Modality-Aware Expert Pruning for MoE-Based Multimodal Large Language Models

Donggeun Kim
Nota Inc.
donggeun.kim@nota.ai

Abstract

Mixture-of-Experts (MoE) enables multimodal large language models (MLLMs) to scale capacity efficiently, but expert parameters dominate memory even though only a few experts are activated per token: they account for over 90% of model size in modern MoE-MLLMs (e.g., 54 GB out of ∼58 GB in BF16 for InternVL3.5-30B-A3B and Qwen3-VL-30B-A3B). Expert pruning has emerged as an effective compression strategy, but existing methods aggregate importance uniformly across tokens, ignoring that the same expert may serve different roles for vision versus language processing. To address this, we propose MAEP (Modality-Aware Expert Pruning), a training-free framework that decomposes expert importance by modality in a single forward pass. MAEP introduces (1) Delta-H Output (ΔH), a redistribution-aware metric measuring the MoE output change when an expert is removed, and (2) low-attention vision tokens as a vision-side pruning signal that is *dataset-stable*: its expert-importance ranking remains stable across different datasets. Text-side importance serves as a preservation signal while vision-side importance serves as a pruning signal, and experts are ranked globally across layers. Experiments on three MoE-based MLLMs across multimodal and text benchmarks show that MAEP outperforms modality-agnostic baselines on the majority of multimodal benchmarks while remaining competitive on text-only tasks, reducing the BF16 expert weight footprint by ∼27 GB (∼47% of total memory) so that the full weight footprint of a ∼58 GB model fits within a single 40 GB GPU. Code is available at https://github.com/nota-github/MAEP.

1 Introduction

Recent advances in large language models (LLMs) have increasingly adopted the Mixture-of-Experts (MoE) architecture as a key strategy for scaling model capacity while maintaining computational efficiency [11, 23, 57]. By activating only a subset of experts per token through sparse routing, MoE models achieve a favorable trade-off between total parameter count and inference cost. This paradigm has naturally extended to multimodal LLMs (MLLMs) [1, 3, 16, 25, 42, 49, 51], where MoE layers handle the joint processing of vision and language inputs.

However, while sparse MoE reduces computation by activating only a few experts per token, all expert parameters must remain in memory regardless of activation—making the MoE architecture inherently memory-intensive. In models such as InternVL3.5-30B-A3B and Qwen3-VL-30B-A3B, expert parameters account for 93–94% of the total model size (54 GB out of ∼58 GB in BF16; see Appendix C), despite only 8 of 128 experts being activated per token. This gap between memory footprint and effective utilization motivates **expert compression**—reducing the number or size of experts through techniques such as pruning, merging, or quantization. For instance, pruning 50% of experts can reduce the BF16 *routed-expert weight footprint* by ∼27 GB (47% of total weight memory), bringing the BF16 weight footprint of a 58 GB model below the nominal capacity of a single 40 GB GPU, thereby enabling deployment in memory-constrained settings.

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

Existing expert compression approaches, including merging [5, 30, 31], pruning [2, 27], and decomposition/quantization [20, 34], have proven effective for text-only MoE LLMs. However, these methods aggregate importance statistics uniformly across all input tokens without distinguishing token modality, making their direct application to MoE-based MLLMs inadequate. Indeed, recent analysis [22] reveals a consistent distributional gap between text and vision tokens in expert layers, with each modality interacting with experts differently, highlighting the need for a modality-aware expert compression strategy.

To this end, we propose **MAEP (Modality-Aware Expert Pruning)**, a training-free framework with two key components. Given only a small calibration set, MAEP computes all importance scores in a single forward pass without any gradient computation or weight updates:

First, we propose **Delta-H Output** (ΔH, where H denotes the MoE layer's hidden-state output) as the expert importance metric, defined as the squared L2 change in this output when an expert is removed with routing redistribution. Unlike metrics based on routing frequency or an expert's isolated contribution, ΔH captures the residual output change after redistributing weight to the remaining experts; smaller values indicate higher functional redundancy among co-activated experts. Because ΔH averages over a token subset, it can be computed separately for text and vision tokens, yielding per-modality importance scores.

Second, we find that not all vision tokens are equally reliable for importance estimation. Evaluating ΔH over the full set of vision tokens yields dataset-dependent rankings because vision content is tied to image-specific objects, making the resulting importance estimates domain-sensitive. Instead, **low-attention vision tokens**—those receiving minimal attention from text tokens—provide a substantially more dataset-stable signal.

We evaluate MAEP across multiple MoE-based MLLMs, pruning ratios, and benchmarks. Our results suggest that modality-aware importance scoring outperforms modality-agnostic baselines, supporting the value of modality decomposition for expert compression in vision-language models.

