TASK-FREE VISUAL-TOKEN REPAIR FOR FROZEN MULTIMODAL LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) rely on an internal interface that maps visual representations into the language model. When these representations are altered by quantization, compression, pruning, or other efficiency constraints, recovering downstream utility would ideally require optimizing them directly against the MLLM's downstream objective. However, this requires task-specific supervision and repeated forward/backward passes through a large MLLM, making end-to-end adaptation expensive and often impractical. We instead study task-free recovery at the visual-token interface. We introduce a lightweight repairer trained only from paired clean and corrupted visual representations, while keeping the visual encoder, multimodal projector, and language model frozen. Training requires neither task labels, prompts, language losses, nor LLM forward passes. Controlled perturbation experiments show that pointwise token fidelity is incomplete: representations with similar reconstruction error can induce substantially different downstream behavior, depending on their structure and direction. We evaluate this approach under practical corruption across Honeybee, LLaVA-v1.5-7B, and Qwen2.5-VL-7B. Under severe static 2-bit visual-token quantization, task-free repair recovers 81.8%, 72.5%, and 52.9% of the POPE degradation on Honeybee, LLaVA-v1.5, and Qwen2.5-VL, respectively, while objectives that preserve richer representation properties can outperform standard pointwise reconstruction. Learned image compression at approximately 0.155 bpp provides a second, upstream corruption regime: repair recovers downstream utility across all three architectures, while robust and structure-aware objectives outperform MSE in matched settings. Overall, our results show that lightweight, task-free visual-token repair provides a practical alternative to end-to-end MLLM adaptation and helps identify which representation properties should be preserved under different forms of visual corruption.

1 INTRODUCTION

Multimodal large language models (MLLMs) have emerged as powerful general-purpose systems for combining visual perception with language understanding and reasoning Alayrac et al. (2022); Li et al. (2023a); Liu et al. (2023); Bai et al. (2025). Despite substantial architectural diversity, many MLLMs share a common organization: a visual encoder maps an input image into a sequence of visual representations, a multimodal projector or connector adapts them to the language model embedding space, and a large language model (LLM) produces the final output conditioned on both visual and textual information. In many such systems, the LLM constitutes the largest and most expensive component, making repeated end-to-end adaptation of the complete multimodal stack costly.

At deployment time, however, the visual representation presented to a pretrained MLLM may differ substantially from the one for which the model was originally trained. Visual tokens may be pruned, merged, or compressed to reduce inference cost Shang et al. (2025); Li et al. (2025); they may be quantized to reduce storage or communication bandwidth; or they may be produced from a degraded or compressed image before entering the visual encoder Kao et al. (2025). Such scenarios arise naturally in resource-constrained and split-computing settings, where visual processing may take place separately from the much larger language model. Importantly, the corruption need not occur directly in token space: an altered input image can itself induce a shifted visual representation. In

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

all these cases, a common problem emerges: *can the resulting visual tokens be repaired before they are consumed by the frozen multimodal model?*

A direct solution would be to optimize the altered representation for the downstream objective of the MLLM itself. This would provide the most direct supervision for preserving task utility, but would require task-specific data and repeated forward and backward passes through a billion-parameter language model. Such optimization can be expensive or impractical, particularly when the LLM is remotely hosted, fixed, or otherwise unavailable for training. This issue also arises in prior work on bridging compressed image latents to MLLMs Kao et al. (2025), where avoiding optimization through the complete downstream MLLM was a central design requirement.

In this work, we ask whether visual representations can instead be repaired *without training through the LLM*. We introduce a lightweight residual repairer at the visual-token interface while keeping the visual encoder, multimodal projector, and language model frozen. The repairer is trained solely from paired clean and corrupted visual representations: it requires no task labels, prompts, language loss, or LLM forward/backward passes. We consider representations around the multimodal interface, including both pre- and post-projector tokens where appropriate, allowing the same formulation to cover corruptions originating directly in feature space as well as those induced upstream by degraded image inputs.

This task-free formulation raises a fundamental question: *what should a repaired visual representation preserve?* Pointwise reconstruction provides an obvious objective, but closeness in token space does not necessarily imply equivalent downstream behavior. Through controlled perturbations, we first show that visual representations with comparable reconstruction error can lead to markedly different downstream outcomes, depending on which tokens are altered, the direction of the perturbation, and the geometry of the resulting representation. These observations motivate repair objectives that go beyond blind pointwise reconstruction and preserve representation properties that are more closely related to downstream utility.

We evaluate this framework across Honeybee, LLaVA-v1.5-7B, and Qwen2.5-VL-7B-Instruct, considering both controlled interventions and practical corruption regimes. Under severe low-bit visual-token quantization, lightweight task-free repair substantially recovers downstream utility across all three architectures, with additional gains observed across several downstream benchmarks, while robust and structure-aware reconstruction objectives can outperform standard pointwise MSE. We further study learned image compression as a qualitatively different upstream source of token corruption. At approximately 0.155 bpp, task-free repair recovers downstream utility across all three architectures, and robust or structure-aware objectives improve over MSE in matched comparisons. Together, these results show that useful visual representations can be recovered without optimizing through the LLM, while highlighting the importance of preserving utility-relevant properties of the visual-token space.

Our main contributions are:

- We formulate *task-free visual-token repair*, in which a lightweight module restores corrupted visual representations while the entire MLLM remains frozen and no downstream supervision or LLM passes are required during training.
- We provide a controlled analysis showing that pointwise token fidelity alone is insufficient to characterize downstream utility, motivating robust and structure-aware repair objectives.
- We evaluate repair across three distinct MLLMs and practical corruption regimes, including severe visual-token quantization and learned image compression, showing recovery under both regimes and benefits from robust or structure-aware objectives over MSE in matched settings.

2 Related Work

Efficient visual interfaces for MLLMs. A growing line of work reduces the cost of multimodal inference by pruning, merging, or repacking visual tokens before they are processed by the language model Shang et al. (2025); Li et al. (2025). These methods primarily change which visual information is retained or how many tokens are propagated. Our focus is complementary: given an

already altered representation, we ask whether a lightweight module can restore utility at the frozen multimodal interface without training through the LLM.

Intermediate-feature compression and split inference. Split-computing systems transmit intermediate network features instead of raw inputs and often learn dedicated bottlenecks to reduce communication cost Shao & Zhang (2020); Matsubara et al. (2022). Such approaches commonly optimize feature compression jointly with a downstream task or tail network. Our setting instead keeps the downstream MLLM fixed and trains the repairer only from paired clean and corrupted visual representations. This also differs from prior work on bridging compressed image latents to MLLMs Kao et al. (2025): here the repair problem is defined directly at a generic MLLM visual interface and is studied across multiple architectures and corruption sources.

Representation matching and lightweight adaptation. Intermediate-representation matching has a long history in knowledge distillation, where student features are trained to reproduce teacher representations Romero et al. (2015). Parameter-efficient multimodal adaptation likewise introduces small trainable components to connect or adapt visual and language models Gao et al. (2023). Our repairer shares the use of a small trainable mapping, but differs in supervision and purpose: it learns from paired clean/corrupted representations, does not update the frozen MLLM, and is trained without language or task-specific objectives.

