# Feature Recovery for Object Understanding After Irreversible Fire Damage

**Aditi Tiwari*** **Sofia Stoica** **Savya Khosla** **David Forsyth** **Heng Ji**
University of Illinois Urbana-Champaign
Urbana, IL 61801
{aditit5, sstoica2, savyak2, daf, hengji}@illinois.edu

## Abstract

Objects in post-fire environments often undergo irreversible physical transformations that change their geometry, material state, and visual appearance. Detecting and identifying these remnants is critical for locating hazards, reconstructing pre-incident contents, and inventorying losses. Unlike standard image corruptions, these degradations affect the physical structure of the object itself. To study this setting, we introduce **TRACE**, a transformation-aware benchmark for post-fire object understanding. TRACE contains 21.4K real-image-grounded synthetic scenes and paired object-level pristine-to-degraded progressions spanning 499 object identities across 189 categories. We define five tasks targeting localization and pre-degradation understanding: degraded-object detection, pristine-state recovery and retrieval, original material recovery, pristine description generation, and functional reasoning. Existing models degrade sharply with severity. From the least to the most severe level, RF-DETR mAP decreases by 71% relative, while InternVL3.5 retrieval R@1 falls from 93.85 to 28.11. To address this, we propose **Feature Recovery Module** (FRM), a lightweight plug-and-play module that maps degraded encoder features to pristine-aligned representations while keeping the host frozen. Trained only with paired feature supervision, FRM improves scene-level detection, CLIP/SigLIP2 feature recovery, and all four object-level VLM tasks, with larger gains under more severe degradation. On 100 held-out real post-fire scenes, FRM also improves detection by 16.1% relative. Across VLM hosts and severity levels, relative gains average 12.5% for retrieval, 20.1% for material recovery, 13.2% for description generation, and 12.4% for functional reasoning.

Project page and dataset: TRACE.

## 1 Introduction

Real-world post-disaster environments present a fundamental challenge for visual perception. Unlike classical image corruptions, post-fire degradation is physical and irreversible. Wood chars, plastic warps, metal oxidizes, and glass fractures, jointly altering geometry, material state, and surface appearance. Recovering object identity, material, and function from such remnants is critical across post-disaster workflows, including firefighters locating hazardous items, investigators reconstructing pre-incident contents, and adjusters inventorying damaged property from partial remains.

Existing benchmarks do not capture this setting. Standard object detection benchmarks [22, 14, 49, 38] contain mostly pristine object instances, while robustness suites [40, 25] perturb appearance over a fixed geometric scaffold, leaving object shape and material identity largely intact. In contrast, post-fire transformation changes structure, material, and appearance together. Across RF-DETR [36], CLIP [31], SigLIP2 [44], and five VLMs, Qwen3-VL [4], Qwen2.5-VL [5], InternVL3.5 [46], Molmo2 [12], and LLaVA-OV-1.5 [2], we observe consistent failure. From the least (L0) to the

*Corresponding author: aditit5@illinois.edu

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

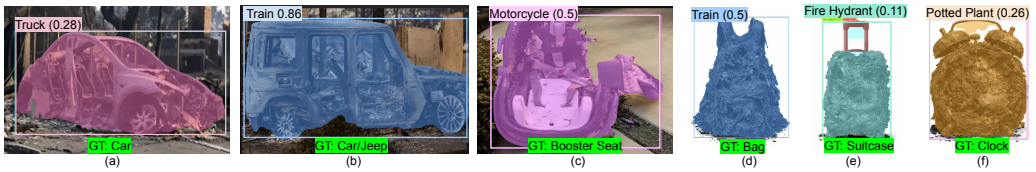



**Figure 1: RF-DETR failures under post-fire degradation.** RF-DETR consistently mislabels detected objects in real post-fire scenes (a-c) and synthetic degradations (d-f), as post-fire transformations alter object structure, material, and surface appearance in ways not seen by the model. This highlights the need for benchmarks that explicitly link pristine objects to their degraded counterparts, which are not available in current datasets.

most severe degradation (L4) level, RF-DETR mAP drops by 71%, and 92.9% of boxes localized at IoU$\geq$0.5 are misclassified (Figure 1). VLMs, despite access to world knowledge, exhibit the same degradation pattern: InternVL3.5 retrieval R@1 falls from 93.85 to 28.11, and Qwen2.5-VL falls from 76.92 to 32.31. Similar drift in frozen encoder features indicates a systematic limitation of current visual representations rather than a model-specific deficiency.

To study this gap, we introduce **TRACE**, a transformation-aware benchmark for post-fire object understanding. TRACE provides 21.4K real-image-grounded synthetic **post-fire scenes** with bounding boxes, material labels, and degradation-severity captions (Figure 2a). It also provides **object degradation trajectories** for 499 instances across 189 object types, each with a pristine reference and five progressively degraded states annotated at the part level (Figure 2b). On TRACE, we define five tasks spanning localization and pre-degradation understanding (Table 1). These are degraded-object detection, pristine-state recovery and retrieval, original material recovery, pristine description generation, and functional reasoning.

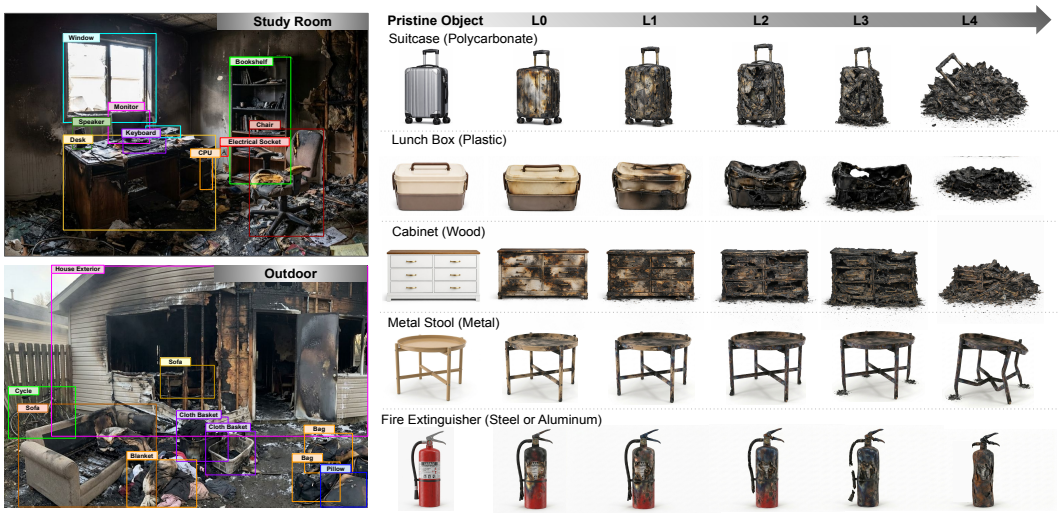
Pixel-space restoration is a natural baseline, but physical degradation makes pristine reconstruction ambiguous. Irreversible changes can destroy object parts and material cues that cannot be uniquely inferred. We fine-tune Restormer [52] on TRACE degraded-pristine pairs and use it before frozen RF-DETR and VLM retrieval, but even with paired supervision, Restormer gives only modest gains and remains below feature recovery. We instead train a lightweight residual **Feature Recovery Module (FRM)** at the host feature interface, leaving all host weights frozen. FRM maps degraded features toward pristine counterparts in the host representation space, leveraging semantics already encoded by pretrained models. Across detectors, encoders, and VLMs, FRM yields severity-scaled gains: RF-DETR detection improves by 30.8% relative, while InternVL3.5 and Qwen3-VL improve by 27% and 12% on average across the four VLM pristine-state tasks. On 100 real post-fire scenes, FRM also improves detection from 0.223 to 0.259 mAP. This gain is smaller than on synthetic scenes, but it shows that FRM transfers beyond synthetic data.

**Contributions.**

- **Benchmark.** TRACE provides 21.4K real-image-grounded synthetic post-fire scenes with object-level annotations and 499 paired pristine-to-degraded object trajectories across 189 types, supporting five evaluation tasks (Table 1).
- **Findings.** On TRACE, state-of-the-art detectors, encoders, and VLMs fail consistently under physical degradation, indicating a representational rather than model-specific limitation.
- **Method and release.** FRM is a plug-and-play residual feature-recovery module that improves all five evaluation tasks in Table 1 across diverse hosts. RF-DETR detection improves by 30.8% relative, and the four VLM pristine-state tasks improve by 12.4 % to 20.1% mean relative across five frozen VLMs. We release the dataset, generation and evaluation pipelines, and training code.

## 2 Related Work

Prior work on visual robustness studies models under image corruptions, domain shift, adverse conditions, synthetic distribution shifts, or pixel-level restoration. These settings assume the underlying object is intact and recoverable from the observation. **i. Benchmarks.** Robustness benchmarks [19, 6, 39, 3, 10, 28, 18, 40, 25] evaluate recognition under appearance shifts (noise, blur, weather, viewpoint) but preserve object structure and material identity. Natural-disaster benchmarks [17, 33, 32, 47, 30] target damage assessment from aerial or social-media imagery at the building or scene level, not at the level of individual objects undergoing irreversible transformation. TRACE is the first benchmark to study object-level physical degradation with paired pristine-to-degraded trajectories, supporting severity-stratified evaluation and feature-space recovery. **ii. Methods.** The methods most relevant to FRM fall into three families, each with a different assumption




(a) Post-fire scenes (b) Object degradation trajectories

**Figure 2: TRACE dataset overview.** TRACE contains real-image-grounded synthetic post-fire scenes and paired pristine-to-degraded object trajectories. Scene annotations include bounding boxes, material labels, and captions; object trajectories include part-level materials and progressive degradation states for tracking identity changes under physical transformation. Detailed object and scene statistics are provided in Appendix A.

**Table 1: Evaluation tasks in post-fire object understanding.** We evaluate five tasks spanning detection, retrieval, and reasoning, with pristine-state recovery split into feature-space and gallery-retrieval protocols. Full prompts are provided in Appendix C.

| Task | Goal | Output | Metric |
|---|---|---|---|
| Degraded-object detection | Localize all objects of interest in scene-level post-fire images | Bounding boxes + mapped COCO classes | COCO-mapped mAP@0.5:0.95 |
| Pristine-state recovery (feature) | Recover CLIP/SigLIP2 features toward paired pristine targets | Recovered visual embeddings | MSE, cosine similarity |
| Pristine-state retrieval (gallery) | Retrieve paired pristine reference from candidate gallery using VLM embeddings | Ranked pristine gallery | Recall@1 |
| Original material recovery | Predict pre-degradation materials from degraded object image | Material label set | Micro-F1 over normalized material classes |
| Pristine description generation | Describe object identity, material, and function before degradation | Single-sentence description | Mean of identity accuracy, material micro-F1, and function accuracy |
| Functional reasoning | Predict pre-degradation function and required restoration materials | Function tags + restoration materials | Mean of function-tag micro-F1 and restoration-material micro-F1 |

about the nature of degradation. *Domain adaptation* aligns features between source and target distributions through covariance matching, pseudo-labeling, or prototype alignment [42, 48, 9, 20, 50, 34]. These methods assume a coherent target distribution to align to; under our setting, each degradation severity induces a different distribution, and the relevant supervision is paired pristine-degraded correspondences rather than aggregate distributional alignment. *Test-time adaptation* adapts models on unlabeled test data through pseudo-supervision, feature-distribution alignment, or dynamic class statistics [37, 24, 53]. These methods optimize at deployment time without paired pristine references; FRM instead trains once on paired data and applies without per-deployment optimization. *Image restoration* recovers high-quality images from degraded image quality observations [51, 21, 52, 11, 26, 13, 35, 50], and feature-quality methods characterize degradation-induced shifts in learned embeddings [8, 1, 7]. Restoration shares FRM 's motivation but operates in pixel space, where physical degradation makes inversion ill-posed and errors can propagate to downstream recognizers. Feature-quality methods quantify degradation without correcting it, whereas FRM uses paired supervision to recover pristine-aligned features in the host representation space while keeping all host weights frozen.

## 3 TRACE

**Design Principles.** TRACE is shaped by three constraints specific to physical transformations of the object. *i. Pairing.* Each object appears in both pristine and degraded states with instance-level correspondence. This enables direct measurement of representation drift under transformation, rather than end-task accuracy under shifted distributions alone. *ii. Severity progression.* Each object is rendered along a discrete five-level severity ladder (L0-L4), where L0 denotes mild surface damage and L4 denotes severe deformation or near-total destruction. This enables severity-stratified evaluation and exposes whether a method interpolates within a trained severity range or extrapolates beyond it

(Sec. 5). *iii. Real-world grounding.* Synthetic generation is conditioned on real post-fire imagery [43], grounding visual statistics, material behavior, and scene composition in observed environments rather than free-form prompts alone. Each sample includes object class, material composition, degradation-state descriptions, failure modes, embeddings, and bounding boxes; see Appendix A. To verify annotation quality, two authors independently assigned accept/reject labels on a stratified 300-object subset, yielding 93.4% agreement and Cohen's $\kappa = 0.836$. These values measure whether the two authors agreed to accept or reject each object, not how closely their bounding boxes overlap.

**Object Inventory.** The object inventory targets two post-fire response needs. The first covers hazardous items such as aerosol cans, gas cylinders, and chemical containers that responders must identify and secure. The second covers personal valuables and everyday objects, including electronics, furniture, bags, and medical equipment, that matter for insurance assessment and inventory reconstruction. Overall, TRACE spans 189 object types and 11 normalized material classes, with plastic (1,315 mentions), metal (948), fabric/textile (390), and wood (287) most represented.

**Scene-Level Subset.** Gemini 3 Pro Image [16] generates each scene from a structured prompt and two to four real post-fire references, sampled from approximately 700 images drawn primarily from Fire360 [43] and research-permissive web sources. References span residential, commercial, institutional, and outdoor environments. Scene prompts cover four environment groups and 30 scene types. A Gemini 3 Flash [15] vision-language judge scores each candidate for realism, material fidelity, structural plausibility, and consistency with the references. Candidates below 0.7 are rejected, yielding 21,400 accepted scenes from approximately 29,000 generations. GroundingDINO initializes object boxes and labels, which the authors inspect and correct when needed. Ambiguous objects are excluded from evaluation. Full prompts, judge criteria, filtering details, and licensing notes are in Appendix B.

