VDIFF-BENCH: DIAGNOSING AND IMPROVING FINE-GRAINED VISUAL DIFFERENCE UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) achieve strong performance on general visual understanding tasks, yet reliably identifying differences between similar images remains challenging. We introduce **VDIFF-BENCH**, a diagnostic benchmark that connects controlled image difference identification with open-ended visual difference discovery through two tracks: **VDIFF-BENCH-MCQ**, which consists of **1,756 multiple-choice questions** across 10 semantic, textual, and low-level change categories; and **VDIFF-BENCH-GEN**, with 300 challenging image pairs for open-ended image difference generation. Evaluation of 12 MLLMs on the MCQ track reveals pronounced category-specific weaknesses, with small open-weight models particularly struggling on low-level changes. To address the observed failure, we propose **Expert-guided Harness Evolution (EHE)**: an expert coding agent uses traces and evaluation feedback to revise the tools, routing, and response handling around a small base agent. Experiments show that EHE successfully improving a 8B Qwen model's MCQ performance by 26.9%, comparable with the GPT-5.4 model. To better evaluate open-ended image difference captioning, we further introduce **VDIFF-JUDGE**, a tool-augmented agentic evaluator that judges generated differences using targeted visual evidence, forming verifiable traces for assessing correctness and completeness. Even the strongest reported proprietary model achieves only 46.8% micro-F1 on the Gen track. Notably, **EHE's performance gain is transferrable to different data and the challenging Gen task**: adapting EHE to the Gen track improves micro-F1 by 155.5%, approaching proprietary performance; adapting EHE to a different benchmark also yields over 90% improvement.

1 INTRODUCTION

Multimodal large language models (MLLMs) have advanced rapidly on image understanding, achieving strong results on single-image captioning and visual question answering (Liu et al., 2023; Bai et al., 2023; Yue et al., 2024). Yet even strong proprietary models fail at accurately identifying differences between two similar images—a familiar task from children's activity books (e.g. bottom right example in Fig. 1). We refer to this capability as fine-grained **Image Difference Identification (IDI)**, which requires comparative perception across two images, sensitivity to small localized or low-level visual differences, and robustness against hallucinating absent changes.

Existing visual difference benchmarks face 2 main limitations: **insufficiently challenging data** and **unreliable evaluation**. For example, CLEVR-Change (Park et al., 2019) uses controlled synthetic scenes with limited diversity; Figure 1 shows Gemini successfully describing their changes, including camera variations absent from the annotations. Meanwhile, lexical-overlap metrics such as CIDEr and ROUGE (Park et al., 2019; Liu et al., 2025) can penalize paraphrased true changes while rewarding false descriptions with similar wording. DiffCap-Bench (Wei et al., 2026) instead uses an LLM judge to assess individual differences, but lacks an explicit procedure for acquiring and checking localized visual evidence. Together with documented biases and unrobustness in LLM judges (Wang et al., 2024; Panickssery et al., 2024; Raina et al., 2024), these limitations motivate challenging visual difference questions with evidence-grounded, verifiable evaluation.

To address these challenges, we introduce **VDIFF-BENCH** with two tracks: VDIFF-BENCH-MCQ, which enables deterministic evaluation of difference recognition through multiple-choice questions,

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

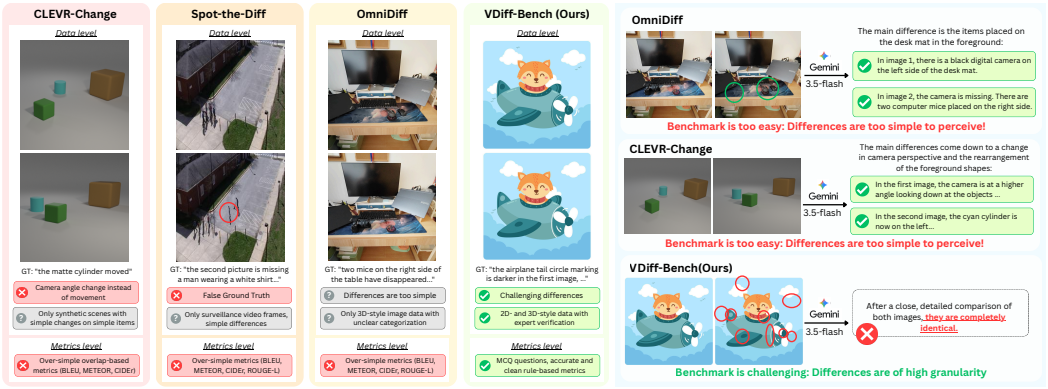


Figure 1: Examples from prior visual difference benchmarks and VDIFF-BENCH. The illustrated responses contrast successful descriptions of simpler changes with failures on densely composed scenes and subtle visual differences, motivating broader diagnostic coverage.

and VDIFF-BENCH-GEN, which tests open-ended image difference captioning on a challenging subset. Atomic annotations cover 10 categories spanning semantic changes (e.g. position), textual changes, and low-level changes (e.g. noise, illumination). The MCQ track contains 1,756 questions while the Gen track contains 300 image pairs. Stratified evaluation of 12 MLLMs exposes category-specific weaknesses: 4 open-weight 7–8B models achieve 52.5–74.5% on semantic differences but only 9.3–46.9% on low-level differences. These deficits motivate the search for methods that can resolve observed limitations, bringing small open-weight models closer to proprietary performance.

Inspired by recent works on agent-assisted harness optimization (Lee et al., 2026; Ursekar et al., 2026), we propose **Expert-guided Harness Evolution (EHE)**: an expert coding agent revises the tools, routing, and response handling around a frozen small model, while paired evaluations select revisions subject to category-level regression checks. On VDIFF-BENCH-MCQ, EHE raises Qwen3-VL-8B-Thinking (Bai et al., 2025) from 65.7% to 83.4% accuracy, a 17.7-percentage-point gain and within 1.5 points of GPT-5.4's 84.9%, proving the effectiveness of tools and expert-optimized harness to strengthen a small model's fine-grained visual understanding.

For evaluation experiments on VDIFF-BENCH-GEN, motivated by our tool-augmented model's gains and the reliability concerns surrounding direct LLM judges, we introduce **VDIFF-JUDGE**, a grounded agentic evaluator that assesses claim–ground truth correspondences through tools and targeted visual evidence, retaining verifiable traces. VDIFF-JUDGE achieves 97% agreement with human judgments on a set of sampled claims, surpassing GPT-5.6-sol and Qwen3-VL-8B as judges. Results on VDIFF-BENCH-GEN exposes substantial limitations even in leading proprietary models: the highest reported micro-F1 is merely 46.8%, and even the strongest result only completely identified all differences in 17.3% of image pairs. Interestingly, EHE shows transferrable performance gains when adapted to the generation setting or to different benchmark: Applying EHE on VDIFF-BENCH-GEN raises Qwen3-VL-8B's micro-F1 from 17.3% to 44.2%, approaching Gemini 3.5 Flash's performance; EHE-optimized Qwen also achieves the highest precision among all models. Applying EHE on a different OmniDiff benchmark improves its main metrics over GPT-5.6-Terra, extending the gains beyond our data distribution. Together, these results demonstrate the progress attainable through better visual evidence procedures, while revealing the remaining difficulty of fully identifying all fine-grained visual changes.

Our main contributions are fourfold:

- We introduce VDIFF-BENCH, a diagnostic image difference identification benchmark spanning 10 change categories, with MCQ and open-ended Gen sub-tracks.
- We introduce VDIFF-JUDGE, an agentic evaluator that utilizes targeted visual verification to assess the correctness and completeness of verbalized image differences.
- We evaluate 12 MLLMs on VDiff-Bench, revealing category-specific perceptual failures and limited open-ended discovery.
- We propose the *Expert-Guided Harness Evolution* (EHE) method, which turns failure diagnoses into executable revisions to a model's tools, inference logics, etc., enabling small models to demonstrate proprietary-level IDI performance without training.

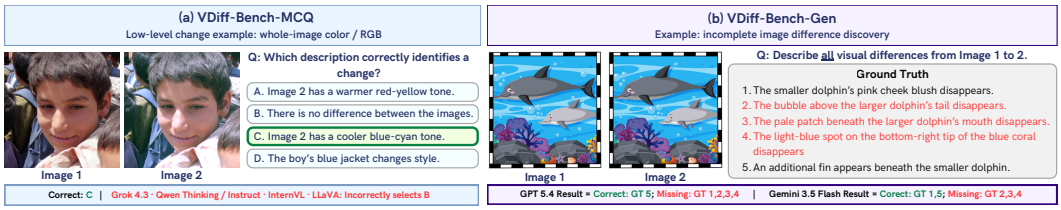


Figure 2: **Examples from the two VDiff-Bench tracks.** Left: five models select "no difference" despite a global color shift in the MCQ example. Right: GPT-5.4 and Gemini 3.5 Flash identify only one and two of five visible changes, respectively, in the Gen track.

2 The VDiff-Bench Benchmark

2.1 Task Definition

Image Difference Identification idi) requires comparing an ordered image pair (I_A, I_B) to identify what changes from I_A to I_B. We represent the annotated changes as a set of atomic difference descriptions $G = \{g_1, \ldots, g_m\}$, each specifying an affected object or region and its change in appearance, state, or image properties. VDiff-Bench evaluates this ability through two complementary tracks: **VDiff-Bench-MCQ** and **VDiff-Bench-Gen**.

VDiff-Bench-MCQ. Given (I_A, I_B) and 4 options $\mathcal{O} = \{o_A, o_B, o_C, o_D\}$, the model selects the true difference from $\hat{y} \in \{A, B, C, D\}$. The options contain one correct description, two false alternatives, and a fixed no-difference distractor; all benchmark pairs contain at least one annotated change. We report choice accuracy, counting invalid or empty outputs as incorrect.

VDiff-Bench-Gen. Given only (I_A, I_B) and an instruction to enumerate their differences, the model produces a set of predicted descriptions $P = \{p_1, \ldots, p_n\}$ without reference choices. This track tests both correctness and completeness: whether predicted changes are visually supported and whether all differences are identified. Our evidence-grounded evaluator compares P with G using the images and reports precision, recall, and F1 (details in Section 4.1).

2.2 Data Collection and Benchmark Construction

We collect and augment image pairs, obtain atomic difference annotations to support open-ended difference generation evaluation, and construct ground truth-conditioned false differences for multiple-choice evaluation. Data sources and their composition are detailed in Appendix A.1.

Image augmentation and annotation. We augment under-represented changes through programmatic transformations (e.g. Gaussian noise, RGB shifts) and model-based image editing (e.g. text changes) and use the change summary as the ground truth difference. For other unannotated pairs, human experts annotate and verify atomic differences. More details are in Appendices A.1–A.2.3.

Negative-choice construction for MCQ. To create distractor options with plausible but false image differences for MCQ questions, we generate reference-conditioned false differences with strong MLLMs. The models are instructed to create changes by applying a change to a nearby entity, reversing a direction or state, or substituting a plausible attribute; these changes are then filtered and validated by structured filters and human expert/ Each final MCQ question combines one ground truth difference, two false differences, and another distractor stating the two images to be identical. More details on option generation, verification, and assembly are in Appendix A.3.

2.3 Dataset Statistics and Benchmark Comparison

The final VDiff-Bench-MCQ track contains **1,756 questions** spanning 10 change categories: position, motion, regional image color, overall image color, appearance/disappearance, noise/resolution, texture, substitution/size, OCR/text, and illumination. VDiff-Bench-Gen contains **300 selected challenging questions**, including 203 questions with multiple differences; selection balances problem difficulty and change categories. Detailed statistics are in Appendix A.4.