3 Problem Formulation

Multimodal interface. We consider an MLLM composed of a visual encoder E, a multimodal projector or connector P, and a language model L. Given an image x and a text prompt t, the standard inference pipeline can be written as

$$V = E(x), \qquad Z = P(V), \qquad y = L(Z, t), \tag{1}$$

where $V \in \mathbb{R}^{N \times d_v}$ denotes the visual representation produced by the encoder and $Z \in \mathbb{R}^{M \times d_\ell}$ denotes the representation passed to the language model. Depending on the architecture and corruption setting, we consider repair either before or after the multimodal projector. For compactness, we use R to denote the clean representation at the selected intervention point, with $R \in \{V, Z\}$.

Corrupted visual representations. At deployment time, the representation presented to the downstream multimodal stack may differ from the one produced by the original model. Let R denote the clean representation at the interface where repair is applied. Corruption may act directly on this representation or arise further upstream in the visual pipeline.

For corruption applied directly at the repair interface,

$$\widetilde{R} = C_R(R), \tag{2}$$

where C_R denotes a transformation in the selected representation space. More generally, let U be an upstream signal or representation and let $G_{U \to R}$ denote the frozen mapping from U to the repair interface. An upstream corruption C_U induces

$$\widetilde{U} = C_U(U), \qquad \widetilde{R} = G_{U \to R}(\widetilde{U}). \tag{3}$$

This formulation covers both image-level corruption ($U = x$) and token-space corruption introduced before a later repair point. For example, in LLaVA the 2-bit corruption is applied to pre-projector visual features while the repairer operates on the corresponding post-projector representation.

Our objective is to recover a representation that remains useful to the original downstream model without modifying that model itself. We introduce a lightweight repairer g_θ at the selected visual interface and apply its prediction through the residual correction

$$\widehat{R} = \widetilde{R} + \alpha \left(g_\theta(\widetilde{R}) - \widetilde{R} \right), \tag{4}$$

where $\alpha \in [0, 1]$ controls the repair strength. In particular, $\alpha = 0$ leaves the corrupted representation unchanged.

Figure 1 summarizes upstream and interface-level corruption together with the common repair formulation at the multimodal interface.

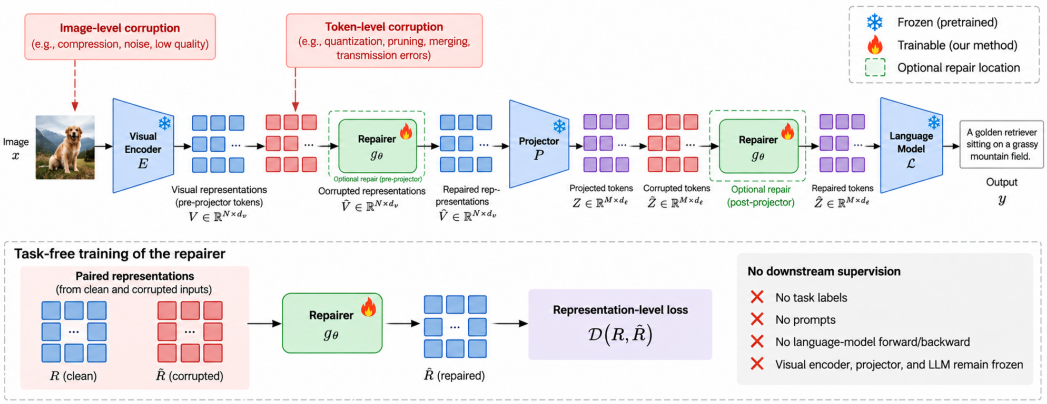


Figure 1: **Task-free visual-token repair.** Visual representations may be altered by image- or token-level corruption. A lightweight repairer operates before or after the multimodal projector while the underlying MLLM remains frozen, and is trained only from paired clean and corrupted representations without downstream supervision or LLM passes.

Task-free repair. A direct way to train g_θ would be to optimize its parameters through the complete MLLM using downstream supervision. For example, one could solve

$$\theta^\star_{\text{task}} = \arg\min_\theta \mathbb{E}\left[\mathcal{L}_{\text{task}}\big(L(\widehat{Z}, t), y\big)\right]. \tag{5}$$

While this directly optimizes downstream utility, it requires task labels, prompts, and repeated forward and backward passes through the language model. For large MLLMs, such adaptation can be computationally expensive and may be impossible when the language model is frozen, remotely hosted, or otherwise unavailable for training.

We instead study a *task-free* setting in which training uses only paired clean and corrupted visual representations. The visual encoder, projector, and language model remain frozen, and no language-model forward pass is required. The repairer is trained as

$$\theta^\star = \arg\min_\theta \mathbb{E}_{(R,\widetilde{R})}\left[\mathcal{D}\big(R, \widehat{R}\big)\right], \tag{6}$$

where $\mathcal{D}$ is a representation-level objective.

This formulation shifts the central question from whether corrupted tokens can simply be reconstructed to *which properties of the clean representation must be preserved to recover downstream utility*. Pointwise reconstruction, such as mean-squared error, is a natural baseline, but it treats all deviations in representation space equivalently. Our controlled analysis shows that this assumption is often insufficient: perturbations with similar magnitude can produce different downstream effects depending on their direction, token location, and representation structure. We therefore study both pointwise and structure-aware objectives, and subsequently evaluate their ability to recover performance under practical corruption regimes.

4 Task-Free Visual-Token Repair

Given a corrupted representation $\widetilde{R}$ at the selected multimodal interface, we learn a lightweight repairer g_θ using only paired clean and corrupted representations $(R, \widetilde{R})$. The visual encoder, multimodal connector, and language model remain frozen, and neither task labels nor language-model outputs are used during repair training.

4.1 Residual Repairer

The repairer is shared across token positions and consists of two residual MLP blocks,

$$h^{(\ell+1)} = h^{(\ell)} + W_2^{(\ell)}\sigma\Big(W_1^{(\ell)}\,\mathrm{LN}(h^{(\ell)}) + b_1^{(\ell)}\Big) + b_2^{(\ell)}, \tag{7}$$

where LN denotes LayerNorm and σ is GELU. The second projection in each block, including its bias, is zero-initialized, so that the complete repairer starts exactly at the identity. The repaired representation is then obtained through the residual mixing rule in Eq. 4.

The residual coefficient α in Eq. 4 controls the strength of the learned correction and is an inference-time scalar rather than a learned repairer parameter. Because representation scales and interfaces differ across architectures, a common value of α is not required. In a fully task-free deployment, it can be selected using only held-out clean and corrupted representations, without downstream scores or language-model outputs; $\alpha = 0$ leaves the corrupted representation unchanged. The operating value of α used in each experimental setting is reported with the corresponding results. With the architecture-specific repair dimensions used here, the two-block repairer contains 8.41M parameters for Honeybee, 8.41M for LLaVA-v1.5, and 7.36M for Qwen2.5-VL; detailed compute accounting is given in Appendix A.3.

4.2 Representation-level objectives

Our controlled analysis in Sec. 6.1 shows that representations with similar pointwise distortion can induce different downstream behavior. We therefore consider complementary constraints on coordinate-wise reconstruction, local token geometry, and channel statistics. All objectives operate only on paired visual representations and require no task-specific supervision or language-model execution.