2 Related Work

2.1 Expert Compression for MoE Models

Prior work on compressing MoE-based LLMs has explored several directions. Among methods that reduce the expert pool, two prominent strategies are *merging*, which combines similar experts [5, 30, 31, 43–45], and *pruning*, which removes redundant experts based on estimated importance. Pruning methods differ mainly in how they identify redundant experts—e.g., router-weighted activation norms [27], differentiable non-uniform layer-wise expert pruning [2], second-order atomic-expert removal [29], or calibration-based search over preserved expert subsets [39]. At the weight level, MoE-Pruner [55] extends activation-aware weight pruning to LLMs by scaling weight-activation scores with router weights. Complementary strategies include replacing pruned experts with lightweight replacement modules [60], neuron-level recombination [64], and quantization [20] or low-rank decomposition [17, 34, 58]. However, most prior pruning and merging methods have been developed primarily for LLMs, leaving expert-compression strategies for MoE-MLLMs relatively underexplored.

Moreover, many importance-based criteria—particularly those derived from activation frequency or routing statistics [39, 55]—aggregate signals uniformly over tokens without distinguishing modality, which can obscure modality-specific expert roles in vision-language models. Vision tokens are themselves substantially redundant [6, 62], which raises the concern that frequency-aggregated importance may overweight non-essential tokens; indeed, calibrating weight pruning on all text tokens but only a subset of visual tokens has been shown to improve pruning quality [33]. This motivates modality-aware importance measures that decompose expert contributions by token type rather than aggregating uniformly across all tokens.

2.2 Efficient MLLMs

As MLLMs grow larger, their memory and compute demands increasingly constrain real-world deployment. Prior work has improved inference efficiency by reducing the number of visual tokens

forwarded to the language model [6, 40, 62], lowering weight precision through quantization [32, 59], and compressing the KV cache to reduce runtime memory [19, 48].

As MoE architectures—which scale model capacity while keeping per-token computation fixed—have also become common in MLLMs [1, 36, 49], improving efficiency for MoE-based architectures has been explored. FastMMoE [53] combines expert activation reduction for visual tokens with routing-aware visual token pruning, while RoE [50] and SMAR [54] optimize expert routing strategies. MoDES [22] is closest in motivation to our work: it accounts for modality-dependent expert utility and applies modality-specific thresholds to skip experts during inference. Existing MoE-MLLM efficiency methods mainly reduce inference-time computation through expert skipping or visual-token pruning; the expert parameters themselves, however, persist in GPU memory. MAEP targets this complementary but underexplored problem: permanent routed-expert pruning for MoE-based MLLMs, using modality-decomposed importance scores.

3 Preliminaries

3.1 Mixture-of-Experts Layers

Within each transformer block, the Mixture-of-Experts (MoE) architecture replaces the dense feed-forward network (FFN) with a set of N parallel expert networks $\{E_1, \ldots, E_N\}$ and a lightweight router network G. For an input hidden state $h \in \mathbb{R}^d$, the router applies a linear projection and a scoring function—typically softmax or sigmoid—to produce per-expert scores $p_i(h)$. Only the top-k highest-scoring experts are activated for each token, and the MoE layer output is their weighted sum:

$$H^{(l)}_{\text{MoE}}(h) = \sum_{i \in \text{top-}k(p(h))} w_i \cdot E_i(h) \tag{1}$$

where $w_i = p_i / \sum_{j \in \text{top-}k} p_j$ are the routing weights. This sparse activation enables MoE models to scale total parameters while maintaining the per-token computation cost of a much smaller model.

3.2 MoE-based MLLM Architecture

MoE-based MLLMs use a vision encoder and a modality projector to map images into vision tokens, which are **concatenated with text tokens** and passed through the shared MoE backbone. Crucially, the router dispatches both vision and text tokens to the **same pool of experts** based solely on the hidden state, without explicit modality conditioning. After the final MoE layer, the model applies an LM head $W \in \mathbb{R}^{V \times d}$ that maps the last hidden state h to the output distribution $p(y \mid h) = \mathrm{softmax}(Wh)$ over a vocabulary of size V. This shared-pool design raises a fundamental question: *is expert importance uniform across modalities, or does it differ between text and vision tokens?* We investigate this empirically in Section 4.

4 Empirical Motivation

4.1 Why We Need a Modality-Aware Proxy Metric

Following OBD [28]'s principle, which measures a component's importance by the loss change upon removal, a direct way to score an expert is to remove it and measure the induced change in the model's next-token distribution. This change can be captured by the LM-head KL divergence $D_{\mathrm{KL}}(p_{\text{orig}} \| p_{\text{pruned}})$ between the unpruned and pruned output distributions. Directly measuring this KL for every expert would capture the effect on output but is prohibitively expensive: with N experts per layer and L MoE layers, it requires $N \times L$ full forward passes per calibration sample (e.g., $128 \times 48 = 6{,}144$ passes for InternVL3.5-30B-A3B). We therefore use **Delta-H Output** (ΔH), the MoE-output perturbation induced by expert removal, as a lightweight proxy. We formalize this proxy in Section 5.1.