**Object-Level Subset.** Each object trajectory contains one pristine render and five progressively degraded states (L0-L4) generated with material-aware prompts that capture wood charring, plastic melting, metal oxidation, and glass fracture. Trajectories span 499 distinct object identities across 189 object types, yielding 2,994 object-level images. Some object types include multiple subtypes or variants, which are treated as distinct identities when constructing train, validation, and test splits. The object-level subset is split 80/10/10 by object identity, ensuring that all renders of an identity remain in the same split.

**Real Post-Fire Evaluation.** To evaluate beyond synthetic data, we manually crop 500 objects from the pool of approximately 700 real post-fire reference images described above. Each crop is annotated with identity and severity labels following the same protocol as the synthetic subset. These crops form a separate real-world evaluation set. We also evaluate 100 real post-fire scenes containing 657 verified COCO-mapped boxes across 24 categories. These scenes were not used to train FRM or condition TRACE generation.

**Evaluation Tasks.** TRACE defines five tasks spanning detection, retrieval, and reasoning in post-fire settings (Table 1). **(1) Degraded-object detection.** First responders must rapidly locate hazardous or structurally compromised objects in post-fire scenes. Given a scene-level image, the task is to detect and localize all objects of interest, evaluated with class-aware COCO-mapped mAP@0.5:0.95 at each degradation level. Only boxes with unambiguous COCO mappings [22] are included in mAP. The mapping covers 55 of the 80 COCO categories. **(2) Pristine-state recovery and retrieval.** Matching a damaged remnant to its undamaged counterpart is the foundation for downstream recovery. This retrieval setting is similar to TOR [43], but TOR defines transformed-object retrieval in video space, while TRACE evaluates paired pristine-to-degraded recovery at the object level. We evaluate this in two protocols. In feature space, CLIP and SigLIP2 features are recovered toward paired pristine targets and measured by MSE and cosine similarity. In gallery retrieval, VLM visual embeddings retrieve the paired pristine reference from a candidate gallery, measured by Recall@1. **(3) Material recovery.** Knowing original materials determines whether an object is restorable or requires replacement and helps environmental teams triage debris for safe disposal. Given a degraded object image, the model predicts the object's pre-degradation materials. Outputs are mapped to our normalized material vocabulary and evaluated using micro-F1 over the 11 material classes. **(4) Pristine description generation.** Reconstruction planning and insurance documentation require a concise account of what the object was, its material composition, and its intended use. The model generates a one-sentence description of the object's pre-degradation identity, materials, and function. We parse the description into identity, material, and function fields. Identity and function are scored by categorical accuracy

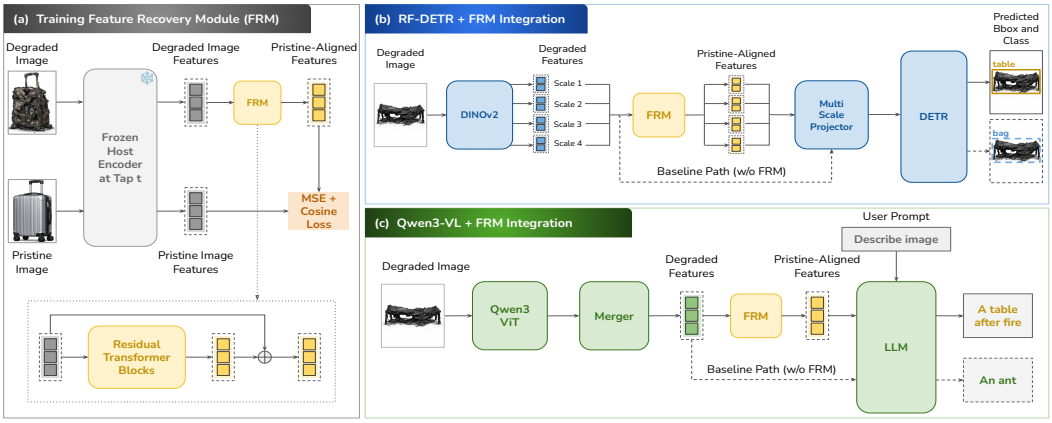



**Figure 3: Feature Recovery Module (FRM).** (a) FRM is a residual Transformer [45] over visual tokens, trained to recover pristine-aligned features from degraded ones with a zero-gated identity initialization; tap $t$ is the frozen host feature interface where FRM is inserted. (b) In RF-DETR, DINOv2 [29] feature maps from four tapped layers are flattened, corrected by a *single shared* FRM head, and reshaped before the unchanged detector. (c) In Qwen3-VL, FRM corrects post-merger visual tokens before they are consumed by the frozen language model. Dashed paths denote the no-FRM baseline.

after synonym normalization, with GPT5 [41] as LLM-judge fallback only for unresolved paraphrases. Materials are scored with micro-F1 over the normalized material vocabulary. The final description score is the unweighted mean of identity accuracy, material micro-F1, and function accuracy, reported on a 0-100 scale. **(5) Functional reasoning.** Fire investigators and restoration teams need structured judgments beyond description, such as whether an object could have served as an ignition source, fuel, or hazard, and what materials are needed for repair. The model predicts pre-degradation function tags and restoration materials, evaluated as the unweighted mean of micro-F1 over function tags and micro-F1 over restoration-material labels. Appendix C specifies the prompts, output normalization, label vocabularies, and exact score computations for all generation-based tasks.

## 4 Feature Recovery Module (FRM)

TRACE provides paired observations of each object instance before and after physical transformation (Figure 3). This pairing enables direct supervision of feature-space recovery: for any frozen visual encoder, the degraded image serves as input and the paired pristine image defines the target representation. We operate in feature space rather than pixel space because severe physical transformations alter geometry, material state, and surface structure in ways that make pixel-level inversion ambiguous. Since FRM modifies only the visual feature representations, the encoder, projector or merger, detector head, and language model remain frozen.

**Feature-Recovery Problem.** Let $x_i^p$ denote the pristine image of object instance $i$, and let $x_i^\ell$ denote a paired image of the same instance after transformation at degradation level $\ell \in \{0, \dots, 4\}$. For a frozen encoder $E$ and a tap point $t$ (an intermediate interface where features can be read and written back to), let $E_t(x) \in \mathbb{R}^{N_t \times D_t}$ denote the features at $t$. FRM learns a shape-preserving function $h_\theta : \mathbb{R}^{N_t \times D_t} \to \mathbb{R}^{N_t \times D_t}$ that maps degraded features toward their pristine counterparts:

$$\tilde{z}_i^{\ell,t} \;=\; h_\theta\big(E_t(x_i^\ell)\big) \;\approx\; E_t(x_i^p).$$

The corrected features $\tilde{z}_i^{\ell,t}$ are forwarded through the unchanged downstream pipeline. As detailed below, $h_\theta$ is implemented as a residual correction $h_\theta(z) = z + \Delta_\theta(z)$ with $\Delta_\theta = 0$ at initialization.

**Architecture.** FRM is a residual Transformer ($M$ blocks, default $M$=4 for VLMs, $M$=1 for RF-DETR) with zero-gated initialization. For any tapped feature tensor $z \in \mathbb{R}^{B \times N \times D}$, where $B$ is batch size, $N$ is the number of visual tokens, and $D$ is feature dimension, let $u^0 = z$. Block $m \in \{1, \dots, M\}$ applies

$$\hat{u}^m = u^{m-1} + \alpha_{\text{attn}}^m \odot \text{MSA}(\text{LN}(u^{m-1})), \quad u^m = \hat{u}^m + \alpha_{\text{mlp}}^m \odot \text{MLP}(\text{LN}(\hat{u}^m)),$$

with $H$ attention heads, an MLP of hidden width $rD$ with expansion ratio $r$, and learned channel-wise residual scales $\alpha_{\text{attn}}^m, \alpha_{\text{mlp}}^m \in \mathbb{R}^D$ initialized to zero. We define the residual correction $\Delta_\theta(z) := u^M - z$, so that $h_\theta(z) = z + \Delta_\theta(z)$. Because the residual scales gate all MSA and MLP outputs to

zero, $\Delta_\theta(z) = 0$ at initialization regardless of inner weights, and FRM acts as the identity on the host interface, which stabilizes early training. Unlike encoder fine-tuning, FRM never touches encoder weights, preserving the downstream model's training-time feature distribution.

**Tap-Point Selection.** The tap point $t$ involves a tradeoff between spatial detail and downstream alignment. Earlier taps in the visual encoder preserve higher spatial resolution, making local degradation patterns easier to identify and correct at the patch level. Later taps, and in particular post-projector or post-merger interfaces, produce features in the same space that the downstream head was trained on, so corrections at these locations translate more directly into task performance. We select the tap point for each host based on this tradeoff. Later taps improve retrieval R@1 from 72.90 to 85.28 on Qwen3-VL and from 63.84 to 74.92 on InternVL3.5 (Table 10).

**Training Objective.** For a paired sample $(x_i^\ell, x_i^p)$, the encoder extracts $z_i^{\ell,t} = E_t(x_i^\ell)$ and $z_i^{p,t} = E_t(x_i^p)$ at the same tap. The pristine branch is computed without gradient tracking, and FRM is applied only to the degraded branch, $\tilde{z}_i^{\ell,t} = h_\theta(z_i^{\ell,t})$. Dropping indices, the recovery loss combines element-wise MSE with token-level cosine alignment:

$$\mathcal{L}_{FRM}(\tilde{z}, z^p) = \tfrac{1}{BND}\|\tilde{z} - z^p\|_2^2 \;+\; \lambda_{\cos}\Big(1 - \tfrac{1}{BN}\sum_{b,n}\cos(\tilde{z}_{b,n}, z^p_{b,n})\Big),$$

with $\lambda_{\cos} = 0.1$ selected on the validation split (Sec. 5.3). The MSE term constrains token magnitudes in the downstream feature space, and the cosine term penalizes directional drift. Spatial augmentations are restricted to transformations that preserve patch-grid correspondence (horizontal flips, 90° rotations, and stride-aligned resizes). The same transformation is sampled once and applied to both $x_i^\ell$ and $x_i^p$ before encoding. Arbitrary-offset crops are excluded because they break paired token correspondence.

**Training Cost and Data Extensibility.** Because only FRM is trained and the encoder is frozen, encoder features can be precomputed once per dataset and reused across epochs. On a single NVIDIA A100, training an FRM head to convergence on approximately **2,000 paired examples** takes approximately **one hour** with cached encoder features and approximately **three hours** when encoder forward passes are recomputed on the fly. Adding new transformation categories or object instances requires only encoding the additional images and continuing FRM training.

**Instantiation in RF-DETR.** RF-DETR uses a DINOv2 encoder, a multi-scale projector, a DETR transformer, and class/box heads. We insert FRM after the encoder and before the projector, and all other components remain frozen. For RF-DETR-Medium with $576 \times 576$ inputs and patch size 16, we extract features from four evenly spaced DINOv2-Small encoder layers $s \in \{3, 6, 9, 12\}$, each producing a $36 \times 36$ grid with $z^s \in \mathbb{R}^{B\times 384\times 36\times 36}$. Each map is flattened to $u^s = \phi(z^s) \in \mathbb{R}^{B\times 1296\times 384}$ and corrected by a single shared FRM head $h_\theta$ applied to all four scales:

$$\tilde{u}^s = h_\theta(u^s), \qquad \tilde{z}^s = \phi^{-1}(\tilde{u}^s), \qquad s \in \{3, 6, 9, 12\}.$$

Sharing $h_\theta$ across scales is possible because all four layers produce the same hidden dimension ($D$=384), and keeps the parameter count independent of the number of tapped layers. The recovery loss $\mathcal{L}_{FRM}$ is applied to each scale independently and averaged over the four tapped layers.

**Instantiation in Qwen3-VL.** Qwen3-VL consists of a frozen vision transformer, a frozen merger that compresses each $2 \times 2$ spatial neighborhood into one token and projects visual features into the LLM-facing space, and a frozen language model. With $448 \times 448$ inputs and patch size 16, the post-merger visual features have shape $z_{\text{post}} \in \mathbb{R}^{B\times 196\times 2560}$. We insert FRM at this post-merger interface, so the corrected tokens $\tilde{z}_{\text{post}}$ are passed directly to the frozen language model. This tap aligns FRM with the representation actually consumed by the LLM. Instantiation details for CLIP, SigLIP2, and the remaining VLM hosts follow the same recipe and are provided in Appendix D.

**Plug-and-Play Scaling and Overhead.** Only the token dimension $D$, capacity $(M, r)$, and tap point are configured per host, where $D$ is determined by the selected tap point. The FRM head has approximately $P_{FRM} \approx M(4 + 2r)D^2$ parameters ($4D^2$ from multi-head self-attention, $2rD^2$ from the MLP). Capacity can be selected from an overhead budget $\varepsilon$, the maximum ratio of FRM parameters to host parameters, via $M(4 + 2r) \le \varepsilon P_{\text{host}}/D^2$. We instantiate FRM with a 1.77M-parameter head for RF-DETR-Medium ($P_{\text{host}}$=33.7M, $D$=384, $M$=1, $r$=4, 5.25% overhead) and a 314.6M-parameter head for Qwen3-VL-4B ($P_{\text{host}}$=4.0B, $D$=2560, $M$=4, $r$=4, 7.86% overhead).