Benchmark	Questions	Image Regime				Target-Change Coverage					Construction	Evaluation Design	
		Real	Edited	Rendered	2D puzzle	Semantic	OCR/ text	Global Col./Ill.	Noise/ res.	Texture	Difficulty	Semantic Variance	Verifiable
Spot-the-Diff (Jhamtani & Berg-Kirkpatrick, 2018)	13,192	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
CLEVR-Change (Park et al., 2019)	79,606	✗	✗	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗
OmniDiff (Liu et al., 2025)	15,598	✓	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗
DiffCap-Bench (Wei et al., 2026)	1,075	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗
VDiff-Bench (ours)	**2,056**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
VDiff-Bench-MCQ	1,756	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
VDiff-Bench-Gen	300	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: **Comparison with existing visual difference benchmarks.** "Global Col./Ill." covers whole-image color and illumination changes. "Semantic Variance" denotes meaning-based claim matching; "Verifiable" denotes retained visual evidence supporting individual judgments.

Table 1 compares coverage and evaluation design with existing benchmarks. VDiff-Bench combines semantic, textual, and low-level changes across real, edited, rendered, and 2D puzzle images. Going beyond n-gram similarity metrics (Liu et al., 2025) and ungrounded MLLM judging (Wei et al., 2026), our MCQ track can be deterministically evaluated and our Gen evaluator verify claims using targeted visual evidence and retains inspectable traces (details in Section 4.1).

3 Diagnosing and Improving Image Difference Identification

We first evaluate 12 MLLMs on VDiff-Bench-MCQ to identify their strengths and failure modes. Guided by these findings, we develop Expert-guided Harness Evolution (EHE), which augments a small open-weight model with tools and optimized inference harness, bringing its performance close to proprietary MLLMs without training.

3.1 Experimental Setup

We evaluate 7 proprietary and 5 open-weight MLLMs. Model details are in Appendix F.1 Every model receives two images with four options and the instruction to return a correct label only. For open-weight models, we set the generation temperature to 0. Appendix B provides details on the prompt template and run configuration.

3.2 VDiff-Bench-MCQ Experiment Results and Failure Analysis

Below, we summarize key observations from experiment results. Additional error breakdowns are provided in Appendix E.1.

Models differ on overall accuracy and persistent weakness patterns. Table 2 shows overall accuracy ranging from 36.0% to 91.3%, compared with the 25% uniform-guess baseline. GPT-5.6-Terra achieves the highest overall score, yet reaches only 74.6% on appearance/disappearance. Even the strongest model in this category achieves just 79.6%, revealing a persistent difficulty in recognizing object additions and removals.

Model size and semantic success does not guarantee low-level difference performance. While other proprietary models like GPT-5.6-Terra and Gemini 3.1 Pro demonstrate strong performance on low-level visual difference questions like noise, Grok and three 7–8B open-weight models are remarkable weaker on identifying these low-level differences. Remarkably, Grok achieves 82.0% on semantic changes but only 41.9% on low-level changes, including a merely 5.3% on noise/resolution. Thus, larger model size and success in recognizing semantic edits does not reliably guarantee success in identifying subtle low-level changes in images.

Missed changes motivate better visual evidence gathering. The four models with the largest performance gaps share a failure mode to select the "no difference" distractor on 48.1–80.9% of low-level questions, although every pair contains a real change. These failures motivate augmenting direct visual comparison with tools and improving how the model uses this evidence.

3.3 Expert-Guided Harness Evolution Improves IDI

We propose *Expert-guided Harness Evolution* (EHE), which improves a frozen small, open-weight MLLM through iterative refinement of its inference harness by an expert agent. Each optimization

Model	Change Category										Overall
	Pos.	Motion	Reg. col.	Whole col.	App./dis.	Noise	Texture	Sub./size	OCR	Illum.	
Random Baseline	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
Closed Source Models											
GPT-5.4	83.5	92.8	92.4	83.7	61.7	**100.0**	83.1	91.5	86.0	80.5	84.9
GPT-5.6-Terra†	**91.8**	90.6	**97.6**	91.8	74.6	**100.0**	**89.8**	95.5	**93.7**	90.8	**91.3**
Gemini 2.5 Flash	57.0	81.9	82.9	57.8	44.8	89.3	26.6	74.9	86.0	65.6	66.5
Gemini 3.1 Pro	88.6	91.3	94.7	86.4	75.1	**100.0**	87.6	94.5	92.8	90.3	89.9
Gemini 3.5 Flash	88.0	88.4	92.9	85.7	**79.6**	**100.0**	83.1	92.5	89.1	95.4	89.4
Grok 4.3	75.9	96.4	89.4	87.1	68.7	5.3	15.3	83.9	90.0	60.0	67.7
Doubao Seed 1.6 Vision	86.1	95.7	94.7	91.8	72.6	97.3	76.8	**97.5**	92.3	80.5	88.1
Open Source Models											
InternVL3.5-8B	56.3	84.8	85.3	6.1	52.7	3.3	3.4	73.9	66.5	48.2	49.3
LLaVA-OneVision-Qwen2-7B	43.7	74.6	73.5	1.4	32.8	7.3	5.1	46.2	52.5	20.5	36.0
Kimi K3	80.4	**97.1**	93.5	82.3	64.2	**100.0**	78.5	89.9	89.1	77.9	84.7
Qwen3-VL-8B Instruct	59.5	87.7	86.5	21.8	53.2	45.3	8.5	71.4	83.3	59.0	58.4
Qwen3-VL-8B Thinking	64.6	89.1	88.8	67.3	54.2	8.0	35.6	80.4	88.2	71.8	65.7
+ EHE	67.7	83.3	88.8	**94.6**	65.2	95.3	72.3	85.4	87.3	**95.9**	83.4
EHE-Baseline Δ	+3.2	−5.8	0.0	+27.2	+10.9	+87.3	+36.7	+5.0	−0.9	+24.1	+17.7

Table 2: **Choice accuracy (%) on VDIFF-BENCH-MCQ by change category and overall.** Δ reports EHE's change over Qwen3-VL-8B Thinking in percentage points, computed before rounding. Invalid outputs count as incorrect. Bold marks the best accuracy in each column.

run starts with an initial harness H_0, executor prompts, available visual tools, expert guidance, and baseline execution feedback. As shown in Figure 3, the *expert coding agent* reads and revises harness code in a persistent workspace based on observations from training set output traces; a *frozen base agent* utilizes the harness and tools to answer image-difference questions through each candidate on the development set; an external, deterministic *selection controller* evaluates paired development results and updates whether the proposed harness change should be accepted. The paired development results also provide aggregate feedback for subsequent rounds. The following four stages repeat within this run, producing a harness that operates without the expert at inference.

3.3.1 EXPERT-GUIDED HARNESS EVOLUTION PIPELINE

1. Diagnose failures. The expert reads both the current accepted harness H_t, current state containing training execution traces using H_t and aggregated development feedback from the last round, as well as the history state containing proposals and decisions accumulated within the evolvement run. It identifies a failure mechanism and formulates a testable intervention. For example, training traces revealed that pixelwise difference computation failed on unequal-sized images, motivating a direct-vision fallback when measurements could not be computed. Appendix C.1.4 presents an excerpt from our EHE run trace, containing the failure examples, expert diagnosis, and the resulting proposed harness change.

2. Implement a candidate harness. The expert then proposes an intervention implementation in a candidate $\widetilde{H}_t$ by *adding*, *editing*, *removing*, or *restoring* a component to fix the observed failure modes. Editable components include evidence aggregation and computation, routing, prompts, response budgets, and answer handling. Restoration reintroduces a mechanism explored earlier in the run while preserving subsequent accepted changes. The output is an executable Python harness and a proposal recording the expert's method hypothesis and expected effect.

3. Execute, evaluate, and select. The next step is to execute the candidate harness with the frozen base agent on fixed training and development splits. For each question, the harness processes the image pair and answer options, gathers visual evidence, invokes the model, and parses its answer. Detailed training traces and reference feedback expose execution failures and incorrect predictions, directly supporting diagnosis and targeted changes in the next optimization round. Development results serve two purposes. First, the controller compares candidate and incumbent predictions on identical questions to measure repairs (incorrect to correct), harms (correct to incorrect), invalid answers, and model calls. It accepts improvements in accuracy, validity, or call efficiency only when category- and source-level regression constraints are also satisfied. Second, aggregate scores and paired comparison summaries record the observed effect of each proposal, providing experience for

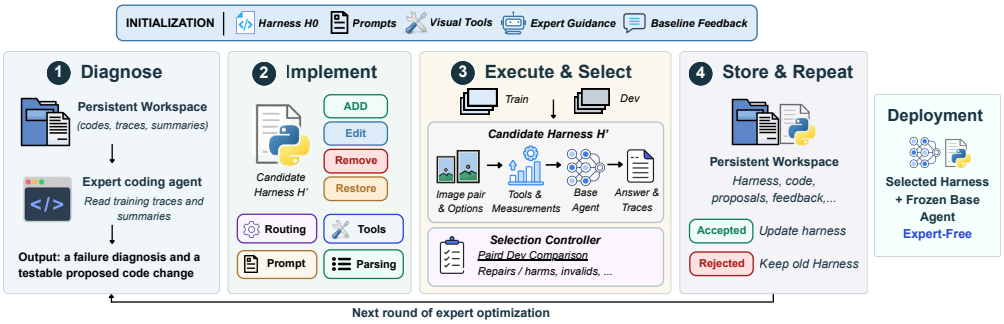


Figure 3: **Expert-guided harness evolution.** An expert coding agent diagnoses failures and modifies a candidate harness. The candidate runs with a frozen base agent, and a deterministic controller selects changes using paired development results. A persistent workspace retains proposals and feedback for the next round; deployment uses only the selected harness and frozen base agent.

subsequent expert optimization. Only these development summaries, rather than individual development predictions or reference answers, are exposed to the expert.

4. Store feedback and repeat. An accepted harness candidate becomes H_{t+1}; otherwise, $H_{t+1} = H_t$. The system stores candidate code, proposals, training traces, and development summaries for both accepted and rejected interventions. The next expert session resumes from this workspace to propose another intervention, continuing the same optimization run until the stopping criterion is met. After optimization, deployment combines the selected harness with the frozen base agent; the expert and selection controller are no longer required.

3.3.2 EHE Experiments and Result Analysis

We instantiate EHE with `GPT-5.6-sol` as the expert coding agent and `Qwen3-VL-8B-Thinking` as the frozen base agent. Starting from a harness that combines global image measurements with direct visual comparison, we optimize on fixed, category-balanced cohorts of 160 training and 200 development questions, with the rest of the VDiff-Bench-MCQ data excluded from optimization. The run allows up to ten rounds and stops after four consecutive rejections. At inference, the frozen harness is being used and no expert is involved. Pseudocode and example optimization trajectory are provided in Appendix C.1.

On VDiff-Bench-MCQ, the final optimized harness improves Qwen's accuracy from 65.7% to 83.4% (+26.9 %), bringing it within 1.5 points of GPT-5.4 performance (Table 2). The largest gains address the low-level weaknesses identified above: noise/resolution accuracy rises from 8.0% to 95.3%, texture from 35.6% to 72.3%, and illumination from 71.8% to 95.9%. Overall, results demonstrate the potential of expert-guided, tool-augmented inference to improve fine-grained visual comparison without updating model weights.