Furthermore, a proxy aggregating uniformly over tokens is insufficient. Prior work has observed modality-dependent expert behavior in MoE-based MLLMs [22, 54, 56]; we find that this extends to expert *importance*: when ranked by ΔH on text tokens versus vision tokens, the resulting expert importance rankings show limited agreement (Spearman ρ between -0.11 and 0.05 at specific layers;

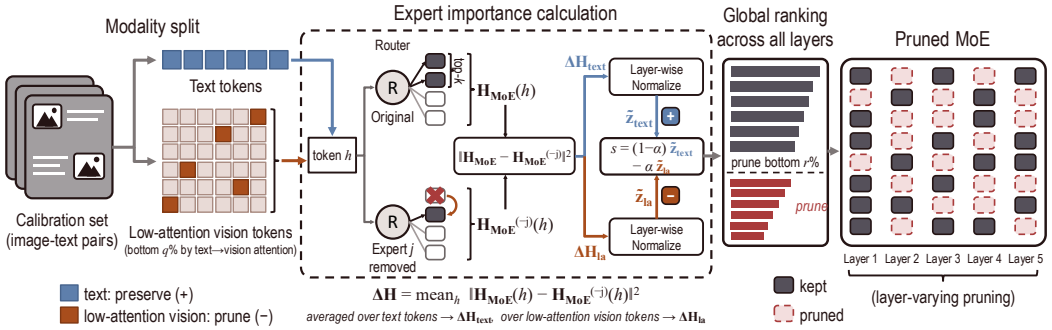


Figure 1: Pipeline of MAEP. Text and low-attention vision tokens are scored separately via $\Delta H^{(l,j)}$, layer-wise normalized, combined with asymmetric signs—text preservation (+) and low-attention vision pruning (−); see Eq. 4—and ranked globally.

Appendix F). The expert removal also induces an asymmetric output-side perturbation: the LM-head output distribution shifts up to 29× more on vision-token positions than on text-token positions, indicating modality-asymmetric representational sensitivity to expert removal. These observations imply that an effective metric must be **computationally efficient** and **modality-decomposable**.

4.2 Vision Token Heterogeneity and Signal Stability

Prior work on vision token pruning [6, 40, 62] has shown that not all vision tokens contribute equally. This suggests that modality decomposition alone is not sufficient on the vision side: the choice of vision tokens used to estimate expert importance also matters. We compute the **attention score of a vision token** as the average text-to-vision attention weight, averaged across all heads at each layer (a single scalar per token, layer, and sample), and compare token subsets by cross-dataset ranking stability.

Specifically, cross-dataset Spearman ρ (between ShareGPT4V [7] and ChartNet [26], averaged across layers; Appendix G) reveals a clear hierarchy: full vision (≈ 0.0), high-attention vision (top $q{\approx}20\%$, ≈ 0.2), text (≈ 0.5), and low-attention vision ($\approx \mathbf{0.6}$). In particular, expert importance ranked by ΔH computed from the full or high-attention vision tokens is sensitive to the calibration dataset. By construction, cross-modal attention highlights tokens relevant to the current text query—a property that text-guided token pruning methods such as SparseVLM [62] rely on. Different calibration datasets carry different content and queries, so high-attention selections drift across datasets. Moreover, not all high-attention tokens are informative: some are visual attention sinks [24] that attract attention despite carrying little semantic content. By contrast, **low-attention vision tokens** (bottom q-th percentile of text-to-vision attention) yield the most stable expert importance rankings across calibration datasets, suggesting they reflect the model's visual processing rather than domain-specific content.

5 Proposed Method

Building on the observations in Section 4, this section formally defines the ΔH proxy and presents the pruning framework (Figure 1). The overall pipeline is entirely training-free: given a small calibration set[1], the full pipeline (importance scoring, modality decomposition, and global pruning) completes within a single forward pass per sample.

Scope. MAEP applies to MoE architectures with top-k routing ($k \geq 2$, required by the redistribution counterfactual; Section 5.1). For architectures that additionally employ always-active shared experts (e.g., DeepSeek-VL2 [51], Kimi-VL [25]), MAEP targets only the routed experts: shared experts are excluded from importance scoring and pruning. The memory savings reported in this paper refer exclusively to the removal of routed expert parameters.

Notation. We distinguish two levels of indexing: $h \in \mathbb{R}^d$ denotes a single token's hidden state, while x denotes a calibration sample (image-text pair) of $|x|$ tokens. At the token level, $H^{(l)}_{\text{MoE}}(h)$ is the

[1]We use 1K samples from ShareGPT4V [7], selected by text length; see Appendix D for details.

MoE layer output (Section 3.1), $p_i(h)$ is the router score, and w_i is the routing weight. Sample-level and dataset-level aggregation of importance scores is defined in Section 5.1.

5.1 Delta-H Output: Expert Importance Metric

We define the importance of expert j in layer l as the expected change in the MoE output when expert j is removed:

$$\Delta H^{(l,j)} = \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{|x|}\sum_{h\in x}\left\|H^{(l)}_{\text{MoE}}(h) - H^{(l,-j)}_{\text{MoE}}(h)\right\|_2^2\right] \tag{2}$$

where $H^{(l,-j)}_{\text{MoE}}$ denotes the MoE output with expert j excluded via **router redistribution**: expert j's router logit is masked and the top-k experts are re-selected and renormalized, as in the pruned model. Formally, the redistributed weight for each remaining expert $i \neq j$ is $w_i^{(-j)} = w_i/(1-w_j+w_{(k+1)})$, where $w_{(k+1)}$ is the $(k+1)$-th ranked expert's router score normalized in the same way as the top-k weights, preserving the relative ratios among the remaining experts.