**Inference and Deployment.** At inference, FRM is inserted at its trained tap. Inference uses only the degraded image. The frozen encoder produces degraded features, FRM recovers pristine-aligned

**Table 2: Baselines and adaptation paradigms.** Detection reports COCO-mapped mAP, classification reports mapped VLM top-1 accuracy, and retrieval reports degraded-to-pristine R@1. Auxiliary classification tests whether pixel restoration preserves object identity and is not a TRACE task. FT = fine-tune, DA = domain adaptation, FR = feature recovery, PR = pixel restoration.

| Paradigm | Method | Score↑ |
|---|---|---|
| *Degraded-object detection, mAP@0.5:0.95* | | |
| No adapt. | Base RF-DETR[36] | 0.338 |
| PR | Restormer[52] + RF-DETR | 0.375 |
| PR | WildIR[27] + RF-DETR | 0.266 |
| DA | SFA [48] | 0.161 |
| DA | DATR [9] | 0.293 |
| FR | Base + FRM | 0.442 |
| Full FT | RF-DETR FT | 0.572 |
| FT + FR | RF-DETR FT + FRM | **0.647** |
| *Auxiliary pristine-object classification, top-1 (%)* | | |
| PR | WildIR[27] + Qwen3-VL | 36.44 |
| PR | WildIR + InternVL3.5 | 22.57 |
| *Paired pristine-gallery retrieval, R@1* | | |
| PR | Restormer + Qwen3-VL | 74.55 |
| FR | Qwen3-VL + FRM | **85.28** |
| PR | Restormer + LLaVA-OV-1.5 | 80.97 |
| FR | LLaVA-OV-1.5 + FRM | **86.09** |

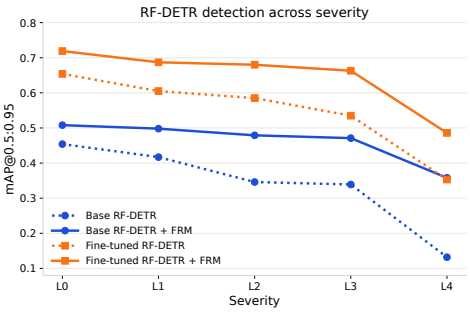



**Figure 4: RF-DETR detection across severity.** FRM improves both frozen and fine-tuned RF-DETR.

**Table 3: Frozen encoder feature recovery.** Degraded-to-pristine recovery on object-level test set (Base → + FRM).

| Encoder | MSE↓ | Cos.↑ | R@1↑ | R@5↑ |
|---|---|---|---|---|
| CLIP ViT-L/14 [31] | 0.6719 → **0.4686** | 0.6537 → **0.7492** | - | - |
| SigLIP2 [44] | 2.4172 → **1.3693** | 0.6045 → **0.7421** | 24.2 → **39.0** | 77.3 → **96.2** |

features, and the corrected features continue through the unchanged downstream pipeline. FRM weights are tied to a specific (encoder, tap, feature space) configuration, so each new host requires a separately trained module.

# 5 Experiments

**Experimental setup.** We evaluate on the scene-level and object-level TRACE subsets from Sec. 3. Degraded-object detection uses the synthetic scene-level test set, with a separate evaluation on 100 real post-fire scenes (Sec. 5.1). The four object-level tasks use held-out degraded trajectories over L0-L4, with a separate 500-crop real post-fire evaluation in Table 5. Tasks and metrics follow Table 1, with prompts, normalization rules, and exact score computations specified in Appendix C. For detection, we compare FRM against RF-DETR fine-tuning, RF-DETR domain adaptation (SFA [48], DATR [9]), and pixel-restoration preprocessors before frozen RF-DETR, including WildIR [27] and Restormer [52] fine-tuned on TRACE. We also evaluate pixel restoration before frozen VLMs: Restormer for degraded-to-pristine gallery retrieval and WildIR for open-ended pristine-state object recovery. All trainable baselines use the TRACE training split; hosts remain frozen unless marked as fine-tuned. For feature-space alignment, we evaluate frozen CLIP ViT-L/14 and SigLIP2 encoders. For VLM tasks, the baseline is the frozen host on degraded inputs, and the pristine ceiling is the same host on pristine inputs. We evaluate five frozen VLM hosts, Qwen3-VL-4B, Qwen2.5-VL-3B, InternVL3.5-4B, Molmo2-4B, and LLaVA-OneVision-1.5-4B, using deterministic decoding and identical prompts, parsers, normalization rules, and scoring scripts with and without FRM. All experiments are run on NVIDIA A100 GPUs.

## 5.1 Degraded-Object Detection and Feature-Space Recovery

**Detection on TRACE scenes.** Table 2 compares adaptation paradigms on the scene-level test set, with severity trends shown in Figure 4. FRM improves the frozen RF-DETR by 30.8% and adds a further 13.1% on top of the fine-tuned host while preserving pristine performance within 0.001 mAP. The additive gain indicates that fine-tuning and FRM address different failure modes. Fine-tuning broadens the encoder's input distribution to include degraded images, while FRM corrects residual feature misalignment that persists after fine-tuning. Statistics-only domain adaptation methods, including SFA and DATR, underperform the frozen baseline, confirming that marginal feature alignment is insufficient when degradation produces structured, per-instance drift requiring paired correction. Pixel restoration is insufficient for this setting. Restormer gives only a small detection gain (0.338→0.375 mAP), while WildIR reduces RF-DETR performance to 0.266 mAP. In an auxiliary classification test, WildIR also yields low pristine-object top-1 accuracy with frozen VLMs (36.4% on Qwen3-VL and 22.5% on InternVL3.5), suggesting that visually restoring local image quality does not reliably recover the pre-degradation identity cues needed for recognition.

**Transfer to real post-fire scenes.** We evaluate zero-shot detection on 100 real post-fire scenes. FRM improves frozen RF-DETR from 0.223 to 0.259 mAP, a 16.1% relative gain. A 95% bootstrap interval

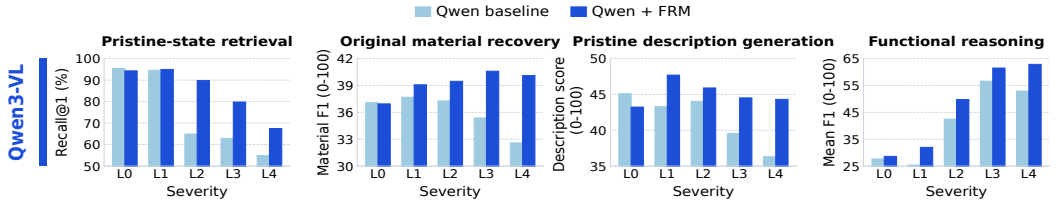



**Figure 5: Per-level Qwen3-VL pristine-state recovery.** Baseline is shown in light color and FRM in dark color across L0-L4. Columns report pristine-state retrieval R@1, original material recovery F1, pristine-description score, and functional-reasoning mean F1. FRM gains are strongest at higher severity.

**Table 4: VLM task performance across five frozen hosts.** Base is mean performance of frozen VLM over L0-L4; $\Delta$ is the gain from FRM. Metrics are reported on a 0-100 scale, with per-level results in Appendix E.

| Model | Base | +FRM | $\Delta$ |
|---|---|---|---|
| ***Task:** Pristine-state retrieval (R@1)* | | | |
| Qwen3-VL | 74.55 | 85.28 | +10.73 |
| Qwen2.5-VL | 60.20 | 72.03 | +11.83 |
| InternVL3.5 | 59.09 | 74.92 | +15.83 |
| Molmo2 | 85.06 | **90.00** | +4.94 |
| LLaVA-OV-1.5 | 83.97 | 86.09 | +2.12 |
| ***Task:** Original material recovery (micro-F1)* | | | |
| Qwen3-VL | 36.00 | **39.24** | +3.24 |
| Qwen2.5-VL | 30.50 | 36.78 | +6.28 |
| InternVL3.5 | 26.10 | 35.91 | +9.81 |
| Molmo2 | 23.56 | 27.52 | +3.96 |
| LLaVA-OV-1.5 | 22.31 | 26.89 | +4.58 |
| ***Task:** Pristine description generation (description score)* | | | |
| Qwen3-VL | 41.67 | **45.15** | +3.48 |
| Qwen2.5-VL | 40.50 | 44.00 | +3.50 |
| InternVL3.5 | 27.01 | 35.42 | +8.41 |
| Molmo2 | 33.80 | 38.13 | +4.33 |
| LLaVA-OV-1.5 | 35.48 | 39.34 | +3.86 |
| ***Task:** Functional reasoning (mean micro-F1)* | | | |
| Qwen3-VL | 41.02 | 46.96 | +5.94 |
| Qwen2.5-VL | 38.11 | 43.62 | +5.51 |
| InternVL3.5 | 36.13 | 40.48 | +4.35 |
| Molmo2 | 42.64 | **47.95** | +5.31 |
| LLaVA-OV-1.5 | 39.76 | 43.07 | +3.31 |

**Table 5: Ablations.** We ablate the recovery loss, cosine weight, and real-crop transfer. Metrics are cosine distance to pristine, identity accuracy, and retrieval R@1. Additional FRM ablations are provided in Appendix G.

| Configuration | Cos.↓ | ID↑ | R@1↑ |
|---|---|---|---|
| **Ablation 1:** *Loss formulation, Qwen3-VL post-merger* | | | |
| No FRM | 0.260 | 32.69 | 74.55 |
| LN-MSE | 0.030 | 36.47 | 57.83 |
| MSE only | 0.027 | 36.81 | 83.17 |
| Huber + $\mathcal{L}_{\cos}$ | 0.026 | 37.08 | 84.63 |
| MSE + $0.1\mathcal{L}_{\cos}$ | **0.025** | **37.31** | **85.28** |
| **Ablation 2:** *Cosine weight $\lambda$, MSE+$\lambda\mathcal{L}_{\cos}$* | | | |
| No FRM | 0.260 | 32.69 | 74.55 |
| $\lambda = 0.01$ | 0.026 | 36.94 | 84.12 |
| $\lambda = 0.10$ | 0.025 | **37.31** | **85.28** |
| $\lambda = 0.50$ | **0.024** | 37.14 | 84.37 |
| $\lambda = 1.00$ | 0.027 | 36.54 | 82.76 |
| **Real crops** | **Base** | **+FRM** | **$\Delta$** |
| **Ablation 3:** *Real-crop transfer, Qwen3-VL* | | | |
| *L2-L4 average:* | | | |
| Retrieval R@1 | 49.16 | 53.03 | +3.87 |
| Material F1 | 25.31 | 29.98 | +4.67 |
| Description gen. | 27.47 | 32.53 | +5.06 |
| Functional reasoning | 29.31 | 33.80 | +4.49 |
| *L4 only:* | | | |
| Retrieval R@1 | 43.28 | 46.03 | +2.75 |
| Material F1 | 22.47 | 26.88 | +4.41 |
| Description gen. | 24.93 | 27.84 | +2.91 |
| Functional reasoning | 26.84 | 30.63 | +3.79 |

for the improvement excludes zero (margin 0.017). The gain is smaller than the 30.8% improvement on synthetic TRACE scenes, showing that a synthetic-to-real gap remains.

**Feature-space recovery generalizes across encoder families.** To isolate feature-space recovery from detector-specific confounds, we apply the same FRM recipe to frozen CLIP and SigLIP2 features (Table 3). Despite different pretraining objectives, contrastive for CLIP and sigmoid contrastive for SigLIP2, FRM consistently moves degraded features toward paired pristine targets. It reduces MSE by 30.3% on CLIP and 43.4% on SigLIP2, with corresponding cosine similarity gains of +0.095 and +0.138. The effect carries through to pristine-state retrieval, where FRM improves SigLIP2 degraded-to-pristine Top-1 by $+14.8$ points and Top-5 by $+18.9$ points, with Top-5 nearly reaching the pristine-query ceiling of $100\%$. These gains support the view that fire-induced degradation produces structured displacement on the feature manifold across encoder families.

### 5.2 Pristine-State Recovery in VLMs

**Retrieval gains concentrate at higher severity.** FRM improves degraded-to-pristine Recall@1 across all evaluated VLM hosts (Table 4, Figure 5; full per-level results in Table 9). Averaged across hosts, retrieval gains are nearly flat at mild degradation ($+0.73$ points over L0-L1) but rise sharply at higher severity ($+14.66$ points over L2-L4). This pattern argues against a fixed correction bias and is consistent with FRM correcting feature displacement that grows with physical degradation. The gains are larger for weaker retrieval baselines and smaller for stronger ones, suggesting that FRM recovers pristine-state signal that weaker encoders lose while stronger encoders partially retain it without correction.

**Lower baseline performance leaves more room for recovery.** The largest absolute gains on original material recovery and pristine description generation occur on InternVL3.5 (Table 4), which has the weakest degraded-input baseline on these tasks (material $+9.81$ points, description generation

+8.41 points). FRM's benefit is better explained by degradation sensitivity than by model capacity. It helps most when the frozen host loses more pristine-state information under physical transformation. Operationally, this suggests that FRM is most valuable for host-task pairs with large degradation-induced performance drops.

**Pristine alignment transfers to functional reasoning.** FRM improves the functional reasoning combined score on every host (Table 4), with a mean aggregate gain of $+4.9$ points across VLMs and larger gains at severe degradation levels (average $+7.6$ at L3-L4, Figure 5). FRM is trained only on paired features with no per-image task labels, yet recovered features support reasoning about pre-degradation function and restoration needs. This suggests that the frozen language models already encode object-function knowledge, and FRM restores the visual evidence needed to invoke it. Without FRM, models often fall back to coarse functional guesses such as "seating," or "container," which can partially match broad tags without recovering the specific object. The high-severity gains suggest FRM recovers object-specific function rather than improving generic guessing.

### 5.3 Ablations

We ablate the recovery loss, compare recovery heads and paired-data scaling, and test real-crop transfer under moderate and severe degradation (Table 5, with additional ablations in Appendix G and Table 10). **i. Loss formulation.** All loss ablations use the Qwen3-VL post-merger tap. LN-MSE reduces cosine distance but drops retrieval below the no-FRM baseline, showing that cosine distance alone is not sufficient and that token magnitude matters for pooled retrieval. MSE alone restores retrieval, while MSE+$\mathcal{L}_{\cos}$ gives the best tradeoff across cosine distance, identity accuracy, and R@1. Huber+$\mathcal{L}_{\cos}$ is close but does not exceed it, suggesting that the quadratic penalty on large deviations helps severe degradation tokens. **ii. Cosine weight.** R@1 peaks at $\lambda = 0.1$ and changes smoothly around it. Although $\lambda = 0.5$ gives the lowest cosine distance, $\lambda = 0.1$ gives the best identity accuracy and retrieval. Across $\lambda \in \{0.01, 0.1, 0.5\}$, R@1 varies by only 1.16 points, so the joint loss is robust to the exact weight in this range. We use MSE+$0.1\mathcal{L}_{\cos}$ throughout. **iii. Real-crop transfer.** We evaluate Qwen3-VL on 500 real post-fire crops from classes present in TRACE at L2-L4, since real post-fire imagery is concentrated in moderate to severe damage and contains few reliable L0-L1 analogs. FRM improves every real-crop task on average and remains positive at L4, where near-total destruction leaves less recoverable evidence. **iv. Recovery heads and scaling.** At the same Qwen3-VL tap and training protocol, low-rank and MLP heads reach 80.63 and 82.52 retrieval R@1, compared with 74.55 without recovery and 85.28 with FRM. With 500 and 995 training pairs, FRM reaches 82.40 and 84.10 R@1, compared with 85.28 using 1,990 pairs. Each training fraction uses one identity-level subsample. Transferring Qwen3-VL FRM weights directly to InternVL3.5 yields 60.8 R@1, compared with 59.1 without FRM and 74.9 with an InternVL3.5-trained FRM. These results show that paired recovery helps with simpler heads and limited data, while learned weights remain host-specific.

