qwen-3.5-4B also improves binary F1 consistently, whereas LLaVA exhibits mixed F1 changes, indicating model-dependent transfer. Results are reported in Appendix J (Table 23).

Additional Analysis. We do in-depth analysis to identify the reason why VLMs do not generalize to the OOD dataset. First, we train linear probes and shallow MLPs on frozen embeddings from CLIP, DINOv2, and SigLIP to understand the train-to-test accuracy drop in Table 5. Training AUC is high across all three backbones, yet drops to near chance on the generator-shifted test set. Increasing probe depth leaves test AUC essentially unchanged (Figure 6a,b), indicating that the bottleneck is not capacity but what the model has learned to rely on. Secondly, as shown in Figure 6(c,d), the spectral patterns differ across editing models, suggesting that fine-tuned models may pick up generator-specific cues that do not generalize well. We further test this with counterfactual region swaps. For

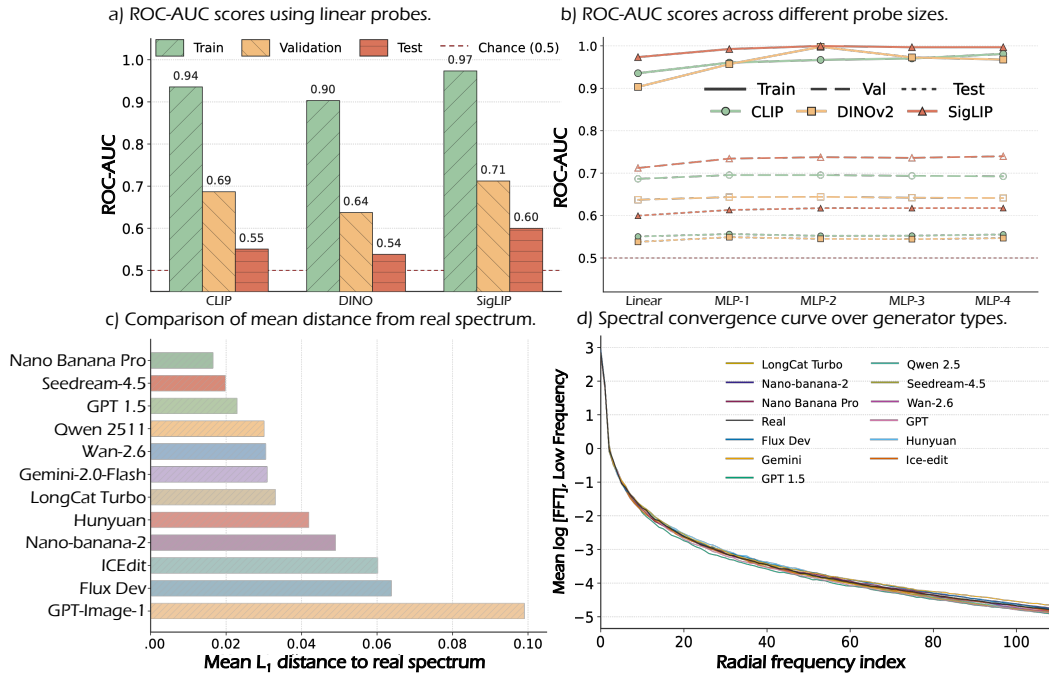


Figure 6: (a) Linear probes on frozen CLIP, DINOv2, and SigLIP embeddings. (b) Increasing probe capacity from linear to MLP-4. (c) Mean L_1 distance from the real-image frequency spectrum per generator. (d) Radial convergence trends per generator.

generators seen during training, regions outside the intended edit lead to more fake predictions than the localized edit itself, while the opposite trend is observed for held-out generators (Appendix J, *Counterfactual analysis*). This further suggests that models are relying on both manipulation cues and generator-specific signals, rather than only the actual edited region. This motivates methods grounded in explicit visual evidence rather than model-specific statistical regularities.

6 Conclusion

We introduced XPlainVerse, a million-scale benchmark for explainable deepfake detection, containing 470K real and 530K manipulated images from twelve generators with 1.53M natural-language explanations in two styles: technical explanations for expert analysis and simple explanations for non-technical users. Our edit-aware filtering pipeline removes low-quality generations, while human evaluation supports both edit and explanation quality. In our experiments, most zero-shot models perform poorly, while fine-tuned models achieve strong in-distribution performance but transfer weakly to held-out generators. EntityScore and EvidenceScore further make entity and evidence-level explanation quality measurable, revealing failures that surface-level semantic metrics can miss when models produce fluent explanations that fail to recover the relevant entities and supporting evidence. Spectral and counterfactual analyses indicate that models remain sensitive to generator-specific signals, while localized manipulation evidence also influences their predictions. This motivates methods grounded in transferable visual evidence rather than generator-specific statistical regularities. XPlainVerse aims to support future models that can detect and explain deepfakes in a grounded, user-appropriate way.

Limitations and Future Directions. XPlainVerse is currently limited to still images and therefore does not capture temporal or audio-visual cues present in video deepfakes. The human evaluation also uses a relatively small and geographically concentrated participant pool, and reliance on a fixed set of VLMs and commercial APIs may affect reproducibility over time. Future work should extend the benchmark to video and audio-visual manipulations, broaden the evaluator pool, and develop stronger protocols for edit-instruction compliance. Our observed cross-generator degradation also motivates detectors that ground predictions in localized manipulation evidence rather than generator-specific cues. As generative technology evolves, periodic benchmark updates will be essential to keep XPlainVerse meaningful.

Ethics Statement

XPlainVerse uses real images from five publicly released research datasets; we collected no new photographs, and source assets remain subject to their original licenses and terms of use. Manipulated images were generated for academic research using off-the-shelf commercial APIs and open-weight image-editing models. Candidate images underwent semantic filtering, automated NSFW screening, manual spot-checks, and the Edit-Check filtering pipeline. No generated image depicting identifiable private individuals was retained, and all materials used for human annotation and evaluation were screened to exclude NSFW or distressing content.

All human annotation and evaluation were conducted under institutional IRB approval and participation was voluntary. Human-evaluation participants received written study information and provided informed consent before participating. No personally identifying information was collected or recorded, and evaluation responses were anonymized before analysis. Compensation rates were determined independently of the labels, ratings, or judgments participants produced.

Deepfake datasets carry inherent dual-use risks, since difficult manipulations and detector failures could potentially inform attempts to evade detection. We mitigate these risks through controlled access under the XPlainVerse EULA, which restricts use to academic, non-commercial, and not-for-profit research and prohibits harmful or deceptive applications, unauthorized redistribution, public hosting, sublicensing, and commercial use without prior approval. Third-party assets remain governed by their upstream licenses and provider terms; where redistribution is not permitted, we release only compatible metadata, identifiers, annotations, or XPlainVerse-specific derived components. We do not release generation-model weights, service credentials, or tools intended to create stronger forgeries. Further details on data sourcing, human participation, licensing, release conditions, dual-use risks, and limitations are provided in Appendix B.

Reproducibility Statement

We support reproducibility at both the benchmark and pipeline levels. The released XPlainVerse materials include split and metadata files, prompts, generated and human annotations, perturbation metadata, and evaluation protocols, enabling reconstruction of the benchmark splits and reported evaluation setup without regenerating the underlying data. Section 3 and Appendices C–G document data assembly, edit generation and filtering, perturbation schedules and parameters, and explanation construction. Appendix H specifies the human-evaluation protocol; Appendix I provides the complete EntityScore and EvidenceScore implementation and evaluator configuration; Appendix J reports model prompts, fine-tuning, inference, scoring, hyperparameters, and compute; and Appendix K provides the full prompt templates used throughout dataset construction and evaluation.

AI use statement

Generative image and language models were used to construct manipulated images, generate captions and edit instructions, filter candidate edits, and generate reference explanations, as described in Section 3 and Appendix K. An LLM evaluator was used for EntityScore and EvidenceScore (Appendix I). AI assistance was also used for manuscript editing and LaTeX formatting. The authors are responsible for the final manuscript and reported results.

References

Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7. IEEE, December 2018. doi: 10.1109/wifs.2018.8630761. URL http://dx.doi.org/10.1109/WIFS.2018.8630761.

Alibaba Cloud. Wan2.6 image generation and editing. Alibaba Cloud Model Studio Documentation, 2026. URL https://www.alibabacloud.com/help/en/model-studio/wan-image-generation-api-reference.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei

Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Black Forest Labs. Flux.2-dev. Hugging Face Model Card, 2026. URL https://huggingface.co/black-forest-labs/FLUX.2-dev.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

ByteDance Seed Team. Seedream 4.5. Official Model Page, 2026. URL https://seed.bytedance.com/en/seedream4_5.

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, 2026a. URL https://arxiv.org/abs/2511.23158.

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, Tiankai Hang, Duojun Huang, Jie Jiang, Zhengkai Jiang, Weijie Kong, Changlin Li, Donghao Li, Junzhe Li, Xin Li, Yang Li, Zhenxi Li, Zhimin Li, Jiaxin Lin, Linus, Lucaz Liu, Shu Liu, Songtao Liu, Yu Liu, Yuhong Liu, Yanxin Long, Fanbin Lu, Qinglin Lu, Yuyang Peng, Yuanbo Peng, Xiangwei Shen, Yixuan Shi, Jiale Tao, Yangyu Tao, Qi Tian, Pengfei Wan, Chunyu Wang, Kai Wang, Lei Wang, Linqing Wang, Lucas Wang, Qixun Wang, Weiyan Wang, Hao Wen, Bing Wu, Jianbing Wu, Yue Wu, Senhao Xie, Fang Yang, Miles Yang, Xiaofeng Yang, Xuan Yang, Zhantao Yang, Jingmiao Yu, Zheng Yuan, Chao Zhang, Jian-Wei Zhang, Peizhen Zhang, Shi-Xue Zhang, Tao Zhang, Weigang Zhang, Yepeng Zhang, Yingfang Zhang, Zihao Zhang, Zijian Zhang, Penghao Zhao, Zhiyuan Zhao, Xuefei Zhe, Jianchen Zhu, and Zhao Zhong. Hunyuanimage 3.0 technical report, 2026b. URL https://arxiv.org/abs/2509.23951.

Liam Cripwell, Joël Legrand, and Claire Gardent. Simplicity level estimate (sle): A learned reference-less metric for sentence simplification, 2023. URL https://arxiv.org/abs/2310.08170.

Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil Jain. On the detection of digital face manipulation, 2020. URL https://arxiv.org/abs/1910.01717.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Erhang Li, Fangqi Zhou, Fangyun Lin, Fucong Dai, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoran Wei, Haowei Zhang, Haowen Luo, Haozhe Ji, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, Jialiang Huang, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jingchang Chen, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexin Huang, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Liang Zhao, Liangsheng Yin, Lihua Guo, Lingxiao Luo, Linwang Ma, Litong Wang, Liyue Zhang, M. S. Di, M. Y Xu, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Panpan Huang, Peixin Cong, Peiyi Wang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, S. H. Liu, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Songyang Zhou, Tao Ni, Tao Yun, Tian Pei, Tian Ye, Tianyuan Yue, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjun Gao, Wentao Zhang, Xi Gao, Xiangwen Wang, Xiao Bi, Xiaodong Liu, Xiaohan

Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyuan Li, Xu Chen, Xuecheng Su, Xuehai Pan, Xuheng Lin, Xuwei Fu, Y. Q. Wang, Yang Zhang, Yanhong Xu, Yanru Ma, Yao Li, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yiliang Xiong, Ying He, Ying Zhou, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuduan Wang, Yue Gong, Yuhan Wu, Yuheng Zou, Yukun Li, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehua Zhao, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhiyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Zizheng Pan, Zongqing Yao, Bei Feng, Hui Li, J. L. Cai, Jiaqi Ni, Lei Xu, Meng Li, Ning Tian, R. J. Chen, R. L. Jin, S. S. Li, Shuang Zhou, Tianyu Sun, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xinnan Song, Xinyi Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, Dongjie Ji, Jian Liang, Jianzhong Guo, Jin Chen, Leyi Xia, Miaojun Wang, Mingming Li, Peng Zhang, Ruyi Chen, Shangmian Sun, Shaoqing Wu, Shengfeng Ye, T. Wang, W. L. Xiao, Wei An, Xianzu Wang, Xiaowen Sun, Xiaoxiang Wang, Ying Tang, Yukun Zha, Zekai Zhang, Zhe Ju, Zhen Zhang, and Zihua Qu. Deepseek-v3.2: Pushing the frontier of open large language models, 2025. URL https://arxiv.org/abs/2512.02556.

Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset, 2020. URL https://arxiv.org/abs/2006.07397.

Upol Ehsan and Mark O. Riedl. Human-centered explainable ai: Towards a reflective sociotechnical approach, 2020. URL https://arxiv.org/abs/2002.01092.

Google. Gemini 2.5 pro. Google AI for Developers Documentation, 2025a. URL https://ai.google.dev/gemini-api/docs/models/gemini-2.5-pro.

Google. Gemini 3 flash: Frontier intelligence built for speed. Google Blog, 2025b. URL https://blog.google/products-and-platforms/products/gemini/gemini-3-flash/.

Google. Experiment with gemini 2.0 flash native image generation. Google Developers Blog, 2025. URL https://developers.googleblog.com/experiment-with-gemini-20-flash-native-image-generation/. Accessed: 2026-05-06.

Google. Build with nano banana 2. Google Blog, 2026. URL https://blog.google/innovation-and-ai/technology/developers-tools/build-with-nano-banana-2/.

Google DeepMind. Introducing nano banana pro. Google Blog, 2025. URL https://blog.google/innovation-and-ai/products/nano-banana-pro/.

Parul Gupta, Shreya Ghosh, Tom Gedeon, Thanh-Toan Do, and Abhinav Dhall. Multiverse through deepfakes: The multifakeverse dataset of person-centric visual and conceptual manipulations. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 13258–13265, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3758283. URL https://doi.org/10.1145/3746027.3758283.

David Güera and Edward J. Delp. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, 2018. doi: 10.1109/AVSS.2018.8639163.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Hanxun Huang, Sarah Erfani, Yige Li, Xingjun Ma, and James Bailey. X-transfer attacks: Towards super transferable adversarial attacks on clip, 2025a. URL https://arxiv.org/abs/2505.05528.

Zhengchao Huang, Bin Xia, Zicheng Lin, Zhun Mou, Wenming Yang, and Jiaya Jia. Ffaa: Multimodal large language model based explainable open-world face forgery analysis assistant, 2024. URL https://arxiv.org/abs/2408.10072.

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, 2025b. URL https://arxiv.org/abs/2412.04292.

Yikun Ji, Hong Yan, Jun Lan, Huijia Zhu, Weiqiang Wang, Qi Fan, Liqing Zhang, and Jianfu Zhang. Interpretable and reliable detection of ai-generated images via grounded reasoning in mllms, 2025. URL https://arxiv.org/abs/2506.07045.

Yikun Ji, Yan Hong, Bowen Deng, Jun Lan, Huijia Zhu, Weiqiang Wang, Liqing Zhang, and Jianfu Zhang. Locate-then-examine: Grounded region reasoning improves detection of ai-generated images, 2026. URL https://arxiv.org/abs/2510.04225.

Xiaojun Jia, Sensen Gao, Simeng Qin, Tianyu Pang, Chao Du, Yihao Huang, Xinfeng Li, Yiming Li, Bo Li, and Yang Liu. Adversarial attacks against closed-source mllms via feature optimal alignment, 2025. URL https://arxiv.org/abs/2505.21494.

Ronak Kosti, Jose Alvarez, Adria Recasens, and Agata Lapedriza. Context based emotion recognition using emotic dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019. ISSN 1939-3539. doi: 10.1109/tpami.2019.2916866. URL http://dx.doi.org/10.1109/TPAMI.2019.2916866.