Redistribution operates on renormalized top-k weights, making it structurally agnostic to the underlying scoring function. We adopt a **fixed top-k counterfactual** (the promoted $(k+1)$-th expert is not evaluated and contributes no output) to avoid an additional expert computation during calibration; this is nearly identical to the promotion variant (mean $\rho > 0.98$ across modalities; Appendix H).

The entire computation requires only a **single forward pass** through the model: cached top-k weights and expert outputs let us evaluate all counterfactuals analytically. Restricting the token set to a specific modality yields per-modality variants of ΔH, supporting fine-grained importance estimation while capturing functional redundancy among co-activated experts.

5.1.1 Why ΔH: Connection to KL Divergence

The following bound shows that $\|\hat{h} - \hat{h}^{(-j)}\|_2^2$ (the squared L_2 change in the MoE output, with $\hat{h} := H^{(l)}_{\text{MoE}}(h)$) arises naturally in a KL upper bound, motivating ΔH as a *within-layer* ranking proxy. It does not itself justify the cross-layer normalization, modality decomposition, or global ranking in Section 5.3; those are described and validated separately.

Proposition 1. *Let $\Phi_l : \mathbb{R}^{T\times d} \to \mathbb{R}^{T\times|V|}$ map MoE layer l's outputs for a T-token sequence through subsequent transformer blocks and the LM head to pre-softmax logits, on a compact activation domain (approximately ensured by RMSNorm/LayerNorm). For the pre-MoE hidden states $h_t \in \mathbb{R}^d$ of the sequence at layer l, let $\hat{h}_t := H^{(l)}_{MoE}(h_t)$ and $\hat{h}_t^{(-j)} := H^{(l,-j)}_{MoE}(h_t)$ denote the MoE outputs before/after removing expert j, and let p_t and $p_t^{(-j)}$ denote the resulting next-token distributions at position t. Then*

$$\frac{1}{T}\sum_{t=1}^{T} D_{\text{KL}}\big(p_t \,\|\, p_t^{(-j)}\big) \le \frac{L_{\Phi_l}^2}{4}\cdot\frac{1}{T}\sum_{t=1}^{T}\|\hat{h}_t - \hat{h}_t^{(-j)}\|_2^2, \tag{3}$$

where $L_{\Phi_l} = \|W\|_{\text{op}}\, L_{\text{norm}} \prod_{i=l+1}^{L}(1+L_{g_i})$ absorbs the LM head, the pre-LM-head normalization, and the local Lipschitz constants of subsequent attention+FFN branches g_i in $f_i(X) = X + g_i(X)$.

The proof (Appendix B) composes Lipschitz post-MoE blocks with the $\frac{1}{2}$-smoothness of $A(z) = \log\sum_i e^{z_i}$. Averaging over the calibration distribution yields $\mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{|x|}\sum_{h\in x} D_{\text{KL}}\right] \le C_l \cdot \Delta H^{(l,j)}$, with $C_l = L_{\Phi_l}^2/4$.

Role of the bound. Since C_l is layer-dependent but expert-independent, ranking experts by the bound within a layer is equivalent to ranking them by ΔH. The bound motivates ΔH—rather than routing frequency or gating entropy—as the quantity that emerges naturally from the Lipschitz–softmax composition of the post-MoE network. Since C_l is a layer-shared constant, within-layer ranking depends only on relative magnitudes of ΔH across experts; L_{Φ_l} thus does not affect ranking outcomes and need not be estimated.

Empirical validation. Per-layer Spearman rank correlation between ΔH and actual KL divergence increases with depth overall for both modalities, reaching >0.8 in deeper layers (vision) and ~ 0.89 (text) (Appendix I). This pattern is qualitatively consistent with the depth dependence in Proposition 1:

deeper layers admit a tighter bound and thus a more reliable proxy. The weaker correlation in shallow layers reflects the bound's depth dependence in Proposition 1; in practice, the global pruning decision aggregates across all layers via the combined score (Eq. 4).

5.2 Low-Attention Vision Scoring

Motivated by the observation in Section 4.2 that full-vision and high-attention signals are dataset-dependent, we use low-attention vision tokens as the basis for the vision-side pruning signal. Per sample and per layer, tokens in the bottom q-th percentile ($q = 2\%$; ablated in Section 6.3.1) of the attention score (Section 4.2) are classified as low-attention; this per-sample selection adapts to varying image resolutions and aspect ratios. We then compute ΔH (Eq. 2) over these tokens only (with $|x|$ rescoped to the subset cardinality), yielding the vision-side score $\Delta H_{\text{la}}^{(l,j)}$ ("la" denoting low-attention; same subscript convention used for $\tilde{z}_{\text{la}}$ in Eq. 4). A high value indicates that expert j is heavily engaged with vision tokens that receive little text-to-vision attention—i.e., experts that specialize in non-salient background visual content. **In contrast to text-side ΔH, where a high value marks an essential expert to preserve, a high $\Delta H_{\textbf{la}}^{(l,j)}$ identifies experts that allocate capacity to less informative tokens and are therefore favorable pruning candidates** (verified at the model output in Appendix I.1); we make this asymmetric usage explicit in Section 5.3. This signal is also more dataset-stable than full or high-attention alternatives (Section 4.2), since low-attention tokens reflect non-salient background content rather than domain-specific features (Appendix G.2).