## 6 Conclusion & Discussion

**Conclusion.** We introduced TRACE, a transformation-aware benchmark for post-fire object understanding, comprising 21.4K real-image-grounded synthetic scenes and object-level pristine-to-degraded trajectories spanning 499 object identities across 189 categories. We also introduced Feature Recovery Module (FRM), a plug-and-play residual module that maps degraded features toward their pristine counterparts in a frozen feature space, enabling downstream models to reason about damaged objects in terms of their pre-degradation state. Across eight frozen hosts spanning RF-DETR, CLIP, SigLIP2, and five VLMs, FRM improves every task, with larger gains at higher degradation severity. Detection improves by 30.8% relative, and mean relative improvements across the four VLM tasks range from 12.4% to 20.1%. Unlike prior robustness work that targets image-quality degradation or distributional shift, TRACE and FRM address irreversible physical transformations where the object itself has changed material state, lost parts, or retained only partial structural evidence. This positions the framework as a step toward real-world post-fire applications, from identifying hazardous objects during emergency response to inventorying destroyed property for insurance claims and reconstructing pre-incident scene contents for investigation.

**Limitations and Future Directions.** The recovery direction is task-dependent: FRM recovers pristine-aligned features, which improves queries about original identity, materials, function, and restoration, but removes exactly the damage signal needed for queries about current state (e.g., "describe the fire damage"). FRM should be disabled for such queries, and a query-aware gating

mechanism would allow a single host to serve both pristine-state and damage-state queries. The module is also tied to a specific (encoder, tap, feature space) configuration, so each new host requires a separately trained module. Learning a shared recovery space across related hosts remains an open direction. Adding a new object identity requires paired supervision from a pristine visual anchor and at least three degraded observations across severity levels; relaxing this requirement is an open direction. Gains on real object crops and scenes remain positive but smaller than synthetic gains. This gap likely reflects incomplete coverage of real-world clutter, viewpoints, lighting, and material failure in TRACE, as well as possible synthetic-domain biases learned by FRM. Mixed real and synthetic training is a natural next step. Finally, identity recovery is only one of the cues humans use under severe degradation: room layout, neighboring objects, and surviving structural parts all carry information that FRM currently ignores. Incorporating spatial context and part-level understanding is a natural next step, as is extending the paired-supervision recipe to other irreversible transformations such as flood damage, structural collapse, and weathering, wherever pre-event visual references exist.

**Acknowledgments.** This research is based upon work supported by the U.S. Defense Advanced Research Projects Agency (DARPA) under the ECOLE Program (No. HR00112390060) and the National Science Foundation (NSF) under Grant No. 2106825. The views and conclusions expressed herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation herein.

**Impact Statement.** This work targets visual recognition under irreversible physical degradation, with applications in emergency response, forensic investigation, and insurance assessment. Reliable detection and identification of objects in post-fire environments could support firefighters in locating hazardous items in collapsed structures, assist investigators in reconstructing pre-incident scene contents, and reduce the manual effort required to inventory damaged property after disasters. Releasing TRACE and the FRM training pipeline lowers the barrier for further research on degraded-object understanding, including extensions to other irreversible transformations such as flood damage and weathering. The deployment risks are largely shared with general-purpose object recognition. Pristine-state predictions are inferences from incomplete physical evidence and can be wrong, particularly at high severity; using them as evidence in legal or insurance contexts requires uncertainty quantification and human oversight that this work does not provide. The benchmark consists of synthetic scenes grounded in real post-fire imagery; while we filter for realism, distributional gaps to specific deployment environments remain, and downstream users should validate on in-domain data before operational use. We are not aware of dual-use concerns specific to this work.

## References

[1] Lorenzo Agnolucci, Leonardo Galteri, Marco Bertini, and Alberto Del Bimbo. Arniqa: Learning distortion manifold for image quality assessment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 189–198, 2024.

[2] Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

[3] Anastasios Arsenos, Vasileios Karampinis, Evangelos Petrongonas, Christos Skliros, Dimitrios Kollias, Stefanos Kollias, and Athanasios Voulodimos. Common corruptions for evaluating and enhancing robustness in air-to-air visual object detection. *IEEE Robotics and Automation Letters*, 9(7):6688–6695, 2024. doi: 10.1109/LRA.2024.3408485.

[4] 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.

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

[6] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/97af07a14cacba681feacf3012730892-Paper.pdf.

[7] Stefan Becker, Simon Weiss, Wolfgang Hübner, and Michael Arens. Self-aware object detection via degradation manifolds. *arXiv preprint arXiv:2602.18394*, 2026.

[8] Simone Bianco, Luigi Celona, and Paolo Napoletano. Disentangling image distortions in deep feature space. *Pattern Recognition Letters*, 148:128–135, 2021. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2021.05.008. URL https://www.sciencedirect.com/science/article/pii/S0167865521001859.

[9] Liang Chen, Jianhong Han, and Yupei Wang. Datr: Unsupervised domain adaptive detection transformer with dataset-level adaptation and prototypical alignment. *IEEE Transactions on Image Processing*, 34:982–994, 2025.

[10] Wenyue Chong. Towards robust semantic correspondence: A benchmark and insights. *arXiv preprint arXiv:2508.00272*, 2025.

[11] Xiaojie Chu, Liangyu Chen, and Wenqing Yu. Nafssr: Stereo image super-resolution using nafnet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1239–1248, June 2022.

[12] Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, et al. Molmo2: Open weights and data for vision-language models with video understanding and grounding. *arXiv preprint arXiv:2601.10611*, 2026.

[13] Haoyou Deng, Zhiqiang Li, Feng Zhang, Qingbo Lu, Zisheng Cao, Yuanjie Shao, Shuhang Gu, Changxin Gao, and Nong Sang. Learning unpaired image dehazing with physics-based rehazy generation. *arXiv preprint arXiv:2506.12824*, 2025.

[14] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. doi: 10.1007/s11263-009-0275-4. URL https://doi.org/10.1007/s11263-009-0275-4.

[15] Google. Gemini 3 flash preview. https://ai.google.dev/gemini-api/docs/models/gemini-3-flash-preview, 2025. Accessed: 2026-05-06.

[16] Google. Gemini 3 pro image preview. https://ai.google.dev/gemini-api/docs/models/gemini-3-pro-image-preview, 2025. Accessed: 2026-05-06.

[17] Ritwik Gupta, Richard Hosfelt, Sandra Sajeev, Nirav Patel, Bryce Goodman, Jigar Doshi, Eric Heim, Howie Choset, and Matthew Gaston. xbd: A dataset for assessing building damage from satellite imagery. *arXiv preprint arXiv:1911.09296*, 2019.

[18] Xiangzhao Hao, Zefeng Zhang, Zhenyu Zhang, Linhao Yu, Yao Chen, Yiqian Zhang, Haiyun Guo, Shuohuan Wang, and Yu Sun. Clear: Unlocking generative potential for degraded image understanding in unified multimodal models, 2026. URL https://arxiv.org/abs/2604.04780.

[19] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[20] Marc-Antoine Lavoie, Anas Mahmoud, and Steven L Waslander. Large self-supervised models bridge the gap in domain adaptive object detection. *arXiv preprint arXiv:2503.23220*, 2025.

[21] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. *arXiv preprint arXiv:2108.10257*, 2021.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[23] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[24] Yabo Liu, Jinghua Wang, Chao Huang, Yiling Wu, Yong Xu, and Xiaochun Cao. Mlfa: Toward realistic test time adaptive object detection by multi-level feature alignment. *IEEE Transactions on Image Processing*, 33:5837–5848, 2024. doi: 10.1109/TIP.2024.3473532.

[25] Yuen Peng Loh and Chee Seng Chan. Getting to know low-light images with the exclusively dark dataset. *Computer Vision and Image Understanding*, 178:30–42, 2019. doi: https://doi.org/10.1016/j.cviu.2018.10.010.

[26] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Controlling vision-language models for universal image restoration. *arXiv preprint arXiv:2310.01018*, 2023.

[27] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Photo-realistic image restoration in the wild with controlled vision-language models. *arXiv preprint arXiv:2404.09732*, 2024.

[28] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019.

[29] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[30] Adiba Proma. Nadbenchmarks - a compilation of benchmark datasets for machine learning tasks related to natural disasters. In *AAAI 2022 Fall Symposium: The Role of AI in Responding to Climate Challenges*, 2022. URL https://www.climatechange.ai/papers/aaaifss2022/9.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[32] Maryam Rahnemoonfar, Tashnim Chowdhury, Argho Sarkar, Debvrat Varshney, Masoud Yari, and Robin Roberson Murphy. Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. *IEEE Access*, 9:89644–89654, 2021.

[33] Maryam Rahnemoonfar, Tashnim Chowdhury, and Robin Murphy. Rescuenet: a high resolution uav semantic segmentation dataset for natural disaster damage assessment. *Scientific data*, 10 (1):913, 2023.

[34] Rindra Ramamonjison, Amin Banitalebi-Dehkordi, Xinyu Kang, Xiaolong Bai, and Yong Zhang. Simrod: A simple adaptation method for robust object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3570–3579, 2021.

[35] Bin Ren, Yawei Li, Xu Zheng, Yuqian Fu, Danda Pani Paudel, Hong Liu, Ming-Hsuan Yang, Luc Van Gool, and Nicu Sebe. Efficient degradation-agnostic image restoration via channel-wise functional decomposition and manifold regularization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jDMAvoLsVj.

[36] Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. Rf-detr: Neural architecture search for real-time detection transformers, 2025. URL https://arxiv.org/abs/2511.09554.

[37] Xiaoqian Ruan and Wei Tang. Fully test-time adaptation for object detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1038–1047, 2024. doi: 10.1109/CVPRW63382.2024.00110.

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[39] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[40] Jenny Schmalfuss, Victor Oei, Lukas Mehl, Madlen Bartsch, Shashank Agnihotri, Margret Keuper, and Andrés Bruhn. Robustspring: Benchmarking robustness to image corruptions for optical flow, scene flow and stereo. *arXiv preprint arXiv:2505.09368*, 2025.

[41] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[42] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV 2016 Workshops*, 2016.

[43] Aditi Tiwari, Farzaneh Masoud, Dac Trong Nguyen, Jill Kraft, Heng Ji, and Klara Nahrstedt. Fire360: A benchmark for robust perception and episodic memory in degraded 360-degree firefighting videos. *arXiv preprint arXiv:2506.02167*, 2025.

[44] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. URL https://arxiv.org/abs/2502.14786.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[46] 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.

[47] Ethan Weber, Dim P Papadopoulos, Agata Lapedriza, Ferda Ofli, Muhammad Imran, and Antonio Torralba. Incidents1m: a large-scale dataset of images with natural disasters, damage, and incidents. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4768–4781, 2022.

[48] Wang Wen, Cao Yang, Zhang Jing, He Fengxiang, Zha Zheng-Jun, Wen Yonggang, and Tao Dacheng. Exploring sequence feature alignment for domain adaptive detection transformers. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.

[49] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. doi: 10.1109/CVPR.2010.5539970.

[50] Jinquan Yan, Zhicheng Zhao, Zhengzheng Tu, Chenglong Li, Jin Tang, and Bin Luo. Compositional-degradation uav image restoration: Conditional decoupled moe network and a benchmark. *arXiv preprint arXiv:2604.09313*, 2026.

[51] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021.

[52] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022.

[53] Lihua Zhou, Mao Ye, Shuaifeng Li, Nianxin Li, Jinlin Wu, Xiatian Zhu, Lei Deng, Hongbin Liu, Jiebo Luo, and Zhen Lei. Bayesian test-time adaptation for object recognition and detection with vision-language models. *arXiv preprint arXiv:2510.02750*, 2025.

# Appendix

The appendix is organized as follows:

- **Appendix A** reports dataset statistics for the object-level and scene-level.
- **Appendix B** describes the object and scene generation pipeline, including prompts, real-image grounding, and filtering.
- **Appendix C** provides the evaluation prompts and scoring protocols for all tasks.
- **Appendix D** details FRM instantiations across VLM and frozen-encoder hosts.
- **Appendix E** reports full per-level VLM results.
- **Appendix F** reports frozen-encoder feature recovery and SigLIP2 retrieval.
- **Appendix G** provides additional ablations on capacity, integration, and tap-point selection.

## A Dataset Statistics

This appendix summarizes the subset statistics, object and scene coverage, and degradation diagnostics for TRACE. Detailed generation prompts, grounding-judge criteria, and filtering protocol are provided in Appendix B.

### A.1 Subset Statistics

Figures 6 and 7 summarize object-level and scene-level coverage. Figure 8 contrasts standard image corruptions with physical transformation. Figure 9 shows the effect of real-image grounding and VLM filtering.

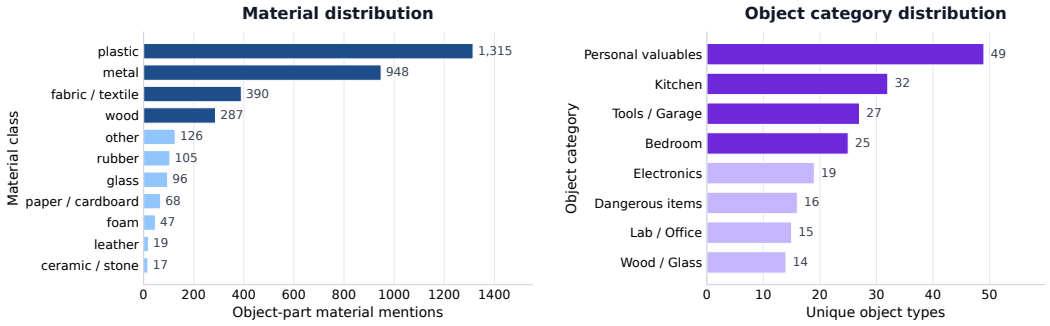



**Figure 6: Object inventory statistics.** The object inventory spans 499 object identities, 189 object types, and 11 normalized material classes. Materials are long-tailed, with plastic, metal, fabric/textile, and wood most represented.