3.3.3 Ablation Study

We conduct an additional ablation study to examine whether EHE performance gain is transferrable to data of other distributions. We adapt EHE to improve the same `Qwen3-VL-8B-Thinking` base agent on OmniDiff (Liu et al., 2025), an Image Difference Caption (IDC) Generation benchmark. We briefly adapt previous EHE initial prompts and harness to OmniDiff's task, and use the same `GPT-5.6-sol` expert agent to optimize the harness around the frozen base agent. The harness generates difference captions in prose, combines visual comparison with luminance-based lighting descriptions, and includes recovery when the model produces no final caption. Optimization uses 100 training pairs and 200 development pairs, balanced across domains; candidate selection requires improved CIDEr without degrading SPICE or increasing blank outputs, together with additional metric and runtime checks. We also select 1000 evaluation data—with 500 on the "Real" image cohort and 500 on the "Rendered" cohort in Omnidiff—that are disjoint from the data used during optimization. We compare the final optimized harness with the outputs from directly prompting Qwen3-VL-8B-Thinking and GPT-5.6-Terra on the same evaluation data. We report CIDEr and SPICE using the same scoring pipeline in OmniDiff's implementation; blank captions receive zero and are counted as errors.

Method	Real		Render		Overall	
	CIDEr ↑	SPICE ↑	CIDEr ↑	SPICE ↑	CIDEr ↑	SPICE ↑
GPT-5.6-Terra	14.29	14.72	4.36	13.78	9.32	14.25
Qwen3-VL-8B Thinking	18.63	13.32	6.45	12.57	12.54	12.94
+ EHE adaptation	**25.80**	**17.74**	**21.86**	**20.83**	**23.83**	**19.29**
EHE-Baseline Δ	+7.17	+4.43	+15.41	+8.26	+11.29	+6.34

Table 3: **Ablation study of EHE on OmniDiff's Image Difference Captioning (IDC) task.** Results are on 1,000 sampled evaluation pairs, including 500 from real image cohort and 500 from rendered cohort. We compare direct prompting models for verbalized differences with the agentic framework optimized by EHE on Qwen3-VL-8B Thinking. CIDEr measures n-gram similarity with ground trut; SPICE measures semantic scene-graph agreement (higher is better). Δ reports absolute gains over direct Qwen; bold marks the best score in each column.

Ablation Results. Table 3 shows improvements in both domains: overall CIDEr increases from 12.54 to 23.83 (+90.0% relative), and SPICE from 12.94 to 19.29 (+49.0%). The adapted 8B model also exceeds the direct GPT-5.6-Terra baseline on both metrics. The largest gains occur on rendered images, where CIDEr rises from 6.45 to 21.86 and SPICE from 12.57 to 20.83. Blank captions decrease from 136 to 2, indicating that output recovery contributes alongside changes to caption generation. These results support EHE's transferrability to a different data distribution and a different task of image difference captioning under conventional reference-based metrics.

4 Open-Ended Difference Discovery and Evaluation

We have demonstrated success and failure modes of models on the IDI task through VDiff-Bench-MCQ with deterministic evaluation. To further challenge the model, we hope to assess whether models can independently identify and verbalize visual differences; however, previously proposed evaluation methods like n-gram overlap metrics and direct LLM judge are not robust, reliable, or verifiable. Inspired by the positive results of our EHE method, which demonstrates that refining the inference harness around a frozen model can substantially improve difference identification, we introduce VDiff-Judge, a tool-augmented agentic framework for evaluating the open-ended Image Difference Captioning (IDC) task on our challenging VDiff-Bench-Gen. VDiff-Judge achieves higher agreement with human judgments than prompting Qwen or GPT-5.6-Sol as judges, while using claim-level verification and reference matching to retain inspectable evidence traces.

4.1 VDiff-Judge: Evidence-Grounded Evaluation

We adapt the visual-evidence tools of the optimized harness in Section 3.3 to build VDiff-Judge. Specifically, we replace option selection with claim–reference judgments and add targeted grounding, semantic consistency checks, and set-level aggregation. Prompt and harness excerpts, visual tools, and additional evaluation details are provided in Appendix D.

VDiff-Judge pipeline. Fig. 4 summarizes five stages of our VDiff-Judge pipeline.

1. Problem Comprehension. A judge agent receives the ordered image pair, model-generated claims to be judged P, and ground-truth differences G.

2. Propose correspondences. Each claim is compared against all ground truths on object identity, attributes, location, change direction, etc.. These correspondences identify what evidence to inspect.

3. Inspect targeted evidence. The harness selects and invokes visual tools according to each claim: whole-image measurement tools for global changes, such as illumination or noise, and grounding and cropping tools for localized changes. The resulting measurement outcomes and paired crops provide additional, localized evidence for the subsequent verification step.

4. Verify correspondences. The judge revisits each candidate match using the tool outputs and semantic consistency checks, allowing equivalent wording while requiring visual support for the claimed change. For example, the illustrated crops in Fig. 4 support the beehive disappearance but not the model-claimed insect appearance. Verdicts and supporting evidence are retained in a verifiable trace; stages 2–4 repeat for each claim.

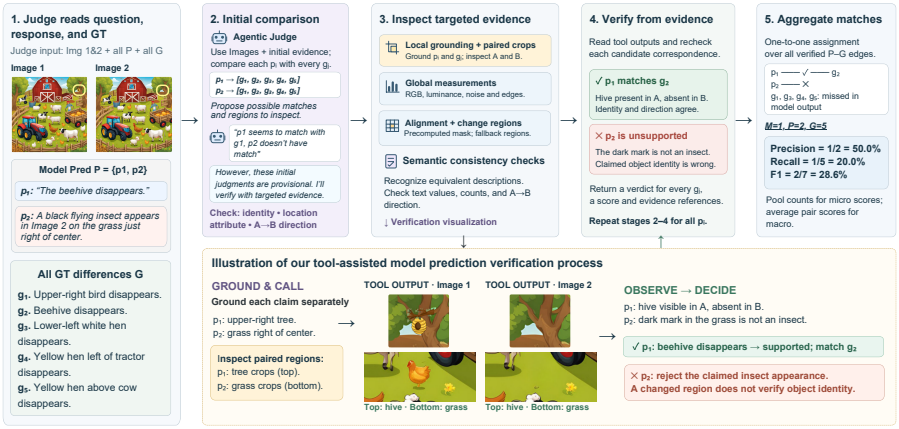


Figure 4: **VDIFF-JUDGE Pipeline: evidence-grounded evaluation of image differences.** The judge compares each predicted claim with all ground truths, verifies correspondences using visual evidence and semantic checks, and compute metrics based on GT matches. The example shows verification of a supported beehive disappearance claim with an unsupported insect appearance claim.

5. Aggregate matches. We use precision to measure the fraction of predictions matched to references, and recall to measure reference coverage. Additionally, we report micro-F1, pair-macro F1, and exact-set accuracy; scoring details are in Appendix D.2.

Evaluator Reliability On 100 model-blinded claims, VDIFF-JUDGE agrees with a human expert on 97% of judgments (Cohen's $\kappa = 0.937$), compared with 84% for Qwen judging and 85% for GPT-5.6-Sol judging. Agreement calculation details are in Appendix D.3.

4.2 OPEN-ENDED BENCHMARK RESULTS

Table 4 reports results for models on VDiff-Bench-Gen.

Holistic IDC is difficult even for proprietary models. Across the reported runs, micro-F1 ranges from 7.1% to 46.8%, and no model exceeds 17.3% exact-set accuracy. Gemini 3.5 Flash achieves the highest reported F1, yet recovers only 54.8% of annotated differences. Thus, even strong proprietary models struggle to correctly and holistically captur fine-grained differences between two images.

Greater coverage does not ensure greater precision. GPT-5.4 achieves higher recall by recovering more correct differences than Gemini 3.1 Pro, but its lower precision yields lower F1. This separates finding more annotated changes from producing a more accurate set of claims: additional descriptions help only when they identify supported differences.

Multiple-change scenes expose a coverage bottleneck. All models have lower recall on multiple-change pairs than on single-change ones. Gemini 3.5 Flash falls from 77.3% to 49.9%, and Kimi K3 from 64.9% to 40.9%, indicating substantial difficulty in identifying co-occurring changes.

EHE improves answer completion and difference detection. Adapting EHE on the open-ended generation task raises Qwen3-VL-8B Thinking's pair-macro F1 from 9.5% to 43.0% and micro F1 from 17.3% to 44.2%. What's more, the EHE-optimized Qwen model now achieves the highest precision on VDiff-Bench-Gen, beating proprietary models.

5 RELATED WORK

5.1 IMAGE DIFFERENCE IDENTIFICATION AND CAPTIONING BENCHMARKS

Image Difference Captioning (IDC) benchmarks prompt models to identify changes between paired images. Spot-the-Diff (Jhamtani & Berg-Kirkpatrick, 2018) introduced image pairs from surveillance footages with crowd-sourced descriptions; CLEVR-Change (Park et al., 2019) utilized rendered images from synthetic scenes. These datasets test MLLMs on change localization and verbalization, but their domains and change inventories are limited. More recently, OmniDiff (Liu et al.,

Model	Micro			Pair-macro	Exact set
	Precision	Recall	F1	F1	Acc.
Proprietary models					
GPT-5.4	15.0	41.0	22.0	26.8	7.0
GPT-5.6-Terra	21.0	30.1	24.7	24.9	7.7
Gemini 2.5 Flash	20.3	52.0	29.2	32.8	5.0
Gemini 3.1 Pro	22.9	34.2	27.4	31.5	10.3
Gemini 3.5 Flash	40.8	**54.8**	**46.8**	**49.5**	**17.3**
Grok 4.3	14.7	23.5	18.1	20.0	4.7
Doubao Seed 1.6 Vision	16.5	16.5	16.5	17.7	9.3
Open-weight models					
InternVL3.5-8B	8.6	20.0	12.0	15.8	5.7
LLaVA-OneVision-Qwen2-7B	4.6	15.4	7.1	8.2	1.3
Kimi K3	19.2	45.2	27.0	32.4	7.7
Qwen3-VL-8B Thinking	21.2	14.6	17.3	9.5	2.7
+ EHE	**41.5**	47.3	44.2	43.0	14.3
Δ (EHE − baseline)	+20.3	+32.7	+26.9	+33.5	+11.7

Table 4: Open-ended image difference captioning results on VDIFF-BENCH-GEN. Micro scores pool counts across pairs; pair-macro F1 averages per-pair F1. Exact-set accuracy requires all references matched and no unmatched predictions after duplicate removal. Bold marks column maxima, excluding the delta row. Qwen3-VL-8B-Thinking with EHE achieves strongest micro precision, whereas Gemini 3.5 flash achieves the best on other metrics.

2025) broadens IDC to real and rendered image pairs across scenarios and change types, with human descriptions. However, it evaluates model-verbalized differences using ngram-based metrics like CIDEr and BLEU, which remain sensitive to paraphrase and might falsely credit to reversed claims. DiffCap-Bench (Wei et al., 2026) addresses this issue by using MLLM judge-reported metrics (Wei et al., 2026), but this method relies heavily on the performance of the judge; a large body of previous works (Zheng et al., 2023; Wang et al., 2024; Panickssery et al., 2024; Raina et al., 2024) have revealed significant issues with unrobustness and biases in LLM judges.

5.2 AGENT AND HARNESS OPTIMIZATION

Agent optimization increasingly targets executable procedures around language models. Zelikman et al. (2024) improves a code-generation scaffolding program through self-modification, while Hu et al. (2025) uses a meta-agent to discover agent implementations. Zhang et al. (2026b) extends self-modification through open-ended search over an archive of coding agents. More directly, Meta-Harness (Lee et al., 2026) searches harness code using historical programs, scores, and execution traces; VeRO (Ursekar et al., 2026) provides versioned snapshots and controlled evaluation for coding-agent optimizers. Self-Harness (Zhang et al., 2026a) connects weakness mining to minimal edits and regression testing, while HarnessBank (Luo et al., 2026) combines a separate evolver with a diverse harness archive and gated screening.