Kartik Kuckreja, Parul Gupta, Muhammad Haris Khan, and Abhinav Dhall. Pixels don't lie (but your detector might): Bootstrapping mllm-as-a-judge for trustworthy deepfake detection and reasoning supervision, 2026. URL https://arxiv.org/abs/2602.19715.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, March 2020. ISSN 1573-1405. doi: 10.1007/s11263-020-01316-z. URL http://dx.doi.org/10.1007/s11263-020-01316-z.

Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Visual social relationship recognition, 2018a. URL https://arxiv.org/abs/1812.05917.

Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection, 2020. URL https://arxiv.org/abs/1912.13458.

Yixuan Li, Xuelin Liu, Xiaoyang Wang, Bu Sung Lee, Shiqi Wang, Anderson Rocha, and Weisi Lin. Fakebench: Probing explainable fake image detection via large multimodal models, 2024. URL https://arxiv.org/abs/2404.13306.

Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking, 2018b. URL https://arxiv.org/abs/1806.02877.

Q. Vera Liao and Kush R. Varshney. Human-centered explainable ai (xai): From algorithms to user experiences, 2022. URL https://arxiv.org/abs/2110.10790.

Q. Vera Liao, Daniel Gruen, and Sarah Miller. Questioning the ai: Informing design practices for explainable ai user experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pp. 1–15, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376590. URL https://doi.org/10.1145/3313831.3376590.

Rhema Linder, Sina Mohseni, Fan Yang, Shiva K. Pentyala, Eric D. Ragan, and Xia Ben Hu. How level of explanation detail affects human performance in interpretable intelligent systems: A study on explainable fact checking. *Applied AI Letters*, 2(4):e49, 2021. doi: https://doi.org/10.1002/ail2.49. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ail2.49.

Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain, 2021a. URL https://arxiv.org/abs/2103.01856.

Si Liu, Zitian Wang, Yulu Gao, Lejian Ren, Yue Liao, Guanghui Ren, Bo Li, and Shuicheng Yan. Human-centric relation segmentation: Dataset and solution, 2021b. URL https://arxiv.org/abs/2105.11168.

LLaVA Team. Llava-next: Improved reasoning, ocr, and world knowledge. Project Blog, 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, Xunliang Cai, Yayong Guan, and Jie Hu. Longcat-image technical report, 2025. URL https://arxiv.org/abs/2512.07584.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2018.07.007. URL https://www.sciencedirect.com/science/article/pii/S0004370218305988.

Abhijeet Narang, Parul Gupta, Liuyijia Su, and Abhinav Dhall. Laylens: Improving deepfake understanding through simplified explanations. In *Proceedings of the 27th International Conference on Multimodal Interaction*, ICMI '25, pp. 788–790, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714993. doi: 10.1145/3716553.3757093. URL https://doi.org/10.1145/3716553.3757093.

Seong Joon Oh, Rodrigo Benenson, Mario Fritz, and Bernt Schiele. Person recognition in personal photo collections, 2015. URL https://arxiv.org/abs/1509.03502.

OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence. OpenAI Blog, 2024. URL https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

OpenAI. Gpt image 1 model. OpenAI API Documentation, 2025a. URL https://developers.openai.com/api/docs/models/gpt-image-1. Accessed: 2026-05-06.

OpenAI. Gpt image 1.5 model documentation. OpenAI API Documentation, 2025b. URL https://developers.openai.com/api/docs/models/gpt-image-1.5.

OpenAI. Introducing gpt-oss. OpenAI Blog, 2025c. URL https://openai.com/index/introducing-gpt-oss/.

OpenAI. Gpt-5 mini model documentation. OpenAI API Documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-5-mini.

Anisha Pal, Julia Kruk, Mansi Phute, Manognya Bhattaram, Diyi Yang, Duen Horng Chau, and Judy Hoffman. Semi-truths: A large-scale dataset of ai-augmented images for evaluating robustness of ai-generated image detectors, 2024. URL https://arxiv.org/abs/2411.07472.

Jeongsoo Park and Andrew Owens. Community forensics: Using thousands of generators to train fake image detectors, 2025. URL https://arxiv.org/abs/2411.04125.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Mireia Ribera Turró and Agata Lapedriza. Can we do better explanations? a proposal of user-centered explainable ai. 03 2019.

Yao Rong, Tobias Leemann, Thai-Trang Nguyen, Lisa Fiedler, Peizhu Qian, Vaibhav Unhelkar, Tina Seidel, Gjergji Kasneci, and Enkelejda Kasneci. Towards human-centered explainable ai: A survey of user studies for model explanations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2104–2122, April 2024. ISSN 1939-3539. doi: 10.1109/tpami.2023.3331846. URL http://dx.doi.org/10.1109/TPAMI.2023.3331846.

Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images, 2019. URL https://arxiv.org/abs/1901.08971.

Junke Wang, Zuxuan Wu, Wenhao Ouyang, Xintong Han, Jingjing Chen, Yu-Gang Jiang, and Ser-Nam Li. M2tr: Multi-modal multi-scale transformers for deepfake detection. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*, ICMR '22, pp. 615–623, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392389. doi: 10.1145/3512527.3531415. URL https://doi.org/10.1145/3512527.3531415.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025. URL https://arxiv.org/abs/2508.18265.

Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection, 2023. URL https://arxiv.org/abs/2303.09295.

Haiquan Wen, Tianxiao Li, Zhenglin Huang, Yiwei He, and Guangliang Cheng. Busterx++: Towards unified cross-modal ai-generated content detection and explanation with mllm, 2026. URL https://arxiv.org/abs/2507.14632.

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, 2025. URL https://arxiv.org/abs/2503.14905.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

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, 2025. URL https://arxiv.org/abs/2410.02761.

Z.ai. Glm-4.6v: Open source multimodal models with native tool use. Z.ai Developer Documentation, 2025. URL https://docs.z.ai/guides/vlm/glm-4.6v.

Baogen Zhang, Sheng Li, Guorui Feng, Zhenxing Qian, and Xinpeng Zhang. Patch diffusion: a general module for face manipulation detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 3243–3251, 2022.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020. URL https://arxiv.org/abs/1904.09675.

Yue Zhang, Ben Colman, Xiao Guo, Ali Shahriyari, and Gaurav Bharaj. Common sense reasoning for deepfake detection, 2024. URL https://arxiv.org/abs/2402.00126.

Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer, 2025. URL https://arxiv.org/abs/2504.20690.

Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection, 2021. URL https://arxiv.org/abs/2103.02406.

Ziyin Zhou, Yunpeng Luo, Yuanchen Wu, Ke Sun, Jiayi Ji, Ke Yan, Shouhong Ding, Xiaoshuai Sun, Yunsheng Wu, and Rongrong Ji. Aigi-holmes: Towards explainable and generalizable ai-generated image detection via multimodal large language models, 2025. URL https://arxiv.org/abs/2507.02664.

Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. Genimage: A million-scale benchmark for detecting ai-generated image, 2023. URL https://arxiv.org/abs/2306.08571.

Appendix

XPlainVerse: A Million-Scale Benchmark for Explainable Deepfake Detection

Supplementary Material

A Appendix Roadmap

This appendix collects the details needed to understand and reproduce XPlainVerse without interrupting the main paper. We keep the construction details first, then the evaluation protocol, and place the full prompts at the end for readability.

Table 6: Appendix organization.

Section	Name	What it contains
Section B	Ethics	Data sourcing, safety checks, human annotation, data release, dual-use risk, and limitations.
Section C	Dataset	Real/fake counts, generator counts, split accounting, release layout, and ID/OOD notes.
Section D	Edit instructions	Edit types, instruction difficulty scores used for model routing, and representative examples.
Section E	Filtering	Difference extraction, edit-alignment judging, failure categories, and retained and rejected examples.
Section F	Perturbations	Perturbation list, parameter ranges, mixture protocol, and clean-perturbed examples.
Section G	Explanations	Complex, simple, authenticity, and silver-standard example checks.
Section H	Human study	Annotation protocol and seven-stage evaluation design.
Section I	Metrics	Entity/Evidence extraction format, coverage rules, and metric configuration.
Section J	Baselines	Zero-shot prompting, fine-tuning settings, and output parsing.
Section K	Prompts	Full prompt text for captioning, edit generation, filtering, explanations, and metric judging.

B Ethics Statement

This section documents the ethical practices followed during the construction of XPlainVerse, covering data sourcing, human participation, compensation, IRB oversight, dual-use risk, and release policy. Both the human annotation study and the user evaluation described below were conducted under the same institutional review board approval shared across the affiliated institutions, consistent with the widely used synthetic media datasets, like FaceForensics (Rössler et al., 2019) and DeepfakeJudge (Kuckreja et al., 2026).

Real-image data. The real subset draws from five publicly available datasets: OpenImages V7, EMOTIC, PIPA, PIC 2.0, and PISC. Each source is used in accordance with its published research license, and all five are cited in the main paper. No images were scraped from sources without a public data release. We did not collect any new photographs of individuals.

Synthetic and edited image data. Manipulated images were generated using commercial image-editing APIs, including Gemini, GPT-Image-1 and GPT-Image-1.5, SeedDream, Flux-Dev, HunyuanImage, LongCat, Wan, Nano Banana, and Qwen-Edit, solely for academic research. Additional manipulated candidates came from MultiFakeVerse, which was also produced for research under a compatible license. All candidate images passed through three filtering stages before inclusion: a semantic filter on the generation prompt to remove fantasy and non-photorealistic content, an automated NSFW classifier, and manual spot-checks by the authors. The filtering pipeline described

in Section 3 and Appendix E removes candidates where the intended edit is absent or the image is dominated by unintended changes; this step also incidentally removes many artifacts that could make images distressing or inappropriate for annotation. No generated image depicting identifiable private individuals was retained.

Human annotation for gold-reference explanations. Twelve annotators wrote explanations for 2,000 manipulated images. All annotators were adults recruited through academic networks and affiliated with university research groups. Before independent work began, each annotator reviewed written task instructions and worked through example images with the research team to calibrate judgment and establish consistent standards. Annotators were assigned non-overlapping subsets and compensated at rates commensurate with local academic norms. No personally identifying information was collected. Participation was voluntary, and annotators could withdraw at any point without consequence. All materials were screened to exclude NSFW or distressing content before annotation began. Inter-annotator agreement is assessed on 50 images, each independently annotated by eight annotators, yielding 400 annotations. Mean EntityScore and EvidenceScore agreement is 0.519 and 0.491, respectively.

Human evaluation study. Thirty adult participants took part in the seven-stage human evaluation described in Appendix H. Participants were recruited from research groups at the authors' institutions. Before participating, each person received a written statement explaining the study's purpose and procedures, with worked examples of each annotation task, and gave informed consent through a study form. Presentation order was randomized, and examples were balanced across sources and generators. No personally identifying data were recorded, and all responses were anonymized before analysis. Participants were compensated for their time. All study materials were reviewed and cleared for NSFW and distressing content before use.

IRB approval. All human participation described above, including both annotation phases and the user evaluation, was conducted under institutional review board approval shared across the affiliated institutions. Participation was voluntary throughout. No special populations were recruited, and no participant was placed at foreseeable risk of harm. Compensation rates were determined before the study began and did not depend on the labels, ratings, or judgments participants produced.

Data and model release. XPlainVerse benchmark assets, including metadata, split files, prompts, generated explanations, human annotations, perturbation metadata, and evaluation protocols, are released under XPlainVerse End User License Agreement (EULA). The EULA permits use only for academic, non-commercial and not-for-profit research. Redistribution, public hosting, mirroring, sublicensing, commercial use, and release of arbitrary dataset images or derived dataset variants are prohibited without prior written approval. Third-party source datasets, images, generated images governed by provider terms, model checkpoints, and software remain subject to their original licenses or provider terms. Where redistribution or relicensing of a third-party asset is not permitted, we release compatible metadata, identifiers, annotations, or XPlainVerse-specific derived components rather than relicensing the original asset.

Dual-use risk and mitigations. XPlainVerse has potential dual-use risks, since examples of difficult manipulations and detector failures could inform attempts to create harder-to-detect synthetic media. We mitigate this through controlled access: the dataset is distributed under a strict research-only EULA for non-commercial academic use, with redistribution, public hosting, commercial use, and harmful or deceptive applications prohibited. The real images come from existing public or research datasets, and the manipulated images are produced using off-the-shelf editing systems; we do not release generation model weights, service credentials, or tools intended to create stronger forgeries. We release XPlainVerse as a defensive benchmark for training, evaluating, and explaining deepfake detectors, and believe its primary benefit is to support trustworthy detection research.

Limitations. XPlainVerse covers still images only; video manipulation is outside its scope. The annotation pool of twelve people and thirty evaluators is geographically and institutionally concentrated, which may introduce cultural or stylistic biases in the human-reference explanations. The explanation generation pipeline relies on a small set of openly released and proprietary VLMs; any systematic tendencies in those models, such as preferences for certain types of cues or certain phrasings, may carry over into the silver-standard explanations. The NSFW filters and manual checks reduce but do not eliminate the possibility that distressing content could appear in the data.

Table 7: Existing assets used in XPlainVerse and their license/terms-of-use handling.

Asset	Use in XPlainVerse	License / terms	Compliance and redistribution
XPlainVerse release	New benchmark metadata, split files, prompts, generated explanations, human annotations, and evaluation protocol.	XPlainVerse End User License Agreement (EULA): academic, non-commercial, not-for-profit research-only use; no redistribution, public hosting, sublicensing, or commercial use without prior written approval. Third-party source assets remain under their original licenses or provider terms.	Distributed through controlled access. Users must accept the EULA, cite the dataset/challenge paper and relevant upstream sources, comply with all upstream terms, and use the data only for approved academic research. Where source/provider terms restrict redistribution, we release compatible metadata, identifiers, annotations, or XPlainVerse-specific derived components rather than restricted assets.
OpenImages V7	Source of real images and annotations.	Images: Creative Commons Attribution 2.0 (CC BY 2.0); annotations: Creative Commons Attribution 4.0 (CC BY 4.0).	Original images and annotations remain subject to their source Creative Commons licenses. We retain source attribution/license metadata where available and follow the applicable CC BY attribution requirements. Associated XPlainVerse metadata, annotations, explanations, and splits are distributed under the XPlainVerse EULA, subject to upstream terms.
EMOTIC	Source of real images.	Non-commercial research and educational use.	Used only for academic research. Original source is cited. EMOTIC source assets remain subject to their original non-commercial terms. Associated XPlainVerse metadata, annotations, explanations, and splits are distributed under the XPlainVerse EULA where compatible with EMOTIC terms.
PIPA	Source of real images.	Flickr/source image-level licenses and terms; no single dataset-wide image license identified.	Used only where access and reuse are compatible with the corresponding source license. Where redistribution is not permitted, released files reference source identifiers or metadata rather than relicensing the original images. Associated XPlainVerse metadata, annotations, explanations, and splits are distributed under the XPlainVerse EULA where compatible with source terms.
PIC 2.0	Source of real images.	Non-commercial use for academic research, teaching, scientific publication, and personal experimentation.	Used for research purposes only. Original source is cited. PIC 2.0 source assets remain subject to their original non-commercial terms. Associated XPlainVerse metadata, annotations, explanations, and splits are distributed under the XPlainVerse EULA where compatible with PIC 2.0 terms.
PISC	Source of real images.	Creative Commons Attribution 4.0 International (CC BY 4.0).	Original PISC assets remain subject to CC BY 4.0, with attribution to the original authors and source. Associated XPlainVerse metadata, annotations, explanations, and splits are distributed under the XPlainVerse EULA, with required upstream attribution.
MultiFakeVerse	Source of manipulated candidates.	MultiFakeVerse EULA / research-only access terms; non-commercial academic use only, no redistribution or modification without permission, and compliance with upstream PIC, PISC, PIPA, and EMOTIC terms required.	MultiFakeVerse assets remain under their original access terms. Associated XPlainVerse metadata, annotations, explanations, and splits are distributed under the XPlainVerse EULA where compatible.
Generation/editing APIs	Candidate manipulated-image generation.	Official provider API/service terms for GPT-Image, Gemini/Nano Banana, Seedream, Wan, and related closed-source systems.	Accessed through official APIs. Generated outputs and associated XPlainVerse metadata or annotations are distributed under the XPlainVerse EULA where provider terms permit; otherwise, only compatible metadata, identifiers, or annotations are released.