**Table 6: Comparison with robustness and disaster datasets.** Existing benchmarks provide clean-corrupted image pairs, matched adverse-condition scenes, or disaster damage annotations. TRACE instead tracks individual objects through progressive physical degradation with a pristine reference.

| Dataset | Setting | Reference pairing | Physical object trajectory |
|---|---|---|---|
| ImageNet-C [19] | Image corruption | Clean image | No |
| COCO-C [28] | Detection corruption | Clean image | No |
| ACDC [39] | Adverse driving conditions | Matched normal scene | No |
| xBD [17] | Disaster building damage | Pre/post satellite | No |
| RescueNet [33] | Post-hurricane UAV damage | No | No |
| Incidents1M [47] | Incident images | No | No |
| TRACE | Post-fire object damage | Pristine object | L0–L4 |

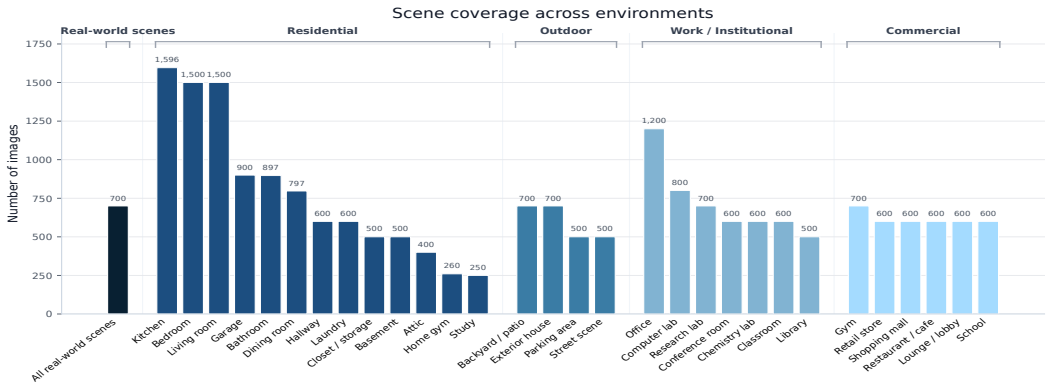



**Figure 7: TRACE scene coverage.** TRACE contains 21,400 accepted synthetic post-fire scenes grounded by approximately 700 real post-fire reference images. Scenes span residential, outdoor, work/institutional, and commercial environments.

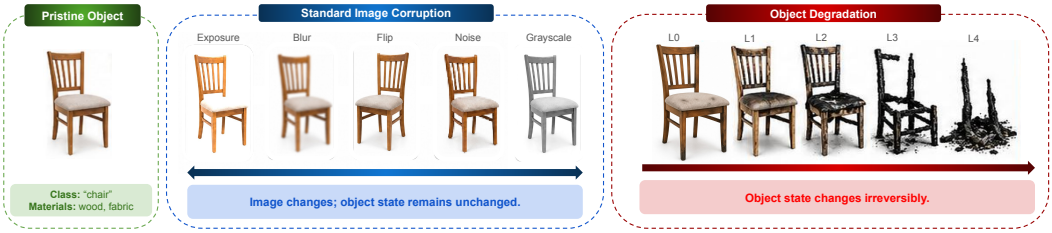



**Figure 8: Physical transformation is not image corruption.** Standard corruptions alter image appearance while preserving object state. Post-fire degradation changes material and structure, requiring recovery of pre-degradation identity and properties.

## A.2 Degradation Severity Metadata

Each object trajectory is annotated with approximate severity metadata for soot coverage, surface or material failure, deformation, and structural integrity. These ranges guide generation and support severity-stratified evaluation (Table 7).

## A.3 Annotation Summary

Object-level metadata includes part-level material composition, object-specific failure notes, and per-level degradation descriptions. Scene-level metadata includes bounding boxes, mapped object labels, visible materials, degradation states, scene captions, and generation metadata. Intended object and material inventories from the generation prompt are stored as metadata and are not used as ground truth for evaluation.

**Table 7: Degradation severity metrics by level.** Approximate ranges used as object-level generation metadata across the five degradation levels.

| Metric | L0 | L1 | L2 | L3 | L4 |
|---|---|---|---|---|---|
| Soot coverage (%) | 30-45 | 45-60 | 60-75 | 72-87 | 87-96 |
| Surface/material failure (%) | 30-50 | 30-45 | 25-40 | 40-60 | 60-80 |
| Deformation (%) | 5-15 | 15-30 | 35-55 | 55-75 | 75-92 |
| Structural integrity (%) | 85-95 | 65-82 | 40-58 | 15-35 | 4-13 |



Our Pipeline-grounded generation



Prompt-only generation

**Figure 9: Effectiveness of the scene-generation pipeline.** Both rows use Gemini 3 Pro Image as the generator. The top row shows scenes conditioned on 2–4 real post-fire reference images with VLM-based filtering. The bottom row uses the same prompts without reference grounding or filtering. Representative real post-fire scenes are shown in Figure 1(a–c).

## B Generation Prompts and Filtering

This appendix provides the prompt templates used to generate the scene-level and object-level subsets of TRACE. All generation uses Gemini 3 Pro Image for image synthesis and Gemini 3 Flash as the grounding judge.

Object Trajectory Metadata Example: Dresser

```
{
  "object": "dresser",
  "scene_category": "bedroom",
  "trajectory": "1 pristine + 5 degraded states",
  "split_unit": "object identity",
  "degradation_levels": ["L0", "L1", "L2", "L3", "L4"]
}
```



```
{
  "part_materials": [
    {"part": "top surface", "material": "MDF or hardwood with veneer"},
```

```
    {"part": "body panels", "material": "MDF laminate or hardwood"},
    {"part": "drawer fronts", "material": "MDF or hardwood"},
    {"part": "drawer boxes", "material": "MDF or plywood"},
    {"part": "drawer runners", "material": "steel slides"},
    {"part": "drawer pulls", "material": "brass, zinc alloy, or ABS plastic"},
    {"part": "base plinth", "material": "hardwood or MDF"},
    {"part": "back panel", "material": "thin MDF or hardboard"},
    {"part": "finish coating", "material": "polyurethane lacquer or paint"}
  ]
}


{
  "material_failure_notes": {
    "wood_MDF": "finish blisters and peels; MDF chars, swells, and crumbles",
    "hardware": "steel drawer slides and metal pulls survive longest",
    "late_stage": "dresser collapses into a charred structure with surviving hardware"
  }
}


{
  "severity_ladder": {
    "L0": "heavy soot, finish blistering, early surface char, structure intact",
    "L1": "deep surface char, cracking, stronger heat damage",
    "L2": "burn-through in weaker members, exposed hardware, partial failure",
    "L3": "collapsing charred skeleton with surviving metal components",
    "L4": "near-total destruction with weak identity cues in hardware and debris"
  }
}
```


### B.1 Object-Level Generation

Each object trajectory is generated iteratively. The pristine image is generated first from a text-only prompt, and each subsequent degradation stage (L0-L4) is conditioned on the previous stage's image. This ensures cumulative, physically grounded damage progression.

**Pristine prompt.** The pristine stage uses a studio product-photo prompt with explicit part-level material specifications and variant identity constraints to ensure distinct physical instances across variants of the same category.

**Pristine Prompt Template**

Single-object studio product photograph of a `{obj}`.
This exact object instance must be: `{variant_style}`.
Design language: `{variant_manufacturing}`.
Color / finish direction: `{variant_colorway}`.
The object is centered, fills roughly 70 percent of the frame, fully visible, isolated on a pure white seamless background. Soft even studio lighting, sharp focus, realistic shadows only under the object, no stylization, no watermark, no text overlay.
Exact part-level construction that must be visible:
`{part_materials}`
Hard constraints:

- Keep the object in a canonical product-photo orientation.
- Keep background unchanged across all stages.
- Do not add hands, props, shelves, tables, walls, flames, smoke clouds, or extra objects.
- Show realistic material texture and manufacturing details.
- This image must depict a DIFFERENT real-world variant than other versions of the same object category, not merely a recolor.
- The silhouette, proportions, detailing, and subtype cues should clearly distinguish this instance from sibling variants.

**Degradation stage prompt.** Each degradation stage is conditioned on the previous stage's image and includes material-specific failure behavior, stage-level severity metrics, and strict constraints ensuring cumulative and spatially uneven damage.

**Degradation Stage Prompt Template**

Generate the next degradation image for a `{obj}` at stage `{stage_level}` using the provided reference image(s) as the SAME physical object instance.
`{variant_identity_block}`
`{consistency_constraints}`
`{global_realism_rules}`
Exact part-level construction that must remain physically grounded:
`{part_materials}`
Material-specific failure behavior that MUST be obeyed per part:
`{material_behavior}`
Object-specific failure notes:
`{object_specific_failure_notes}`
Stage description - THIS IS THE TARGET STATE:
`{stage_description}`
Critical image constraints:

- Make stage `{stage_level}` CLEARLY and DRAMATICALLY different from the previous stage.
- Every visible surface must show fire exposure - no pristine areas anywhere.
- Keep the white studio background unchanged.
- Do not add flames, smoke plumes, sparks, people, tables, or debris outside the object footprint.
- Damage must be spatially uneven and material-aware - not a uniform dark overlay.

**Consistency and realism constraints.** The following constraints are injected into every degradation stage prompt to enforce material-aware failure behavior.

**Material Failure and Realism Rules**

**Material-specific failure rates:** *Paper and cardboard.* Among the first materials to fail. Curl and brown at edges, then char progressively inward. Thin sheets fragment into ash flakes. Cardboard delaminates at glue seams before charring. Final state is gray-white ash with occasional partially-surviving fragments in shielded regions.
*Fabric and textile.* Synthetic fabrics (polyester, nylon) melt, shrink, and fuse into hard irregular beads or films. Natural fabrics (cotton, linen) char and disintegrate into fragile ash. Blended fabrics show both behaviors simultaneously. Seams and hems often survive longer than flat panels. Collapse follows loss of structural fiber integrity.
*Foam and padding.* Melts, shrinks, and collapses rapidly. Exposed foam recedes from edges inward, leaving skeletal frames or springs visible underneath. Polyurethane foam produces sticky, re-solidified residue. Memory foam chars into brittle black mass.
*Leather.* Shrinks dramatically (up to 50% area loss), curls at edges, and develops deep surface cracking. Grain pattern distorts but remains partially visible. Blackens progressively from tan to dark brown to black. Does not melt or produce liquid residue.
*Wood and MDF.* Surface scorches first, producing alligator-skin char patterns. Deeper charring follows with visible char depth increasing per stage. MDF delaminates at adhesive layers. Structural wood cracks along grain lines. Final stages show brittle, crumbling charcoal with ash accumulation at the base.
*Plastics (ABS, polycarbonate, polypropylene, PVC).* Soften and deform at relatively low temperatures. ABS blisters and warps. Polycarbonate yellows, then browns, then deforms. Polypropylene melts and pools into irregular puddles. PVC blackens and becomes brittle. Re-solidified plastic forms irregular lumpy masses with frozen drip patterns. Thin-walled plastic containers collapse inward.
*Rubber and silicone.* Blacken and shrink. Rubber hardens and cracks into segments. Silicone maintains shape longer but eventually chars and crumbles. Tire-like rubber develops deep fissures before fragmenting. Rubber coatings on handles and grips peel and curl away from underlying metal or plastic.
*Metal (steel, aluminum, copper, iron).* Most durable common material. Steel heat-tints through straw, blue, and purple before developing surface oxide scale. Aluminum softens, sags, and can partially melt at extreme temperatures, losing structural rigidity. Copper and brass develop heavy green-black oxidation. Iron rusts rapidly in post-fire moisture. Thin sheet metal warps and buckles. Fasteners (screws, hinges) often survive intact and become identifying features. Metal does not become ash.
*Glass.* Shatters from thermal shock (rapid heating or cooling with water). Tempered glass produces small granular fragments. Plate glass cracks in radiating patterns before falling out of frames. At extreme temperatures, glass may soften and slump but does not melt into liquid. Soot deposits on surviving glass surfaces.

*Ceramic and stone.* Ceramic glazes craze (develop fine crack networks) and may spall. Unglazed ceramic and stone crack under thermal stress but maintain overall form. Neither material melts. Porcelain can shatter from thermal shock. Stone surfaces accumulate soot and may show heat-induced color changes (reddening of limestone, darkening of granite).
*Electronics and composite materials.* PCBs char and delaminate, exposing copper traces. Solder melts and beads on surfaces. LCD and LED screens crack and blacken. Battery cells swell, rupture, and may leave chemical residue. Wire insulation melts away, exposing bare copper conductors. Integrated circuits and ceramic chip packages may survive as identifiable components.
*Gemstones.* Diamond, sapphire, ruby, and similar crystalline stones survive fire temperatures and remain as intact loose stones among debris. Settings (typically metal) may deform or melt around them. Softer stones (opal, turquoise) may crack or discolor.
**Failure rate ordering (earliest to latest):** Paper and cardboard → thin fabric and foam → thick fabric and leather → wood and MDF → plastic and rubber → glass → ceramic and stone → metal → gemstones.
**Stage consistency:** Damage must accumulate from the previous stage only. Each new stage must be clearly and dramatically worse. Viewpoint, framing, scale, and orientation are preserved until physical collapse makes it impossible. Do not invent parts not present in the pristine image. Different materials on the same object must fail at different rates, so a single object at intermediate stages will show a mix of material states (e.g., fabric seat cover charred while the metal frame beneath shows only heat-tinting).

**Degradation stage definitions.** Each object trajectory progresses through five degradation stages (L0-L4). The following definitions specify the target physical state at each stage and are injected into the stage prompt as `{stage_description}`.