5.3 Combined Scoring and Global Pruning

The two signals play **asymmetric roles**: text-side ΔH acts as a *preservation signal* (high $\rightarrow$ retain), while low-attention vision-side ΔH acts as a *pruning signal* (high $\rightarrow$ remove). Since raw ΔH values follow a heavy-tailed distribution and vary in magnitude across layers by up to 10^3–$10^4\times$ (Appendix J), we first apply a log-transform followed by layer-wise z-score normalization: $\tilde{z}^{(l,j)} = (\log(\Delta H^{(l,j)} + \epsilon) - \mu_l)/\sigma_l$, $\epsilon = 10^{-8}$. Here μ_l and σ_l denote the mean and standard deviation of $\log(\Delta H^{(l,j)} + \epsilon)$ over experts j in layer l. The final score for expert j in layer l is:

$$s(l, j) = (1 - \alpha) \cdot \tilde{z}_{\text{text}}^{(l,j)} - \alpha \cdot \tilde{z}_{\text{la}}^{(l,j)} \tag{4}$$

where α controls the modality balance. **Experts with the lowest $s(l, j)$ are pruned first.**

Correlation-based α selection. Since models exhibit varying degrees of text–vision importance ranking correlation, the modality balance α is model-specific; we obtain it from a closed-form expression on calibration statistics. Let $\bar{\rho}_{x,y} = \frac{1}{L}\sum_l \rho_l(\Delta H_x^{(l)}, \Delta H_y^{(l)})$ denote the layer-averaged Spearman correlation between expert rankings under modalities $x, y \in \{\text{t}, \text{v}, \text{la}\}$, where t, v, and la index ΔH on text, full-vision, and low-attention vision token subsets respectively. We adopt $\alpha = a/(a + b)$ with $a := \bar{\rho}_{\text{t,v}} - \bar{\rho}_{\text{t,la}}$ and $b := \bar{\rho}_{\text{v,la}}$ (both positive on the evaluated models, keeping $\alpha \in [0, 1]$; Table 8). Intuitively, a captures the *differential alignment* of text-side importance with the full-vision versus low-attention signal: when text and full-vision experts overlap but text and low-attention experts do not, low-attention vision is complementary to text and α tends to be larger. b captures *within-vision redundancy*: when full-vision and low-attention rankings agree, favoring low-attention adds little new information and α tends to be smaller. Section 6.3.3 reports the closed-form α for each model and verifies that these values lie within the empirical safe range; per-model $\bar{\rho}$ values are reported in Appendix F.

Global pruning. Expert redundancy varies across layers and modalities [2, 22], suggesting that a fixed per-layer budget is suboptimal. We instead rank all experts across all layers by $s(l, j)$ and prune the bottom $\lfloor r \cdot N_{\text{total}} \rfloor$ entries, so that no layer is forced to prune experts that its own score distribution does not mark as redundant (Section 6.3.1, Appendix K). We compare this against uniform per-layer pruning in Section 6.3.1.

Proposition 1 justifies ΔH only as a *within-layer* ranking, since $C_l = L_{\Phi_l}^2/4$ differs across layers and is not estimated. The z-score above serves a separate purpose: it makes the per-layer score distributions comparable in shape so that ΔH can be used as a global ranking signal without recovering C_l (Section 6.3.1, Appendix K). The asymmetric text/vision combination reflects the modality gap of Section 4.1.

Table 1: Main results at 25% and 50% expert pruning ratios. †Unpruned-model results are reproduced with our evaluation pipeline. "Avg. (%)" is the mean retention of each method's per-benchmark score relative to the unpruned baseline, averaged over all 11 benchmarks.