6 CONCLUSION

We introduced **VDIFF-BENCH**, a diagnostic benchmark for fine-grained Image Difference Identification in MLLMs. VDIFF-BENCH consists of two challenging and complementary tracks: VDIFF-BENCH-MCQ and VDIFF-BENCH-GEN. VDIFF-BENCH-MCQ consists of 1,756 questions about differences in image pairs across 10 semantic, textual, and low-level change categories. VDIFF-BENCH-GEN further challenges models to correctly and holistically identify all changes between an image pair on 300 hard cases. Evaluation of 12 MLLMs reveals distinct weaknesses across change categories, with most open-source small models suffering on low-level visual changes like noises. Inspired by recent works on harness evolvement, we propose a training-free **Expert-guided Harness Evolution (EHE)** framework that improves a small-scale 8B Qwen model from 65.7% to 83.4% accuracy on VDiff-Bench-MCQ, an 26.9% performance gain and approaching proprietary model performance. We then adapted the optimized harness to build **VDiff-Judge**, an evidence-grounded evaluator linking visual verification with structured matching and scoring for open-ended image difference generation evaluation. Using this verifiable evaluator, we score the MLLMs on the challenging VDIFF-BENCH-GEN; even the strongest model only fully capture differences in 17.3% of the evaluation problems, exposing remarkable limitations even for proprietary models.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing methods, providing feedback on experiments, processing data, and interpreting results. As part of benchmark construction, generative models supported image augmentation and the generation of reference-conditioned MCQ distractors, followed by the filtering and human verification described in the paper. Generative models also constitute components of the methods under study: an expert coding agent proposes and implements inference-harness revisions in EHE, and an MLLM performs tool-assisted claim verification in VDIFF-JUDGE. These uses are documented in the corresponding methodology and experimental sections. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, or assist with translation.

Additionally, we used AI assistants for identify relevant literature, brainstorming, and edit a research paper to improve readability.

We have reviewed all AI-assisted work. The authors reviewed and revised AI-assisted manuscript texts, checked factual claims and citations against their sources, and checked quantitative summaries against recorded experimental outputs. AI-generated codes were checked through execution and inspection, and benchmark annotations and evaluator decisions underwent the human review described in the paper. We take responsibility for the final content of this work, including its text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

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

ByteDance Seed. Technical introduction to the Seed1.6 model series. `https://seed.bytedance.com/en/seed1_6`, June 2025. Accessed: 2026-09-05.

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

Gemini Team. Gemini 3.1 Pro: A smarter model for your most complex tasks. *Google Blog*, February 2026. URL `https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/`. Published February 19, 2026.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations*, volume 2025, pp. 21344–21377, 2025.

Harsh Jhamtani and Taylor Berg-Kirkpatrick. Learning to describe differences between pairs of similar images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4024–4034, 2018. doi: 10.18653/v1/D18-1436.

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

Koray Kavukcuoglu, Jeff Dean, Oriol Vinyals, and Noam Shazeer. Gemini 3.5: Frontier intelligence with action. *Google Blog*, May 2026. URL `https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-5/`. Published May 19, 2026; accessed August 17, 2026.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. doi: 10.48550/arXiv.2607.24653. URL https://arxiv.org/abs/2607.24653.

Yoonho Lee, Roshen Sanjay Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-harness: End-to-end optimization of model harnesses. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=tmbOUyFx3R.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Yuan Liu, Saihui Hou, Saijie Hou, Jiabao Du, Shibei Meng, and Yongzhen Huang. OmniDiff: A comprehensive benchmark for fine-grained image difference captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21440–21449, 2025.

Xiaotian Luo, Fengxingyu Wang, Chuanrui Hu, Dizhan Xue, and Yafeng Deng. Self-evolving agent harnesses via gated semantic quality-diversity. *arXiv preprint arXiv:2607.13683*, 2026.

Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognition—RRC-MLT-2019. In *2019 International Conference on Document Analysis and Recognition*, pp. 1582–1587, 2019. doi: 10.1109/ICDAR.2019.00254.

OpenAI. GPT-5.4 Thinking System Card. Technical report, 2026a. URL https://openai.com/index/gpt-5-4-thinking-system-card/.

OpenAI. GPT-Image-2 Model. OpenAI API documentation, April 2026b. URL https://developers.openai.com/api/docs/models/gpt-image-2. Released April 21, 2026; accessed September 4, 2026.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 68772–68802. Curran Associates, Inc., 2024. doi: 10.52202/079017-2197. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/7f1f0218e45f5414c79c0679633e47bc-Paper-Conference.pdf.

Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust change captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4624–4633, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021.

Vyas Raina, Adian Liusie, and Mark Gales. Is LLM-as-a-judge robust? investigating universal adversarial attacks on zero-shot LLM assessment. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7499–7517, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.427. URL https://aclanthology.org/2024.emnlp-main.427/.

Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8802–8812, 2021.

Mischa Anouk Smith. AI edited my face without asking: At first i laughed it off—then i started seeing myself differently. *Marie Claire UK*, July 2026. URL https://www.marieclaire.co.uk/life/ai-edited-my-face-without-asking. Published July 11, 2026; accessed August 17, 2026.

Varun Ursekar, Apaar Shanker, Veronica Chatrath, Yuan Xue, and Samuel Marc Denton. Vero: A harness for agents to optimize agents. *arXiv preprint arXiv:2602.22480*, 2026.

Yixin Wan, Lei Ke, Wenhao Yu, Kai-Wei Chang, and Dong Yu. Motionedit: Benchmarking and learning motion-centric image editing. *arXiv preprint arXiv:2512.10284*, 2025.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.511. URL https://aclanthology.org/2024.acl-long.511/.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

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

Yuancheng Wei, Haojie Zhang, Linli Yao, Lei Li, Jiali Chen, Tao Huang, Yiting Lu, Duojun Huang, Xin Li, and Zhao Zhong. DiffCap-Bench: A comprehensive, challenging, robust benchmark for image difference captioning. *arXiv preprint arXiv:2605.04503*, 2026.

xAI. Grok 4.3. https://docs.x.ai/developers/models/grok-4.3, 2026. Accessed: 2026-09-05.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. Self-taught optimizer (STOP): Recursively self-improving code generation. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=46Zgqo4QIU.

Hangfan Zhang, Shao Zhang, Kangcong Li, Chen Zhang, Yang Chen, Yiqun Zhang, Lei Bai, and Shuyue Hu. Self-harness: Harnesses that improve themselves. *arXiv preprint arXiv:2606.09498*, 2026a.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, volume 2026, pp. 104223–104294, 2026b.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

A Additional Details on Dataset Construction

This section details the construction of VDiff-Bench, including data sources, image augmentation and human annotation, and the generation and verification of false differences for MCQ distractors. We also provide stratified dataset statistics for the MCQ and Gen tracks.

A.1 Data Source Composition

Below, we provide details on the sources of data to construct our benchmark. Table 5 reports counts for both questions and distinct image pairs from different source datasets. Our raw data comes from three main sources: Existing image pairs with annotated differences, exisiting image pairs with un-annotated differences, and single-image data that requires augmenting into image pairs.

Existing annotated pairs. We draw from complementary paired-image resources: fixed-camera scenes from Spot-the-Diff (Jhamtani & Berg-Kirkpatrick, 2018), motion-centric edits from MotionEdit (Wan et al., 2025), color and position edits from OmniEdit (Wei et al., 2025), and illumination, substitution, size, and OCR cases from OmniDiff (Liu et al., 2025). We use source difference annotations or editing instructions as provenance for the intended change, then normalize each selected statement and map it to the current VDiff-Bench taxonomy.

Existing un-annotated pairs. Inspired by the "spot-the-difference" puzzle games in childrens' playbooks, we collect a set of 196 "find-the-difference" puzzle pairs curated from various online sources. These drawn cartoon scenes contain many small and deliberately challenging visual differences per pair of images.

Unpaired image data. Inspired by the user-identified failure modes in image editing models to preserve human skin texture (Smith, 2026), we sample from the FFHQ dataset (Karras et al., 2019) with high-quality facial images and augment them by applying low-level visual changes (see Appendix A.2.1). Additionally, we augment our dataset with state-of-the-art image editing models on scene-text images from MLT19 (Nayef et al., 2019) and TextOCR (Singh et al., 2021), as well as sampled input images from OmniEdit (Wei et al., 2025).

Source subset	Questions	Pairs	Primary category/categories
Spot-the-Diff	108	55	appearance, position
MotionEdit	137	67	motion
OmniEdit color	137	137	regional color
OmniEdit position	120	120	position
Custom illumination	44	44	illumination
Kids games	102	12	mixed puzzle differences
JustFamilyFun puzzles	56	56	mixed puzzle differences
Kids puzzles	34	34	mixed puzzle differences
Gaussian perturbation	150	150	noise/resolution
Smoothing transformation	150	150	texture
RGB shift	147	147	whole-image color
OCR/text	221	221	OCR/text
OmniDiff illumination	151	151	illumination
OmniDiff substitution/size	199	199	substitution/size
Total	**1,756**	**1,543**	

Table 5: Source composition of the current benchmark inventory. Source names are reported for provenance; main-paper coverage is summarized by change category.

A.2 Data Augmentation

A.2.1 Augmentation with Programmatic Visual Changes

Adding Gaussian Noise We perform the first type of low-level visual transformation on 150 FFHQ images (Karras et al., 2019) by adding Gaussian noise. For each image, we independently

sample additive noise for every pixel and RGB channel from a zero-mean Gaussian distribution with standard deviation 15 in 8-bit pixel space, and clip the result to $[0, 255]$. This yields global low-level corruption while preserving the image content and layout.

Changing RGB The second type of RGB-shift transformation starts from the same 150-image FFHQ image pool. As a first augmentation step, we apply a stronger global RGB perturbation to each image. As a second step, we compute the mean per-channel difference between this perturbed image and the original, remove its average channel offset to isolate chromatic rather than brightness change, scale the resulting color direction by 0.55, and cap the maximum absolute channel offset at 12 pixel values before applying it to the final image. This yields subtle global tone shifts while keeping the transformed image close to the original. The final benchmark retains 147 examples after excluding three construction-time flagged images.

Changing Texture / Smoothing The texture transformation / smoothing transformation uses the same 150-image FFHQ image pool. We apply a skin- and edge-aware smoothing transform: the image is blurred with a Gaussian kernel whose standard deviation is sampled from $[2.4, 3.6]$, while a soft skin mask, estimated from YCbCr color thresholds, increases smoothing on skin-like regions. Edges are protected using a luminance high-pass signal, and residual detail is mixed back with a random scale in $[0.18, 0.36]$. After smoothing, we add a small luminance-dependent highlight term to bright skin-like regions, reduce saturation by mixing the image toward grayscale, lower contrast around the mid-gray point, and apply a gamma correction sampled close to one. Together, these operations suppress fine texture and create a smoother, plastic-like appearance while preserving the image layout and object identities.

Changing Illumination We augment the OmniDiff (Liu et al., 2025) illumination subset by constructing images with more subtle illumination changes. We use the first 50 sorted images from the Country211 test split (Radford et al., 2021) as the seed split. Each image is adjusted by a gamma-plus-linear transform, $\mathrm{clip}(255 \cdot x^{\gamma} + b)$, where x is the normalized RGB image. For brighter examples, $\gamma \sim U(0.70, 0.85)$ and $b \sim \{5, \ldots, 15\}$; for darker examples, $\gamma \sim U(1.15, 1.30)$ and $b \sim \{-15, \ldots, -5\}$.