Continued on next page

Asset	Use in XPlainVerse	License / terms	Compliance and redistribution
Open-weight generation/editing models	Candidate manipulated-image generation.	Model-specific licenses, including FLUX non-commercial license for FLUX.2-dev, Tencent Hunyuan Community License for HunyuanImage, and Apache 2.0 for LongCat-Image and Qwen-Image/Edit where applicable.	Used according to model-specific licenses. Generated outputs and associated XPlainVerse metadata or annotations are distributed under the XPlainVerse EULA where model terms permit; otherwise, only compatible metadata or annotations are released.
Filtering and explanation VLMs	Captioning, filtering, explanation generation, and metric judging.	Official API terms for closed-source models; open-weight licenses where applicable, including Apache 2.0 or MIT-style licenses for relevant open models.	Used for dataset construction, filtering, evaluation, and explanation generation. Closed-source model weights are not redistributed. XPlainVerse textual outputs, annotations, and evaluation records are distributed under the XPlainVerse EULA where model/API terms permit.
Baseline model checkpoints	Zero-shot and fine-tuned baselines.	Individual model-card licenses and terms for Qwen, LLaVA, InternVL, BusterX++, and related VLM checkpoints.	Used for evaluation and fine-tuning under their respective model-card terms. The paper reports results and does not redistribute third-party model weights unless explicitly permitted.
Software libraries	Training, inference, feature extraction, and scoring.	Package-specific open-source licenses, including Apache 2.0 for `ms-swift`, `vLLM`, Hugging Face `transformers`, DINOv2, and SigLIP/Big Vision tooling; MIT for BERTScore and OpenAI CLIP.	Cited and used according to package licenses. The released code includes attribution and license notices where required.

C Dataset Assembly and Split Details

Dataset accounting. Edit-Check processes approximately 609K candidate fake edits and retains 265K fake base images. The real subset contains 235K base images. One perturbed counterpart is created for each base image, yielding $265\text{K} \times 2 = 530\text{K}$ final fake images and $235\text{K} \times 2 = 470\text{K}$ final real images, for 1.00M images in total. Each fake image has a complex and a simple explanation, while each real image has one authenticity explanation. Consequently, the total number of image–explanation pairs is $2 \times 530\text{K} + 470\text{K} = 1.53\text{M}$. Candidate edits, retained base images, final images, and image–explanation pairs therefore denote different quantities.

Real-Image Distribution The real-image subset contains 470,000 final images from five source corpora, comprising 235,000 base images and their perturbed counterparts. Table 8 reports the final real data split.

Table 8: Final real-image source inventory by split.

Source	Train	Val.	Test	Total images
OpenImages	200,000	28,000	80,000	308,000
PIPA	43,106	7,130	3,244	53,480
PISC	35,382	5,974	2,730	44,086
EMOTIC	33,398	5,918	2,604	41,920
PIC 2.0	18,114	2,978	1,422	22,514
Total	**330,000**	**50,000**	**90,000**	**470,000**

Fake-Image Distribution The fake-image subset contains 530,000 final images from twelve generators, comprising 265,000 retained base images and their perturbed counterparts. Table 9 gives the final retained counts by split. Candidate and discard counts are recomputed from the visual-difference and judge records described in Appendix E; the released benchmark includes only retained fake samples unless otherwise noted for analysis.

Final Split Composition and ID/OOD Assignment Table 10 shows the final data-split across the train, validation, and test sets. Training and validation are used for model development. The test split emphasizes held-out or newer generator families and is used as the primary OOD evaluation. Our ID/OOD evaluation is defined with respect to generator families. Manipulated images produced by generators represented during training are treated as ID, while images produced by held-out

Table 9: Final retained fake-image inventory by generation model.

Generator	Train	Val.	Test	Total
Gemini-2.0-flash	354,256	19,612	29,438	403,306
Flux Dev	–	–	53,222	53,222
Qwen 2511	–	22,296	6,422	28,718
Longcat Turbo	–	11,282	3,140	14,422
Hunyuan	–	6,422	7,040	13,462
ICEdit	3,050	186	304	3,540
GPT-Image-1	2,694	202	258	3,154
Nano Banana 2	–	–	3,260	3,260
Wan 2.6	–	–	2,190	2,190
Seedream 4.5	–	–	1,790	1,790
Nano Banana Pro	–	–	1,500	1,500
Gpt 1.5	–	–	1,436	1,436
Total	**360,000**	**60,000**	**110,000**	**530,000**

generators are treated as OOD. Accordingly, ID/OOD assignment follows the generator used to produce each manipulated image; source-image identity is not used to determine ID/OOD status. The goal of this protocol is to measure generalization across manipulation generators.

Table 10: Final split composition used in the paper.

Split	Real images	Fake images	Total images	Complex explanations	Simple explanations
Train	330,000	360,000	690,000	360,000	360,000
Validation	50,000	60,000	110,000	60,000	60,000
Test	90,000	110,000	200,000	110,000	110,000
Total	470,000	530,000	1,000,000	530,000	530,000

D Edit Instruction Qualitative Examples

Table 11 gives representative edit-instruction examples used for data generation in XPlainVerse.

E Quality Filtering

This section documents how the multi-stage filtering pipeline behaves across generators, edit types, and failure modes. It also includes qualitative examples of retained and discarded candidates.

Filtering Decision Rules Filtering is applied to each generated candidate using two structured records: a visual-difference record and an alignment-judge record. The visual-difference extractor compares the paired original and edited images and lists only visible, localized before/after differences. The alignment judge then compares the requested edit instruction against the difference record and returns a final score, a match score, an unrelated-change penalty, and a discrete verdict.

A candidate is retained when the generated image visibly realizes the requested edit and the difference JSON provides localized evidence that can support an explanation. Candidates are rejected when the target edit is missing, dominated by off-target scene rewrites, affected by severe low-level artifacts, or too ambiguous for consistent explanation supervision.

Failure Categories for Discarded Candidates Table 12 lists the main reasons a generated candidate can be rejected during filtering.

Filtering records. Each generated candidate is audited with the same two records: a visual-difference JSON comparing the paired original and edited images, and an edit-alignment judge

Table 11: Representative edit-instruction examples by edit type.

Edit type	Example instruction	Why it can be hard
Object change	**Scene:** people in helmets and orange life jackets seated in a blue inflatable raft. **Instruction:** Replace one of the paddles in the raft with a long, thin surfboard, adjusting the angle to make it look naturally placed. **Target:** paddles; blue inflatable raft.	Local replacement with contact, depth, and shadow constraints.
Move or resize	**Scene:** a retro computer desk with a framed photo and a small potted plant. **Instruction:** Resize the potted plant to be significantly larger, making it appear as if it is taking up more space on the desk. **Target:** potted plant.	The resized object must keep its identity, position, shadows, and occlusion relationships.
Copy-move	**Scene:** musicians in gray jackets playing brass instruments in a crowd. **Instruction:** Duplicate the musician on the right and place the duplicate slightly behind the original, adjusting the angle to create a sense of depth. **Target:** musician on the right.	Requires identity preservation, shadows, and depth ordering.
Shape change	**Scene:** a white hatchback car parked on pavement with trees in the background. **Instruction:** Replace the sporty alloy wheels with oversized, cartoonish wheels that are unnaturally large and round, while keeping the car body consistent. **Target:** white hatchback car; sporty alloy wheels.	Shape edit with strict wheel-well, ground-contact, and material constraints.
Depth-order change	**Scene:** a black coupe parked on a paved lot with visible tire and bumper details. **Instruction:** Introduce a small, realistic puddle of water on the paved lot near the front tire of the coupe, reflecting the car and surrounding environment. **Target:** paved lot.	The new object must sit below the car while preserving believable reflections and contact cues.
Text edit	**Scene:** large white letters spelling `BONDI BEACH` in front of tall city buildings. **Instruction:** Replace the letters spelling `BONDI BEACH` with `CITY PARK` in the same large white font, ensuring the new text fits seamlessly. **Target:** large white letters.	Text must remain legible, scaled correctly, and physically integrated with the structure.
Planar warp	**Scene:** the Gibeau Orange Julep sign mounted below a large orange icon. **Instruction:** Apply a local planar perspective warp to the black and white sign to make it appear slightly tilted, revealing a portion of the brick wall behind it. **Target:** black and white sign; brick building.	Planar geometry must remain locally plausible after the warp.

Table 12: Failure categories for discarded fake candidates.

Failure mode	Discard criterion
Missing edit	The requested target edit is not visibly present, or the difference extractor cannot identify any concrete on-target before/after evidence.
Partial edit	The output contains a weak fragment of the intended edit but misses the core semantic change.
Prominent off-target change	An unrelated addition, removal, replacement, identity shift, or text/content change is one of the dominant visible differences.
Identity / attribute drift	A person, object, or key attribute changes beyond the requested local manipulation.
Background or layout rewrite	Camera framing, background, or scene structure changes enough that the sample no longer isolates the requested manipulation.
Low-level artifact	Severe warping, boundary errors, text corruption, blur, or rendering artifacts prevent reliable explanation supervision.
Unsafe / ambiguous sample	The target region is too small, occluded, ambiguous, or unsuitable for release and evaluation.

JSON comparing the requested edit against the extracted differences. The judge returns `KEEP`, `BORDERLINE`, or `DISCARD`, along with match and off-target-change scores. Final split counts are computed after this unified filtering pass and split assignment.

Visual-Difference Examples Figure 7 shows two visual-difference records tied to visible evidence.



Original

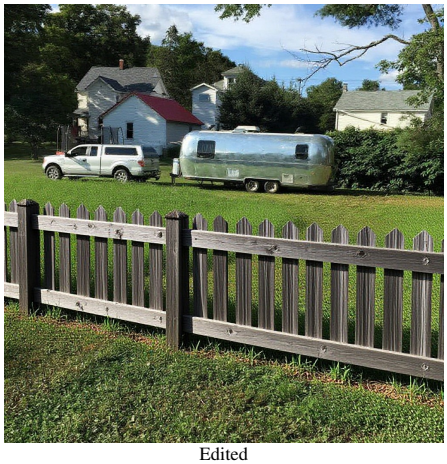


Edited

Summary: A large, colorful feathered hat is added on the model's head.
Visual Difference: `Hat`, salience 5: A: No hat on the model's head; B: A large, colorful feathered hat is present on the model's head.



Original



Edited

Summary: The primary difference is the replacement of a stone wall in the foreground with a wooden picket fence, along with changes to the grass texture and height in the immediate foreground.
Visual Difference: `Foreground barrier`, salience 5: A: A stacked stone wall is visible on the right side of the foreground; B: A wooden picket fence spans across the entire foreground.
`Grass in foreground`, salience 4: A: The grass is tall and unevenly cut, with a distinct mowed line separating long and short sections; B: The grass is shorter, more uniform in height, and lacks the distinct mowed line seen in Image A.

Figure 7: Visual-difference extraction examples. The examples show that the pipeline records localized before/after evidence rather than relying on generic descriptions of manipulation.

Judge Examples Figure 8 shows `KEEP` and `DISCARD` decisions.



Original



Edited

Edit: Add a small realistic puddle near the front tire of the coupe, with reflections of the car and surroundings.
Visual Difference: A dark puddle appears under the front bumper, with reflected car details visible in the water.
Judge Score/rationale: match=5, penalty=0, final=5. The added puddle and its local reflections directly support the requested edit.



Original



Edited

Edit: Tilt the grandstand slightly right and reveal a hidden event banner behind it.
Visual Difference: A large car-show banner appears. A new roof-like support structure is also added.
Judge Score/rationale: match=4, penalty=2, final=2. The banner is present, but the grandstand is not tilted. And the new support frame is a prominent off-target change.

Figure 8: Judge scoring and rationale examples.

F Perturbation Suite

This section summarizes the perturbations applied to XPlainVerse. Corruptions are sampled with deterministic per-split schedules, and adversarial stress tests include FOA-style and xTransfer attacks. The validation set builds upon the training perturbation set, and the test set further expands the validation and training set with additional adversarial attacks and corruption methods. Table 13 groups the perturbations by the kind of image change they introduce, while Table 14 groups perturbations by train, validation, and test split. Table 15 gives representative parameter ranges of each perturbation family for reproducibility. Figure 9 shows three pairs of clean images with their perturbed versions under different perturbation settings.

Table 15: Parameter ranges for individual perturbations. Rows use the perturbation names and split membership from `add_noise_for_mapping_fast_skip_existing.py`; FOA and xTransfer are generated by separate adversarial pipelines.

Perturbation	Split(s)	Parameter range	Implementation notes
`jpeg_recompression`	Train / Val. / Test	JPEG quality 50–85	OpenCV JPEG encode/decode
`gaussian_noise`	Train / Val. / Test	mean 0.0; standard deviation 7.0–15.0	Additive pixel-space Gaussian noise

Continued on next page

Table 15: Parameter ranges for individual perturbations (continued).

Perturbation	Split(s)	Parameter range	Implementation notes
poisson_noise	Train / Val. / Test	Poisson peak 40.0–100.0	Poisson sampling after scaling image to $[0, 1]$
speckle_noise	Train / Val. / Test	mean 0.0; standard deviation 0.05–0.12	Multiplicative speckle noise
gaussian_blur	Train / Val. / Test	kernel 3 or 5; σ 0.8–1.5	Full-image Gaussian blur
sharpen	Train / Val. / Test	amount 0.5–1.0	Unsharp masking using a Gaussian-smoothed image
bit_depth_reduction	Train / Val. / Test	bits 4, 5, or 6	Uniform quantization of RGB intensities
gamma_tone_curve	Train / Val. / Test	gamma 0.85–1.25; gain 0.95–1.05	Global tone curve
chromatic_aberration	Train / Val. / Test	channel shift 1 px	Shifts red and blue channels in opposite directions
vignetting	Train / Val. / Test	radial Gaussian σ from image width$/3$ to image width	Runtime default in the perturbation function
color_channel_mixing	Train / Val. / Test	alpha 0.02–0.05; beta 0.02–0.05	Small RGB channel crosstalk
color_jitter	Train / Val. / Test	saturation factor 0.85–1.15; hue shift ± 5; value factor 0.9–1.1	HSV-space color jitter
sparse_pixel_flip	Train / Val. / Test	density 0.00005–0.0003; mode lsb_flip or delta; bit 0 or 1; max delta 1 or 2	Sparse near-invisible pixel edits
tiny_local_blur	Train / Val. / Test	relative patch size 0.015–0.03; kernel 3 or 5; σ 0.8–1.5; alpha 0.8–1.0	Blends a small local patch with its blurred version
lsb_intensity_perturbati	Train / Val. / Test	density 0.002–0.01; mode signed_delta or bit_toggle; step 1 or 2; bit 0 or 1	Dense but tiny intensity perturbation
double_jpeg	Val. / Test	first JPEG quality 82–92; second JPEG quality 65–82	Two-stage JPEG recompression
mild_denoise	Val. / Test	method bilateral, median, or nlm; bilateral $d = 5$, σ_{color} 20–40, σ_{space} 20–40; median kernel 3 or 5; NLM h and h_{color} 3–7	Restoration-like denoising
motion_blur	Val. / Test	kernel 7, 9, or 11; angle 0–180 degrees	Directional blur kernel
defocus_blur	Val. / Test	disk radius 3, 4, or 5	Defocus-style disk blur
tiny_local_jpeg_corruptiv	Val. / Test	relative patch size 0.012–0.03; JPEG quality 35–65; number of patches 1, 2, or 3	Local JPEG artifacts in tiny regions
tiny_local_mask	Val. / Test	relative patch size 0.012–0.03; mode blur_fill, neighbor_copy, or mean_fill; blur-fill kernel 5 and σ 1.0–2.0	Small local masking or replacement
patch_replacement_nearby	Val. / Test	relative patch size 0.012–0.03; max offset fraction 0.03–0.08; blend alpha 0.85–1.0	Replaces a tiny patch with a nearby patch
grayscale_or_desaturate	Test	mode full_gray or partial_desat; partial desaturation factor 0.2–0.8	Held-out color-removal stressor
resize_resample	Test	resize scale 0.6–0.9	Downsample with area interpolation, then upsample with nearest, linear, or cubic interpolation
small_geometric	Test	rotation ± 3 degrees; scale 0.97–1.03; x/y translation fraction ± 0.02	Small affine transform with reflected borders
local_block_compression	Test	block size 32 or 64; JPEG quality 60–85; number of blocks 2–5	Local block-level JPEG artifacts
ringing_halo	Test	σ 1.2–2.2; amount 0.25–0.6; edge boost 0.7–1.0	Edge-conditioned halo/ringing artifact
tiny_patch_shuffle	Test	relative patch size 0.015–0.03; block size 2 or 4	Shuffles blocks within a tiny local patch
checkerboard_highfreq	Test	strength 0.8–1.8; period 1 or 2; alpha 1.0	Weak checkerboard high-frequency residual
fourier_domain_perturbat	Test	amplitude 0.3–1.0; low cutoff 0.35–0.5; high cutoff $\max(\text{low} + 0.1, 0.85)$–1.0	Band-limited Fourier-domain residual
Separate adversarial pipelines			
xTransfer	Train / Val. / Test	ℓ_∞ non-targeted, $\epsilon = 12/255$; base/mid/large attacker ratio 0.2/0.2/0.6; seed 42	XTransferBench mapping pipeline
FOA-style attack	Val. / Test	MIFGSM, $\epsilon = 8$, 100 steps, $\alpha = 1$, input resolution 224; seed 2023	CLIP B16/B32/LAION surrogate ensemble, cluster number 10

Mixture Protocol For each image, the perturbation sampler uses one of three schedules: *single*, *pair*, or *chunk*. A *single* schedule samples one operation from the split-specific allowed set. A *pair* schedule samples two random operations from the allowed set. A *chunk* schedule samples a larger set without replacement from all operations allowed for the split. The default schedule probabilities are 85/10/5 for train, 75/15/10 for validation, and 70/20/10 for test, with chunk sizes increasing from 6-8

Table 13: Perturbation inventory by group. Validation includes all training operations; test includes all validation operations.