Pointwise reconstruction. For $R, \widehat{R} \in \mathbb{R}^{N\times d}$, the standard reconstruction baseline is

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{Nd}\sum_{i,c}\left(\widehat{R}_{ic} - R_{ic}\right)^2. \tag{8}$$

We additionally consider the robust Smooth-L1 objective

$$\mathcal{L}_{\mathrm{rob}} = \frac{1}{Nd}\sum_{i,c}\rho_\beta\left(\widehat{R}_{ic} - R_{ic}\right), \tag{9}$$

where

$$\rho_\beta(e) = \begin{cases} e^2/(2\beta), & |e| < \beta, \\ |e| - \beta/2, & \text{otherwise.} \end{cases} \tag{10}$$

Local token relations. Let $s_{ij}(R)$ denote the cosine similarity between tokens i and j. For each anchor i, $\mathcal{N}_K(i)$ contains its $K_i = \min(K, N-1)$ most similar clean tokens, excluding itself. We preserve this local geometry through

$$\mathcal{L}_{\mathrm{rel}} = \frac{1}{N}\sum_i \frac{1}{K_i}\sum_{j\in\mathcal{N}_K(i)}\left[s_{ij}(\widehat{R}) - s_{ij}(R)\right]^2. \tag{11}$$

Neighbor sets are defined from the clean representation and kept fixed during optimization.

Channel-statistic calibration. Let μ_c and σ_c denote the mean and population standard deviation of channel c across visual tokens. We define

$$\begin{aligned}\mathcal{L}_{\mathrm{cal}} = \frac{1}{d}\sum_c \Bigg[&\left(\frac{\mu_c(\widehat{R}) - \mu_c(R)}{\mathrm{sg}(\max(\sigma_c(R), \epsilon))}\right)^2 \\ &+ \left(\log\max(\sigma_c(\widehat{R}), \epsilon) - \log\max(\sigma_c(R), \epsilon)\right)^2\Bigg],\end{aligned} \tag{12}$$

where sg denotes stop-gradient and $\epsilon > 0$ is a stabilization constant.

Structured repair objective. We combine the complementary representation constraints as

$$\mathcal{L}_{\mathrm{struct}} = \mathcal{L}_{\mathrm{rob}} + \lambda_{\mathrm{rel}}\mathcal{L}_{\mathrm{rel}} + \lambda_{\mathrm{cal}}\mathcal{L}_{\mathrm{cal}}. \tag{13}$$

Where visual self-attention is naturally available, we additionally consider an attention-aware variant that reweights pairwise relation terms according to architecture-specific visual-token importance. Exact attention extraction, alignment, and weighting procedures are provided in Appendix C. Experimental hyperparameters are specified in Sec. 5.2.

Table 1: **MLLM corruption and repair interfaces.** "Pre" and "post" refer to the architecture-specific multimodal connector. Qwen2.5-VL uses dynamic resolution.

Model (connector)	Corruption	Repair
Honeybee (D-Abstractor)	pre: 256×1024	post: 144×4096
LLaVA-v1.5 (MLP)	pre: 576×1024	post: 576×4096
Qwen2.5-VL (PatchMerger)	pre: $N\times1280$	post: $(N/4)\times3584$

5 Experimental Setup

5.1 Models and repair interfaces

We evaluate three MLLMs with distinct visual-language interfaces: Honeybee Cha et al. (2024), LLaVA-v1.5-7B Liu et al. (2024), and Qwen2.5-VL-7B-Instruct Bai et al. (2025). Table 1 distinguishes the representation at which the 2-bit corruption is injected from the representation on which the repairer is trained and applied.

Corruption and repair need not occur at the same representation. In the quantization experiments, visual features are corrupted before the architecture-specific multimodal connector, while the repairer operates on the corresponding post-connector representation. For Honeybee this maps 256×1024 pre-D-Abstractor features to 144×4096 repair tokens; for LLaVA-v1.5, 576×1024 pre-projector features to 576×4096 repair tokens; and for Qwen2.5-VL, $N \times 1280$ pre-PatchMerger features to $(N/4) \times 3584$ repair tokens. Image compression acts still further upstream, at the input image, and the induced shift is repaired at the same architecture-specific repair interface. Exact tensor shapes and attention alignment are provided in Appendix A.

5.2 Repair training protocol

All repairers use bottleneck dimension $d_b = 512$ and are trained for 30 epochs with AdamW, learning rate 10^{-4}, weight decay 0, batch size 4, and gradient clipping at 1. Training uses 512 COCO train2014 images Lin et al. (2014), split into 448 training and 64 representation-validation images. Final repair configurations use seeds $\{0, 1, 2\}$ unless noted otherwise, and repaired scores are averaged across these runs. For variable-length Qwen representations, losses are computed independently for each image before batch averaging.

For the structured objective, we set $\beta = 0.5$, $K = 16$, $\epsilon = 0.05$, $\lambda_{\text{rel}} = 0.25$, and $\lambda_{\text{cal}} = 0.1$. These values are kept fixed across architectures and downstream benchmarks. For the static 2-bit quantization experiments, we additionally fix the residual repair strength to $\alpha = 0.5$ across models, objectives, seeds, and downstream tasks.

5.3 Corruption regimes

Controlled diagnostic perturbations. To study whether pointwise token fidelity is sufficient to characterize downstream utility, we use two controlled perturbation families on LLaVA-v1.5 visual tokens. First, we replace 75% of the 576 pre-projector patch tokens with either zeros, the clean image-wise token mean, or an all-ones vector. Second, we construct a rank-64 PCA subspace from visual tokens of 512 disjoint COCO images and compare Gaussian perturbations projected onto this subspace or its orthogonal complement. We evaluate matched relative Frobenius norms $\rho \in \{0.20, 0.40\}$; the main-text comparison below uses $\rho = 0.40$. Full construction details are given in Appendix E.

2-bit visual-token quantization. Our primary practical token-space corruption is static symmetric per-channel 2-bit quantization applied immediately before the architecture-specific multimodal connector. Per-channel scales are calibrated from the absolute 0.999 quantile of 32,768 visual tokens from disjoint COCO calibration images. The implementation uses three effective reconstruction levels, $\{-s_d, 0, +s_d\}$, with no zero-point. The resulting corrupted features are then propagated to the repair interface listed in Table 1. Packed 2-bit codes correspond to an $8\times$ theoretical payload reduction relative to FP16. Exact calibration and quantize–dequantize equations are given in Appendix B.

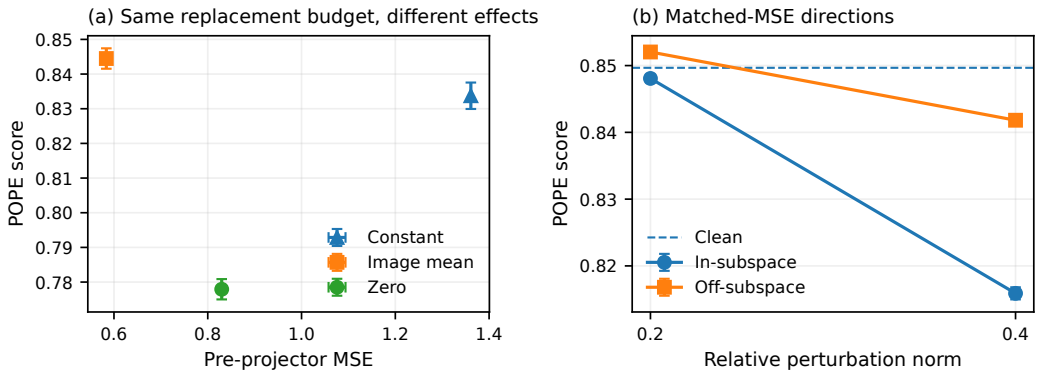


Figure 2: **Token-space fidelity does not uniquely determine downstream utility.** (a) Different replacements under the same 75% token budget yield a non-monotonic relationship between pre-projector MSE and POPE performance. (b) In- and off-subspace perturbations at matched relative norms $\rho \in \{0.20, 0.40\}$ yield different downstream scores; the main-text comparison reports the $\rho = 0.40$ condition. Points show mean $\pm$ standard deviation over three perturbation seeds.