			Multimodal								*Text-only*			Avg.
Model	Ratio	Method	MME	MMBench	MMStar	ChartQA	TextVQA	RealWorldQA	CV-Bench	POPE	MMLU-Redux	IFEval	ARC-C	(%)
InternVL3.5 30B-A3B	—	Original†	2442.2	85.2	71.4	87.3	80.2	72.2	84.6	89.4	80.9	78.9	95.2	100.0
	25%	HC-SMoE	2344.8	79.8	67.1	**87.0**	76.2	71.6	**84.5**	**89.8**	67.4	45.6	87.4	91.9
		REAP	2326.6	**84.0**	68.9	83.2	78.8	**72.5**	83.6	89.3	69.3	69.7	92.7	95.9
		MAEP	**2364.9**	83.2	**69.3**	**87.0**	**78.9**	71.6	84.3	89.7	**72.3**	**77.1**	**94.1**	**97.7**
	50%	HC-SMoE	1907.5	66.1	56.2	75.3	66.5	68.0	79.8	88.3	33.9	20.9	53.2	74.1
		REAP	2109.6	78.7	**63.3**	65.3	74.4	71.2	**83.9**	89.1	53.9	**66.4**	85.7	88.5
		MAEP	**2160.3**	**80.7**	**63.3**	**76.7**	**75.6**	**71.6**	83.3	**89.7**	**59.8**	64.9	**90.6**	**91.2**
Qwen3-VL 30B-A3B	—	Original†	2416.6	87.3	66.9	85.3	84.6	73.5	86.0	88.5	79.0	86.0	96.7	100.0
	25%	HC-SMoE	2085.4	83.9	65.3	**84.3**	83.0	71.9	81.3	**88.8**	**73.6**	81.7	94.9	96.0
		REAP	2133.5	83.3	**65.7**	84.2	82.9	68.4	73.9	88.3	73.3	80.9	95.1	94.8
		MAEP	**2287.3**	**84.4**	**65.7**	82.5	**83.9**	**72.5**	**82.8**	88.4	68.7	**83.0**	**95.3**	**96.6**
	50%	HC-SMoE	1923.9	75.1	59.9	**74.3**	76.1	69.4	80.5	85.2	**56.0**	65.2	74.8	85.5
		REAP	2026.6	**80.3**	59.9	67.1	78.1	66.0	70.1	88.2	**56.0**	**73.8**	**85.9**	86.6
		MAEP	**2042.8**	**80.3**	**60.2**	69.6	**79.7**	**69.7**	**85.2**	**89.5**	53.0	69.3	83.1	**88.3**
Gemma4 26B-A4B	—	Original†	2333.2	88.9	71.1	69.2	71.8	76.3	85.1	88.1	80.3	88.2	97.1	100.0
	25%	HC-SMoE	1970.1	22.0	37.4	66.7	63.8	53.1	71.3	85.5	62.2	67.3	94.9	77.2
		REAP	2204.9	84.9	**65.8**	66.6	**72.1**	75.1	82.6	**88.3**	**70.6**	83.7	**95.9**	**96.0**
		MAEP	**2224.9**	**85.8**	64.6	**68.4**	70.8	**76.2**	**84.0**	88.2	64.5	**86.5**	95.5	**96.0**
	50%	HC-SMoE	553.4	9.8	27.6	35.6	14.0	40.1	49.0	49.6	25.5	26.1	58.7	39.3
		REAP	1939.3	61.0	50.7	**63.4**	65.2	66.8	75.9	87.9	41.0	67.8	77.7	80.9
		MAEP	**1975.4**	**78.2**	**52.3**	54.4	**70.7**	**72.3**	**82.7**	**88.4**	**51.8**	**73.6**	**88.1**	**86.7**

6 Experiments

6.1 Experimental Setup

Models and baselines. We evaluate on three MoE-based MLLMs: InternVL3.5-30B-A3B [49], Qwen3-VL-30B-A3B-Instruct [1], and Gemma4-26B-A4B-Instruct [15].[2] All three share the same expert pool for both vision and text tokens; detailed model configurations are provided in Appendix C. We compare against (1) HC-SMoE [5], which merges similar experts via hierarchical clustering, and (2) REAP [27], which prunes experts by router-weighted activation norms.

Benchmarks. We report results on several multimodal benchmarks—ChartQA [41], TextVQA [46], MMStar [8], MMBench [38], RealWorldQA [52], MME [13], POPE [35], and CV-Bench [47] (average of 2D and 3D splits)— covering chart understanding, visual question answering, hallucination evaluation, and 2D/3D vision-centric perception. To measure the effect of pruning on language-only capabilities, we additionally evaluate on three text-only benchmarks: MMLU-Redux [14], IFEval [63], and ARC-Challenge [9]. Further details on hardware and evaluation pipeline are provided in Appendix D.

6.2 Main Results

As shown in Table 1, MAEP retains near-original performance at 25% pruning and achieves the best or tied-best average retention on every model–ratio configuration, with the gap widening as the pruning ratio increases.

Multimodal benchmarks. At 25% pruning, MAEP stays close to the unpruned baseline across models, with the best or tied-best average retention on every configuration ("Avg." column in Table 1). At 50%, MAEP maintains the highest average retention across all three models, while HC-SMoE drops sharply and REAP trails consistently. The advantage of modality-aware pruning grows with sparsity: at 25% REAP stays within $\sim$2 points of MAEP, while at 50% the gap widens to 2–6 points, reflecting that the cost of modality-uniform treatment grows with sparsity. HC-SMoE's similarity-based merging is competitive at 25% on InternVL3.5 and Qwen3-VL but drops sharply on Gemma4, where its modality-uniform clustering conflates experts with distinct modality roles. REAP's router-weighted activation scoring degrades more gradually but treats vision and text tokens uniformly, missing the asymmetric importance documented in Section 4 and yielding consistently lower multimodal retention than MAEP. MAEP's text/vision decomposition addresses both: text-side

[2]"AxB" denotes $\sim x$ billion active parameters per token under top-k routing (e.g., 30B-A3B = 30B total, 3B active per token).

Table 2: Ablation studies. (a) Scoring components and pruning strategy on InternVL3.5 (50% pruned). (b) Calibration set design on Qwen3-VL (50% pruned): composition and number of samples. Bold marks the best value per column; *collapse* indicates average retention below 30% of the unpruned baseline. $^{\dagger}\alpha$ re-tuned for the uniform budget (Section 6.3.1).