Group	Operations
Noise and pixel-level changes	Gaussian noise; Poisson noise; speckle noise; sparse pixel flip; LSB intensity perturbation
Blur and restoration-like changes	Gaussian blur; sharpen; tiny local blur; mild denoise; motion blur; defocus blur
Compression	JPEG recompression; double JPEG; local block JPEG/compression
Color and tone changes	Bit-depth reduction; tone curve; chromatic aberration; vignetting; channel mixing; color jitter; grayscale/desaturation
Adversarial stress tests	FOA-style attack; xTransfer attack
Local patch changes	Tiny local JPEG corruption; tiny local mask; nearby patch replacement; tiny patch shuffle
Geometry and resampling	Resize/resample; small rotation/scale/translation transform
Frequency artifacts	Ringing halo; checkerboard high-frequency pattern; Fourier-domain perturbation

Table 14: Perturbation distribution by data split.

Split	Single operation	Pair	Chunk
Train	85%	10%	5% of 6–8 operations
Validation	75%	15%	10% of 8–12 operations
Test	70%	20%	10% of 13–17 operations

to 8-12 to 13-17 operations, as specified in Table 14. This design makes the perturbation distribution cumulative while increasing severity at validation and test time.

G Explanation Data Details and Qualitative Examples

This section illustrates the explanation supervision provided in XPlainVerse. It includes complex explanations and simple explanations for fake images, and authenticity explanations for real images.

Explanation Types and Acceptance Criteria Table 16 summarizes the explanation targets used in the dataset.

Table 16: Explanation types in XPlainVerse and criteria for acceptable outputs.

Explanation type	Audience	Expected content	Typical length
Complex fake explanation	Technical / forensic user	Multiple concrete visual cues, manipulated entities, physical inconsistencies, localized evidence, and minimal unsupported speculation	2-4 sentences
Simple fake explanation	Non-technical user	One salient reason in plain language; no technical jargon; accessible wording	1-2 sentences
Authenticity explanation	General user	Natural details supporting perceived authenticity, e.g., physical contact, lighting, texture, small imperfections, and object interactions	1-3 sentences
Human reference explanation	Gold-standard evaluation subset	Human-written visible manipulation cues grounded in fake-image evidence	2-4 sentences

Qualitative Examples for Reasoning Figure 10 provides fake-image explanation examples while Figure 11 shows real-image explanation examples in XPlainVerse.



Clean



Perturbed

Pair perturbation example using `gaussian_blur` and `tiny_local_jpeg_corruption`.



Clean



Perturbed

Chunk perturbation example combining `color_jitter`, `gamma_tone_curve`, `gaussian_noise`, and `resize_resample` etc.



Clean



Perturbed

Single perturbation example using `checkerboard_highfreq`.

Figure 9: Qualitative perturbation examples. Perturbations alter low-level image statistics while preserving the semantic content needed for detection and explanation evaluation.



Original



Edited

Edit/diff: Add a large construction crane in the background, partially obscured by the partially demolished building, ensuring it blends with the existing lighting and perspective.

Complex: The image contains several visual inconsistencies that suggest manipulation. Most notably, the large yellow construction crane in the center background appears to be floating; its base does not realistically connect to the ground or the surrounding construction debris, and it lacks any contact shadows on the terrain below. The crane's lattice structure also shows signs of digital artifacts, with some of the thin metal beams appearing slightly blurred or disconnected against the blue sky. Additionally, the lighting on the crane is overly uniform and does not perfectly match the harsh, directional sunlight seen on the buildings and taxis in the foreground, making it look like a flat overlay rather than a physical part of the scene.

Simple: The tall yellow crane in the middle looks like a sticker that was placed on top of the picture. Its bottom doesn't touch the ground or sit on the dirt like a real heavy machine should, so it looks like it is just floating in the air.



Original



Edited

Edit/diff: Replace the microphone labeled '14' with a microphone labeled '99', ensuring the new label matches the original style and lighting.

Complex: The image contains a subtle digital artifact on the microphone flag held in the center. The number '99' appears unnaturally sharp and flat compared to the textured, slightly weathered surface of the black flag it is printed on. Additionally, the white strokes of the numbers lack the consistent grain and soft lighting fall-off seen on the adjacent colorful logo, making the digits look like a clean digital overlay rather than a physical part of the equipment.

Simple: The number 99 on the microphone looks like a white sticker that was just pasted on top. It is too clean and flat, while the rest of the microphone looks like a real, worn-out object.

Figure 10: Fake-image explanation examples. **Complex** explanations preserve multiple grounded cues; **Simple** explanations select the single clearest cue and use non-technical language.



Real image

This looks real because the road isn't perfect. You can see little cracks and spots on it, like a real road that people ride bikes on every day.



Real image

This looks real because the girl's hair is a little bit messy and sticking out, like real hair does.

Figure 11: Authenticity explanation examples for real images. Each explanation should point to visible physical evidence rather than merely asserting that the image is real.

H Human Annotation and Evaluation

The main paper reports aggregate human-evaluation findings. This appendix section discusses the human annotation and user study in detail.

Human Reference Annotation Protocol Twelve annotators wrote explanations for 2,000 randomly selected manipulated images. Each task displayed the original and edited images side by side. Annotators described visible manipulation cues in the edited image, using the original only for context and avoiding unsupported claims. Inter-annotator consistency was assessed on 50 shared images, with eight independent annotations per image.

Seven-Stage Human Evaluation Protocol Thirty adult evaluators recruited from research groups at the authors' institutions participated in the study. None of these evaluators participated in the reference annotation. Presentation order was randomized, and evaluation examples were balanced across sources and generators. Table 17 summarizes the tasks, while Table 18 reports stage-specific sample counts and sample-balanced results. Recruitment and ethics details are provided in Appendix B.

Table 17: Seven-stage human evaluation protocol.

Stage	Goal	Questions / criteria	Response type
1. Image realism	Judge whether individual images appear real or manipulated	Real/fake judgment, confidence, visual realism	Choice + 1–5 scale
2. Edit quality	Evaluate paired original and edited images	Core edit presence, edit realism, unrelated off-target changes	1–5 scale
3. Explanation quality	Rate one explanation for one image	Grounding, specificity, completeness, convincingness, unsupported claims	1–5 scale
4. Explanation preference	Compare two explanations for the same image	Grounding, specificity, completeness, fewer unsupported claims, overall usefulness	A/B/Tie
5. Human vs. generated comparison	Compare human-written and generated complex explanations	Grounding, specificity, completeness, fewer unsupported claims, overall preference	A/B/Tie
6. Readability	Evaluate simple/user-facing explanations	Ease of understanding, memorability, cognitive load, liking	1–5 scale
7. Audience usefulness	Compare explanations for different users	Personal preference, usefulness for ordinary users, usefulness for technical analysts, clarity/evidence balance, default system choice	Choice

Table 18: Human evaluation results by stage. Values are sample-balanced: we first aggregate responses within each evaluated sample and then average across samples. The n column reports unique samples / total ratings. For pairwise rows, percentages are ordered as the listed choice columns.

Stage	Measure	n	Result
1	Real images	84 / 253	Judged real 66.8%, manipulated 28.2%, not sure 5.0%; realism mean 3.97/5.
1	Retained fake images	83 / 247	Judged real 57.4%, manipulated 35.4%, not sure 7.2%; realism mean 3.72/5.
1	Discarded fake images	33 / 100	Judged real 27.7%, manipulated 64.9%, not sure 7.4%; realism mean 2.98/5.
2	Filtered edit quality	50 / 154	Composite 3.77/5; edit presence 3.95/5; edit realism 3.76/5; off-target changes 2.38/5.
2	Discarded edit quality	50 / 146	Composite 2.66/5; edit presence 2.80/5; edit realism 2.86/5; off-target changes 3.68/5.
3	Complex explanation quality	118 / 360	Composite 3.86/5; grounding 4.05; specificity 4.03; completeness 3.82; convincingness 3.80; unsupported claims 2.40 (lower is better).
4	Complex vs. SIDA: grounding	118 / 360	Complex 64.9%, tie 6.2%, SIDA 28.9%; $\Delta = +36.0$.
4	Complex vs. SIDA: specificity	118 / 360	Complex 58.7%, tie 5.7%, SIDA 35.6%; $\Delta = +23.1$.
4	Complex vs. SIDA: completeness	118 / 360	Complex 57.9%, tie 6.3%, SIDA 35.8%; $\Delta = +22.1$.
4	Complex vs. SIDA: fewer unsupported claims	118 / 360	Complex 54.9%, tie 14.0%, SIDA 31.1%; $\Delta = +23.7$.
4	Complex vs. SIDA: overall choice	118 / 360	Complex 59.2%, tie 8.8%, SIDA 32.0%; $\Delta = +27.2$.
5	Complex vs. human: grounding	120 / 360	Complex 63.7%, tie 13.3%, human 23.0%; $\Delta = +40.7$.
5	Complex vs. human: specificity	120 / 360	Complex 61.6%, tie 12.7%, human 25.7%; $\Delta = +35.9$.
5	Complex vs. human: completeness	120 / 360	Complex 63.2%, tie 14.2%, human 22.6%; $\Delta = +40.6$.
5	Complex vs. human: fewer unsupported claims	120 / 360	Complex 58.0%, tie 17.3%, human 24.7%; $\Delta = +33.4$.
5	Complex vs. human: overall preference	120 / 360	Complex 59.5%, tie 15.2%, human 25.3%; $\Delta = +34.2$.
6	Simple explanation readability	119 / 360	Composite 3.90/5; ease of understanding 3.99; memorability 3.97; cognitive load 2.13 (lower is better); liking 3.77.
7	Personal preference	119 / 360	Simple 30.0%, complex 53.1%, both/either 10.3%, neither 6.5%.
7	Ordinary-user usefulness	119 / 360	Simple 48.5%, complex 35.5%, both/either 8.8%, neither 7.2%.
7	Technical-analysis usefulness	119 / 360	Simple 12.3%, complex 72.3%, both/either 9.0%, neither 6.4%.
7	Evidence/readability balance	119 / 360	Simple 15.3%, complex 58.6%, both/either 15.5%, neither 10.6%.
7	Default system choice	119 / 360	Simple 27.0%, complex 47.3%, both/either 16.4%, neither 9.3%.

Image-level preference analysis. Across the 120 human–generated explanation pairs, generated explanations receive more votes than human explanations on 80 images for grounding, 81 for completeness, and 74 for fewer unsupported claims; human explanations receive more votes on 24, 22, and 27 images, respectively. The remaining 16, 17, and 19 images have equal generated/human vote counts.

Participant-Facing Instructions and Interface We reproduce below the participant-facing instructions used in the gold-reference annotation task and the human-evaluation study. The study was administered through a form-based interface. All participants gave informed consent before beginning the task. Compensation rates were determined before the study began, were at or above the applicable local minimum-wage equivalent, and did not depend on the labels, ratings, or judgments participants produced.

Gold-reference annotation instructions. Annotators were shown the original image and the edited image side by side. The instruction text was:

> You will be shown an original image and an edited image. Your task is to write an explanation describing why the edited image appears manipulated. Focus only on visible evidence in the edited image. You may use the original image for context, but your explanation should refer to cues that can be observed in the edited image itself. Mention concrete regions, objects, people, boundaries, textures, lighting, geometry, or other visible artifacts. Do not speculate about hidden causes, model names, or metadata. If multiple cues are visible, describe all important ones clearly.

Human-evaluation instructions. Participants in the human-evaluation study received the following general instruction:

> You will evaluate images and explanations produced for a deepfake-detection benchmark. Some images are real and some are manipulated. Please answer based only on what is visible in the image and the explanation shown to you. There are no trick questions. Use your best judgment, and select "not sure" or a middle score when the evidence is unclear. Do not include any personally identifying information in your responses.

Stage-specific instructions. For image realism, participants judged whether each image appeared real, manipulated, or uncertain, and rated visual realism on a 1–5 scale. For edit quality, participants compared the original and edited images and rated whether the intended edit was present, visually realistic, and free of unrelated off-target changes. For explanation quality, participants rated grounding, specificity, completeness, convincingness, and unsupported claims. For pairwise comparisons, participants chose which explanation was better or selected tie. For readability, participants rated ease of understanding, memorability, cognitive load, and liking. For audience usefulness, participants judged which explanation was more useful for ordinary users and for technical analysts.

I Metric Implementation Details

This section describes the implementation details for the automatic metrics used to evaluate XPlain-Verse explanations. For complex explanations, we evaluate whether a generated explanation identifies the same manipulated regions and recovers the same supporting evidence described in the reference. We therefore compute two intent-aware metrics: *EntityScore*, which measures coverage of diagnostic image entities, and *EvidenceScore*, which measures coverage of concrete visual evidence claims. Both metrics are reference-based: they evaluate agreement in manipulation-relevant entities and evidence claims between the predicted and reference explanations, rather than relying only on overall sentence-level similarity. Both metrics use an offline Qwen3.5-4B evaluator for structured extraction and semantic coverage judging. We additionally report BERTScore-F1 as a complementary measure of full-sentence semantic similarity. For simple explanations, we report BERTScore-F1 for semantic fidelity and a normalized SLE score for simplicity.

Entity and evidence extraction. Given a predicted explanation y^p and a reference explanation y^r, we first convert each free-form explanation into a structured set of diagnostic entities and evidence

claims. This extraction is performed independently for the prediction and reference using the same offline Qwen3.5-4B evaluator.