Learned image compression. As a qualitatively different upstream corruption, we compress the input image before visual encoding using the CompressAI `bmshj2018_hyperprior` implementation of the scale-hyperprior codec (Ballé et al., 2018), at quality level 1. This operating point yields an aggregate bitrate of approximately 0.15486 bits per pixel (bpp). Clean and compressed versions of the same 512 COCO training images provide paired representations for repair training; the visual encoder, connector, and LLM remain frozen. Additional compression-specific protocol details are given in Appendix F.

5.4 Evaluation and Statistics

POPE Li et al. (2023b) is our primary cross-model benchmark and the only benchmark used for the controlled three-model comparison. We additionally use MME Fu et al. (2023), SEED-Bench Li et al. (2024), ScienceQA-IMG Lu et al. (2022), and MMMU Yue et al. (2024) where supported; benchmark coverage therefore differs by architecture. The 500 POPE evaluation images are disjoint from repair-training images, and prompts, decoding, splits, and scoring are held fixed across clean, corrupted, and repaired conditions. For aligned predictions we use paired bootstrap resampling over images. We report downstream scores together with the fraction of corruption-induced degradation recovered, $(S_{\text{repair}} - S_{\text{raw}}) / (S_{\text{clean}} - S_{\text{raw}})$. Values above one are interpreted as returning to the clean operating range, not as evidence that corruption improves the model. Full confidence intervals, seed-level results, and bootstrap details are reported in Appendix D.

6 Experimental Results

6.1 Controlled Diagnostics: Token Fidelity is Incomplete

Figure 2 reports the controlled perturbation experiments introduced in Sec. 5.3. Under the fixed 75% token-replacement budget, image-mean replacement gives the lowest pre-projector MSE (0.584) and a POPE score of 0.844, whereas zero replacement has higher MSE (0.830) but substantially lower performance (0.778). All-ones replacement has the largest MSE (1.361) while retaining a POPE score of 0.834. Thus, lower pointwise distortion does not imply higher downstream utility.

The principal-subspace diagnostic leads to the same conclusion under matched perturbation magnitude. At $\rho = 0.40$, in- and off-subspace perturbations each have Frobenius norm $0.40\|V\|_F$ and produce essentially identical pre-projector MSE (0.1771), yet their mean POPE scores differ by about 2.6 points (0.8159 versus 0.8418 over three perturbation seeds). Hence, downstream behavior depends not only on perturbation magnitude but also on its direction relative to the geometry of

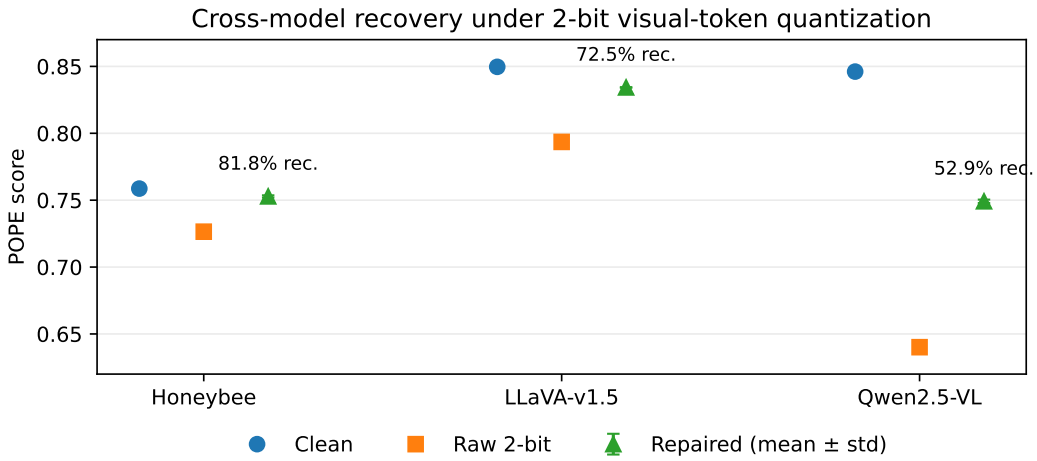


Figure 3: **Task-free repair under severe 2-bit visual-token quantization.** POPE performance for clean, raw quantized, and repaired representations. Repaired points show mean ± standard deviation over three independently trained repairers; clean and raw conditions are fixed references. Labels report the fraction of raw corruption-induced degradation recovered. The structured relation weighting is architecture-specific: attention-derived for Honeybee/LLaVA and uniform for the reported Qwen multi-seed configuration.

the visual-token space. The lower $\rho = 0.20$ condition is shown in Fig. 2 and fully documented in Appendix E.

6.2 Task-free repair under 2-bit visual-token quantization

Figure 3 shows that severe 2-bit quantization reduces POPE from 0.759 to 0.726 for Honeybee, 0.850 to 0.794 for LLaVA-v1.5, and 0.846 to 0.640 for Qwen2.5-VL. Structured repair raises the scores to 0.753, 0.834, and 0.749, respectively, recovering 81.8%, 72.5%, and 52.9% of the raw degradation. Repaired scores are averaged over three independently trained repairers, with standard deviation across training seeds shown as error bars in Fig. 3; paired-bootstrap intervals for the primary POPE gains exclude zero. Full intervals and seed-level values are given in Appendix D.

The cross-model comparison validates the repair framework rather than a single unchanged loss instantiation. Honeybee and LLaVA use attention-derived structured relation weighting, whereas the reported Qwen2.5-VL multi-seed configuration uses uniform relation weighting. The later Qwen attention-edge variant is analyzed separately in Appendix D.4.

The representation objective also matters within this otherwise fixed repair setting. On LLaVA-v1.5 POPE, the raw quantized model scores 0.794; MSE repair reduces performance to approximately 0.660, Smooth-L1 reaches approximately 0.823, and the structured objective attains the highest mean score among the evaluated objectives at 0.834. The structured objective improves over Smooth-L1 by $+0.0112$ POPE points, with a paired 95% confidence interval of $[+0.0073, +0.0152]$. Together with the cross-model recovery above, this supports using robust and structure-aware constraints rather than relying on pointwise MSE alone. A separate exploratory Qwen2.5-VL seed-0 comparison shows that the relative gain of individual objective components can vary across downstream tasks. Full objective comparisons and additional cross-task quantization results are reported in Appendix D.4 and Appendix D.3, respectively. Exploratory LLaVA-v1.5 seed-0 transfer results on MME and ScienceQA are reported separately in Appendix D.2; their paired confidence intervals currently include zero, so they are not used as primary evidence.

6.3 Repair under learned image compression

We next test whether the same representation-only training principle extends to an upstream corruption that acts before visual encoding. We use the CompressAI `bmshj2018_hyperprior`

Table 2: **Representative recovery under learned image compression** with CompressAI `bmshj2018_hyperprior` quality 1 (≈ 0.155 bpp). Honeybee uses MSE at $\alpha = 0.1$; LLaVA uses normalized Smooth-L1 with shrinkage at $\alpha = 0.1$; Qwen uses the same robust objective at $\alpha = 1.0$, reported as the mean over three repairer-training seeds. Recovery is the fraction of the clean-to-compressed degradation restored.