(a) Scoring components and pruning strategy (InternVL3.5, 50% pruned)

Configuration	MME	MMStar	ChartQA	TextVQA
Default	2160	**63.3**	**76.7**	75.6
Modality decomposition				
w/o modality split	2034	51.1	41.2	66.0
Vision-side signal				
w/ high-attn (top $\{2, 5, 10\}\%$)		*collapse*		
w/ high-attn (top 20%)	**2226**	55.5	54.5	76.1
w/ full vision	2152	48.3	38.8	66.9
Pruning strategy				
w/ uniform per-layer budget (α=0.5)†	2104	62.2	72.1	**76.5**

(b) Calibration set design (Qwen3-VL, 50% pruned)

Calibration setting	MME	MMStar	ChartQA	TextVQA
Default (ShareGPT4V, 1K)	2042.8	60.2	69.6	79.7
Varying composition (1K)				
+ ChartNet (7:3)	2034.2	**64.3**	**83.5**	79.4
Varying # samples (ShareGPT4V)				
100	2006.0	58.7	68.2	78.9
500	**2047.2**	60.7	66.1	**80.3**

preservation protects language-critical experts, while low-attention vision scoring isolates vision-side redundancy.

Text-only benchmarks. While all methods suffer substantial degradation on text-only benchmarks at 50% pruning, MAEP remains competitive on most cells and achieves the highest aggregate retention despite trailing REAP on the three Qwen3-VL text-only benchmarks (separate multimodal and text-only averages are given in Appendix E.1).

6.3 Ablation Study and Analysis

6.3.1 Modality Decomposition, Vision Signal, and Pruning Strategy

Modality decomposition. Table 2(a) shows that the modality decomposition itself is essential: aggregating ΔH uniformly over all tokens (no text/vision split) substantially degrades MMStar, ChartQA, and TextVQA, indicating that the asymmetric text-preserve / vision-prune signaling cannot be recovered from a single undifferentiated importance score.

Vision signal alternatives. Table 2(a) confirms that low-attention's dataset-stability advantage over the high-attention and full-vision alternatives is reflected in downstream performance: replacing it with either alternative substantially degrades visual reasoning benchmarks. For these alternatives we flip the sign in Eq. 4 ($-\alpha\,\tilde{z}_{\text{la}} \rightarrow +\alpha\,\tilde{z}_{\text{vision/ha}}$) because high-attention and full-vision signals capture *important* vision tokens (preservation signal), whereas low-attention captures *unimportant* ones (pruning signal). Overall, high-attention is comparable on benchmarks close to the calibration distribution (MME, TextVQA) but drops substantially on distant benchmarks like ChartQA (-22.2). Appendix G.1 extends this comparison to representation-drift and diversity-based token selectors.

Low-attention threshold. We vary the quantile threshold q used to define low-attention vision tokens, considering $q \in \{2\%, 5\%, 10\%, 20\%, 30\%\}$. As q increases, more semantically relevant tokens enter the low-attention set, making the vision-side ranking converge toward the text-side ranking. Figure 2 (left) confirms this: ρ rises from negative correlation at q=2% to near zero at q=30%, with TextVQA accuracy tending to decrease as q grows—modality-decomposed scoring (Eq. 4) relies on the two signals carrying distinct information. The right panel further shows that low-attention vision tokens yield the most dataset-stable expert rankings: cross-dataset ρ (ShareGPT4V vs. ChartNet) reaches 0.54 at q=2%, compared to 0.33 for the size-matched high-attention subset (top 2%) and $\approx$0 for full vision tokens. This suggests that the strictest threshold isolates vision tokens whose expert importance is both modality-discriminative and dataset-stable.

Global vs. uniform pruning. Prior work has shown that pruning sensitivity is highly non-uniform across layers [21, 61], and expert redundancy likewise varies substantially across MoE layers [2]. Uniform pruning, which removes the same number of experts from every layer, ignores this non-uniformity; in contrast, our global ranking exploits per-layer differences in score-distribution shape (layer-wise z-scoring centers each layer's $\tilde{z}$ at zero, so the cross-layer pruning rate reflects skewness and tail thickness, not absolute magnitude). Table 2(a) shows that global allocation still leads uniform pruning on MME and ChartQA on InternVL3.5, even with α re-tuned for the uniform budget (α=0.5;