A *diagnostic entity* is an image region, subject, object, text element, body part, clothing item, background component, or other semantically meaningful visual element that is explicitly used as evidence in the explanation. Entities are not extracted merely because they are mentioned; they must be connected to the explanation's real/fake judgment. For example, "the sign text," "the cyclist's hands," "the metal support bars," or "the reflection in the window" may be diagnostic entities if the explanation uses them as evidence. In contrast, generic mentions such as "the image," "the scene," or contextual objects that are not tied to a visual cue are excluded.

An *evidence claim* is a compact, checkable visual observation about a diagnostic entity. Evidence claims describe what is visually suspicious or authenticity-supporting about the entity, such as distorted lettering, inconsistent boundaries, unnatural texture, impossible geometry, mismatched lighting, or physically plausible interaction. Claims are linked to their corresponding entities whenever possible. The extractor is instructed to prefer coarse, meaningful entities over overly fine-grained fragments, to avoid unstated visual inferences, and to exclude generic conclusions such as "this image is fake" unless they are grounded in a specific visual observation.

The expected extraction output is valid JSON:

Entity and evidence extraction output schema

```
{
  "diagnostic_entities": [
    {"entity_id": "E1", "name": "..."}
  ],
  "evidence_claims": [
    {"claim_id": "C1", "entity_id": "E1", "claim": "..."}
  ]
}
```

We denote the extracted entity sets from the prediction and reference as $\mathcal{E}^p$ and $\mathcal{E}^r$, respectively. Similarly, we denote the extracted evidence-claim sets as $\mathcal{V}^p$ and $\mathcal{V}^r$. We use $\mathcal{V}$ rather than $\mathcal{F}$ to emphasize that these are visual evidence claims rather than general factual statements.

Bidirectional semantic coverage. After extraction, we compute semantic coverage in two directions. The first direction measures precision: whether the diagnostic entities and evidence claims mentioned by the prediction are supported by the reference explanation. The second direction measures recall: whether the diagnostic entities and evidence claims in the reference explanation are recovered by the prediction.

For precision-side coverage, we run one Qwen3.5-4B evaluator pass using the extracted JSON of the predicted explanation and the full reference explanation. The extracted prediction JSON contains the predicted diagnostic entities $\mathcal{E}^p$ and predicted evidence claims $\mathcal{V}^p$, with evidence claims linked to their corresponding entities. Given this full predicted structure and the reference explanation y^r, the evaluator judges which predicted entities and evidence claims are semantically supported by the reference. This yields entity and evidence precision, measuring whether the prediction introduces reference-supported diagnostic content rather than additional unsupported diagnostic content.

For recall-side coverage, we run a second Qwen3.5-4B evaluator pass in the reverse direction, using the extracted JSON of the reference explanation and the full predicted explanation. The extracted reference JSON contains the reference diagnostic entities $\mathcal{E}^r$ and reference evidence claims $\mathcal{V}^r$, again with evidence claims linked to their corresponding entities. Given this full reference structure and the predicted explanation y^p, the evaluator judges which reference entities and evidence claims are semantically recovered by the prediction. This yields entity and evidence recall, measuring whether the prediction captures the diagnostic regions and visual evidence present in the reference.

Formally, let

$$\mathcal{G}^p = (\mathcal{E}^p, \mathcal{V}^p) \qquad \text{and} \qquad \mathcal{G}^r = (\mathcal{E}^r, \mathcal{V}^r)$$

denote the extracted JSON structures for the predicted and reference explanations, respectively. Here, $\mathcal{E}$ denotes the set of diagnostic entities and $\mathcal{V}$ denotes the set of entity-linked evidence claims. Let

$\mathrm{cover}(z \mid \mathcal{G}, y)$ be an indicator function that returns 1 if source item z, interpreted within the full extracted structure $\mathcal{G}$, is judged to be semantically supported by target explanation y, and 0 otherwise.

Entity precision and recall are computed as

$$\mathrm{EntityPrecision} = \frac{1}{|\mathcal{E}^p|} \sum_{e \in \mathcal{E}^p} \mathrm{cover}(e \mid \mathcal{G}^p, y^r), \tag{1}$$

$$\mathrm{EntityRecall} = \frac{1}{|\mathcal{E}^r|} \sum_{e \in \mathcal{E}^r} \mathrm{cover}(e \mid \mathcal{G}^r, y^p). \tag{2}$$

Evidence precision and recall are computed analogously:

$$\mathrm{EvidencePrecision} = \frac{1}{|\mathcal{V}^p|} \sum_{v \in \mathcal{V}^p} \mathrm{cover}(v \mid \mathcal{G}^p, y^r), \tag{3}$$

$$\mathrm{EvidenceRecall} = \frac{1}{|\mathcal{V}^r|} \sum_{v \in \mathcal{V}^r} \mathrm{cover}(v \mid \mathcal{G}^r, y^p). \tag{4}$$

The matching is wording-invariant but evidence-sensitive. For entities, exact lexical overlap is not required: for example, "the CITY PARK letters" and "the large white sign text" may be counted as the same entity if they refer to the same visible region. However, overly broad entities are not accepted when they fail to localize the diagnostic region. For evidence claims, coverage is stricter: the target explanation must express the same visual observation, not merely mention the same object. Thus, a prediction may receive entity credit for identifying the matching region while still receiving low evidence credit if it gives a different or incomplete visual reason.

We compute EntityScore and EvidenceScore as harmonic means of their corresponding precision and recall terms:

$$\mathrm{EntityScore} = \frac{2 \cdot \mathrm{EntityPrecision} \cdot \mathrm{EntityRecall}}{\mathrm{EntityPrecision} + \mathrm{EntityRecall}}, \tag{5}$$

$$\mathrm{EvidenceScore} = \frac{2 \cdot \mathrm{EvidencePrecision} \cdot \mathrm{EvidenceRecall}}{\mathrm{EvidencePrecision} + \mathrm{EvidenceRecall}}. \tag{6}$$

If both precision and recall are zero, the corresponding harmonic-mean score is set to zero.

EntityScore captures whether the explanation recovers the manipulated or authenticity-relevant regions in the reference, while EvidenceScore captures whether it recovers the corresponding visual reasons.

Metric extraction example. Table 19 gives a compact example of entity and evidence extraction followed by semantic coverage judging.

Table 19: Illustrative entity and evidence extraction example.

Item	Example
Reference explanation	The large CITY PARK letters look flat and poorly integrated with the metal supports, with bars appearing to pass through parts of the text.
Extracted entities	[E1: CITY PARK letters; E2: metal support bars]
Extracted evidence claims	C1/E1: letters look flat; C2/E1: letters are poorly integrated; C3/E2: support bars appear to pass through the letters.
Candidate explanation	The big white letters look like they are pasted on, and the metal bars go through them instead of sitting behind them.
Coverage result	Entity match: E1 covered, E2 covered. Evidence match: C1 covered, C2 covered, C3 covered. This yields illustrative EntityScore $= 1.00$ and EvidenceScore $= 1.00$.

Table 20: Handling of common edge cases in EntityScore and EvidenceScore.

Case	Handling rule	Reason
Empty predicted entity set	Entity precision is set to 0 unless both prediction and reference contain no diagnostic entities.	Penalizes vacuous explanations.
Empty reference entity set	The example is excluded from entity/evidence scoring for fake explanations.	Avoids undefined recall.
Synonymous region names	Count as covered if the evaluator judges that both phrases refer to the same visible region.	Allows natural wording variation.
Overly broad entity	Count as not covered if the phrase fails to identify the diagnostic region specifically.	Discourages generic explanations.
Unsupported extra evidence	Counts against precision-side EvidenceScore.	Penalizes evidence claims not supported by the reference.
Correct entity, wrong evidence	Entity coverage may be positive while evidence coverage remains negative.	Separates region identification from reasoning quality.
Correct evidence, different wording	Count as covered if the same visual observation is expressed semantically.	Avoids over-penalizing paraphrases.

Coverage-judging edge cases. Table 20 summarizes how we handle common edge cases during automatic coverage judging.

BERTScore and simple-explanation scoring. For both complex and simple explanations, we compute BERTScore-F1 between the full generated explanation and the corresponding reference explanation. BERTScore captures overall semantic similarity, but it does not explicitly test whether the explanation identifies the same manipulated entity or recovers the same evidence claims as the reference. We therefore treat BERTScore as complementary to EntityScore and EvidenceScore rather than as a replacement.

For simple explanations, we additionally compute the SLE simplicity score (Cripwell et al., 2023). Since raw SLE values can fall outside a convenient reporting range, we clip the raw score to $[-1, 4]$ and normalize it to $[0, 1]$:

$$\mathrm{SLE}_{\mathrm{clip}} = \min\left(4, \max\left(-1, \mathrm{SLE}_{\mathrm{raw}}\right)\right), \tag{7}$$

$$\mathrm{SLE}_{\mathrm{norm}} = \frac{\mathrm{SLE}_{\mathrm{clip}} + 1}{5}. \tag{8}$$

Higher normalized SLE indicates simpler and more accessible language. We report normalized SLE for simple explanations so that the score is directly comparable across models and remains bounded between 0 and 1.

Automatic metric configuration. Table 21 summarizes the automatic evaluators used for scoring.

Cross-family judge validation. Because EntityScore and EvidenceScore use Qwen3.5-4B as the default semantic evaluator, we additionally test whether the resulting comparisons are sensitive to the choice of judge model. We recompute both metrics using three judges from different model families: Qwen3.5-4B, Molmo2-O-7B, and Llama-3.1-Nemotron-Nano-VL-8B. As a targeted sensitivity analysis, we evaluate explanations produced by two fine-tuned models, Qwen3.5-4B and LLaVA-v1.6-Mistral-7B.

Across all three judges, the relative ordering of the two evaluated models is preserved for both EntityScore and EvidenceScore. Pairwise judge agreement is also consistently positive: example-level Spearman correlations range from 0.506–0.671 for EntityScore and 0.604–0.673 for EvidenceScore. When scores are aggregated by image generator, the corresponding correlations increase to 0.588–0.890 and 0.841–0.984, respectively. These results indicate that the tested comparative conclusions

Table 21: Automatic metric configuration.

Metric	Evaluator	Configuration
BERTScore-F1	BERTScore scorer	Computed between aligned prediction and reference explanations.
EntityScore	`Qwen/Qwen3.5-4B` evaluator	Extract diagnostic entities from prediction and reference explanations, judge bidirectional semantic coverage, and compute the harmonic mean of entity precision and recall.
EvidenceScore	`Qwen/Qwen3.5-4B` evaluator	Extract entity-linked visual evidence claims from prediction and reference explanations, judge bidirectional semantic coverage, and compute the harmonic mean of evidence precision and recall.
Normalized SLE	SLE scorer	Clip raw SLE to $[-1, 4]$ and normalize using $(\mathrm{SLE}_{\mathrm{clip}} + 1)/5$.
Detection Accuracy / F1	Label parser and scorer	Parse real/fake labels from model outputs and compute standard classification metrics.

are not specific to the default Qwen evaluator, although absolute metric values can vary across judge models.

J Baseline and Training Details

Model references. The evaluated model families are GPT-5-mini (OpenAI, 2026), Gemini-3-Flash (Google, 2025b), Qwen3.5 (Qwen Team, 2026), Qwen3-VL (Bai et al., 2025), BusterX++ (Wen et al., 2026), InternVL3.5 (Wang et al., 2025), and LLaVA-NeXT (LLaVA Team, 2024).

Evaluation protocol. All baselines follow the same detection-and-explanation protocol. For each input image, the model receives a single image and is prompted to output a binary prediction, either *real* or *fake*, together with a natural-language explanation. We evaluate two explanation tracks separately. The *complex* track asks for detailed forensic reasoning, while the *simple* track asks for short, jargon-free explanations. Both tracks are evaluated on the same image lists, but use separate prompts tailored to the requested explanation style.

We evaluate on both in-distribution (ID) and out-of-distribution (OOD) test sets. ID fake images are sampled from generator families represented during training, while OOD fake images are sampled from held-out generators introduced only at evaluation time. Each evaluation set contains 18,500 images, consisting of 10,000 fake images and 8,500 real images. The same ID and OOD splits are used for both explanation tracks, ensuring that detection and explanation metrics are computed on aligned examples.

Prompt templates. Both zero-shot and fine-tuned baselines use the same task instruction and output schema, with separate prompts for the complex and simple explanation tracks. For the complex track, we use:

```
Detect whether the image is real or fake and provide
reasoning for it.
Respond in the following format:
<reasoning>your reasoning here</reasoning>
<answer>real or fake</answer>
```

For the simple track, we use:

```
Detect whether the image is real or fake and provide
reasoning for it in simple, easy-to-understand, jargon-free
language.
```

```
Respond in the following format:
<reasoning>your reasoning here</reasoning>
<answer>real or fake</answer>
```

Zero-shot protocol. Zero-shot baselines are evaluated directly using the prompt templates above. The text inside `<reasoning>` is used for explanation evaluation, while the label inside `<answer>` is used for detection evaluation.

Fine-tuning setup. For fine-tuned baselines, we use a balanced 50% subset of the training split, sampled evenly across real and fake images. We construct separate JSONL training files for the complex and simple explanation tracks using the same prompt templates and output schema as in zero-shot evaluation. The two files use the same sample identifiers and image paths, but differ in the target explanation style. Each track is fine-tuned independently so that the model is optimized for the corresponding explanation format.

Scoring. For each run, we parse the predicted label from the `<answer>` field and the explanation from the `<reasoning>` field. Detection F1 is computed with the fake class as the positive class. Explanation metrics are computed separately for the complex and simple tracks using the corresponding generated and reference explanations across the evaluation set.

Compute accounting. GPU-hours are computed as the number of GPUs multiplied by the wall-clock runtime. The values in Table 22 report approximate compute for the final runs used in the paper and exclude small debugging runs. Dataset construction also required caption generation, edit-instruction generation, candidate image generation, filtering, explanation generation, and perturbation generation. For closed-source API models used during dataset construction and evaluation, provider-side compute is not observable; we therefore report processed-example counts/API requests in the project code and dataset documentation rather than provider-side GPU-hours.

Cross-dataset detection transfer. We evaluate Qwen-3.5-4B and LLaVA-1.6-Mistral-7B on 10K-sample subsets of SIDA (Huang et al., 2025b), GenImage (Zhu et al., 2023), and Community-Forensics-Eval (Park & Owens, 2025). *Vanilla* denotes the released checkpoint, while *Fine-tuned* denotes its counterpart fine-tuned on XPlainVerse training data. Table 23 reports detection accuracy and binary F1. This experiment evaluates transfer to external datasets rather than explanation quality.

Fine-tuning improves accuracy for both models across all three datasets. Qwen-3.5-4B also improves binary F1 consistently, by 23.00, 7.02, and 3.61 percentage points on SIDA, GenImage, and Community-Forensics-Eval, respectively. LLaVA exhibits mixed F1 changes: decreases of 4.24 and 5.11 points on SIDA and GenImage, respectively, but an increase of 9.47 points on Community-Forensics-Eval. These results demonstrate benefits from XPlainVerse fine-tuning beyond the benchmark itself, while showing that transfer remains model- and dataset-dependent.

Broader generator training. We conduct a supplementary training study to examine how broader generator coverage affects cross-generator detection. The fake-image portion spans five generators: Gemini-2.0-Flash, LongCat, Hunyuan, ICEdit, and GPT-Image-1. Seven additional generators are excluded from fine-tuning and treated as held out: FLUX-dev, GPT-Image-1.5, Nano Banana 2, Nano Banana Pro, Qwen-Image-Edit, Seedream 4.5, and Wan 2.6. Class-balanced here refers to the real/fake classes and does not imply uniform sampling across generators.

We evaluate Qwen3.5-4B and BusterX++ on images from all twelve generators. In addition to overall accuracy and fake-class F1, we compute fake recall separately for each generator and average it over the five training-exposed generators and the seven held-out generators.