A.2.2 Augmentation with Image Editing Differences

We additionally augment the OCR change and object position change categories in our evaluation data. Inspired by the recent advancement in high-performance image editing models, we utilize frontier commercial image editing model to help with this data augmentation process.

Changing OCR Text The OCR subset contains 221 examples: 50 existing OCR/text-related pairs imported from OmniDiff, and 171 custom text edits initialized from MLT19 and TextOCR images. For the custom OCR edits, we select readable, approximately horizontal text regions with moderate size, avoiding boxes smaller than 50×18 pixels or covering more than 40% of the image. We propose one of four localized textual edits: replacing one same-type character, replacing one digit, swapping adjacent characters, or replacing a word with another same-script word visible in the image. The selected crop is padded by 24 pixels, and then edited with *gemini-3-pro-image* which is instructed to also preserve font, color, perspective, background, lighting, and texture, and pasted back with an 8-pixel feathered boundary. We then validate that the target text appears and that non-text scene content remains unchanged. Below, we provide the prompt for the image editing model on the OCR edit task.

OCR Edit Prompt

You are looking at a crop from a scene image that contains text in {`script`} script.
Change the text "{`orig_text`}" to "{`new_text`}".
Match the original font, color, size, alignment, perspective, and texture exactly.
Keep the background, lighting, and every other element pixel-identical to the input.
Only the text changes. Output the edited image.

Changing Object Position We further augment the "position change" category in the evaluation data. We take 150 seed images from the "swap" task of the OmniDiff dataset to begin with; the reason for selecting the "swap" images is because the nature of this task guarantees at least 2 clearly visible objects in images, whose positions can be cleanly changed. For each seed image, we first utilize *gpt-5.4* to proposes 5 candidate edits, each specifying a visible, countable object, its original location, a plausible new location, an editing instruction, and a reference difference statement beginning with "In the second image,". The prompt explicitly disallows adding, removing, recoloring, resizing, rotating, duplicating, replacing, or deforming objects, so the intended change is restricted to spatial relocation. We use the first candidate by default and append preservation constraints requiring the same scene, camera angle, lighting, style, object identities, colors, and object counts. The selected instruction is then passed to *gpt-image-2* (OpenAI, 2026b) to produce an edited image. Below, we provide the prompts for the MLLM to propose potential positional edits, as well as the final image editing prompt template.

Position Edit Proposal Prompt

You are designing image-edit prompts for a rigorous visual difference benchmark.
Look at the image and propose exactly 5 candidate edits where ONE visible object changes position.
Each candidate must move an existing visible object to a different plausible location in the same scene.
Do not add, remove, recolor, resize, restyle, rotate, deform, duplicate, or replace objects.
Prefer clearly localized, countable objects whose position change would be easy to verify.
Avoid vague background regions.
The `edit_prompt` must be clear and effective to instruct an image editing model to preserve everything except that single object's position.
The `difference_description` must be a short dataset annotation beginning with 'In the second image,'.

Original OmniEdit swap instruction for context only:
- {`original_instruction_1`}
- {`original_instruction_2`}

Position Edit Prompt

{`selected__candidate__edit__prompt`}

Keep the same scene, camera angle, lighting, style, object identities, colors, and object counts. Only change the selected object's position.

A.2.3 Human Annotation and Cross-Validation

For curated puzzle pairs without sufficiently detailed source annotations, we collected ground-truth difference descriptions from volunteer domain experts who are fluent in English. Annotators inspected each ordered image pair side by side and enumerated all visible differences. Each description was required to identify a single change, specify the affected object or region using distinguishing visual attributes and spatial cues, and state explicitly how the first and second images differ. Annotators were asked to capture not only added or missing objects, but also localized changes in color, shape, orientation, and fine-grained pattern or texture, while avoiding speculative or overly vague descriptions. To ensure the quality of the annotated image differences, every annotation subsequently underwent a second review to be cross-validated by a different expert. The reviewer re-examined the image pair for coverage and visual support, and rewrote descriptions that were inaccurate, ambiguous, overly broad, grammatically unclear, or inconsistent in comparison direction.

The resulting reviewed descriptions form the ground-truth pool from which reference differences are selected during question construction.

A.3 Additional Details on False Differences Generation for VDiff-Bench-MCQ

We use Gemini 2.5 Pro and GPT-5.5 to generate false differences conditioned on image pairs and their reference annotations. Structured filters remove duplicates and exact truth matches.

Human Verification Model-generated false differences might be too semantically unplausible or too vague to be judged, therefore not acting as challenging "negative" choices that an evaluated MLLM needs to distinguish from. Therefore, we invite a human expert to inspect the image pairs and generated false differences and refine, rewrite, or discard low-quality ones.

Question assembly. To construct the final multiple-choice questions in our VDiff-Bench dataset, we retain the ground truth difference caption, two false difference statements for each image pair, append the fixed no-difference distractor option, and shuffle the four options to be randomly ordered. To control for option-position bias and rule out fixed response strategies (e.g., always selecting option D), we shuffle the four choices using a fixed random seed; consequently, both the correct-answer labels and the no-difference distractor positions are approximately balanced across A–D. The generation prompt is provided below.

Generation Prompt. For each image pair, the model receives a list of ground truth differences in them and are prompted to use these as references to create 2 to 6 statements that are factually incorrect but visually plausible, formatted similarly as the references. The prompt instructs the model to consider 4 strategies: swapping the target object, swapping the source object, changing an attribute of the target, and introducing a plausible alternative change to an entity visible in at least one image. The prompt also explicitly prohibits invented entities, indirect references to the true change, and complex relational wording. The prompt additionally requests up to five potentially missing true differences for audit; these suggestions are not treated as annotations, but any overlap with a displayed candidate alternative is conservatively flagged for visual adjudication. Fig. 5 below presents the prompt for the model to generate the false differences.

False Differences Generation Prompt

Role
You are a precision image-analysis engine specialized in generating hard-negative distractors for change detection datasets.

Context
Dataset Category: {`category_type`}
Expected Change Logic: {`allowed_family_text`}

Input Data
1. Images: [Provided]
2. Ground Truth (GT) Differences:
{`truth_bullets`}

Task 1: Generate False Differences (Hard Negatives)
Create {`distractor_count`} to {`max_false_differences`} statements that are FACTUALLY INCORRECT but visually plausible based on the scene.
Each false difference should be phrased and formatted just like the GT.
Use these strategies:

- **GT Swap:** Take a GT change and swap the ending object, e.g., B in "A is changed to B", to another object that is also present in the image, preferably located close to A / B.
- **Input Swap:** Take a GT change and swap the beginning object, e.g., A in "A is changed to B", to another object that is visible in the image, preferably located close to A / B.

- **GT change:** Take a GT change and change characteristics of the ending object, e.g., `adj_Y B` in "`adj_X A` is changed to a `adj_Y B`", to another characteristic, e.g., "`adj_X A` is changed to a `adj_P B`", such as color, material, shape, etc.
- **Non-Substitution changes:** Randomly add false changes from non-substitution categories, such as color changes, left/right/closer/further movement, character or shape changes, illumination changes, noise changes, or RGB tone changes. Make sure the object you mention exists in at least one of the images.

Constraints:

- Your false differences should be phrased and formatted just like the GT differences.
- The subject of your false statement MUST exist and be visible in at least one of the images. Do not invent new objects.
- Preserve the main changed object from the GT in at least one but at most two generated false differences.
- Do NOT mention the ground truth. For example, if the ground truth is "the white SUV has disappeared", do NOT describe something else as moved relative to "where the SUV was".
- Do NOT use complex relational phrasing such as "behind where X was", "next to where Y had been", or "in the spot formerly occupied by".
- Do NOT generate more than two GT Swap differences that look similar to each other.

Task 2: Identify Potential Missing True Differences
List up to 5 real visual changes visible in the images that were NOT captured in the provided Ground Truth list.
If the GT is exhaustive, return an empty list.

Output Format (JSON)
Return the response in this exact JSON structure:

```
{
  "false_differences": [
    {
      "statement": "The false sentence here.",
      "based_on_true_difference":
        "The exact GT sentence this is mimicking.",
      "strategy_used":
        "Object Swap / Spatial Error / State Error"
    }
  ],
  "potential_missing_true_differences": [
    "Statement 1",
    "Statement 2"
  ]
}
```

Figure 5: Prompt used to generate hard-negative distractors.

A.4 Stratified Dataset Statistics for VDiff-Bench-MCQ and VDiff-Bench-Gen

Table 6 summarizes detailed statistics of data in the two tracks of VDiff-Bench using their respective evaluation units: MCQ questions and Gen image pairs with atomic reference differences.

B Inference Prompts

B.1 VDiff-Bench-MCQ Inference Prompt

All MCQ inferences use the following textual template, with the four MCQ options substituted from the corresponding per-example inference artifact. Images are supplied directly to the model in A–B order; no difference mask is used. All runs use temperature zero.:

VDiff-Bench-MCQ		VDiff-Bench-Gen		
Category	**Questions**	**Selection group**	**Pairs**	**References**
Position	158	Position	26	39
Motion	138	Motion	54	117
Regional color	170	Regional color	17	17
Whole-image color	147	Appearance/disappearance	40	73
Appearance/disappearance	201	Texture	3	9
Noise/resolution	150	Substitution/size	17	17
Texture	177	OCR/text	17	17
Substitution/size	199	Noise + whole-image color	42	84
OCR/text	221	Texture + whole-image color	42	84
Illumination	195	Illumination + whole-image color	42	84
Total	**1,756**	**Total**	**300**	**541**

Table 6: Statistics of VDIFF-BENCH-MCQ and VDIFF-BENCH-GEN. The two halves are independent inventories; rows do not imply correspondence. Gen contains 100, 168, 23, and 9 pairs with one, two, three, and four reference differences, respectively (mean: 1.80).

VDiff-Bench-MCQ Evaluation Prompt

You are given image A and image B.

Choose the option that correctly describes a real difference between image A and image B.
Options:
A. [option A]
B. [option B]
C. [option C]
D. [option D]

Answer ONLY one of the choice labels (e.g. 'A.') without other text:

B.2 VDIFF-BENCH-GEN INFERENCE PROMPT

For direct open-ended generation on VDIFF-BENCH-GEN, models receive the ordered image pair without answer choices or reference annotations and are asked to list atomic visible changes in bullet points to allow easy parsing, since some models might output broken JSON formatting. The template below requests change direction and location while separating independent differences.

Open-Ended Generation Prompt

Compare the first image with the second image very carefully. Every image pair in this benchmark is guaranteed to contain at least one real, directly visible difference. Find as many visible differences as you can. Describe one atomic difference per line: one changed object, attribute, relation, text span, or global property. State the first-to-second-image direction and a concise location when useful. Do not mention similarities, uncertainty, reasoning, or possible differences. Do not combine independent changes.

Report at most 10 of the strongest, clearly visible differences. When many similar instances share one systematic change (for example, an entire palette, repeated pattern, region, or group changes color), describe that collective change once instead of making one bullet per instance. Likewise, describe a global or regional color, lighting, noise, or smoothing change as one difference rather than repeating it for individual objects.

Output only a Markdown bullet list. Begin every item with "- ". Do not output JSON, a heading, reasoning, or an empty list. Example format:

- A small bird near the upper-left is present in the first image but missing in the second

image.
- The car is red in the first image and blue in the second image.