**Degradation Stage Definitions**

**L0 - Mild surface damage.** Early-stage fire exposure. Light soot deposits on exposed surfaces. Minor discoloration or browning of heat-sensitive materials (paper, thin fabric). Plastic surfaces may show slight warping or surface bubbling. Metal shows no visible change or faint heat tinting. Object silhouette, proportions, and all parts remain fully intact. The object is immediately recognizable.
**L1 - Moderate surface damage.** Sustained heat exposure across all surfaces. Paper and thin fabric are partially charred or missing. Plastic shows visible warping, bubbling, or early melting. Wood surfaces show surface scorching and early char patterns. Metal begins to show heat-tint discoloration. Leather shrinks slightly and darkens. Foam begins to recede from edges. Object silhouette remains intact but surface appearance has changed substantially. Recognition requires attention to shape and remaining material cues.
**L2 - Structural onset.** Fire has begun to compromise structural integrity of weaker materials. Fabric and foam are largely gone or reduced to residue, exposing internal frames and components. Plastic has melted and re-solidified into deformed masses. Wood shows deep charring with visible char depth and cracking along grain. Rubber has hardened and cracked. Metal shows moderate oxidation and early warping of thin sections. Some parts may have detached or collapsed. Object is still recognizable from overall shape but individual material surfaces are heavily altered.
**L3 - Severe degradation.** Major structural failure of non-metallic materials. Wood is deeply charred and crumbling. Plastic has pooled or fused to other surfaces. Fabric, paper, foam, and leather are largely absent. Glass is shattered or missing. Metal components show heavy oxidation, soot coverage, and warping. Significant parts of the object may have collapsed, detached, or disintegrated. Object identity is recoverable primarily from surviving metal skeleton, overall proportions, and spatial arrangement of remaining components.
**L4 - Near-total destruction.** Approaching complete loss of original form. Non-metallic materials are reduced to ash, charred rubble, or re-solidified residue. Metal components are heavily deformed, oxidized, and may have partially melted (aluminum) or collapsed. Object silhouette is barely recognizable or unrecognizable without context. Identification requires expert-level inference from surviving fragments, hardware, and spatial cues. Some objects at this stage consist primarily of a debris pile with identifiable metal remnants.

### B.2 Scene-Level Generation

Each scene is generated from a structured prompt conditioned on two to four real post-fire reference photographs. The prompt specifies scene type, damage level, material profile, clutter level, lighting, and socioeconomic band, with explicit instructions that references serve only as inspiration for realism and material fidelity.

**Scene Generation Prompt Template (abbreviated)**

You are generating a completely NEW, original real-world post-fire scene.
The reference images are ONLY for inspiration about realism, material degradation fidelity, structural damage realism, and scene complexity. Do NOT copy, recreate, or closely match any specific layout, room geometry, viewpoint, object arrangement, composition, or damage pattern from the references. The generated scene must differ significantly in geometry, camera angle, and object placement from all references.
Generate a realistic `{scene_type}` after a fire incident. The image should be novel, physically grounded, hyper-realistic, and clearly distinct from the references.

**TARGET SCENE CONTROL:**

- Scene type: `{scene_type}`
- Damage level: `{damage_level}`
- Material profile: `{material_profile}`
- Clutter level: `{clutter_level}`
- Lighting mode: `{lighting}`
- Socioeconomic / aesthetic band: `{wealth_style}`
- Layout style: `{layout_variant}`

**PHYSICAL AND MATERIAL ACCURACY (CRITICAL):** All visible materials must degrade according to real-world fire behavior.

- *Wood:* charred, cracked, structurally weakened, sometimes partially collapsed. Load-bearing beams may sag or fracture. Exposed grain shows deep alligator-pattern charring. Ash accumulates at the base of vertical wooden surfaces.
- *Plastic:* melted, warped, deformed, softened, pooled, or fused to adjacent surfaces. Thin-walled containers collapse inward. Re-solidified plastic forms irregular lumpy masses with frozen drip patterns on floors and shelving below.
- *Fabric and upholstery:* burned away unevenly, collapsed, torn, or reduced to residue. Synthetic fabrics melt and fuse into hard beads or films. Natural fabrics char and fragment. Springs and frames become exposed as covering material fails.
- *Foam and padding:* melted, collapsed, shrunken, missing in sections, or exposed beneath charred fabric. Polyurethane foam produces sticky re-solidified residue. Mattresses and cushions show deep cavities where foam has receded.
- *Metal:* structurally present but heat-discolored, oxidized, soot-covered, or slightly warped. Steel shows straw-to-blue heat tinting. Aluminum may sag or partially melt. Thin sheet metal (appliance panels, ductwork) buckles and warps. Fasteners, hinges, and handles often survive intact.
- *Glass:* shattered, fractured, partially missing, or heat-cracked. Window frames may retain jagged fragments. Tempered glass produces granular piles on the floor below. Mirrors crack in radiating patterns. Soot deposits on surviving glass surfaces.
- *Paper and cardboard:* curled, charred, blackened, fragmented, reduced to ash, or partially surviving in shielded regions such as inside drawers or behind other objects.
- *Rubber:* warped, softened, heat-damaged, partly melted, or degraded at edges. Cable insulation melts away exposing wire conductors. Rubber seals and gaskets shrink and crack.
- *Concrete, tile, and masonry:* scorching, cracking, soot staining, and spalling where surfaces faced direct heat. Grout may crack and crumble. Floor tiles may lift or shatter from thermal expansion. Concrete walls show smoke staining gradients that indicate fire direction and intensity.
- *Insulation and ceiling materials:* exposed fiberglass or mineral wool sags and discolors. Ceiling tiles collapse or hang partially detached. Drywall paper facing chars while the gypsum core remains as crumbling white residue.

**SPATIAL DAMAGE CONSISTENCY:**

- Fire damage must be uneven and spatially coherent, with heavier damage near the likely fire origin and fading outward.
- Upper surfaces, ceilings, and the tops of walls show heavier soot accumulation and heat damage than lower surfaces (hot gas layer rises).
- Visible soot gradients and runoff streaking on vertical surfaces.
- V-patterns and inverted-cone burn patterns on walls indicate upward fire spread from point sources.
- Floors show debris accumulation from collapsed shelving, ceiling, and wall-mounted objects.

- Adjacent rooms or hallways visible through doorways should show diminishing damage with distance from the fire origin.

**SECONDARY EFFECTS:**

- Realistic firefighting residue including white or gray extinguisher powder and water pooling or streaking on floors.
- Some surfaces may appear damp, washed, or partially streaked from hose water.
- Debris, ash, and fragmented material accumulate naturally around collapse points, along walls, and at the base of furniture.
- Electrical wiring may be exposed where wall or ceiling coverings have burned away.
- Smoke detectors, light fixtures, and ceiling-mounted objects may hang partially detached.

**OBJECT DISTRIBUTION:**

- Scene should feel asymmetric, naturally disturbed, and realistic.
- Include a mix of partially recognizable objects and heavily degraded objects.
- No central hero object and no staged or overly clean composition.
- Natural occlusion and layering should occur (objects partially buried under debris, leaning against walls, fallen from shelves).
- At least 2-3 objects should be identifiable to a trained observer despite damage.

**STRUCTURAL DETAILS:**

- Possible partial collapse of ceilings, shelving, partition walls, door frames, or built-in cabinetry.
- Exposed structural elements (studs, joists, wiring, plumbing) where wall and ceiling coverings have failed.
- Doors may be open, partially burned, or missing. Door hardware (knobs, hinges) typically survives.
- Windows may be broken out by thermal shock or firefighting ventilation. Curtain rods may remain while curtain fabric is gone.

**CAMERA STYLE:** Realistic handheld or inspection-style photograph, approximately 24-35mm equivalent, natural framing imperfections, documentary-like realism. No cinematic grading or artistic stylization.

**STRICT CONSTRAINTS:** No flames, no active smoke or haze effects, no stylization, no painterly or artistic rendering, no cinematic grading, no unrealistic symmetry, no pristine clean areas unless physically justified (e.g., shielded by another object), no duplicated obvious objects or synthetic visual artifacts.

`{scene_specific}`

The output must be indistinguishable from a real-world post-fire documentation photo.

To reduce visual homogeneity, scene prompts vary scene type, damage level, material profile, clutter, lighting, socioeconomic style, and layout, and explicitly prohibit copying the reference layout or viewpoint.

### B.3 Grounding Judge and Filtering

After generation, each scene candidate is scored by a Gemini 3 Flash vision-language judge that compares the candidate against its real-image references on a 0-1 grounding scale.

**Grounding Judge Prompt**

You are a grounding judge. Compare the candidate post-fire scene image against its reference real-world images. Score grounding on a 0 to 1 scale based on: (1) realism, (2) material and fire-physics fidelity, (3) structural damage plausibility, and (4) overall consistency with real-world post-fire visual characteristics while preserving novelty.

Return strict JSON only:

`{"grounding_score": <float>, "reasoning": "<short>", "pass": <bool>}` Set pass = true only if grounding_score $\geq 0.7$.

Candidates scoring below 0.7 are rejected. This filtering yields 21,400 accepted scenes from approximately 29,000 generated candidates, for an acceptance rate of 73.8%. The grounding score is not used as an evaluation label. It is used only to filter generated scenes for realism, material and fire-physics fidelity, structural damage plausibility, and consistency with the real post-fire reference set while

preserving novelty. Object boxes and labels are then handled as annotations for the detection task rather than as outputs of the grounding judge.

GroundingDINO [23] initializes object boxes and labels for accepted scenes. The authors inspect each annotation for consistency between the visible object remnant, its bounding box, and its class label. Incorrect boxes or labels are corrected, and ambiguous objects are removed. Objects without an unambiguous COCO mapping are excluded from mAP evaluation. Separately, two authors independently assigned accept/reject quality-control labels on a stratified 300-object subset, yielding 93.4% agreement and Cohen's $\kappa = 0.836$. These statistics measure agreement on annotation acceptance, not bounding-box overlap.

### B.4 Generation Failures and Prompt Refinement

During prompt development, we observed failures in material behavior and object consistency. Examples included book covers degrading at the same rate as pages, invented contents appearing inside damaged containers, and implausible damage to glass and appliances. We refined material-specific and part-consistency instructions to address these failures. Samples with inconsistent object structure or material behavior were regenerated or removed. These checks improve generation plausibility but cannot eliminate the synthetic-to-real gap.

## C Evaluation Task Details

This appendix specifies the five evaluation tasks from Sec. 3, namely degraded-object detection, pristine-state recovery and retrieval, original material recovery, pristine description generation, and functional reasoning.

### C.1 Degraded-Object Detection

Given a scene-level post-fire image, the model detects and localizes objects of interest. This task evaluates whether a detector can recover object remnants in cluttered, physically degraded scenes.

We evaluate RF-DETR outputs using class-aware COCO-mapped mAP@0.5:0.95, reported overall and separately by degradation level. A manually constructed lookup table maps TRACE object classes to the nearest COCO category for compatibility with COCO-pretrained detectors. Only boxes with unambiguous COCO mappings are included in mAP computation. The mapping covers 55 of the 80 COCO categories. The 55 mapped COCO categories cover the most common indoor object types (furniture, electronics, containers, appliances). Excluded categories are primarily TRACE-specific items without clear COCO counterparts (e.g., aerosol cans, gas cylinders, specific tool subtypes). The full mapping table is released with the evaluation code. A prediction is counted as correct only when it satisfies both localization and mapped-category correctness. Detection is evaluated on the scene-level subset and no natural-language prompt is used.

### C.2 Pristine-State Recovery and Retrieval

Given a degraded object image, the goal is to recover its paired pristine state. We evaluate this task with two complementary protocols: feature-space alignment and gallery retrieval.

**Feature-space alignment.** For frozen encoder diagnostics, CLIP ViT-L/14 and SigLIP2 features are recovered toward paired pristine targets. Given degraded features $z^{\ell}$ and paired pristine features $z^{p}$, FRM produces recovered features $\tilde{z}^{\ell}$. We report mean squared error (MSE) and mean token cosine similarity between $\tilde{z}^{\ell}$ and $z^{p}$. Lower MSE and higher cosine similarity indicate stronger pristine alignment.

**Gallery retrieval.** For VLM retrieval, the model retrieves the corresponding pristine reference from a gallery of pristine objects. For consistency with the VLM setting, we use the following natural-language query:

```
Find the pristine object image in the gallery that corresponds to
this physically degraded object.
```

The prompt defines the retrieval intent, but scoring is performed in feature space: the model's visual encoder extracts features for the degraded query and all pristine gallery images, and gallery images are ranked by cosine similarity. Each degraded query has exactly one paired pristine target. We report Recall@1 as the main gallery-retrieval metric.

### C.3 Original Material Recovery

Given a degraded object image, the model predicts the object's pre-degradation materials. The prompt is:

```
What were the main materials of this object before degradation?
Reply with a comma-separated list of single-word materials only (for
example:  wood, steel, plastic).  Do not describe the damage.
```

Ground-truth materials are parsed from the object's part-level material annotations and mapped to the 11 normalized material classes in TRACE. Model responses are lowercased, stripped of punctuation, singularized, and mapped to the same material vocabulary using a fixed synonym table. Let $Y_i^{\mathrm{mat}}$ and $\hat{Y}_i^{\mathrm{mat}}$ denote the ground-truth and predicted material sets for object $i$. We compute micro-F1 across all objects and material labels:

$$P_{\mathrm{mat}} = \frac{\sum_i |\hat{Y}_i^{\mathrm{mat}} \cap Y_i^{\mathrm{mat}}|}{\sum_i |\hat{Y}_i^{\mathrm{mat}}|}, \qquad R_{\mathrm{mat}} = \frac{\sum_i |\hat{Y}_i^{\mathrm{mat}} \cap Y_i^{\mathrm{mat}}|}{\sum_i |Y_i^{\mathrm{mat}}|},$$

$$\mathrm{F1}_{\mathrm{mat}} = \frac{2P_{\mathrm{mat}}R_{\mathrm{mat}}}{P_{\mathrm{mat}} + R_{\mathrm{mat}}}.$$

If a model predicts no labels for a task, precision is set to zero; if the ground-truth set is empty, recall is set to zero. In our benchmark, ground-truth material and function sets are non-empty. We report $100 \cdot \mathrm{F1}_{\mathrm{mat}}$.