Model	Task	Clean	Compressed	Repaired	Recovery
Honeybee	MME	1536.1	1476.3	1502.7	44.1%
LLaVA-v1.5	MME	1872.9	1782.7	1809.4	29.7%
Qwen2.5-VL	MME	2322.6	2232.1	2257.9	28.5%

codec at quality level 1, corresponding to approximately 0.155 bpp. Repairers are trained only from paired clean and compressed visual representations, with the full MLLM frozen and no task labels, prompts, or language-model execution during training. For compression we additionally consider a channel-normalized Smooth-L1 objective, obtained by scaling coordinate errors by clean training-set channel statistics before the robust penalty in Eq. 9. A shrinkage-regularized variant further discourages correction energy that is large relative to the observed clean-to-compressed representation shift.

Repair strength is applied only at inference and need not be shared across architectures. We use $\alpha = 0.1$ for Honeybee and LLaVA-v1.5 and a stronger $\alpha = 1$ setting for the Qwen2.5-VL MME result; in a fully task-free deployment, this scalar can instead be chosen from held-out paired representations using a representation-level validation criterion. These choices do not alter the task-free training of the repairer parameters.

Table 2 summarizes representative recovery results. At $\alpha = 0.1$, MSE repair recovers 44.1% of the compression-induced MME degradation on Honeybee. On LLaVA-v1.5, normalized Smooth-L1 with shrinkage reaches 1809.4 MME, recovering 29.7% of the degradation compared with 19.0% for MSE; the structured objective reaches 28.4% recovery and exceeds MSE by 8.50 MME points at the same repair strength. On Qwen2.5-VL, normalized Smooth-L1 with shrinkage at $\alpha = 1$ reaches a mean score of 2257.9 over three repairer-training seeds, recovering 28.5% of the clean-to-compressed gap. A matched-strength Qwen ablation further shows structured repair exceeding MSE by 21.58 MME points at $\alpha = 0.5$.

A conservative Qwen POPE setting also improves the compressed baseline: channel-normalized Smooth-L1 at $\alpha = 0.1$ raises the score from 0.8042 to 0.8064, with a paired 95% bootstrap interval of $[+0.0011, +0.0033]$ for the repair gain. Together, these results show that task-free visual-token repair extends from direct token quantization to learned image compression, while robust and structure-aware objectives can provide additional gains over pointwise MSE. Full objective comparisons, seed-level Qwen results, and repair-strength ablations are reported in Appendix F.

7 Conclusion

We studied task-free visual-token repair for frozen multimodal LLMs. Controlled perturbations show that pointwise token fidelity does not fully characterize downstream utility, motivating objectives that also preserve robust coordinate-wise behavior, channel statistics, and local token geometry. Under severe 2-bit visual-token quantization, a lightweight repairer trained only from paired visual representations recovers substantial downstream performance across three architecturally distinct MLLMs, without task labels or language-model training. Robust and structure-aware reconstruction is generally effective in this setting, while the relative contribution of individual objective components can vary across tasks and architectures. The same repair formulation also recovers downstream utility under upstream learned image compression at approximately 0.155 bpp across the three evaluated architectures, with robust and structure-aware objectives outperforming MSE in matched settings.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript editing, code inspection, and figure preparation during development. All AI-assisted text, analyses, and artifacts were reviewed and verified by the authors, who take responsibility for the final content of the paper.

ETHICS STATEMENT

This work introduces no new human-subject study or user-data collection. The experiments analyze existing multimodal models and established evaluation datasets. The proposed repair mechanism may reduce resource requirements for MLLM deployment but does not change the underlying model's safety properties; those remain those of the frozen downstream system.

REPRODUCIBILITY STATEMENT

Sections 4 and 5 specify the repair method, training protocol, corruption regimes, and evaluation procedure. Appendix A gives architecture-specific implementation details, Appendix B specifies the quantizer and payload accounting, Appendix E defines the controlled perturbations, and Appendix D reports seed-level and uncertainty details. Code and configuration files will be included in the supplementary material. Appendix F gives learned-compression details and additional results

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, 2022.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13817–13827, 2024.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. LLaMA-Adapter V2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.

Chia-Hao Kao, Cheng Chien, Yu-Jen Tseng, Yi-Hsin Chen, Alessandro Gnutti, Shao-Yuan Lo, Wen-Hsiao Peng, and Riccardo Leonardi. Bridging compressed image latents and multimodal large language models. In *International Conference on Learning Representations*, 2025.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. SEED-Bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13299–13308, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, 2023a.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. TokenPacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, 133:6794–6812, 2025.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, 2023b.

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*, pp. 740–755, 2014.

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

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, 2022.

Yoshitomo Matsubara, Ruihan Yang, Marco Levorato, and Stephan Mandt. Supervised compression for resource-constrained edge computing systems. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2685–2695, 2022.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *International Conference on Computer Vision*, 2025.

Jiawei Shao and Jun Zhang. Bottlenet++: An end-to-end approach for feature compression in device-edge co-inference systems. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2020. doi: 10.1109/ICCWorkshops49005.2020.9145068.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024.

A Additional Method and Implementation Details

A.1 Visual interfaces and repair points

The three evaluated MLLMs expose substantially different visual interfaces. Honeybee follows `pixels` → `vision encoder` → `pre-D-Abstractor features` → `D-Abstractor` → `bridge tokens` → `LLM`. Token-space corruption is applied to the 256×1024 pre-D-Abstractor representation. The corrupted features are then passed through the frozen D-Abstractor, and the repairer operates on the resulting 144×4096 post-D-Abstractor bridge-token representation, using the corresponding clean post-D-Abstractor features as targets.

LLaVA-v1.5-7B follows `pixels` → `CLIP vision tower` → `patch features` → `MLP projector` → `bridge tokens` → `LLM`. At the standard evaluation resolution, quantization is applied to 576 pre-projector patch tokens of dimension 1024. These corrupted features are then passed through the frozen MLP projector, and the repairer operates on the resulting 576 post-projector tokens of dimension 4096. Thus the relevant path is

$$V \to \widetilde{V} \to P(\widetilde{V}) \to g_\theta(P(\widetilde{V})), \tag{14}$$

Table 3: **Repairer overhead.** Counts use the architecture-specific repair interfaces in Table 1. Qwen2.5-VL uses the mean post-PatchMerger POPE token count of 352.842. These are analytical operation counts, not measured latencies.

Model	Repair d	Params	MACs/token	MACs/image
Honeybee	4096	8.414M	8.389M	1.208G
LLaVA-v1.5-7B	4096	8.414M	8.389M	4.832G
Qwen2.5-VL-7B	3584	7.363M	7.340M	2.590G

with the representation loss comparing the repaired output against the clean post-projector target $P(V)$. This distinction is important: the repairer need not invert a corruption in the same feature space in which it was introduced.

Qwen2.5-VL-7B-Instruct uses a visual transformer followed by a PatchMerger. Because the model uses dynamic resolution, the pre-PatchMerger representation has a variable number N of tokens with feature dimension 1280. Token-space corruption is applied at this pre-PatchMerger interface. The corrupted features are then passed through the frozen PatchMerger, and the repairer operates on the post-PatchMerger representation of shape $(N/4) \times 3584$, using the corresponding clean post-PatchMerger representation as target.

For image-level corruption, the input image is transformed before visual encoding. The resulting representation shift is then repaired at the architecture-specific repair interface while the downstream multimodal stack remains frozen.

A.2 Repairer architecture and optimization

All repairers consist of two token-wise residual MLP blocks shared across token positions, with bottleneck dimension 512. Each block applies LayerNorm, a linear projection, GELU, and a second linear projection. The weight and bias of the second projection are initialized to zero, making the complete repairer exactly identity-initialized.