C Additional Details on EHE

We summarize the executable components of EHE and their adaptations to OmniDiff data and the image difference generation task. The pseudocode and prompt excerpts below simplify the saved implementations while preserving their control flow; they are not verbatim transcripts. Appendix C.1 covers MCQ initialization, optimization, and the final harness, including the selection rules and development trajectory; Appendix C.1.4 illustrates the process with a recorded failure and recovery.

C.1 EHE Experiments on VDiff-Bench-MCQ

C.1.1 Brief Overview

The harness is a Python program surrounding a frozen Qwen3-VL-8B Thinking model. It receives two images and answer choices, constructs evidence and a route-specific prompt, calls the model, and parses its answer. Reference labels are used by the external scorer, not supplied to the answering model.

Initialization and data splits. We instantiate the framework with GPT-5.6-sol as the optimizer and Qwen3-VL-8B Thinking as the frozen executor. The optimization run starts from a supplied global-signal harness H_0 and uses fixed, category-balanced cohorts of 160 training and 200 development questions. Here, training data provide optimization feedback rather than parameter updates. Other data from VDiff-Bench-MCQ are excluded from the optimization loop.

Initial Harness (pseudo-code) The initial option-text router detects alternatives describing global brightness, color tone, noise, or sharpness changes. These questions receive numerical evidence; the remaining questions use direct visual comparison.

```
def initial_harness(A, B, choices):
    if describes_global_change(choices):
        evidence = global_measurements(A, B)
        prompt = global_prompt(choices, evidence)
    else:
        prompt = direct_prompt(choices)
    response = Qwen(prompt, images=[A, B], budget=1200)
    return parse_label_or_explicit_answer(response)
```

Initial base agent prompt Both routes request one option label. The global route additionally explains how to interpret the measurements.

> **Direct route.** Compare image A and image B and select exactly one supported difference. Check the direction of change and avoid using an unrelated region. Return JSON with an `answer` field containing one option letter.
>
> **Global route.** Act as a global image-signal specialist. Select exactly one option using the images and full-frame measurements. RGB and luminance differences are computed as B minus A. A larger median residual suggests more noise; edge energy can increase with either noise or sharpening. Check that the claimed change is global. Return only {`"answer": "LETTER"`}.
>
> **Supplied content.** Answer choices; measurements for the global route; both images for either route.

Initial tools The workspace exposes frozen-model calls, whole-image pair statistics, and corresponding crops specified by normalized coordinates; the expert may also implement deterministic image operations in candidate code. The initial harness uses pair statistics and local numerical routines for RGB means, luminance means and medians, median-filter residuals, edge energy, and the fraction of changed luminance pixels. Paired crops are available to the expert but are not invoked by this initial MCQ harness.

C.1.2 Expert-guiderd Harness Evolution Rules

The expert edits reusable inference code, including routing, prompts, tool handling, response budgets, and parsing. It receives the current harness, detailed training traces, aggregate development feedback, and records of earlier proposals. The following condensed instruction captures its role:

Harness	Correct / 200	Accuracy	Invalid
Initial harness	160	80.0%	7
+ Measurement fallback (R2)	163	81.5%	4
+ Extended response budget (R5)	175	87.5%	1

Table 7: **Harness evolution on adaptive development data.** Accepted modifications improve the frozen 8B model sequentially. These results describe an optimization trajectory, rather than independent ablations or held-out performance.

> **Objective.** Improve the frozen model's reusable comparison procedure without changing its weights.
> **Inspect.** Read successes and failures in training traces, including images, choices, reference answers, model responses, and tool errors. Use only aggregate development feedback.
> **Propose and implement.** Make one falsifiable component modification: add, edit, remove, or restore a mechanism. Explain its hypothesis, expected benefit, failure criterion, cost, and invalid-answer risk. Tools must not use reference labels or dataset metadata.
> **Check and submit.** Mock modified routes and failure paths, including malformed responses and unequal image sizes. Submit candidate Python code and a proposal record for external evaluation.

The controller executes the candidate on the fixed training and development cohorts and compares its development answers with those of the incumbent. Accepted candidates replace the incumbent; rejected candidates are recorded without replacing it. Detailed training traces and aggregate development comparisons enter the workspace for the next expert session (Figure 6).
Figure 6 separates expert code generation from controlled execution and selection. Each call to `ExpertCodingAgent` starts a fresh session with the saved workspace. `Evaluate` executes the harness on image pairs and options, then scores predictions against references externally; the answering model does not receive those references. `PairedGate` returns aggregate repairs, harms, validity and call statistics, and protected-group regressions. The stopping routine tracks consecutive rejections and resets its counter after acceptance.

Acceptance and stopping. The selection gate accepts a candidate through one of three paths: at least three net repairs without additional invalid answers; at least two fewer invalid answers without an accuracy loss; or at least 15% fewer model calls with unchanged accuracy and no additional invalid answers. All paths reject losses exceeding two correct answers in any category or source group with at least ten examples. The search permits at most ten rounds and stops after four consecutive rejections.

C.1.3 Expert-guided Harness Evolution Process

What changes during EHE. Two accepted changes illustrate what evolves. First, an exception handler transfers questions to direct vision when global measurements fail, including pixelwise comparisons between unequal-sized images. Second, the response budget increases from 1,200 to 1,800 tokens to reduce unfinished answers. Other proposed changes, such as retries and alternative answer-recovery mechanisms, were evaluated but not retained. The measured trajectory is reported in Table 7.

Adaptive development trajectory. Table 7 shows an improvement from 80.0% to 87.5%, a gain of 7.5 percentage points over the initial harness. Round 2 introduces a visual fallback for measurement exceptions. Round 5 increases the specialist response ceiling from 1,200 to 1,800 tokens, yielding 14 repairs and two harms relative to Round 2 while reducing invalid answers from four to one. Both accepted candidates make 200 Qwen calls, although the larger response budget can increase token consumption. Later modifications are not uniformly beneficial: Round 8 restores an earlier retry mechanism but falls to 164/200 and is rejected. Four consecutive rejections terminate the search with Round 5 as the incumbent. The development cohort guides candidate selection, so these results describe adaptive optimization rather than held-out performance.

C.1.4 Trace Excerpt: Recovering from a Measurement Failure

The following excerpt illustrates the diagnosis–intervention loop using a recorded training case and the Round 2 proposal. Inputs and reference wording are shortened; the expert diagnosis and change

EHE: one optimization run

Inputs: initial harness H_0, frozen model Q, tool interfaces T, expert guidance I, fixed training/development cohorts, and baseline results.
Output: selected inference harness H.

```
H, R_dev = H0, baseline_dev_results
W = InitializeWorkspace(H0, I, baseline_feedback)

for round in optimization_budget:
    # 1-2. Expert reads history and implements one change
    H_new, proposal = ExpertCodingAgent(W)

    # 3. External evaluation; reference labels stay in scorer
    R_train = Evaluate(H_new, Q, T, train_cohort)
    R_new   = Evaluate(H_new, Q, T, dev_cohort)
    decision = PairedGate(R_dev, R_new)
    if decision.accept:
        H, R_dev = H_new, R_new

    # 4. Save both accepted and rejected proposals
    W.store(H_new, proposal, R_train.detailed_feedback,
            R_new.aggregate_feedback, decision.summary, H)
    if StopCriterionMet(decision):
        break

return H  # Deploy with frozen Q; no expert calls
```

Figure 6: **Pseudocode for expert-guided harness evolution.** The controller retains incumbent development results for paired selection. Only detailed training feedback and aggregate development feedback enter the expert's workspace; held-out data are excluded.

are paraphrased from the submitted proposal, not reproduced as a reasoning transcript. Dataset identifiers, file paths, and unrelated execution fields are omitted.
This training example supports the proposed failure mechanism; acceptance is determined separately on the fixed development cohort. Round 2 yields 12 repairs and nine harms (160 to 163 correct out of 200), reduces invalid answers from seven to four, and causes no protected-group regression, so the controller accepts it. The single recovered case is illustrative and does not account for the entire development gain.

Recorded training case: unequal-sized image pair

Input to the answering model. Two airplane-puzzle images with spatial array shapes $(1405, 1412)$ and $(1405, 1409)$, and four options (abridged):

A. White curved motion lines beside the propeller disappear.
B. Birds to the left of the plane disappear.
C. There is no difference.
D. An oval wing marking becomes lighter.

Ground truth (scoring and training feedback only). A: the white curved motion lines to the right of the propeller are present in the first image and missing in the second.

Before the modification: execution failure. The measurement route raises `ValueError: operands could not be broadcast together` for the unequal array shapes. Recorded prediction: `null`; valid: `false`; correct: `false`. No model call occurs, so there is no Qwen-generated answer in this failed execution.

Expert diagnosis (proposal summary). The Round 2 proposal identifies four training invalids caused by unequal-shape measurement errors before model invocation. Unavailable numerical evidence should be discarded, allowing the original images and options to be assessed by direct vision.

Proposed modification: ADD an exception fallback. Wrap global measurement computation in exception handling. If measurements succeed, retain the existing measurement-assisted prompt; if

they fail, invoke the direct visual comparison prompt on the original images and options. This changes harness control flow rather than model weights.

After the modification: recorded recovery. The trace records `measurement_fallback` with reason `global measurement unavailable:ValueError`, followed by direct vision. Qwen returns {`"answer": "A"`}; valid: `true`; correct: `true`.

C.2 Adapting EHE on VDiff-Bench-Gen

We adapt the MCQ EHE-optimized harness by replacing option selection with an atomic list of differences and extending its evidence collection to provide separate color, brightness, and texture cues, together with full images and corresponding center crops. Unlike the option-conditioned MCQ router, this procedure collects evidence without answer choices. The generator receives neither reference differences nor evaluator decisions at inference. VDiff-Judge remains a separate, fixed evaluator during this experiment.

Final prompt and harness (excerpts)

Generation pseudo-prompt. Compare A to B and find clearly visible changes. Check global color balance separately from brightness, smoothing, and noise. For local changes, identify the same object in both images; distinguish object changes from framing changes. Quote changed text when legible. Report at most ten nonduplicated Markdown bullets, one independently verifiable change per bullet, with its location and A-to-B direction. Treat numerical measurements as fallible cues and confirm them visually.

```
evidence = color_brightness_texture_statistics(A, B)
crops = paired_center_crops(A, B)
response = Qwen(gen_prompt(evidence), [A, B] + crops)
claims = parse_bullets(response)
if not claims:
    claims = edit_supported_changes_from_saved_notes()
if not claims:
    claims = retry_visual_generation(A, B)
return claims
```

Tools and output handling The center crops cover the same central region in each image and retain native detail. Color statistics separate a shared brightness shift from channel-specific changes; high-frequency energy supplies a texture cue rather than a definitive label. If no bullets are recovered, a conditional text-only editor converts saved comparison notes into an answer, followed, if necessary, by a fresh visual call. These recovery calls can increase inference cost relative to direct generation.

C.3 Adapting EHE on OmniDiff

For OmniDiff, we adapt the MCQ EHE-optimized harness by replacing option selection with prose captioning and separating object-level comparison from global lighting assessment. The final Real and Render harnesses use domain-appropriate caption examples. Visual generation describes object-level changes, while a deterministic luminance-ratio rule supplies a lighting sentence when a domain-specific threshold is crossed. Thresholds are fitted on training data. The caption assembly stage filters model-generated lighting statements before adding this sentence. Reference captions are not provided during inference.