### C.4 Pristine Description Generation

Given a degraded object image, the model generates a one-sentence description of the object's pre-degradation state. The prompt is:

```
Describe what this object most likely was before it was damaged.
Mention its object type, main material(s), and intended use.  Do not
focus on the damage.  Answer in one concise sentence.
```

Ground-truth identity is taken from the object manifest, materials from the part-level annotations, and function from the canonical function tag assigned to the object category. Each generated sentence is parsed into three fields: object identity, materials, and function. Identity and function predictions are lowercased, stripped of punctuation and articles, singularized, and matched to the canonical category or function vocabulary using fixed synonym tables. Materials are normalized as in Appendix C.3.

For object $i$, identity is scored as

$$S_i^{\mathrm{id}} = \mathbb{1}[\hat{y}_i^{\mathrm{id}} = y_i^{\mathrm{id}}],$$

where $y_i^{\mathrm{id}}$ is the ground-truth object category and $\hat{y}_i^{\mathrm{id}}$ is the normalized predicted category. Function is scored analogously:

$$S_i^{\mathrm{func}} = \mathbb{1}[\hat{y}_i^{\mathrm{func}} = y_i^{\mathrm{func}}].$$

Materials are scored using micro-F1 over the normalized material vocabulary, following Appendix C.3. The final pristine-description score is the unweighted mean of identity accuracy, material micro-F1, and function accuracy:

$$S_{\mathrm{desc}} = 100 \cdot \frac{1}{3}\left(\frac{1}{N}\sum_{i=1}^{N} S_i^{\mathrm{id}} + \mathrm{F1}_{\mathrm{mat}} + \frac{1}{N}\sum_{i=1}^{N} S_i^{\mathrm{func}}\right).$$

When deterministic synonym normalization cannot resolve a paraphrase, we use a deterministic LLM equivalence judge as a fallback. The judge receives only the predicted phrase, the ground-truth label, and the relevant synonym policy, and returns a binary equivalent/not-equivalent decision. It is

run with temperature $0.0$ and is used only for unresolved identity and function paraphrases, not for material scoring.

The synonym table resolves $5{,}562/6{,}250$ identity predictions ($89.0\%$) and $5{,}312/6{,}250$ function predictions ($85.0\%$) without invoking the judge. The GPT-5 fallback is triggered for $688/6{,}250$ identity evaluations ($11.0\%$) and $938/6{,}250$ function evaluations ($15.0\%$), mainly for paraphrases such as "writing surface" for "desk." Replacing the fallback with a strict no-match rule changes the aggregate description score by less than $0.8$ points.

### C.5 Functional Reasoning

Given a degraded object image, the model answers two structured questions about the object's pre-degradation function and restoration needs without being told its category.

**Prompt.**

```
You are looking at an object that has been damaged in a fire.
Reason about what the object originally was and answer two questions
about its pre-damage identity and restoration potential.

FUNCTION: In one short phrase (5 words or fewer), what was this
object's original function or primary use?

REPLACEMENTS: List the main materials that would be needed
to restore this object to working condition.  Reply as a
comma-separated list of single-word items.

Respond in this exact format with no extra text:
FUNCTION: <your answer>
REPLACEMENTS: <comma-separated list>
```

The prompt is identical across all hosts and across FRM-on/off conditions. All calls use temperature $0.0$ and a maximum of 150 output tokens.

**Prompt sensitivity.** Functional reasoning is the only prompted task that explicitly mentions fire. Replacing "damaged in a fire" with "damaged" changes the Qwen3-VL FRM gain from $+5.94$ to $+5.57$ points. The other prompted tasks use fire-neutral wording, while retrieval is evaluated through visual-feature similarity. Thus, explicit fire context does not explain the FRM improvement.

**Ground truth. i. Function labels.** We define 20 canonical function tags, such as seating, lighting, cooking_appliance, food_storage, general_storage, safety_equipment, and communication_device. Each object type in TRACE is mapped to one or two function tags. The mapping is constructed with LLM assistance and verified by the authors. The full mapping is released as supplementary material.

**ii. Restoration-material labels.** The canonical material set annotated for each object type serves as the ground-truth restoration-material set. This reuses existing material annotations and introduces no additional task-specific labels.

**Scoring.** The FUNCTION response is normalized by lowercasing, stripping punctuation, singularizing, and mapping to the fixed function-tag vocabulary using keyword and synonym rules. When this deterministic mapping is ambiguous, we use the same deterministic LLM fallback described in Appendix C.4. The judge receives the model response and the fixed function-tag vocabulary, and returns one or more canonical tags. It is not given the ground-truth tag. Let $Y_i^{\text{func}}$ and $\hat{Y}_i^{\text{func}}$ denote the ground-truth and predicted function-tag sets for object $i$. Function performance is computed as micro-F1:

$$P_{\text{func}} = \frac{\sum_i |\hat{Y}_i^{\text{func}} \cap Y_i^{\text{func}}|}{\sum_i |\hat{Y}_i^{\text{func}}|}, \qquad R_{\text{func}} = \frac{\sum_i |\hat{Y}_i^{\text{func}} \cap Y_i^{\text{func}}|}{\sum_i |Y_i^{\text{func}}|},$$

$$\text{F1}_{\text{func}} = \frac{2P_{\text{func}}R_{\text{func}}}{P_{\text{func}} + R_{\text{func}}}.$$

The REPLACEMENTS response is parsed as a comma-separated material set and normalized to the same 11 material classes used for material recovery. Let $Y_i^{\text{rest}}$ and $\hat{Y}_i^{\text{rest}}$ denote the ground-truth and

**Table 8: FRM parameter overhead across VLM and frozen-encoder hosts.** For each host, $D$ is the hidden dimension at the selected tap point. All listed hosts use four Transformer blocks with MLP ratio $r$=4, giving $P_{FRM} \approx M(4+2r)D^2$. Overhead is reported relative to the host parameter count where the host size is fixed in our experiments.

| Host | Tap | $D$ | $M$ | $r$ | $P_{FRM}$ | Overhead |
|---|---|---|---|---|---|---|
| Qwen3-VL-4B | post-merger | 2560 | 4 | 4 | 314.57M | 7.86% |
| Qwen2.5-VL-3B | post-merger | 2048 | 4 | 4 | 201.33M | 6.71% |
| InternVL3.5-4B | post-projector | 2560 | 4 | 4 | 314.57M | 7.86% |
| Molmo2-4B | post-projector | 2560 | 4 | 4 | 314.57M | 7.86% |
| LLaVA-OV-1.5-4B | post-merger | 2560 | 4 | 4 | 314.57M | 7.86% |
| CLIP ViT-L/14 | pre-projection | 1024 | 4 | 4 | 50.33M | – |
| SigLIP2-base | pre-projection | 768 | 4 | 4 | 28.31M | – |

predicted restoration-material sets. Restoration-material performance is computed as micro-F1:

$$P_{\text{rest}} = \frac{\sum_i |\hat{Y}_i^{\text{rest}} \cap Y_i^{\text{rest}}|}{\sum_i |\hat{Y}_i^{\text{rest}}|}, \qquad R_{\text{rest}} = \frac{\sum_i |\hat{Y}_i^{\text{rest}} \cap Y_i^{\text{rest}}|}{\sum_i |Y_i^{\text{rest}}|},$$

$$\text{F1}_{\text{rest}} = \frac{2P_{\text{rest}}R_{\text{rest}}}{P_{\text{rest}} + R_{\text{rest}}}.$$

The final functional-reasoning score is the unweighted mean of function-tag micro-F1 and restoration-material micro-F1:

$$S_{\text{func}} = 100 \cdot \frac{1}{2}\left(\text{F1}_{\text{func}} + \text{F1}_{\text{rest}}\right).$$

If the denominator of precision or recall is zero, the corresponding value is set to zero.

**Baseline artifact.** At high severity, models sometimes default to coarse answers such as "container" or "furniture." Such responses may partially match broad function tags even when the predicted object identity is underspecified. We therefore interpret functional-reasoning results together with pristine-description and retrieval performance. FRM's high-severity gains, when accompanied by improved identity and material recovery, indicate more specific pre-degradation recovery rather than only coarse functional guessing.

# D FRM Implementation Details

This appendix details the host-specific tap points used to instantiate FRM. All hosts follow the same formulation from Sec. 4: degraded features and paired pristine targets are extracted at the same frozen feature interface, FRM maps $z^\ell \in \mathbb{R}^{B\times N\times D}$ to $\tilde{z}^\ell$, and only FRM is trained. The token dimension $D$ is determined by the selected tap point, while $N$ is the number of visual tokens at that interface.

Table 8 reports the resulting FRM sizes for VLM and frozen-encoder hosts. Qwen3-VL, InternVL3.5, Molmo2, and LLaVA-OV-1.5 use $D$=2560 at their LLM-facing post-merger or post-projector interfaces, yielding 314.57M FRM parameters. CLIP and SigLIP2 use earlier pre-projection vision tokens, giving 50.33M and 28.31M FRM parameters respectively. We omit percentage overhead for CLIP and SigLIP2 because the relevant denominator depends on whether one counts only the vision tower or the full contrastive model.

For SigLIP2 retrieval, the query is the degraded object image and the gallery contains candidate pristine object images (Figure 11).

## D.1 FRM Instantiation in VLM Hosts

**InternVL3.5-4B.** For InternVL3.5, FRM is trained at the post-projector visual-token interface. The frozen InternVL visual encoder first produces image tokens and the frozen multimodal projector maps them into the language-model input space. We apply FRM after this projector, so both degraded features and paired pristine targets have shape $z \in \mathbb{R}^{B\times N\times 2560}$. With $448 \times 448$ inputs, the token count $N$ is determined by the model preprocessing and projector configuration, and the feature dimension is the text hidden size $D$=2560. During training, pristine post-projector features are

cached once, degraded images are encoded on the fly, and FRM is optimized with the same MSE plus cosine recovery loss used in Sec. 4. At inference, the InternVL host remains frozen and the wrapper replaces the live degraded post-projector tokens with FRM-corrected tokens before generation. The baseline and FRM conditions therefore differ only in the visual feature path consumed by the frozen language model.

**LLaVA-OneVision-1.5-4B.** For LLaVA-OneVision-1.5, FRM is trained at the post-merger visual-token interface. The frozen Rice ViT first produces patch tokens, and the frozen RicePatchMerger spatially downsamples and projects them into the LLM-facing hidden space. We insert FRM after this merger, so the tapped tensor has shape $z \in \mathbb{R}^{B\times 256\times 2560}$ for the standard $448 \times 448$ input setting. Here $N{=}256$ follows from $(448/14/2)^2$, and $D{=}2560$ is the merger output dimension. The implementation reads the exact token count and feature dimension from the model configuration. Pristine post-merger tokens are cached before training, degraded tokens are encoded on the fly, and FRM is trained with the same paired feature objective as in Sec. 4. At inference, the wrapper patches the visual forward pass only inside the FRM-enabled context, applies FRM independently to each packed image token sequence, and then restores the original path. Thus, LLaVA generation, prompting, processor behavior, and language-model weights remain unchanged.

**Molmo2-4B.** For Molmo2, FRM is trained at the post-projector visual-token interface used by the host before language generation. The frozen vision stack produces packed visual tokens, the frozen projector maps them into the LLM-facing feature space, and FRM is applied to those post-projector tokens. Thus, the tapped tensor has shape $z \in \mathbb{R}^{B\times N\times 2560}$, where $N$ is the post-projector visual-token count and $D{=}2560$ is the LLM-facing hidden dimension read from the host configuration. We use the same VLM FRM configuration with $M{=}4$, 8 attention heads, and MLP ratio $r{=}4$.

Training follows the paired feature-recovery setup in Sec. 4. Pristine post-projector features are cached once, degraded images are encoded on the fly, and FRM is optimized with the same MSE plus cosine alignment loss against the paired pristine target. At inference, the wrapper enables FRM only inside the FRM-on condition. It runs the frozen visual path, splits packed visual features by image using runtime grid metadata, applies FRM independently to each image token sequence, and concatenates the corrected features before language generation. Baseline, FRM, and pristine-ceiling evaluations therefore share the same prompts, decoding, and frozen language model, differing only in whether degraded post-projector visual tokens are corrected before being consumed by the model.

### D.2 FRM Instantiation in Frozen Encoders

**CLIP ViT-L/14.** For CLIP, FRM is trained on pre-projection visual tokens from the frozen CLIP vision tower. We tap the CLIPVisionModel last hidden state before the visual projection into the text-aligned embedding space. For CLIP only, we drop the CLS token so that FRM operates on spatial patch tokens. With $224 \times 224$ inputs and patch size 14, the tapped feature tensor has $N{=}256$ patch tokens and hidden dimension $D{=}1024$, giving $z \in \mathbb{R}^{B\times 256\times 1024}$. Pristine patch-token features are cached once, degraded images are encoded on the fly, and FRM predicts recovered tokens in the same $B \times N \times D$ space. The frozen CLIP encoder is never updated, and evaluation compares recovered degraded tokens to paired pristine CLIP tokens using MSE, cosine similarity, and pooled-token retrieval.

**SigLIP2.** For SigLIP2, FRM is also trained at the pre-projection visual-token interface. We tap the frozen SigLIP2 vision model's last hidden state and use patch tokens rather than pooled image embeddings or text-aligned outputs. With google/siglip2-base-patch16-224, the input size is $224{\times}224$, the patch size is 16, and the visual grid is $14 \times 14$. The tapped tensor therefore has $N{=}196$ patch tokens and hidden dimension $D{=}768$, giving $z \in \mathbb{R}^{B\times 196\times 768}$. As in Sec. 4, FRM is shape preserving: degraded features $z^\ell$ are mapped to recovered features $\tilde{z}^\ell$ and supervised against cached paired pristine features $z^p$ from the same tap. We keep the SigLIP2 encoder frozen and report feature-space recovery metrics as well as pristine-state retrieval using pooled visual features.

**Integration and tap point.** Correcting Qwen3-VL's DeepStack stream slightly reduces identity accuracy, so we keep FRM at the primary visual-token interface. For tap-point selection, downstream-facing taps outperform earlier visual-token taps. Qwen3-VL improves from 72.90 R@1 with a pre-merger tap to 85.28 with a post-merger tap, while InternVL3.5 improves from 63.84 R@1 with a pre-projector tap to 74.92 with a post-projector tap (Table 10). These results support correcting the features consumed by the frozen language model.