Repairers are trained for 30 epochs with AdamW, learning rate 10^{-4}, weight decay 0, batch size 4, and gradient clipping at 1. Training uses 512 COCO train2014 images, split into 448 repair-training and 64 representation-validation images. No downstream labels, prompts, answers, language-model losses, or LLM forward/backward passes are used during repair training.

A.3 Repairer size and compute

For repair-interface dimension d and bottleneck dimension d_b, each residual block contains LayerNorm parameters, two affine projections, and their biases. With two blocks, the total parameter count is

$$4dd_b + 2d_b + 6d. \tag{15}$$

For $d_b = 512$, this gives 8.414M trainable parameters for Honeybee ($d = 4096$), 8.414M for LLaVA-v1.5 ($d = 4096$), and 7.363M for Qwen2.5-VL ($d = 3584$). For the 7B-scale models, the repairer therefore contains at most 8.414M trainable parameters, approximately 0.12% of a 7B-parameter backbone. The dominant linear-layer cost is $4dd_b$ multiply–accumulate operations (MACs) per repaired token; LayerNorm, GELU, bias, and residual-add costs are not included in this count. Table 3 summarizes the resulting parameter and analytical compute overhead.

We do not report a latency speedup because the current experiments simulate the repairer and quantize–dequantize operations without an optimized packed-kernel or communication implementation. The table therefore characterizes model and arithmetic overhead rather than wall-clock deployment latency.

A.4 Loss reduction and numerical details

Representation losses are first computed independently for each image and then averaged uniformly across images. This is particularly important for Qwen2.5-VL, whose dynamic-resolution inputs

produce variable token counts, and prevents examples with more visual tokens from receiving greater weight.

Channel standard deviations use the population convention (`unbiased=False`). Cosine-similarity denominators are clamped by 10^{-8}, and the clean top-K neighbor sets in Eq. 11 remain fixed during optimization.

For the low-bit quantization experiments, we use

$$K = 16, \qquad \beta = 0.5, \qquad \epsilon = 0.05, \qquad \lambda_{\mathrm{rel}} = 0.25, \qquad \lambda_{\mathrm{cal}} = 0.1.$$

These values are kept fixed across architectures and downstream benchmarks. The residual repair strength is also fixed to $\alpha = 0.5$ for all static 2-bit quantization experiments.

B Static 2-Bit Visual-Token Quantization

B.1 Calibration and quantize–dequantize rule

Our primary token-space corruption is static symmetric per-channel quantization with a 2-bit code budget. Calibration uses 512 COCO train2014 images. We deterministically sample 64 visual tokens per image, giving 32,768 calibration tokens in total.

For each feature channel d, let

$$a_d = Q_{0.999}(|x_d|), \tag{16}$$

where the quantile is evaluated over the sampled calibration activations. The channel scale is

$$s_d = \max\left(\frac{a_d}{q_{\max}}, 10^{-8}\right), \qquad q_{\max} = 2^{b-1} - 1. \tag{17}$$

For $b = 2$, $q_{\max} = 1$. At inference time, each activation is quantized and immediately dequantized as

$$q_d = \mathrm{clip}\left(\mathrm{round}(x_d/s_d), -1, 1\right), \qquad \widetilde{x}_d = q_d s_d. \tag{18}$$

No zero-point is used. Consequently, the three effective reconstruction levels for channel d are

$$\{-s_d,\, 0,\, +s_d\}.$$

Thus, while the storage budget assigns 2 bits per code, the implemented symmetric quantizer has three effective reconstruction levels. We use "INT2" only as a compact label for this 2-bit setting in figures and result tables.

The experiments simulate quantize–dequantize in floating-point execution. They do not implement physical bit packing, network transmission, or measured communication throughput.

B.2 Theoretical payload reduction

Ignoring the static scale metadata, replacing FP16 activations by packed 2-bit codes reduces the representation payload by

$$\frac{16}{2} = 8\times, \tag{19}$$

corresponding to an 87.5% theoretical reduction.

Table 4 reports the corresponding representation sizes. For Qwen2.5-VL, whose token count varies across images, we report the mean token count observed on POPE. Per-channel scales are static and shared across images, so their storage cost is normally amortized.

The checkpoint stores the static scales in FP32, giving 4 KiB of scale metadata for the 1024-channel Honeybee/LLaVA interfaces and 5 KiB for the 1280-channel Qwen2.5-VL interface. Because these scales are shared across images, their cost is normally amortized. Even if they are conservatively charged to every image, the effective reductions are approximately $7.53\times$ for Honeybee, $7.78\times$ for LLaVA-v1.5, and $7.91\times$ for Qwen2.5-VL. These numbers quantify representation storage only and should not be interpreted as measured end-to-end system speedups.

Table 4: **Theoretical representation payload under 2-bit quantization.** Sizes assume packed 2-bit codes. Scale metadata are stored in FP32 by the current checkpoints and are static and shared across images. Qwen2.5-VL uses its mean POPE token count.

Model	Representation	FP16	2-bit codes	Scales
Honeybee	256×1024	512 KiB	64 KiB	4 KiB
LLaVA-v1.5	576×1024	1.125 MiB	144 KiB	4 KiB
Qwen2.5-VL	1411.368×1280	3.446 MiB	441.1 KiB	5 KiB

Table 5: **Recovery under 2-bit visual-token quantization on POPE.** Recovery is measured relative to the degradation of the raw quantized representation from the clean model.

Model	Clean	Raw 2-bit	Repaired	Recovery (%)
Honeybee	0.7587	0.7264	0.7528	81.8
LLaVA-v1.5	0.8497	0.7936	0.8340	72.5
Qwen2.5-VL	0.8461	0.6401	0.7491	52.9

C Attention-Aware Relation Weighting

For architectures for which a visual-token importance distribution a_i can be aligned to the repair representation, we reweight the relation-preservation term. For anchor-weighted relations,

$$\mathcal{L}_{\text{rel}}^{\text{CLS}} = \sum_i a_i \frac{1}{K_i} \sum_{j \in \mathcal{N}_K(i)} \left[s_{ij}(\widehat{R}) - s_{ij}(R) \right]^2, \qquad \sum_i a_i = 1. \tag{20}$$

For the attention-edge variant, each selected directed relation receives

$$w_{ij} = \frac{a_i + a_j}{2}, \tag{21}$$

and the relation loss becomes

$$\mathcal{L}_{\text{rel}}^{\text{attn}} = \frac{\sum_i \sum_{j \in \mathcal{N}_K(i)} w_{ij} \left[s_{ij}(\widehat{R}) - s_{ij}(R) \right]^2}{\max\left(\sum_i \sum_{j \in \mathcal{N}_K(i)} w_{ij}, 10^{-8} \right)}. \tag{22}$$

For LLaVA-v1.5, token importance is obtained from normalized last-layer CLS-to-patch visual attention. Because the MLP projector preserves the 576 patch positions, these weights are aligned one-to-one with the post-projector repair tokens. For Qwen2.5-VL, which has no visual CLS token at the repair interface, incoming attention from the final full-attention vision block is aggregated through the 2×2 PatchMerger groups before constructing symmetric edge weights. The original Qwen2.5-VL multi-seed experiment reported in the main paper instead uses uniform relation weighting.

For Honeybee, the structured configuration used in the reported experiments uses visual attention information aligned to the architecture-specific token representation. The corresponding extraction and alignment are performed entirely within the visual stack and require no language-model execution.

D Additional 2-Bit Quantization Results

D.1 Cross-model POPE recovery

Table 5 reports the primary cross-model POPE result numerically. Repaired values are means over three independently trained repairers with seeds 0, 1, and 2.