As shown in Table 24, broader generator exposure substantially improves cross-generator transfer. For Qwen3.5-4B, mean fake recall on the same seven held-out generators rises from 15.3% with the original checkpoint to 49.0% after broader training. However, performance on held-out generators still remains 33.8 percentage points lower than on generators seen during training for Qwen3.5-4B, with a similar 35.1-point difference for BusterX++. These results show that broader generator exposure substantially improves transfer, but does not eliminate cross-generator sensitivity.

Table 22: Key training, inference, and compute settings for fine-tuned VLM baselines.

Setting	Value
Fine-tuning	
Framework	`ms-swift` supervised fine-tuning
Hardware	4 AMD MI210 GPUs, each with 64 GB VRAM
Fine-tuning method	LoRA
Fine-tuning tracks	Complex and simple tracks trained separately
Training subset	Balanced 50% subset of the training split
LoRA rank / alpha	16 / 32
Target modules	`all-linear`
Vision encoder	Frozen
Epochs	1
Per-device batch size	8
Gradient accumulation	2 steps
Effective batch size	64
Learning rate	2×10^{-4}
Learning-rate schedule	Cosine decay with warmup ratio 0.05
Maximum sequence length	2048 tokens
Precision	bfloat16
Optimizer	AdamW, with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay 0.1
Maximum gradient norm	1.0
Runtime / compute per fine-tuning run	4B model: 6–7 h, 24–28 GPU-h; 7B/8B-class models: 9–10 h, 36–40 GPU-h; 14B model: ≈13 h, ≈52 GPU-h. Each run uses 4 GPUs.
Total fine-tuning compute	10 runs: 5 baselines × 2 explanation tracks; approximately 368–400 GPU-h, assuming BusterX++ follows the 7B/8B-class runtime.
Inference	
Inference backend	vLLM through `ms-swift`
Hardware	1 AMD MI210 GPU
Maximum new tokens	2048
Decoding	Greedy decoding with temperature 0.0
Precision	bfloat16
Maximum model length	4096 tokens
Maximum batch size	16
Runtime / compute per inference run	3–4 h per model-track evaluation on 1 GPU, corresponding to 3–4 GPU-h.
Total inference compute	20 open-source inference runs: 5 zero-shot baselines and 5 fine-tuned baselines, each evaluated on 2 explanation tracks; approximately 60–80 GPU-h.
Automatic scoring	
Metric computation	EntityScore, EvidenceScore, BERTScore-F1, and SLE
Scoring backend	Offline Qwen3.5-4B evaluator for EntityScore/EvidenceScore; local metric computation for BERTScore-F1 and SLE
Runtime / compute per scoring run	3–4 h per scoring run on 1 GPU, corresponding to 3–4 GPU-h.
Total scoring compute	Approximately 60–80 GPU-h across the corresponding open-source evaluation outputs.

Table 23: Cross-dataset detection results (%) before and after fine-tuning on XPlainVerse.

		SIDA		GenImage		Community-Forensics-Eval	
Model family	Setting	Accuracy	Binary F1	Accuracy	Binary F1	Accuracy	Binary F1
Qwen-3.5-4B	Vanilla	56.64	25.68	62.40	43.34	73.52	69.57
	Fine-tuned	65.51	48.68	66.62	50.36	78.31	73.18
LLaVA-1.6-Mistral-7B	Vanilla	63.24	55.31	56.67	50.02	57.33	51.04
	Fine-tuned	66.91	51.07	64.18	44.91	70.86	60.51

Table 24: Detection after broader five-generator fine-tuning. Exposed and held-out values report mean fake recall across the corresponding generator groups. Gap is the exposed-to-held-out difference in percentage points (pp).

Model	Overall Acc.	Fake F1	Exposed	Held-out	Gap
Qwen3.5-4B	75.9%	71.8%	82.8%	49.0%	33.8 pp
BusterX++	75.7%	71.7%	83.9%	48.8%	35.1 pp

Counterfactual analysis. To further examine what drives the detector's prediction, we construct counterfactual images that separate the intended edit from changes outside the edited region. Let R denote the original real image, F the edited image, and M the mask of the intended edit. We construct

$$\begin{aligned} I_{\text{loc}} &= M \odot F + (1 - M) \odot R, \\ I_{\text{out}} &= M \odot R + (1 - M) \odot F, \end{aligned} \tag{9}$$

where I_{loc} keeps only the localized edit while restoring the rest of the image from the real image, and I_{out} restores the edited region but keeps the generator-processed pixels outside it. We use quality-controlled localized edits and manually verify the masks and counterfactual images.

We report the percentage of images predicted as fake under each condition. These are fake-response rates rather than accuracies, since the counterfactual images are diagnostic interventions rather than standard real/fake test samples.

Table 25 reports the original checkpoints, whose task-training data cover Gemini-2.0-Flash, ICEdit, and GPT-Image-1. The broader-training control instead uses the five-generator exposure set described above and treats the remaining seven generators as held out.

For both models, the pattern changes with generator exposure. For generators seen during training, the outside-edit regions produce substantially more fake predictions than the localized edit itself. For held-out generators, the opposite trend is observed. The difference between these two conditions changes by 37.6 percentage points for Qwen3.5-4B (95% CI: 22.4–52.5) and 65.8 points for BusterX++ (95% CI: 52.9–78.2).

We observe the same overall crossover after the broader five-generator training experiment. For Qwen3.5-4B, the localized-minus-outside difference is -17.6 points for training-exposed generators and $+30.5$ points for held-out generators; for BusterX++, the corresponding values are -13.9 and $+30.5$ points. The trend also remains under binary masks, equal generator weighting, and when restricting the analysis to addition, removal, and replacement edits.

These results suggest that the detectors use both cues from the actual manipulated region and generator-associated signals elsewhere in the image. Which of these contributes more to the final prediction appears to depend on whether the generator was represented during training. The effect is not identical for every generator; for example, LongCat remains more responsive to the localized edit despite being represented during training.

Table 25: Fake-response rates (%) under counterfactual region swaps. Exposed generators are represented during fine-tuning, while held-out generators are not. Δ denotes the localized-minus-outside response difference in percentage points.

Model	Exposure	Real	Full edit	Localized	Outside	Δ
Qwen3.5-4B	Exposed (3)	5.6	58.3	12.5	41.7	-29.2
	Held-out (9)	2.7	24.7	15.9	7.4	$+8.4$
BusterX++	Exposed (3)	1.4	76.4	5.6	63.9	-58.3
	Held-out (9)	5.4	17.6	14.5	7.1	$+7.4$

K Prompt Templates and Model Inputs

This final section records the prompts used for captioning, edit-instruction generation, instruction-difficulty rating, visual-difference extraction, alignment judging, explanation generation, and metric judging. We place the prompt text at the end so that the main appendix reads continuously.

Table 26: Prompt and model-version index.

Stage	**Model(s)**	**Output**
Caption generation	GPT-4o-mini family	Dense image caption
Edit-instruction generation	GPT-4o-mini family	Structured edit-instruction JSON
Instruction difficulty audit	GPT-4o-mini family	Difficulty rating and rationale
Visual-difference extraction	Gemini-3-Flash, GPT-5-mini, GLM-4.6V, Qwen3.5-27B	Difference JSON with changed entities, locations, and evidence
Alignment judging	DeepSeek-V3.2, GPT-OSS-120B	`KEEP`, `BORDERLINE`, or `DISCARD` plus rationale
Complex explanation	Gemini-3-Flash, GPT-5-mini, GLM-4.6V, Qwen3.5-27B	Expert-facing explanation
Simple explanation	Gemini-3-Flash, GPT-5-mini, GLM-4.6V, Qwen3.5-27B	User-facing explanation
Authenticity explanation	Gemini-3-Flash, GPT-5-mini, GLM-4.6V, Qwen3.5-27B, Gemini-2.5-Pro	Real-image authenticity explanation
Metric extraction / judging	Qwen/Qwen3.5-4B	Entity/Evidence extraction and semantic coverage decisions

K.1 Caption Generation Prompt

User prompt

```
You are a professional vision-language annotator. Review the photo and
    produce a concise, information-dense caption (2-3 sentences) that:
- Clearly names the primary subject(s) as well as important secondary
    objects that are visible.
- Describes appearance details such as colors, materials, and notable
    textures, along with approximate positions
    (foreground/background/left/right).
- Mentions the environment or setting, lighting conditions, and any visible
    interactions or relationships between objects or people.
- Avoids speculation about unseen details, emotions, or origins, and never
    introduces objects that do not appear in the image.
- Uses straightforward, photographic language suitable for a professional
    photo-editing workflow.
Output only the caption text.
```

K.2 Edit-Instruction Generation Prompt

System prompt

```
You are an expert photo-editing director. Given a caption of a real
     photograph, you must propose concrete edit instructions that change
     visible content in the image and would be considered a
     manipulated/deepfake-like edit.

Key constraints:
- Avoid edits that are ONLY global/camera transforms (e.g.,
     rotating/resizing/warping the entire image). Edits must modify scene
     content or localized geometry/spatial relationships.
- You MAY include people in the scene; however, at least ONE edit must
     target a NON-person object/region (environment, objects, signage,
     screens, etc.).
- Keep the result photorealistic (consistent lighting/materials/shadows).
     You MAY introduce "geometrically impossible" object shapes (e.g.,
     square wheel, unnaturally thin object) as long as it still looks like a
     realistically manipulated photo (not fantasy/magic).
- Prefer 2-4 edits total, each unambiguous and easy to execute.
```

Few-shot messages

```
Example 1 user message:
Caption: A red vintage bicycle leans against a brick wall beside a green
     wooden door, while a potted fern rests on the stone steps.

Example 1 assistant message:
{
  "edits": [
    {
      "instruction": "Add a small wicker basket to the front handlebars of
     the red bicycle.",
      "target_objects": [
        "red bicycle"
      ]
    },
    {
      "instruction": "Repaint the green wooden door to a deep navy blue
     color.",
      "target_objects": [
        "green wooden door"
      ]
    },
    {
      "instruction": "Remove the potted fern from the stone steps and
     realistically fill in the exposed stone step surface.",
      "target_objects": [
        "potted fern",
        "stone steps"
      ]
    }
  ]
}

Example 2 user message:
Caption: Three coworkers gather around a glass conference table reviewing a
     printed sales chart, while a whiteboard behind them displays quarterly
     figures in blue marker.

Example 2 assistant message:
{
  "edits": [
    {
      "instruction": "Add a fourth team member standing on the left side of
     the glass conference table holding a tablet.",
      "target_objects": [
        "glass conference table"
      ]
    },
    {
      "instruction": "Change the sales chart on the table to display the
     title 'Q4 Forecast' instead of the current heading.",
      "target_objects": [
        "printed sales chart"
      ]
    },
    {
      "instruction": "Erase the blue marker text on the whiteboard and
     redraw it in green marker showing a 15% upward trend line.",
      "target_objects": [
```

```
        "whiteboard"
      ]
    }
  ]
}

Example 3 user message:
Caption: A skateboard rests on a concrete sidewalk next to a metal drain
     grate, with a paper coffee cup nearby.

Example 3 assistant message:
{
  "edits": [
    {
      "instruction": "Change the skateboard's front wheels so they appear
     square-shaped instead of round, keeping the wheel material, dirt, and
     lighting consistent.",
      "target_objects": [
        "skateboard",
        "front wheels"
      ]
    },
    {
      "instruction": "Duplicate the paper coffee cup using a copy-move style
     edit, place the duplicate closer to the drain grate, and slightly
     rotate it so it looks naturally dropped; inpaint the original source
     area realistically.",
      "target_objects": [
        "paper coffee cup",
        "concrete sidewalk",
        "drain grate"
      ]
    }
  ]
}

Example 4 user message:
Caption: A storefront window has a large poster taped inside the glass, with
     a small sticker on the lower corner and brick wall around the frame.

Example 4 assistant message:
{
  "edits": [
    {
      "instruction": "Apply a local planar perspective warp to the large
     poster so it appears slightly tilted as if viewed more from the right
     side, without warping the entire window or bricks.",
      "target_objects": [
        "large poster"
      ]
    },
    {
      "instruction": "Use the newly revealed strip created by the tilt to
     reveal a faded graffiti on the glass, and blend it realistically.",
      "target_objects": [
        "newly revealed area behind the poster",
        "storefront window glass"
      ]
    },
    {
      "instruction": "Adjust tape edges/shadows so the poster still looks
     taped to the glass after tilting, and keep reflections/materials
     consistent.",
      "target_objects": [
        "tape",
        "large poster",
        "storefront window glass"
      ]
    }
  ]
}
```

Final user prompt template

```
Using the caption below, propose 2-4 concrete editing operations that
     intentionally change visible content in a deepfake-like way. Follow the
     rules:
```

```
- Reference the same object names or regions mentioned in the caption so the
    editor knows exactly what to modify.
- Avoid edits that are ONLY global/camera transforms (no whole-image
    rotate/resize/crop/perspective warp).
- At least ONE edit must target a NON-person object/region (objects,
    environment, signage, screens, printed text, etc.).

Allowed edits include:
  * add/remove/replace an object (and realistically inpaint the exposed
    background)
  * reposition/resize a specific object (localized, not whole-image)
  * copy-move duplication of an object/region, optionally with small
    rotate/scale/shear to blend
  * object shape deformation / "impossible geometry" (square wheel,
    unnaturally thin object, bent rigid object) while staying photorealistic
  * occlusion/depth-order change: make object A appear in front of/behind
    object B, generating newly revealed (disoccluded) areas realistically
  * local planar perspective warp on a planar object
    (poster/sign/screen/paper).
    - If the warp creates newly revealed (disoccluded) areas, DO NOT just "
    invent random stuff".
      Instead, specify ONE of:
        (a) extend the same surface's design/texture into the newly visible
    region (e.g., continue poster artwork), OR
        (b) reveal a physically plausible previously-hidden detail
    behind/under it (e.g., older poster layer, graffiti, tape residue,
    hidden label, crack), OR
        (c) reveal a object that could realistically have been occluded
    (e.g., a sticker under the poster edge), keeping it consistent with
    lighting and contact/occlusion.

- Keep edits photorealistic (materials/lighting/shadows consistent).
- Avoid fantasy/supernatural elements and generic enhancements (like "
    improve lighting").
- Return only 2-4 edits.

Format the answer as JSON with this structure (no markdown, no explanation):
{{"edits": [
  {{"instruction": "<clear edit action>", "target_objects": ["<object
    name>", ...]}},
  ... up to four entries total ...
]}}

Caption:
{caption}
```

K.3 Instruction Difficulty Rating Prompt

System prompt

```
You are an edit-instruction difficulty rater for routing edits to
    image-editing models.

You will receive:
- A caption describing a real image (you do NOT see the image)
- A list of edit instructions

Your job:
For each instruction, estimate the probability that a typical image-editing
    model (given the real image + the instruction) will successfully
    achieve the intended change WITHOUT breaking other parts of the image
    that should remain unchanged.
Convert that probability into a difficulty score from 1 to 5.

CRITICAL PRINCIPLE:
Difficulty is driven primarily by how *constraint-tight and brittle* the
    instruction is, especially for *subtle/surgical* changes.
Large "generative" changes are often easier when they allow freedom and do
    not require matching the exact original instance details.

BASELINE ASSUMPTION:
Assume all edits are intended to remain photorealistic unless explicitly
    stated otherwise.
Do NOT increase difficulty just because the instruction says "realistic", "
    match lighting", or "preserve texture".
Only increase difficulty for constraints beyond baseline photorealism.

Score = success probability band:
```

```
1 = Very easy  (P(success) > 0.85)
2 = Easy       (0.70-0.85)
3 = Moderate   (0.50-0.70)
4 = Hard       (0.30-0.50)
5 = Very hard  (P(success) < 0.30)

How to estimate P(success): weigh factors in this order (most important
     first)

1) Subtlety / under-edit risk (MOST IMPORTANT)
- Is the intended change small, fine-grained, or easy for models to
     ignore/half-apply?
- If yes, reduce P(success) aggressively, even if the edit sounds "small".

2) Instance/identity lock (replacement not allowed)
- Must remain the same subject/object instance (not swapped for a
     different-looking one)?
- Strongly reduce P(success) if identity must remain stable while only one
     attribute changes.

3) Locality + collateral-damage intolerance
- Must be confined to a specific region while everything else stays
     identical?
- Higher intolerance to spillover => lower P(success).

4) Structure continuity / reveal-cover requirements
- Does the edit require plausible reconstruction of previously occluded
     areas or seamless continuation of edges/textures/geometry?
- If yes, lower P(success), especially when the reconstructed region must
     align with existing structures.

5) Caption underspecification
- If the caption doesn't provide enough detail to judge what "correct" means
     (placement/appearance), treat the instruction as harder (lower
     P(success)).

Magnitude rule (secondary):
- Do NOT automatically rate large changes as hard. If the instruction
     permits freedom (any plausible result acceptable), it can have higher
     P(success) than a subtle, brittle edit.

Bundling rule:
- If an instruction bundles multiple required outcomes, rate it by the
     hardest required outcome.
- If missing any sub-part fails the instruction, prefer the higher
     difficulty.

Output MUST be valid JSON only (no markdown, no extra text), exactly:
{
  "difficulty_scores": [
    { "instruction": "<exactly copy instruction text>", "difficulty_1_to_5":
     1-5, "reason": "<one short sentence>" }
  ]
}
```