Final prompt and harness (excerpts)

Caption pseudo-prompt. Compare the before and after images. Describe visible changes using object identity, before/after states, and scene-relative locations. Return only a prose caption following the supplied examples. Leave global lighting to the measurement component. For the selected Real prompt, scan the scene systematically, reconcile appearances and disappearances into movements where appropriate, and check peripheral details after the most obvious change.

```
lighting = lighting_rule(luminance_ratio(A, B), domain)
caption = Qwen(caption_prompt(domain), [A, B])
if empty(caption):
    caption = edit_caption_from_saved_notes()
if empty(caption):
    caption = retry_visual_caption(A, B)
```

```
    if empty(caption):
        caption = salvage_caption_from_saved_responses()
    caption = clean_and_limit_length(caption)
    caption = combine_object_caption_and_lighting(
        caption, lighting)
    if domain == "Real" and empty(caption):
        caption = "The scene has changed."
    return caption
```

Tools and output handling Both domain harnesses use full-image comparison and luminance statistics, without paired crops. The Real prompt uses a systematic before/after inventory to capture secondary changes; the Render prompt requests a short caption of two to four sentences. Output processing removes formatting, filters lighting statements, and limits caption length. If generation remains empty after a text-only editing call and a visual retry, the harness attempts to recover a caption from saved responses. The Real harness additionally uses a generic nonempty fallback, which provides no specific change evidence. These procedures adapt the output contract and evidence handling to captioning rather than changing the base model's weights.

D Additional Results on VDiff-Judge

This section complements the pipeline in Section 4.1 with prompt and harness excerpts, tool descriptions, scoring conventions, and the human-validation protocol. The pseudocode abstracts conditional checks and error handling; it is not a verbatim execution trace. Unlike the generation prompts in Appendix B.2, the judge receives reference differences as well as the image pair and predicted claims.

D.1 Prompt, Harness, and Visual Tools

D.1.1 Prompt and harness (excerpts)

The judge uses frozen Qwen3-VL-8B Instruct. For each predicted claim, it compares against the complete reference list, first using the images and an automatically constructed evidence panel, then conditionally revisiting the comparisons with targeted evidence. The following condensed prompt preserves the main comparison criteria and output contract.

> **Inputs.** Ordered images A and B; one predicted difference; every numbered reference difference; visual evidence and its metadata.
>
> **Comparison instruction.** Compare the prediction independently with every reference. Different wording and specificity are allowed when they identify the same visible change. Object identity, changed attribute, location, and A-to-B direction must not conflict. A vague related description is only a partial match. Do not match two claims merely because each describes a real change somewhere in the images.
>
> **Grounding instruction.** For local matches or ambiguous comparisons, localize the prediction's object and the reference's object separately using normalized bounding boxes. Request closer inspection when needed; return no box if the object cannot be localized.
>
> **Output for each reference.** Reference index; verdict (match, partial, contradiction, unrelated, or uncertain); score; short justification; visual-verification flag; cited region identifiers; prediction and reference boxes; crop request.

```
evidence = analyze_pair_and_render_panel(A, B)
for p in predictions:
    comparisons = judge(p, all_references, A, B, evidence)
    semantics = recheck_wording_if_needed(p, all_references)
    if local_evidence_needed(p, comparisons):
        boxes = collect_prediction_and_reference_boxes(
            comparisons)
        crops = paired_crops_or_region_fallback(A, B, boxes)
        comparisons = judge_again(p, all_references, crops)
    comparisons = combine_visual_and_semantic_checks(
        comparisons, semantics)
    comparisons = verify_presence_and_apply_guards(
        p, comparisons, A, B)
    comparisons = recheck_unmatched_if_supported(
        p, comparisons, A, B)
    save_comparisons_and_evidence(p, comparisons)
return one_to_one_assignment_and_duplicate_aware_scores()
```

D.1.2 TOOLS AND OUTPUT HANDLING

Global evidence and candidate regions. Deterministic image analysis supplies alignment information, RGB and luminance statistics, noise and edge cues, difference maps, and candidate changed regions. An evidence panel combines the images and these maps. Measurements guide interpretation of global changes; a changed region alone does not establish which object or attribute changed.
Grounding and paired crops. Grounding boxes are proposed by the visual judge itself, separately for the prediction and reference, rather than produced by a dedicated external grounding model. The harness converts valid boxes from normalized coordinates into corresponding A/B crops, adds surrounding context, and renders an enlarged comparison panel. When usable boxes are unavailable, it can use regions from the precomputed difference analysis. The judge then reassesses the correspondence and cites the inspected regions. Local claims judged visually plausible can receive this inspection even if the initial reference comparison found no match.
Semantic and directional checks. A conditional text-only recheck distinguishes equivalent descriptions from related but different changes. It is combined with visual verification rather than treated as an independent proof that the change occurs. Additional checks address quoted text transitions, count and scope, object identity, and appearance/disappearance direction. Presence checks inspect the claimed object in A and B independently of the reference wording. Visually supported but unmatched claims may undergo a further image-based correspondence recheck; a possibly missing annotation is recorded separately from a successful reference match. These checks are fallible and do not make the judge a substitute for human verification.
Structured outputs and traces. The harness validates JSON responses and retries malformed outputs. It stores raw judge responses, per-reference verdicts and scores, grounding boxes, evidence-panel and crop artifacts, and subsequent checks. Final comparisons form the prediction–reference score matrix used below. Empty predictions receive zero credit; traces make individual decisions inspectable without implying that every decision is correct.

D.2 VDIFF-JUDGE SCORING DETAILS

One-to-one assignment first maximizes the number of eligible claim–reference matches and then their total score, preventing repeated credit for the same annotated difference. An eligible correspondence has a final match verdict and a score of at least 0.72. An additional prediction verified against an already credited reference is treated as a duplicate and excluded from the effective prediction count, rather than counted as another match or a false positive. Micro precision and recall pool matched, effective prediction, and reference counts across image pairs; micro-F1 is their harmonic mean. Pair-macro F1 averages the F1 computed separately for each image pair. Exact-set accuracy measures the fraction of pairs with all references matched and no remaining unmatched predictions after verified duplicates are collapsed. An unmatched claim can reflect an incorrect prediction, a missing reference, or an evaluator error, so it is not automatically a hallucination. We calibrate prompts, evidence routing, and consistency checks without updating the judge's weights; the human agreement study and its relation to calibration are described in Appendix D.3.

D.3 HUMAN VALIDATION PROTOCOL

We evaluate judge agreement on 100 sampled generated difference claims from 100 distinct image pairs containing 155 atomic ground-truth differences. The claims originate from ten generation models, with one claim selected per pair; generator identities are hidden in the evaluation packet. Each item contains the ordered image pair, the selected claim, and the complete reference list for that pair. A human expert annotator and three automated judges—direct Qwen3-VL-8B-Instruct, direct GPT-5.6-Sol, and VDIFF-JUDGE, our evidence-grounded Qwen evaluator—decide whether the claim aligns with at least one reference difference, considering object identity, attributes, location, and change direction. Each judge returns an aligned/not-aligned decision for each claim. Agreement is the fraction of the 100 decisions matching the human labels; Cohen's κ additionally adjusts for chance agreement under the observed label frequencies. Our evaluator yields 97% agreement with human labels. This study assesses claim-level reference alignment, rather than complete-set recovery.

Evaluator	Agreement	Cohen's κ
Direct Qwen	84.0	0.651
GPT-5.6-Sol	85.0	0.679
VDIFF-JUDGE	**97.0**	**0.937**

Table 8: Agreement scores with human claim labels on 100 distinct image pairs.

E ADDITIONAL RESULTS

E.1 FAILURE CASES

E.1.1 VDIFF-BENCH-MCQ FAILURE CASES

No-Difference Selection on Low-Level Changes Table 9 measures explicit selection of the no-difference distractor, rather than all incorrect answers. Qwen3-VL-8B Instruct, InternVL3.5-8B, and LLaVA-OneVision-Qwen2-7B select it on 24.8%, 36.8%, and 43.1% of questions, respectively. Their rates exceed 70% on whole-image color and texture changes. This behavior is not shared by every small checkpoint: Qwen3-VL-8B Thinking selects no difference on only 2.1% overall. Grok 4.3 shows a different category-specific pattern, selecting no difference on 79.3% of noise and 72.3% of texture questions despite a 4.1% rate on whole-image color. Thus, low-level failures include both missed-change decisions and selection of an incorrect descriptive alternative; a low no-difference rate alone does not imply high accuracy.

Model	Whole color	Noise	Texture	Illum.	Overall
GPT-5.4	9.5	0.0	9.6	14.4	5.2
Qwen3-VL-8B Thinking	0.7	0.0	1.1	11.8	2.1
GPT-5.6-Terra	2.0	0.0	0.6	6.7	1.4
Gemini 2.5 Flash	24.5	0.0	39.0	21.0	12.1
Gemini 3.1 Pro	1.4	0.0	2.3	4.6	1.6
Gemini 3.5 Flash	2.0	0.0	0.0	1.0	0.5
Grok 4.3	4.1	79.3	72.3	35.4	21.8
Doubao Seed 1.6 Vision	0.0	0.0	4.0	6.2	1.3
Qwen3-VL-8B Instruct	74.8	22.0	74.0	35.4	24.8
InternVL3.5-8B	93.2	87.3	90.4	42.1	36.8
LLaVA-OneVision-Qwen2-7B	93.9	88.7	72.3	72.8	43.1
Kimi K3	15.0	0.0	10.7	15.9	5.9

Table 9: Rate (%) at which each model selects the no-difference distractor. Category denominators are 147 (whole color), 150 (noise), 177 (texture), and 195 (illumination); "Overall" covers all 1,756 questions, including invalid responses in the denominator. High rates indicate a missed-change selection rather than confusion with another detailed description.

Representative Qualitative Failure Cases Figure 7 illustrates three distinct sources of difficulty. The smoothing operation changes local texture without altering scene semantics; the RGB shift distributes evidence across the entire image; and the surveillance example requires locating small missing entities in a low-resolution frame. The repeated no-difference choice is consistent with the aggregate behavior in Table 9.



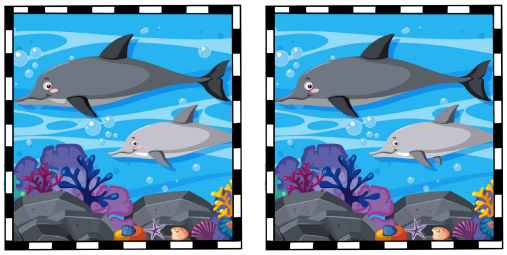
(a) Subtle smoothing (b) Global RGB shift (c) Two people disappear

Figure 7: Representative no-difference selections. (a) Smoothing: Qwen, Gemini, LLaVA, and InternVL select no difference; GPT-5.4 selects the keyed answer. (b) Global RGB shift: Qwen, LLaVA, and InternVL select no difference. (c) Two people disappear: LLaVA and InternVL select no difference.

E.1.2 VDIFF-BENCH-GEN FAILURE CASES

Open-ended generation can fail through both incomplete discovery and incorrect interpretation of a changed region. Figure 8 illustrates these behaviors with saved GPT-5.4 responses. In the puzzle, the model correctly identifies one added flipper but omits four other annotated changes. In the surveillance pair, it describes a disappearing sign instead of the pedestrians' movement. These cases show why evaluation must assess both coverage and the identity and direction of each claimed change.

(a) Incomplete discovery



A B

GT (abridged): Five changes: the smaller dolphin's blush, the bubble above the larger dolphin's tail, the pale patch beneath its mouth, and a blue coral spot disappear; a second flipper appears beneath the smaller dolphin.