The paired-bootstrap confidence interval for the repair gain over the raw quantized condition is $[0.0154, 0.0370]$ for Honeybee and $[0.0347, 0.0467]$ for LLaVA-v1.5. For Qwen2.5-VL, each of the

Table 6: **Exploratory LLaVA-v1.5 seed-0 transfer under 2-bit quantization.** The repairer uses the same structured configuration and $\alpha = 0.5$ as the main POPE experiment. Confidence intervals refer to the paired repair gain over the raw quantized condition.

Benchmark	Clean	Raw 2-bit	Repaired	Recovery (%)
MME	1786.52	1573.44	1619.37	21.6
ScienceQA	0.6786	0.6656	0.6718	47.3
ScienceQA-IMG	0.6644	0.6371	0.6500	47.3

Table 7: **Additional results under 2-bit visual-token quantization.** Values are mean downstream scores where multi-seed results are available. A repaired score slightly above the clean point estimate is interpreted as recovery to the clean operating range rather than evidence that corruption improves the model.

Model	Benchmark	Clean	Raw 2-bit	Structured
Honeybee	POPE	0.7587	0.7264	0.7528
	SEED	0.4360	0.3365	0.4413
Qwen2.5-VL	POPE	0.8461	0.6401	0.7491
	MME	2322.6	1903.6	1867.2
	SEED-IMG	0.7800	0.6897	0.7147
	ScienceQA	0.8810	0.7650	0.7895

three repairer training seeds improves over the raw condition by approximately 0.108–0.110 POPE points, and the corresponding paired-bootstrap intervals exclude zero.

These bootstrap intervals quantify uncertainty over aligned evaluation examples, whereas variation across repairer seeds reflects training stochasticity. We report both sources separately rather than treating the number of evaluation examples as additional repairer-training runs.

D.2 Exploratory LLaVA cross-task transfer

Additional LLaVA-v1.5 evaluations are currently available for the structured repairer from training seed 0. They are reported separately from the main multi-seed POPE result because the corresponding seed-1 and seed-2 downstream evaluations have not yet been completed. Table 6 summarizes these exploratory transfer results.

All three point estimates improve over the raw quantized condition, but the paired-bootstrap 95% confidence intervals for the gains include zero: MME $+45.93$ with CI approximately $[-22.03, 115.06]$, ScienceQA overall $+0.00613$ with CI approximately $[-0.00047, 0.01250]$, and ScienceQA-IMG $+0.01289$ with CI approximately $[-0.00050, 0.02628]$. We therefore treat these as exploratory transfer results rather than multi-seed confirmation.

D.3 Cross-task behavior

Table 7 reports additional downstream benchmarks. The Qwen2.5-VL column labeled "Structured" corresponds to the original multi-seed uniform structured objective used for the main cross-model result.

Honeybee structured repair returns SEED performance to the clean operating range. On Qwen2.5-VL, the uniform structured objective improves POPE, SEED-IMG, and ScienceQA over the raw quantized representation but does not improve MME. Thus, the structured family provides useful recovery across several benchmarks, while the size of the gain remains task- and architecture-dependent.

D.4 Objective dependence

Figure 4 reports the objective comparison omitted from the main paper for space. Panel (a) shows the multi-seed LLaVA-v1.5 POPE experiment: MSE repair is harmful under severe 2-bit quantiza-

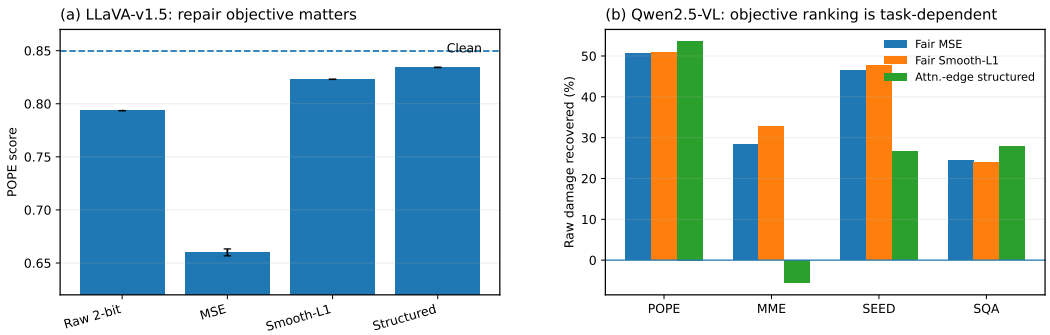


Figure 4: **Repair effectiveness depends on the representation objective.** (a) LLaVA-v1.5 POPE under severe 2-bit quantization. MSE repair is harmful, whereas Smooth-L1 and structured repair recover substantial utility; error bars show standard deviation over three repairer-training seeds. (b) Exploratory Qwen2.5-VL seed-0 comparison using image-balanced objectives. The relative effectiveness of pointwise and structured objectives varies across downstream tasks.

tion, whereas Smooth-L1 and structured repair recover substantial utility. Direct paired comparison shows that structured repair exceeds Smooth-L1 by $+0.0112$ POPE points, with a 95% confidence interval of $[+0.0073, +0.0152]$.

Panel (b) is a separate exploratory Qwen2.5-VL seed-0 experiment in which all objectives use the fair image-balanced reduction. In this comparison, the attention-edge structured variant has the highest observed POPE and ScienceQA scores among the evaluated objectives, whereas pointwise objectives have higher observed scores on MME and SEED-IMG. We therefore use the comparison to characterize how the benefit of individual objective components varies across downstream tasks.

The original Qwen2.5-VL multi-seed result in Table 5 uses uniform relation weighting. The attention-edge configuration in Fig. 4 was evaluated later under the fair image-balanced protocol and should therefore not be interpreted as the same experimental configuration.

On MMMU multiple choice, the available differences among the raw, fair MSE, fair Smooth-L1, and structured conditions are small and their confidence intervals overlap. We therefore do not draw an objective-specific conclusion from that benchmark.

E Controlled Diagnostic Details

E.1 Fixed-budget token replacement

The replacement diagnostic is performed on the 576 LLaVA-v1.5 pre-projector patch tokens. Given $V \in \mathbb{R}^{N \times D}$, we sample

$$m = \mathrm{round}(0.75N) \tag{23}$$

token indices uniformly without replacement. With $N = 576$, exactly 432 tokens are replaced. An eventual CLS token is excluded from the candidate set.

For each selected token i, we consider three replacement rules:

$$V_i' = \begin{cases} 0, & \text{zero replacement,} \\ \dfrac{1}{N}\displaystyle\sum_{j=1}^{N} V_j, & \text{image-mean replacement,} \\ \mathbf{1}, & \text{constant replacement.} \end{cases} \tag{24}$$

The image mean is computed from all clean patch tokens of the same image, and the constant replacement value is 1.0. Token selection is generated deterministically from the perturbation seed and sample index, allowing aligned comparison across replacement rules.

E.2 PRINCIPAL-SUBSPACE PERTURBATIONS

The PCA diagnostic uses 512 COCO train2014 images disjoint from the POPE evaluation set. Each image contributes 576 LLaVA visual tokens of dimension 1024, yielding

$$512 \times 576 = 294{,}912$$

tokens. After subtracting the global token mean μ, we estimate a low-rank PCA basis and retain the leading $r = 64$ directions $U_r \in \mathbb{R}^{1024\times 64}$.