User prompt template

```
Rate each edit instruction independently.

CAPTION:
{caption}

EDIT INSTRUCTIONS:
1) {instruction_1}
2) {instruction_2}
...
```

K.4 VISUAL-DIFFERENCE EXTRACTION PROMPT

System prompt

```
You are a Visual Difference Extractor.

INPUTS:
- Image A: original/real image
- Image B: generated/edited image
```

```
GOAL:
List only the differences that are VISUALLY VERIFIABLE between Image A and
     Image B.
Differences may involve people, objects, text, surfaces, local geometry,
     spatial relationships, or scene structure.

OUTPUT (STRICT):
Return ONLY a valid JSON object (no markdown, no code fences, no extra text).
The JSON MUST have exactly these two keys:
1) "overall_summary": string
2) "changes": array of objects

Each object in "changes" MUST have exactly:
- "entity": string
- "salience": integer from 1 to 5
- "differences": array of strings

FORMAT for each string in "differences":
"A: <what is visible in Image A>; B: <what is visible in Image B>."

HOW TO COMPARE (be thorough):
- Compare Image A and Image B systematically by region: foreground,
     midground, background, and text-bearing regions.
- For each candidate changed region, compare not only object identity, but
     also:
  • presence/absence
  • count
  • position
  • orientation
  • size
  • outline / contour / silhouette
  • local shape
  • surface continuity
  • boundary alignment
  • overlap with nearby objects
  • support/contact with nearby surfaces
  • depth-order relationship
  • local perspective on flat or structured surfaces
  • text/content if legible

PAY SPECIAL ATTENTION TO SUBTLE BUT REAL DIFFERENCES SUCH AS:
- localized shape deformation rather than full object replacement
- bends, bulges, compressions, stretching, warping, or unnatural
     straightening/curving
- changes in edge flow, contour consistency, or region symmetry
- duplicated, missing, or newly introduced localized elements
- local repositioning of one object/part while the rest of the scene remains
     the same
- changes in which object overlaps, blocks, or reveals another
- changes in whether an object appears attached to, resting on, behind, or
     in front of another object
- planar or perspective inconsistencies on signs, screens, papers, posters,
     labels, packaging, walls, floors, tables, or other structured surfaces
- changed text, numbers, symbols, or logos when legible
- changes to pose, hands, face, gaze, clothing, or accessories only if
     clearly visible

IMPORTANT COMPARISON BEHAVIOR:
- When a region looks like the same object in both images, first ask whether
     its SHAPE, POSITION, RELATION, or STRUCTURE changed before describing
     it as a different object or a different appearance.
- Prefer describing a localized structural or spatial difference over
     inventing new semantic detail.
- If the main change is geometric, relational, or positional, report that
     directly.
- Distinguish between:
  • object identity change
  • object attribute change
  • local geometry change
  • spatial relationship change
  • text/content change
- Treat texture, pattern, material, and fine detail conservatively: only
     mention them if they are clearly visible in both images and are central
     to the difference.

STRICT VERIFIABILITY RULES:
- Do NOT include any statement unless you are confident it is directly
     supported by both images.
- Do NOT guess. Do NOT infer intent, cause, or editing process.
- Do NOT replace a subtle visible change with a generic object description.
```

```
- Do NOT invent fine-grained semantic details that are not clearly visible.
- Do NOT use vague filler such as:
  "modified", "repositioned", "different shapes", "changed in arrangement",
     "appears altered"
  unless you name the specific object/region and describe the concrete
     before/after difference.
- Do NOT invent changes for clutter or ambiguous background details.
- Only report background changes if they are specific and clearly visible.
- For subtle pose/expression changes: include only if the relevant body
     parts are clearly visible.
- For text: quote it only if legible.
- If a detail is small, blurry, occluded, or ambiguous, leave it out.

GROUPING RULES:
- Group differences under one entity only if they belong to the same
     localized object or region.
- Separate unrelated changes into separate entities.
- Prefer concrete entity names over vague labels.

SALience GUIDELINES:
- 5 = dominant change that strongly affects the image
- 4 = large obvious local change
- 3 = clearly visible secondary change
- 2 = subtle but reliable change
- 1 = minor but still verifiable detail

WRITING STYLE:
- Be concrete, visual, and concise.
- Describe the exact visible before/after difference.
- Name the object or region explicitly.
- Prefer structural, positional, and relational descriptions when those are
     the true source of the difference.
- Avoid generic recognition-style descriptions that do not identify the
     actual change.

- If you find no clear differences, output "changes": [] and say so in
     "overall_summary".
```

User prompt

```
Compare Image A (original) and Image B (generated). Produce the JSON
     response exactly as specified.
```

K.5 EDIT-INSTRUCTION ALIGNMENT JUDGE PROMPT

System prompt

```
ROLE: TEXT-ONLY JUDGE (no images)

SCENARIO:
Two versions of the same scene exist:
- Image A = before/original
- Image B = after/edited

You do NOT see images. You only see:
(1) EDIT_INSTRUCTION_JSON: the requested edit that Image B is supposed to
     contain relative to Image A.
(2) DIFFERENCE_JSON: another model's claimed A→B differences (may be lossy,
     incomplete, noisy, over-split, or somewhat imprecise in how it
     describes a local change).

IMPORTANT:
- Treat "A:" and "B:" only as labels inside the provided evidence lines.
- Do not assume anything not written.
- DIFFERENCE_JSON may contain false positives, missed details, partial
     descriptions, or slightly imperfect wording for the same underlying
     local edit.
- Judge semantic consistency carefully, but do not demand exact wording
     overlap.
- Prefer localized, concrete evidence over generic statements.

SEMANTIC COMPARISON RULE:
- Do NOT compare text literally or word-for-word.
- Judge meaning at the level of visual gist, localized scene content, and
     coarse attribute family.
```

```
- Treat near-synonyms, paraphrases, level-of-detail differences, and
    approximate descriptor wording as potentially referring to the same
    visual change.
- Small wording drift in color, material, texture, shape, position, or
    relation should NOT by itself count as a mismatch, contradiction, or
    unrelated change if the overall localized meaning is still consistent.
- Prefer semantic equivalence over exact lexical overlap.
- Only treat two descriptions as truly different if they imply a materially
    different visible change in the scene.

GOAL:
Decide whether the claimed A→B differences are consistent with the
    requested edit.

-------------------------
STEP 0: Determine target scope
From EDIT_INSTRUCTION_JSON:
- Identify the requested edit action.
- Identify the listed target_objects.
- Determine the LOCAL EDIT SCOPE.

IN-SCOPE includes:
- direct changes to the named target object(s)/region(s)
- direct local consequences of the edit on the same localized area, such as:
  - changed overlap/contact between the target and nearby objects
  - newly visible or newly occluded local regions caused by the target edit
  - local geometry/perspective changes on the same surface
  - local text/content changes on the edited object
  - local color/material/pattern/appearance changes if they plausibly belong
    to the requested edit
  - nearby local relational changes that naturally follow from editing the
    target

OUT-OF-SCOPE includes:
- unrelated changes to other objects, people, text, or background regions
    not plausibly tied to the requested edit
- broad scene rewrites beyond the local target area
- unrelated semantic additions/removals elsewhere in the scene

Important:
- Scope should be based on the requested edit as a localized manipulation,
    not the whole image.
- Do NOT treat every nearby difference as off-target if it is a natural
    local consequence of the intended edit.

-------------------------
STEP 1: Build a single CORE EDIT INTENT (do NOT over-split)
Create ONE sentence capturing the intended visual change from
    EDIT_INSTRUCTION_JSON.

Then extract 2-6 SUPPORTING CUES from the instruction, but treat them as:
- illustrative hints for what evidence might look like,
- NOT a checklist,
- NOT individually required,
- and not to be scored by counting matches.

Supporting cues may include:
- edited object identity
- add/remove/replace action
- before/after content
- relative location or relation
- text content change
- color/material/pattern/appearance change
- shape/deformation/structure change
- local perspective/warp change
- depth-order/occlusion change
- duplication/count change
- localized pose/part change
- local interaction/contact change

IMPORTANT MATCHING PRINCIPLE:
- The cues describe ONE intended edit; cues often overlap.
- If the core intent is supported, missing fine-grained cues are allowed.
- Merge/normalize cues that express the same underlying change.
- Judge semantic consistency, not wording overlap.
- A partially correct or somewhat imprecise description may still support
    the core intent if it is clearly about the same localized edit.
- Normalize wording into coarse visual meaning before judging.
- Collapse superficially different phrases that likely describe the same
    localized appearance or change.
```

```
- Differences in specificity are allowed: a broader or slightly narrower
    description may still match if it points to the same local edit.
- Do not require exact agreement on low-level descriptive words when the
    underlying visual change is the same.

-------------------------
STEP 2: Evidence matching (semantic, not word-for-word)
Search DIFFERENCE_JSON for evidence lines.

Classify evidence strength as:

- STRONG:
  concrete before/after about a named object/region and its visible change;
  explicit add/remove/replace;
  explicit text/content change;
  explicit position/overlap/depth change;
  explicit color/material/shape/structure change;
  or a clearly localized statement that directly supports the core intent.

- MODERATE:
  evidence is localized to the correct object/region and plausibly supports
    the intended edit,
  but is incomplete, somewhat imprecise, less specific than the instruction,
    or does not fully spell out the before/after.
  This includes semantically adjacent descriptions that likely refer to the
    same local change.

- WEAK:
  vague claims without a clear object/region anchor, such as "looks
    different", "appears altered", "changed layout",
  "more detailed", or similarly generic statements that do not clearly
    identify the edited content.

Decide CORE_MATCH as one of:
- MATCHED_STRONG: strong evidence directly supports the core intent.
- MATCHED_IMPLIED: moderate-to-strong evidence supports the same localized
    intended edit, even if the wording is incomplete or somewhat imprecise.
- PARTIAL_MATCH: some on-target support exists, but it is limited,
    underspecified, or ambiguous.
- NOT_MATCHED: insufficient evidence for the core intent.

Rules:
- STRONG evidence can support MATCHED_STRONG.
- STRONG or MODERATE evidence can support MATCHED_IMPLIED.
- MODERATE evidence, or a combination of weakly aligned but correctly
    localized evidence, can support PARTIAL_MATCH.
- WEAK evidence alone should not support MATCHED_STRONG or MATCHED_IMPLIED.
- Do not fail the sample just because one fine detail is absent, if the core
    edit is reasonably supported.
- If evidence supports the same localized edit but phrases it differently,
    count it as support.
- If evidence is somewhat imprecise but clearly refers to the same target
    area and same underlying edit family, allow partial or implied support.
- Compare the intended edit and the evidence after mentally normalizing
    paraphrases and approximate descriptors into the same coarse visual
    meaning.
- A wording difference is not a mismatch if both descriptions plausibly
    refer to the same localized visible change.
- Prefer agreement on target region + edit gist over exact agreement on
    attribute words.
- Do NOT treat small descriptor mismatches as contradictions if they are
    plausibly alternate descriptions of the same local appearance.
- If two descriptions differ mainly in wording granularity, specificity, or
    approximate attribute naming, prefer a partial/implied match rather
    than a failure.
- Contradiction requires a genuinely incompatible visual claim, not merely
    different wording.
- If the evidence points to the opposite of the requested change, treat that
    as contradiction.

-------------------------
STEP 3: Unrelated changes and hard-discard rule
Consider only OUT-OF-SCOPE changes with STRONG or clearly MODERATE evidence.
Ignore purely WEAK-only claims.

Important caution:
- DIFFERENCE_JSON may over-report tiny side details.
- Do NOT penalize local secondary effects that are natural consequences of
    the requested edit.
- Do NOT penalize slightly imperfect descriptions of the same local edit as
    separate off-target edits.
```

```
- Do NOT penalize wording drift that likely refers to the same underlying
    local edit.
- Do NOT create an off-target penalty from a merely imprecise attribute
    description.
- Before penalizing, ask whether the claimed difference is truly a separate
    semantic edit, or just a noisy textual rendering of the same local
    change.
- Only penalize changes that are clearly separate semantic edits outside the
    local target scope.

Classify OUT-OF-SCOPE changes as:

A) COSMETIC/NO-OP (penalty 0):
   - low-salience distractors
   - minor style/layout differences with the same semantic content
   - small text imperfections or character-level noise plausibly due to
     imperfect rendering
   - tiny local inconsistencies that do not amount to a separate semantic
     edit
   - nearby secondary details plausibly caused by the intended local edit
   - approximate wording differences that preserve the same coarse visual
     meaning
   - minor descriptor drift in color/material/pattern/shape/position wording
     that does not imply a separate semantic edit
   - alternate phrasings of the same localized change

B) MODERATE OFF-TARGET (penalty 1):
   - clear semantic change outside scope that is noticeable but not dominant
   - a separate added/removed/replaced object outside the target area
   - a noticeable unrelated text/content change when text is not the
     intended target
   - one clear off-target change elsewhere in the scene

C) PROMINENT OFF-TARGET / REWRITE (penalty 2 → hard discard):
   - major unrelated object/person addition, removal, or replacement
   - dominant unrelated scene rewrite
   - multiple strong off-target semantic changes
   - a clear separate edit affecting one of the most noticeable regions of
     the image
   - a dominant unrelated text/content change central to the scene

Compute unrelated_change_penalty:
- 2 if any C exists
- else 1 if any B exists (or multiple moderate off-target changes exist)
- else 0

HARD DISCARD:
- If unrelated_change_penalty == 2, verdict MUST be DISCARD.

SEMANTIC PENALTY GATE:
Before adding any penalized_unrelated_change, verify that it describes a
    genuinely separate visible edit outside the intended local scope.
If it could reasonably be an imprecise, approximate, or differently-worded
    description of the intended local change, do NOT penalize it.

-------------------------
STEP 4: Scoring and verdict
If unrelated_change_penalty == 2:
  final_score = 0..2 (choose based on how well core intent matches), verdict
    = DISCARD.
Else:
  Assign match_score 0..5 based on CORE_MATCH (do NOT score by counting
    matched cues):

  5 = MATCHED_STRONG
      core intent directly supported by strong, localized evidence; no
    contradiction

  4 = MATCHED_IMPLIED
      core intent supported by strong/moderate evidence that is clearly
    on-target and semantically consistent,
      even if not perfectly specific

  3 = PARTIAL_MATCH
      core intent has meaningful on-target support, but evidence is
    incomplete, underspecified, or somewhat ambiguous

  2 = weak/minimal overlap with intent
      some limited alignment exists, but support is too vague, too partial,
    or too uncertain
```

```
  1 = almost no support

  0 = contradiction
      evidence suggests the opposite of the requested edit, or clearly
     mismatches the target/edit family

  final_score = clamp(match_score - unrelated_change_penalty, 0..5)

Verdict:
- KEEP if final_score >= 4
- BORDERLINE if final_score == 3
- DISCARD otherwise

-------------------------
GUIDANCE FOR matched_requirements / missing_requirements
- matched_requirements:
  list the main parts of the core intent that are supported, at the level of
     meaning, not exact wording.
- missing_requirements:
  list only important missing parts of the core intent.
  Do NOT list every tiny omitted detail.
  Do NOT mark something missing only because the wording differs if the
     semantic gist is already covered.
  If the core intent is broadly supported, this list may be short or empty.

-------------------------
OUTPUT (STRICT JSON ONLY)
Return ONLY a JSON object with exactly these keys:
- "final_score": integer 0..5
- "match_score": integer 0..5
- "unrelated_change_penalty": integer 0..2
- "verdict": "KEEP" | "BORDERLINE" | "DISCARD"
- "matched_requirements": array of strings
- "missing_requirements": array of strings
- "penalized_unrelated_changes": array of strings
- "evidence_used": array of strings
```