GPT-5.4 (complete answer): "A small dark fin under the lower dolphin's belly near its tail is absent in the first image and present in the second image."

Failure: Finds the added flipper but omits four annotated changes.

(b) Wrong referent



A B

GT: "the people in the foreground have moved closer to the camera"

GPT-5.4 (complete answer): "The purple banner/sign on the pole by the lower-left sidewalk is present in the first image but missing in the second image."

Failure: Describes a disappearing sign rather than tracking the pedestrians' movement.

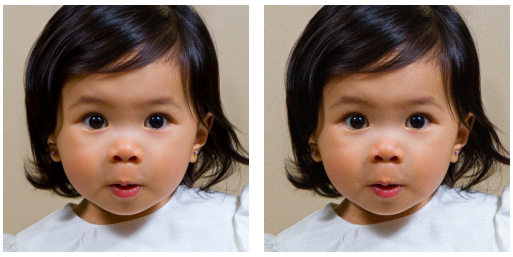
Figure 8: **Generation failures on original image pairs.** A and B follow the evaluated input order. GPT-5.4 outputs are reproduced in full; the puzzle references are abridged. The examples distinguish incomplete coverage from an incorrect object interpretation.

E.2 EHE IMPROVEMENT EXAMPLES

E.2.1 MCQ IMPROVEMENTS

Figure 9 compares direct Qwen3-VL-8B Thinking with its EHE harness on two questions answered incorrectly by the direct model. EHE identifies added noise instead of increased clarity and corrects the direction of a whole-image color shift. Both use the measurement-assisted route in the saved harness traces. The model weights and image pairs are unchanged.

(a) Noise versus clarity

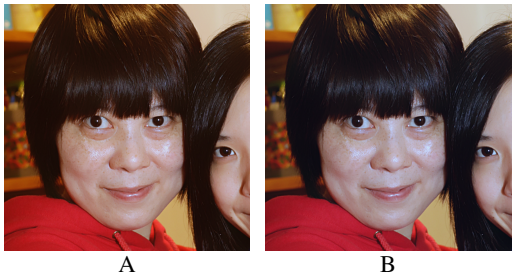


A B

A. The second image is clearer than the first.
B. The first image has a warmer tone.
C. The second image is more noisy.
D. There is no difference.

Ground truth: C.
Direct Qwen Thinking: A (incorrect).
Qwen Thinking + EHE: C (correct).

(b) Color-shift direction



A B

A. There is no difference.
B. The woman's skin appears more tanned in B.
C. B has a cooler blue–cyan tone than A.
D. B has a warmer yellow–red tone than A.

Ground truth: C.
Direct Qwen Thinking: D (incorrect).
Qwen Thinking + EHE: C (correct).

Figure 9: **EHE corrects MCQ decisions with the same frozen Qwen3-VL-8B Thinking model.** The selected answers change from an incorrect alternative to the keyed difference. Options are shortened while preserving their meaning and original labels.

E.2.2 Generation Improvements

Figure 10 illustrates two distinct gains from the adapted generation harness. First, a pair with no parsed direct answer receives a correct disappearance description. Second, EHE reports both global color and noise changes, whereas the direct response omits noise and spreads its description across individual scene elements. These selected examples complement the aggregate results; they do not establish that every harness modification improves every example.

(a) Recovering a usable answer



A B

GT: "the people in the first image are gone"

Direct Qwen Thinking: No parsed difference was returned.

Qwen Thinking + EHE: "The group of people on the sidewalk in Image A is absent in Image B."

Reference coverage: $0/1 \rightarrow 1/1$.

(b) Recovering both global changes



A B

GT (abridged): B has more noise and a stronger blue, weaker green overall tone.

Direct Qwen Thinking (excerpts from 10 claims): "The skin tone is more yellowish in the first image and more neutral in the second image."; "The hair color is lighter in the first image and darker in the second image."

Qwen Thinking + EHE (complete answer):
"Global color balance: B has less red, less green, more blue (chromatic shift)"
"Noise level: B has increased visible noise/grain"

Reference coverage: $1/2 \rightarrow 2/2$.

Figure 10: **EHE improves open-ended output on the same image pairs.** (a) An empty parsed answer becomes a supported disappearance claim. (b) The direct output omits noise; EHE describes both annotated global changes. Coverage counts use the saved VDiff-Judge evaluation. These examples illustrate output recovery and difference coverage, rather than isolating the contribution of an individual harness component.

E.3 Additional Qualitative Results

Below, we provide more qualitative results on how different models perform on data from each category of our benchmark.

F Supplementary Evaluation Details

F.1 MCQ Model Configuration

We evaluate seven proprietary MLLMs: GPT-5.6-Terra and GPT-5.4 (snapshot `2026-03-05`) (OpenAI, 2026a), Gemini 2.5 Flash (Comanici et al., 2025), Gemini 3.1 Pro (Preview) (Gemini Team, 2026), Gemini 3.5 Flash (Kavukcuoglu et al., 2026), Grok 4.3 (xAI, 2026), and Doubao Seed 1.6 Vision (ByteDance Seed, 2025). For Doubao Seed 1.6 Vision, we use the `doubao-seed-1-6-vision-250815` snapshot. We additionally evaluate five open-weight checkpoints: Qwen3-VL-8B Instruct and Thinking (Bai et al., 2025), InternVL3.5-8B (Wang et al., 2025), LLaVA-OneVision-Qwen2-7B (Li et al., 2024), and Kimi K3 (Kimi Team, 2026).

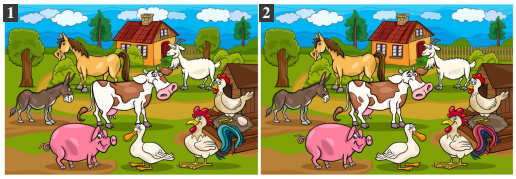
G Discussion

Recognition, generation, and verification. Multiple-choice identification and open-ended description probe complementary capabilities. Answer options supply candidate changes, whereas generation requires models to discover and articulate differences without this support. The observed performance gap therefore cautions against treating strong multiple-choice accuracy as evidence of reliable difference captioning. Our harness results further suggest that measured performance depends on both the underlying model and the inference procedure used to expose visual evidence.

Implications for image-editing evaluation. MLLM-based image-editing evaluation similarly requires identifying intended changes and unintended alterations. Our findings suggest that fluent judgments should be supported by localized visual evidence and explicit checks of object identity,

GPT-5.4 | Gemini 3.1 | Gemini 3.5 | Grok 4.3 | Qwen3-VL-8B

POSITION



A: The white duck in the center is facing left in Image 1 but facing front in Image 2.
B: The pig's tail points downward in Image 1 but curls upward in Image 2.
C: The horse's tail changes color from brown in Image 1 to black in Image 2.
D: No difference between Image 1 and Image 2.

MODELS ✓ ✗ ✗ ✓ ✗

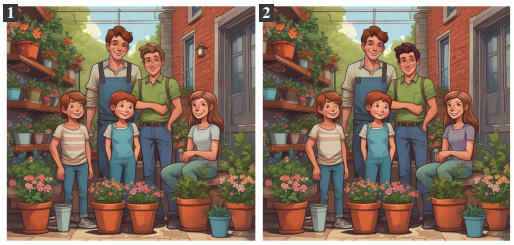
MOTION



A: In Image 2, the woman's eyes are looking down.
B: The beer glass is now tilted and has a higher liquid level.
C: The woman is now drinking from the glass and looking up.
D: No difference between Image 1 and Image 2.

MODELS ✗ ✓ ✓ ✓ ✓

REGIONAL COLOR



A: A flower pot on the middle shelf on the left side is green in Image 1 but grey in Image 2.
B: No difference between Image 1 and Image 2.
C: The small pot on the far right is white in Image 1 and changes to blue in Image 2.
D: The overalls worn by the child in the center are light blue in Image 1 and have changed to a darker blue in Image 2.

MODELS ✗ ✓ ✗ ✓ ✗

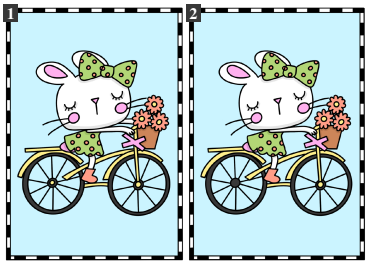
WHOLE-IMAGE COLOR



A: No difference between Image 1 and Image 2.
B: Image 2 has a cooler blue–green color tone than Image 1.
C: The baby's facial expression is different in Image 2.
D: Image 2 has a warmer red–magenta color tone than Image 1.

MODELS ✗ ✓ ✓ ✓ ✗

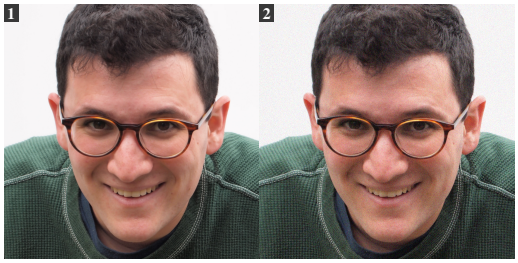
APPEAR / DISAPPEAR



A: No difference between Image 1 and Image 2.
B: The ribbon of the rabbit is missing a round pattern in Image 2 relative to Image 1.
C: The right eye of the rabbit is missing an eyelash in Image 2 relative to Image 1.
D: The bicycle is missing a pedal in Image 2 relative to Image 1.

MODELS ✗ ✗ ✗ ✓ ✓

NOISE / RESOLUTION



A: No difference between Image 1 and Image 2.
B: Image 2 has a warmer color temperature than Image 1.
C: Image 2 is clearer than Image 1.
D: Image 2 is noisier than Image 1.

MODELS ✓ ✓ ✓ ✗ ✗

Figure 11: **Qualitative VDIFF-BENCH results across all ten change categories (part 1 of 2).** Each panel shows the complete image pair at a common height, all four answer choices with the ground truth colored, and correctness for six representative MLLMs. Green checks and red crosses indicate correct and incorrect selections.

attributes, and change direction. Extending our evaluator to this setting would additionally require conditioning verification on the editing instruction and assessing preservation of unrelated content.

GPT-5.4 Gemini 3.1 Gemini 3.5 Grok 4.3 Qwen3-VL-8B

TEXTURE



A: No difference between Image 1 and Image 2.
B: The man's brows changed to a lighter color in Image 2.
C: Image 2 is smoother than Image 1.
D: Image 2 is sharper than Image 1.

MODELS ✓ ✓ ✓ ✗ ✗

SUBSTITUTION / SIZE



A: The large weight on the right tray of the scale in Image 1 is changed to a small weight in Image 2.
B: The large weight on the right tray of the scale in Image 1 is changed to a black package in Image 2.
C: The black package on the left tray of the scale in Image 1 is changed to a small weight in Image 2.
D: No difference between Image 1 and Image 2.

MODELS ✗ ✓ ✓ ✗ ✗

OCR / TEXT



A: No difference between Image 1 and Image 2.
B: The Latin text "amour" is changed to "amour" with an extra letter.
C: The Latin text "mon" is changed to "men".
D: The Latin text "mon" is changed to "man".

MODELS ✓ ✓ ✗ ✗ ✗

ILLUMINATION



A: Image 2 is brighter than Image 1.
B: Image 2 is sharper than Image 1.
C: No difference between Image 1 and Image 2.
D: Image 2 is darker than Image 1.

MODELS ✗ ✓ ✓ ✗ ✗

Figure 11: **Qualitative VDIFF-BENCH results across all ten change categories (part 2 of 2).** Layout and symbols follow Figure 11. Examples are selected to illustrate mixed model outcomes rather than category-level prevalence.