**Capacity scaling.** Smaller FRM heads retain much of the retrieval improvement. A single block with $r=2$ uses 16.7% of the default parameters and recovers 82.6% of its R@1 gain over the frozen baseline. The default four-block configuration performs best among the tested capacities.

## E Per-Level VLM Results

Figure 10 and table 9 provide the full per-level breakdown of FRM gains across all five frozen VLM hosts and four object-level pristine-state tasks. Gains are small or occasionally negative at L0-L1, where degradation is mild and baseline features are already close to the pristine state. Gains concentrate at L2-L4, where physical transformation causes larger feature displacement and the learned correction becomes more useful. This pattern supports the view that FRM is most beneficial when degradation induces structured feature drift.

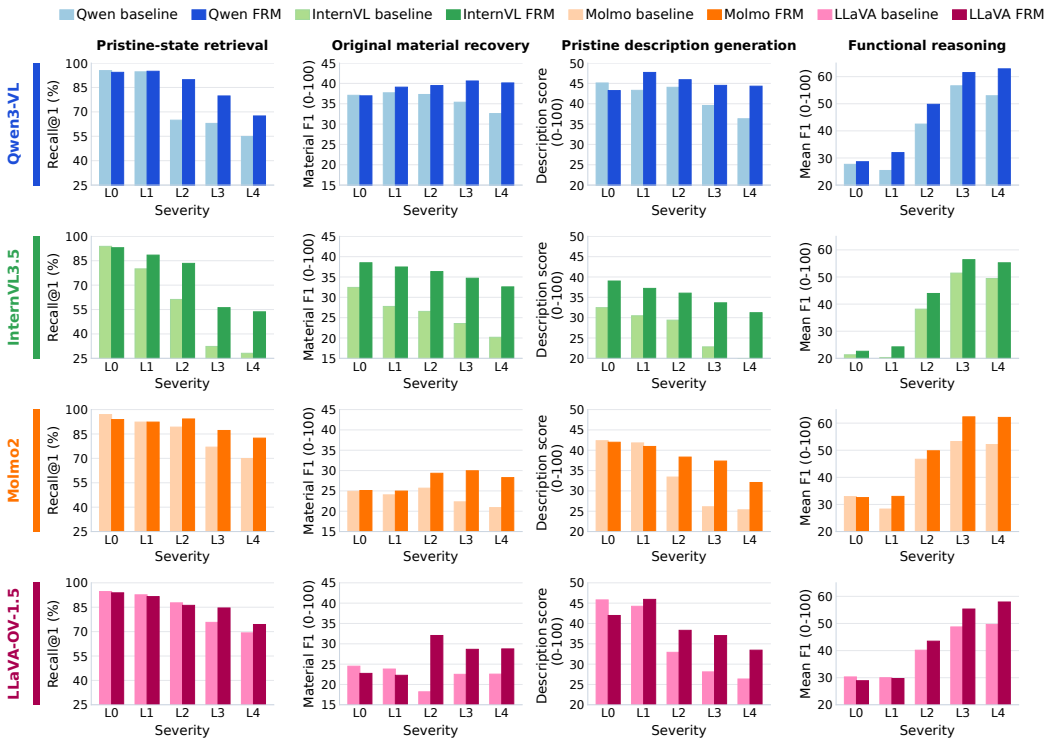



**Figure 10: Per-level VLM pristine-state recovery across additional hosts.** Baseline models are shown in light colors and FRM-enabled models in dark colors across L0-L4. Columns report pristine-state retrieval (R@1), original material recovery (F1), pristine description generation (score), and functional reasoning (mean F1). FRM gains concentrate at higher severity across hosts.

## F Frozen-Encoder Feature Recovery

We evaluate FRM on frozen CLIP ViT-L/14 and SigLIP2 encoders to isolate feature-space recovery from detector and language-model effects. The input is a degraded object image, and the target is the paired pristine image of the same object. We compare recovered degraded features against paired pristine features at the encoder tap point.

For retrieval, the query is the degraded object image and the gallery contains candidate pristine object images. SigLIP2 ranks gallery images by visual-feature similarity, with one paired pristine target per query. The pristine-query ceiling is 100% because querying with the pristine image retrieves the identical reference from the gallery.

**Table 9: Per-level VLM pristine-state recovery.** All scores are on a 0-100 scale. Mean Base and Mean +FRM report the average score over L0-L4. Mean Δ is the average improvement, while ΔL0-ΔL4 report per-level changes from adding FRM to the frozen host. Gains are strongest at L2-L4, while L0-L1 can show smaller gains, matched performance, or occasional drops.

| Model | Mean Base | Mean +FRM | Mean Δ | ΔL0 | ΔL1 | ΔL2 | ΔL3 | ΔL4 |
|---|---|---|---|---|---|---|---|---|
| **Pristine-state retrieval (R@1)** | | | | | | | | |
| Qwen3-VL | 74.55 | **85.28** | +10.73 | −1.06 | +0.36 | +24.88 | +16.88 | +12.58 |
| Qwen2.5-VL | 60.20 | **72.03** | +11.83 | +0.90 | +4.36 | +17.33 | +19.78 | +16.78 |
| InternVL3.5 | 59.09 | **74.92** | +15.83 | −0.83 | +8.46 | +22.18 | +23.84 | +25.48 |
| Molmo2 | 85.06 | **90.00** | +4.94 | −3.07 | +0.00 | +5.00 | +10.23 | +12.53 |
| LLaVA-OV-1.5 | 83.97 | **86.09** | +2.12 | −0.77 | −1.09 | −1.54 | +8.85 | +5.15 |
| **Original material recovery (F1)** | | | | | | | | |
| Qwen3-VL | 36.00 | **39.24** | +3.24 | −0.13 | +1.39 | +2.19 | +5.22 | +7.53 |
| Qwen2.5-VL | 30.50 | **36.78** | +6.28 | +0.93 | +4.29 | +6.50 | +8.50 | +10.90 |
| InternVL3.5 | 26.10 | **35.91** | +9.81 | +6.06 | +9.70 | +9.79 | +11.14 | +12.39 |
| Molmo2 | 23.56 | **27.52** | +3.96 | +0.20 | +0.93 | +3.66 | +7.64 | +7.37 |
| LLaVA-OV-1.5 | 22.31 | **26.89** | +4.58 | −1.77 | −1.54 | +13.85 | +6.15 | +6.20 |
| **Pristine description generation (score)** | | | | | | | | |
| Qwen3-VL | 41.67 | **45.15** | +3.48 | −1.83 | +4.40 | +1.86 | +4.95 | +8.01 |
| Qwen2.5-VL | 40.50 | **44.00** | +3.50 | +1.43 | +3.45 | +2.00 | +4.87 | +6.90 |
| InternVL3.5 | 27.01 | **35.42** | +8.41 | +6.55 | +6.73 | +6.64 | +10.83 | +11.32 |
| Molmo2 | 33.80 | **38.13** | +4.33 | −0.37 | −0.87 | +4.91 | +11.24 | +6.72 |
| LLaVA-OV-1.5 | 35.48 | **39.34** | +3.86 | −3.83 | +1.70 | +5.41 | +8.93 | +7.09 |
| **Functional reasoning (mean F1)** | | | | | | | | |
| Qwen3-VL | 41.02 | **46.96** | +5.94 | +0.96 | +6.59 | +7.30 | +4.88 | +9.93 |
| Qwen2.5-VL | 38.11 | **43.62** | +5.51 | +1.12 | +5.60 | +4.68 | +7.70 | +8.45 |
| InternVL3.5 | 36.13 | **40.48** | +4.35 | +1.30 | +3.92 | +5.78 | +4.95 | +5.80 |
| Molmo2 | 42.64 | **47.95** | +5.31 | −0.36 | +4.67 | +3.12 | +9.11 | +10.01 |
| LLaVA-OV-1.5 | 39.76 | **43.07** | +3.31 | −1.40 | +0.29 | +3.33 | +6.60 | +8.27 |

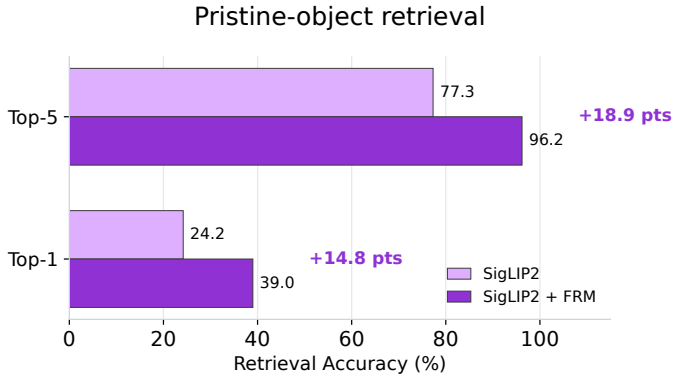



**Figure 11: SigLIP2 pristine-state retrieval.** Given a degraded object query, SigLIP2 ranks candidate pristine gallery images by feature similarity. FRM improves degraded-to-pristine retrieval, increasing Top-1 from 24.2% to 39.0% and Top-5 from 77.3% to 96.2%, approaching the pristine-query ceiling of 100%.

## G Additional FRM Ablations

**DeepStack and tap-point selection.** Table 10 reports additional integration ablations. For Qwen3-VL, applying FRM to the DeepStack stream slightly reduces identity accuracy compared with correcting only the primary visual-token interface. For tap-point selection, later LLM-facing interfaces outperform earlier visual-token interfaces. Qwen3-VL improves from 72.90 R@1 with a pre-merger tap to 85.28 with a post-merger tap, and InternVL3.5 improves from 63.84 R@1 with a pre-projector tap to 74.92 with a post-projector tap.

**Table 10: Additional FRM ablations.** DeepStack correction does not improve Qwen3-VL identity accuracy. Downstream-facing taps outperform earlier visual-token taps. Smaller FRM configurations retain substantial retrieval gains.

| Configuration | Score |
|---|---|
| **DeepStack, Qwen3-VL, ID acc.↑** | |
| No FRM | 32.69 |
| FRM, DeepStack off | **37.31** |
| FRM, DeepStack on | 36.54 |
| Pristine ceiling | 43.08 |
| **Tap point, retrieval R@1↑** | |
| Qwen3-VL post-merger | **85.28** |
| Qwen3-VL pre-merger | 72.90 |
| InternVL3.5 post-projector | **74.92** |
| InternVL3.5 pre-projector | 63.84 |
| **Capacity, Qwen3-VL, retrieval R@1↑** | |
| $M{=}1,\ r{=}2$ (52.4M) | 83.42 |
| $M{=}1,\ r{=}4$ (78.6M) | 84.11 |
| $M{=}2,\ r{=}4$ (157.3M) | 84.79 |
| $M{=}4,\ r{=}4$ (314.6M) | **85.28** |

## NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction identify the benchmark, five evaluation tasks, main failure trends, and FRM's scope. The reported claims are supported by the corresponding detection, feature-recovery, and VLM experiments, and the text specifies that results are measured on TRACE with frozen host models.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses limitations related to synthetic generation, real-reference licensing, host-specific FRM training, severity generalization, and task dependence. All information is in Section 6. Additional dataset and implementation details are provided in the appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present theoretical results or formal proofs. The mathematical content defines the FRM architecture, training loss, and parameter overhead, and is used to specify the method rather than to state or prove a theorem.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper documents dataset construction, identity-disjoint splits, evaluation tasks, metrics, host architectures, feature-recovery objectives, and experimental protocols. Appendix B provides generation prompts and filtering procedures, Appendix C specifies evaluation and scoring, and Appendices D and G describe host-specific implementations and ablations.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We will open-source the dataset, annotations, prompts, metadata, evaluation scripts, and FRM training code. External real post-fire source images are not redistributed unless licenses permit it, and we provide links or identifiers where allowed.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The main paper (Table 5 and Section 3) and Appendix C specify train/validation/test splits, held-out object identities, task prompts, metrics, tap points, training loss, decoding settings, host architectures, and compute setup. Hyperparameters and host-specific implementation details are included in the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: We report a 95% confidence interval for the real-world detection improvement using bootstrap resampling over the 100 evaluated scenes. The interval excludes zero. Other experiments use fixed benchmark splits, with results reported across hosts, tasks, and degradation levels. We do not report repeated-training confidence intervals for every model.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper reports that experiments are run on NVIDIA A100 GPUs (Section 5.1) and provides approximate FRM training costs in terms of time with and without cached features. Appendix D reports host-specific model sizes, tap dimensions, and FRM parameter overhead.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work is intended to support post-disaster object understanding and does not involve human-subject experimentation. We use licensed or research-permissive external references where applicable and avoid redistributing external source images unless permitted.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses positive uses in search-and-rescue, forensic analysis, and insurance assessment. It also notes risks from synthetic data bias, incorrect automated assessments, and overreliance on model predictions in high-stakes settings (Section 6).

    Guidelines:

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

11. **Safeguards**

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

    Answer: [Yes]

    Justification: The release focuses on generated post-fire scenes, annotations, prompts, metadata, and evaluation code. Real post-fire references are not redistributed unless their licenses permit it, and generated candidates are filtered for realism and physical plausibility before inclusion.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper cites existing models, datasets, and baselines used in generation and evaluation, including Fire360, RF-DETR, CLIP, SigLIP2, Qwen, InternVL, Molmo, LLaVA, Restormer, WildIR, SFA, and DATR.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: TRACE is documented with subset statistics, object identities, scene coverage, material classes, severity levels, annotations, prompts, filtering protocol, and evaluation tasks. The release will include generated data, annotations, prompts, metadata, and code documentation.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or human-subject research. Dataset verification was performed by the authors rather than by external crowd workers.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The work does not involve human-subject research, participant recruitment, or crowdsourced annotation. Therefore, IRB approval is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: The paper describes the use of Gemini models for image generation and grounding-based filtering, VLMs for benchmark evaluation, and a GPT-5 judge fallback for unresolved paraphrases in structured scoring. These uses are part of the dataset construction and evaluation protocol, which is documented in the main paper and appendix.

    Guidelines:

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