K.6 Complex Fake-Explanation Prompt

System prompt

```
ROLE: Authenticity Critic: Generated-Image-Only Explanation

You will be given:
- GENERATED_IMAGE (this is the ONLY thing you may describe in the final
     explanation)
- AUXILIARY REFERENCES (for private guidance only): REAL_IMAGE,
     EDIT_INSTRUCTION_JSON, DIFFERENCE_JSON

Purpose of auxiliary references:
- REAL_IMAGE: a reliable reference of the original scene.
- EDIT_INSTRUCTION_JSON: the intended semantic change (the generated image
     may follow it partially).
- DIFFERENCE_JSON: a VLM-produced candidate list of changes (helpful as a
     checklist, but not authoritative).

TASK:
Write a natural, human-style explanation of why the GENERATED_IMAGE looks
     fake/synthetic/manipulated,
as if you ONLY saw the generated image.

NON-NEGOTIABLE RULES (no leakage):
1) Do NOT mention or imply: any other image, comparisons, "before/after", "
     edit instruction", "difference JSON",
   "Image A/B", "original/real/reference", or "compared to".
2) Every claim must be directly and clearly supported by visible evidence in
     the GENERATED_IMAGE alone.
   - If you only know it because of the auxiliary references, DO NOT include
     it.
3) You may use the auxiliary references ONLY to decide where to inspect more
     carefully.
   - Do NOT copy wording from DIFFERENCE_JSON.
   - Do NOT introduce claims that you cannot see in the GENERATED_IMAGE.

WHAT TO LOOK FOR (examples):
- warped geometry (hands/face asymmetry, bent straight lines)
- texture issues (plastic skin, smeared detail, patchy/over-smoothed areas)
- edge/blending artifacts (halos, cutout edges, seams)
```

```
- lighting/shadow inconsistencies (wrong direction, missing contact shadow)
- repeated/tiling patterns or unnatural background
- text artifacts (garbled letters, inconsistent strokes)
- inconsistent occlusion/reflections or depth cues
- noise/sharpness mismatch (one region too sharp/too blurry)

INTERNAL PROCESS:
- First inspect the GENERATED_IMAGE and form your own observations.
- Then consult the auxiliary references only as a checklist for regions to
    re-check.
- Keep only strong, visible cues.

OUTPUT FORMAT (STRICT):
Return ONLY valid JSON with exactly one key:
{
  "explanation": string
}

STYLE CONSTRAINTS:
- Mention concrete cues with lightweight location hints in-text (e.g., "
    around the face...", "near the text...", "in the background...").
- No speculation about editing steps; stay visual.

FINAL SELF-CHECK:
- If any forbidden terms appear
    (real/original/reference/instruction/difference/compared/before/after/Image
    A/Image B),
  rewrite before answering.
```

User prompt

```
Generate the explanation JSON exactly as specified.
Remember: ONLY describe the GENERATED_IMAGE in the final explanation.
```

K.7 Simple Fake-Explanation Prompt

System prompt

```
You are a Digital Detective. Your specialty is spotting image manipulations
    and explaining them in a way a five-year-old can understand, without
    any technical jargon

Your job is to look at an edited image and explain, in a very simple way,
    why one part of it feels fake, strange, or out of place.

You will be given:
- GENERATED_IMAGE
- COMPLEX_EXPLANATION
- DIFFERENCE_JSON
- EDIT_INSTRUCTION_JSON

Your goal is to write one short explanation that even a five-year-old could
    understand.

IMPORTANT IDEA:
This simple explanation should NOT just be a shorter version of the complex
    explanation.
It can say something new.
But it should stay focused on the same main changed thing, person, object,
    or part of the scene suggested by the COMPLEX_EXPLANATION and/or
    DIFFERENCE_JSON.

You do NOT need to explain every changed thing. You can pick only one thing.

MASTER RULE:
Trust what you can see in the GENERATED_IMAGE most.

The other inputs are only clues:
- COMPLEX_EXPLANATION helps you know the main changed thing
- DIFFERENCE_JSON helps you know what likely changed most
- EDIT_INSTRUCTION_JSON helps you know what kind of change may have been
    wanted

But the final explanation must sound like it came from looking at the
    GENERATED_IMAGE only.
```

```
Use:
- very easy words
- short sentences
- everyday ideas
- one clear reason
- an analogy if it helps

Do NOT use hard or technical words.

DO NOT use words like:
- lighting
- reflection
- shadow
- smooth
- sharp
- texture
- blend
- artifact
- proportion
- occlusion
- illumination
- forensic
- manipulated region
- inconsistent
etc.

Instead, say things in simple ways, like:
- too shiny
- too blurry
- too stiff
- does not match
- looks pasted on
- looks like a toy
- looks like a sticker
- looks mixed up
- does not fit the moment
- does not belong there

Look for simple reasons like:
- a face that does not match the moment
- a body pose that looks odd
- hands or fingers that look wrong
- words that look mixed up
- clothes that do not fit the place
- something that looks pasted on
- something that looks like a cartoon in a real picture
- something that feels out of place in an easy, common-sense way

GOOD EXAMPLES:
- "A person is standing on a sandy beach, but their feet aren't sinking into
     the sand at all and they don't have any footprints behind them. It
     looks like they are floating on top of the sand, which doesn't happen
     in real life."
- "A man is wearing a big, thick winter coat and a wool hat, but he is
     standing on a sunny beach with palm trees. This doesn't make sense
     because people wear swimsuits at the beach, not winter clothes."
- "There is a fluffy, cartoon-style cat sitting in a field of real grass.
     The cat looks like it came from a TV show, while the grass looks like a
     real photograph. The two styles don't match."

BAD EXAMPLES:
- "The image contains lighting inconsistencies."
- "The face has blending artifacts."
- "The manipulated region is unrealistic."
- "This looks fake."
- "The subject is unnatural."

VERY IMPORTANT RULES:
1) Write the final explanation as if you only saw the GENERATED_IMAGE.
2) Do NOT mention or hint at:
   - the original image
   - a reference image
   - before/after
   - edit instruction
   - difference JSON
   - intended change
   - "A" or "B"
3) Do NOT use comparison phrases like:
   - changed from
   - used to
   - originally
```

```
   - compared to
   - should be
   - restored
4) Do NOT introduce a totally new main person, object, or story that is not
     connected to the COMPLEX_EXPLANATION and/or DIFFERENCE_JSON.
5) You MAY say new simple things about the same changed thing, as long as
     they are easy to see in the GENERATED_IMAGE.
6) If you are unsure, choose the clearest and most obvious reason only.
7) Do not try to sound smart. Sound clear.

OUTPUT FORMAT:
Return ONLY valid JSON with exactly one key:
{
  "explanation": string
}

FINAL CHECK BEFORE ANSWERING:
- Is this about just one main changed thing?
- Is this easy for a five-year-old to understand?
- Did I avoid hard words?
- Did I explain WHY it feels off, instead of just saying it is fake?
- Does it sound like I only looked at the GENERATED_IMAGE?

If any answer is no, rewrite it before returning.
```

User prompt

```
Generate the explanation JSON exactly as specified.

Remember:
- Focus on only one main changed thing.
- Pick the easiest and most obvious one to explain.
- Keep the explanation very simple.
- It should sound like something a normal person, or even a child, could
     understand.
- It may say something new, but only about the same main changed thing
     suggested by the COMPLEX_EXPLANATION and/or DIFFERENCE_JSON.
- Use the GENERATED_IMAGE as the main evidence.
- Do not mention hidden inputs or compare to another image.
```

K.8 Real-Image Authenticity Prompt

System prompt

```
You are an expert "Digital Detective." Your special skill is explaining why
     a picture is real (that is, not AI-manipulated), using super simple
     language that even a five-year-old can understand.
You will be given one input:
[Image]: A single, real-world photograph.
Your Mission:
Write a short, simple, and convincing explanation for why the image appears
     authentic. Your entire output must be a single JSON object.
Guiding Principles & Rules:
Your goal is to write a simple one or two-sentence paragraph that explains
     why the image feels like a real snapshot of a moment. You must point
     out a compelling, physical detail that a child could see and understand.
To build your case, look for real-world details that are hard to fake. Focus
     only on simple, physical things:
Little Messes and Imperfections: Is there a wrinkle in a shirt, a scuff on a
     shoe, a crumb on a table, or a little bit of dirt on someone's face?
     Real life isn't perfectly clean.
How Things Touch and Interact: Does a person's hair look messy because of
     the wind? Do their feet sink into the sand? Does a spoon make a dent in
     ice cream?
Real Textures and Patterns: Does the cat's fur look fuzzy and soft? Does the
     bark on the tree look rough and bumpy? Does a sweater look like it's
     really made of yarn?
The most important rule is to use zero grown-up words. Explain what you see
     in the simplest way possible.
Required Output Format:
Your entire output must be a single, valid JSON object with one key
{
  "authenticity_explanation": "Your simple and convincing explanation of why
     the image appears authentic."
}
Example of a High-Quality Output:
```

```
(Input: Image of a person writing in a notebook at a wooden desk)
{
  "authenticity_explanation": "This looks real because the wooden desk isn't
     perfect. You can see the little lines that all wood has, and there are
     even some tiny scratches on it, like a real desk that someone uses
     every day."
}
(Input: Image of a dog playing with a ball in a park)
{
  "authenticity_explanation": "You can tell this is a real dog because he's
     a little bit messy. There's some mud on his nose and his paws are
     dirty, which shows he was really outside playing in the park."
}
```

User prompt

```
Analyze the REAL_IMAGE and return the JSON exactly in the required format.
Only use visible evidence from the image.
```

K.9 Metric Extraction and Coverage Prompts

Entity and fact extraction prompt

```
You are an information extraction system.

Task:
Given an explanation of why an image looks fake or real, extract only:

1. diagnostic_entities:
   the entities that are actually used as evidence for the authenticity
     judgment

2. evidence_claims:
   the specific claims that explain why those entities make the image look
     fake or real

Important:
Do NOT extract all entities mentioned in the explanation.
Do NOT extract all factual statements.
Extract only the entities and claims that function as evidence for the
     fake/real judgment.

Definitions:

Diagnostic entity:
A person, object, text element, clothing element, body region, or scene
     component that is explicitly used as evidence for why the image looks
     fake or real.

Evidence claim:
A compact, checkable claim about a diagnostic entity that directly supports
     the authenticity judgment.

What to exclude:
- entities mentioned only for context
- descriptive facts that do not support the fake/real reasoning
- generic conclusions such as "the image looks fake" or "the image looks
     real"
- identity, location, color, or presence details unless they are themselves
     part of the evidence

Entity selection rule:
Include an entity only if the explanation uses it as evidence for
     authenticity or manipulation.

Claim selection rule:
Include a claim only if removing it would weaken the explanation for why the
     image looks fake or real.

Granularity policy:
- Prefer coarse, meaningful entities
- Prefer compact, meaningful claims
- Do not over-split one natural observation into many tiny claims
- But do split clearly distinct pieces of evidence into separate claims

Examples of the intended behavior:
```

```
- If a person is mentioned only to identify another evidence-bearing item,
     do not include the person as an entity
- If a phrase is only contextual and not part of the authenticity reasoning,
     do not extract it as a claim
- If an entity has multiple distinct evidence-based issues, extract multiple
     claims for that entity
- If multiple descriptors belong to one natural evidence unit, keep them
     together

Rules:
1. Use only information explicitly stated in the explanation
2. Do not infer unstated causes, intentions, or visual details
3. Extract only evidence-bearing entities
4. Extract only evidence-bearing claims
5. Each claim must be specific and checkable
6. Keep wording normalized and concise
7. Avoid redundancy
8. Do not create entities for minor sub-parts unless the explanation is
     clearly centered on them
9. Do not include non-evidential context
10. Return valid JSON only

Output schema:
{
  "diagnostic_entities": [
    {
      "entity_id": "E1",
      "name": "..."
    }
  ],
  "evidence_claims": [
    {
      "claim_id": "C1",
      "entity_id": "E1",
      "claim": "..."
    }
  ]
}

Additional guidance:
- Prefer the entity that actually carries the evidence
- Do not include broader context entities when the evidence is really about
     a more specific visual element
- Exclude claims whose role is only to identify or locate an entity
- Include only claims that explain why the image appears fake or real
- If a claim can be removed without weakening the fake/real reasoning,
     exclude it

Now process the following explanation:

{{EXPLANATION_TEXT}}
```

Semantic coverage prompt

```
You are an evaluation system.

Task:
You will be given:
1. a reference JSON containing:
   - diagnostic_entities
   - evidence_claims
2. a candidate explanation

Your job is to determine, for each diagnostic entity and each evidence claim
     in the reference JSON, whether it is present in the candidate
     explanation.

This is a LENIENT SEMANTIC MATCHING task.
Do NOT require exact wording.
Match based on core meaning.

What counts as a match:

Entity present:
A diagnostic entity is present if the candidate explanation clearly refers
     to the same evidence-bearing thing, either directly or indirectly, even
     if the wording is different.

Claim present:
```

```
An evidence claim is present if the candidate explanation conveys the same
     core evidence or substantially the same observation, even if:
- the wording is different
- the phrasing is less specific
- the claim is merged with another claim
- the claim is split across multiple sentences
- the candidate uses synonyms or paraphrases
- the candidate expresses the same idea more simply

Be lenient:
- Accept paraphrases
- Accept indirect references if clear enough
- Accept slightly broader or slightly less specific wording if the same core
     idea is preserved
- Accept one candidate sentence covering multiple reference claims
- Accept one reference claim being covered across multiple candidate
     sentences

But do NOT mark present if:
- the candidate only mentions the entity without the relevant evidence
- the candidate gives only generic criticism or praise without the actual
     reference idea
- the candidate discusses a related but meaningfully different issue
- the match is too vague or speculative

Decision rule:
Ask:
"Would a reasonable human judge say that the candidate explanation covers
     this same evidence, even if phrased differently?"
- If yes, mark present
- If only weakly related or too vague, mark false

Important:
Focus on evidence, not wording.
Do not invent support that is not actually present in the candidate
     explanation.

Output format:
Return valid JSON only.

Use this schema:
{
  "entity_matches": [
    {
      "entity_id": "E1",
      "entity_name": "...",
      "present": true,
      "matched_text": "...",
      "reason": "brief explanation"
    }
  ],
  "claim_matches": [
    {
      "claim_id": "C1",
      "entity_id": "E1",
      "reference_claim": "...",
      "present": true,
      "matched_text": "...",
      "reason": "brief explanation"
    }
  ]
}

Output requirements:
- `present` must be either true or false
- `matched_text` should be the shortest relevant span from the candidate
     explanation if present, otherwise ""
- `reason` should be short and based on semantic matching
- Evaluate each entity and claim independently
- The same candidate text may support multiple items
- If the candidate covers only a very generic version of the claim without
     the core evidence, mark false

REFERENCE_JSON:
{{REFERENCE_JSON}}

CANDIDATE_EXPLANATION:
{{CANDIDATE_EXPLANATION}}
```