For each evaluated image, a Gaussian perturbation E is sampled once and decomposed into components parallel and orthogonal to this principal subspace:

$$E_{\|} = EU_rU_r^{\top}, \qquad E_{\perp} = E - E_{\|}. \tag{25}$$

The same underlying E is used for the two conditions. Each component is then normalized independently on a per-image basis:

$$\widetilde{E}_c = \frac{E_c}{\max(\|E_c\|_F, 10^{-8})}\rho\|V\|_F, \qquad c \in \{\|, \perp\}, \tag{26}$$

with $\rho \in \{0.20, 0.40\}$. The main-text numerical comparison uses $\rho = 0.40$. The perturbed representations are

$$V' = V + \widetilde{E}_{\|} \qquad \text{or} \qquad V' = V + \widetilde{E}_{\perp}.$$

The PCA mean is used when fitting the basis, but is not subtracted from V during the perturbation itself: the projected quantity is the displacement vector E, not the representation V. This construction matches perturbation magnitude while changing only its orientation relative to the principal visual subspace. Figure 2 reports both relative-norm conditions; the numerical comparison highlighted in Sec. 6.1 uses $\rho = 0.40$.

The controlled studies are intended as diagnostics rather than practical deployment corruptions. Their role is to isolate the observation that pointwise distortion magnitude alone does not determine downstream utility.

F LEARNED IMAGE COMPRESSION DETAILS AND ADDITIONAL RESULTS

Codec and protocol. We use the CompressAI `bmshj2018_hyperprior` learned image codec at quality level 1, applied to the RGB image before visual encoding. Across the cached compression set, this operating point yields an aggregate bitrate of approximately 0.15486 bpp. Repair training uses the same 512 disjoint COCO train2014 images as the other practical-corruption experiments and requires only paired clean/compressed representations. The visual encoder, multimodal connector, and LLM remain frozen.

In addition to MSE and the structured objective defined in Sec. 4.2, the compression study evaluates robust cached-token variants of the same Smooth-L1 penalty. *Normalized Smooth-L1* first scales each feature dimension by its clean training-set channel standard deviation before applying Eq. 9. A *shrinkage-regularized* variant additionally penalizes correction energy relative to the observed clean-to-compressed representation shift, while the *cosine* variant adds a cosine-distance term. All normalization statistics are estimated exclusively from the clean repair-training representations. We use these descriptive names consistently below.

Repair strength α is an inference-time control and is not learned with the repairer. Compression experiments therefore report the value used for each operating point rather than requiring a single scalar to transfer across architectures. We use $\alpha = 0.1$ for the Honeybee and LLaVA results and $\alpha = 1$ for the Qwen MME multi-seed result. In a fully task-free deployment, α can be selected from held-out clean/compressed representation pairs by minimizing a representation-level validation criterion, without running the LLM. The repairer parameters themselves are trained task-free in every case.

Fixed-α objective comparison. At $\alpha = 0.1$, Honeybee MME is improved by all four cached objectives, with MSE recovering 44.1% of the clean-to-compressed degradation and normalized Smooth-L1 recovering 42.7%. LLaVA shows a complementary result: normalized Smooth-L1 with shrinkage is strongest at 29.7% recovery, while structured repair reaches 28.4% and exceeds MSE by 8.50 MME points at the same $\alpha = 0.1$. Table 8 reports the full fixed-strength objective comparison.

Table 8: **Learned-compression MME objective comparison.** Scores use $\alpha = 0.1$ unless otherwise indicated. Recovery is relative to the clean and raw compressed conditions.

Model	Objective	Score	Gain vs. raw	Recovery
Honeybee	MSE	1502.66	+26.35	44.1%
Honeybee	Norm. S-L1	1501.85	+25.53	42.7%
Honeybee	Norm. S-L1 + shrink.	1493.27	+16.96	28.4%
Honeybee	Norm. S-L1 + cosine	1482.72	+6.41	10.7%
LLaVA-v1.5	MSE	1799.80	+17.13	19.0%
LLaVA-v1.5	Norm. S-L1	1807.30	+24.63	27.3%
LLaVA-v1.5	Norm. S-L1 + shrink.	1809.43	+26.76	29.7%
LLaVA-v1.5	Norm. S-L1 + cosine	1798.03	+15.37	17.0%
LLaVA-v1.5	Structured	1808.30	+25.63	28.4%
Qwen2.5-VL	MSE	2225.68	-6.37	-7.0%
Qwen2.5-VL	Norm. S-L1	2218.62	-13.43	-14.8%
Qwen2.5-VL	Norm. S-L1 + shrink.	2220.78	-11.28	-12.4%
Qwen2.5-VL	Norm. S-L1 + cosine	2219.47	-12.59	-13.9%

Table 9: **Qwen2.5-VL normalized Smooth-L1 with shrinkage under Ballé-Q1** at $\alpha = 1$.

Seed	MME	Gain vs. raw	Recovery
0	2281.00	+48.95	54.0%
1	2261.14	+29.09	32.1%
2	2231.51	-0.55	-0.6%
Mean	2257.89	+25.83	28.5%

Qwen repair strength and multi-seed confirmation. Qwen MME favors a stronger correction than $\alpha = 0.1$. With normalized Smooth-L1 with shrinkage at $\alpha = 1$, the three repairer-training seeds score 2281.00, 2261.14, and 2231.51, for a mean of 2257.89 compared with 2232.05 for the raw compressed model. The mean gain is $+25.83$ MME points, corresponding to 28.5% recovery of the clean-to-compressed degradation. Table 9 reports the seed-level values.

The importance of the objective is also visible in a matched-strength Qwen ablation. At $\alpha = 0.5$, structured repair reaches 2246.12 MME, whereas MSE reaches 2224.54; structured therefore improves over MSE by 21.58 MME points at the same repair strength. On LLaVA, structured similarly exceeds MSE by 8.50 points at the shared $\alpha = 0.1$. Honeybee provides the complementary case: its exact structured objective scores 1475.38 at $\alpha = 0.1$, whereas the simpler MSE and normalized Smooth-L1 objectives provide the strongest recovery.

Qwen POPE. A conservative repair strength is effective on Qwen POPE. Raw Ballé-Q1 compression yields 0.80422, while normalized Smooth-L1 at $\alpha = 0.1$ reaches 0.80644. The paired repair gain is $+0.00222$, with a 95% bootstrap confidence interval of $[+0.00111, +0.00333]$. MSE and the other normalized variants also remain close to or above the raw compressed score at this conservative strength.

Overall, the compression experiments provide a second practical corruption regime in which representation-only repair recovers downstream utility on all three MLLMs. They also reinforce the objective comparison from the quantization study: robust or structure-aware objectives can outperform plain MSE, while the repair strength provides a simple mechanism for controlling how strongly a learned correction is applied.

G Reproducibility Details

Repair training uses 512 COCO train2014 images, split into 448 training and 64 representation-validation images. Final repair configurations are trained with seeds 0, 1, and 2 unless explicitly identified as exploratory single-seed experiments.

The 500 POPE evaluation images are disjoint from the repair-training images. Within each benchmark, image splits, prompts, decoding settings, and scoring are held fixed across clean, corrupted, and repaired conditions. Where predictions are aligned, uncertainty over evaluation examples is estimated using paired bootstrap resampling.

For controlled perturbation experiments, perturbation seeds are distinct from repairer-training seeds: the former quantify variability in the synthetic perturbation itself, whereas the latter quantify stochasticity in repairer training.

No task labels, downstream answers, language-model losses, or LLM forward/backward passes are used to optimize the repairer.