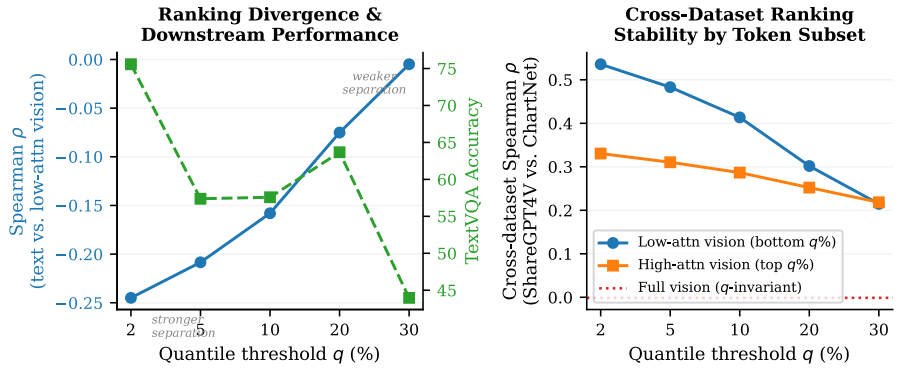


Figure 2: Effect of quantile threshold q on InternVL3.5 at 50% pruning. Left: Spearman ρ between text and low-attention vision rankings (more negative = more distinct) and TextVQA accuracy. Right: cross-dataset Spearman ρ (ShareGPT4V vs. ChartNet) by token subset under matched-size symmetric quantiles (low-attn = bottom $q\%$, high-attn = top $q\%$).

a forced per-layer quota leaves more of the protection to the text term, so it calls for a smaller α; full sweep in Appendix L). Appendix K visualizes the resulting per-layer distribution.

6.3.2 Calibration Set Design: Domain and Size

With the default ShareGPT4V calibration, Qwen3-VL and Gemma4 at 50% pruning show a pronounced weakness on ChartQA. We trace this to *calibration domain mismatch*: ShareGPT4V contains primarily natural-image conversations with little chart or document content, so experts specialized for chart reasoning receive limited activation during calibration and are consequently under-scored by ΔH. This implies that MAEP's importance scores are responsive to deployment-domain signal in the calibration set—a property practitioners can exploit to target a specific deployment distribution.

To demonstrate this, we construct a mixed calibration set of 700 ShareGPT4V [7] and 300 ChartNet [26] reasoning samples, where ChartNet pairs chart images with extended reasoning text that matches ShareGPT4V's text-token density. Under this mixed calibration, MAEP on Qwen3-VL at 50% pruning (Table 2(b)) recovers **13.9 points on ChartQA**, while also improving MMStar by 4.1 points; MME and TextVQA are nearly identical. The absence of a substantial trade-off supports two claims: (i) the ChartQA gap is attributable to calibration domain mismatch; and (ii) MAEP can be *targeted* at a deployment domain by augmenting calibration with in-domain data.

This is consistent with the REAP's [27] strategy of combining general pre-training corpora with domain-specific data for calibration: a diverse mixture prevents catastrophic drops on specialized distributions and, as we observe, can further improve in-domain performance. We leave to future work calibration-mixing strategies and extension to reasoning-heavy generative benchmarks (e.g., coding, mathematical problem solving).

Calibration size. Beyond composition, we test how far the calibration budget can be reduced by subsampling ShareGPT4V from the default 1,000 down to 100 samples (Table 2(b), bottom block). These results suggest MAEP is robust to calibration set size: even at 100 samples, MAEP retains most of the default-1K performance, with some fluctuation on individual benchmarks (e.g., ChartQA).

6.3.3 α Sensitivity and Safe Range

Section 5 fixes α deterministically, which yields $0.75/0.33/0.08$ for InternVL3.5 / Qwen3-VL / Gemma4—rounded to $0.8/0.3/0.1$ (the "Used α" column in Table 3) for simplicity. We verify that these values lie within an empirically stable range by sweeping α at both pruning ratios and reporting the average performance retention over MME, MMStar, ChartQA, and TextVQA. Table 3 shows that the empirical safe range narrows for models with higher within-vision redundancy (the $b := \bar{\rho}_{\text{v,la}}$ feature in Section 5): InternVL3.5 tolerates a wide range of α, Qwen3-VL exhibits a sharp degradation once α exceeds its operating point, and Gemma4 degrades gradually as α grows.

This is consistent with the redundancy interpretation of b: when full-vision and low-attention rankings agree, a large α overemphasizes a vision-side signal that offers little additional information. Across all three models, the closed-form α lies within a stable region of the sensitivity sweep: it coincides with the row's top-performing swept value for Qwen3-VL and Gemma4, and for InternVL3.5 it lies at the upper end of the stable region ($\alpha \leq 0.7$ in the sweep; retention collapses at α=0.9), consistent with the main results in Table 1. We adopt a single fixed α per model for simplicity; exploring layer-wise modality weights is a promising direction, since modality sensitivity may vary across layers.

Table 3: α sensitivity at 25/50% pruning. Cells: % retention of unpruned baseline (avg. over MME, MMStar, ChartQA, TextVQA).

Model	Used α	Ratio	α sweep (% retention)				
			0.1	0.3	0.5	0.7	0.9
InternVL3.5	0.8	25%	97.5	98.3	98.1	98.0	98.1
		50%	84.3	85.9	86.9	89.1	65.1
Qwen3-VL	0.3	25%	96.4	97.1	97.1	29.0	26.0
		50%	84.4	87.6	25.4	29.6	23.0
Gemma4	0.1	25%	95.9	91.1	72.5	67.6	66.7
		50%	83.8	74.1	60.9	51.1	41.0

7 Conclusion

We presented MAEP, a training-free framework for pruning MoE experts in MLLMs by decomposing their importance into text and vision contributions. By combining a redistribution-aware importance metric (ΔH) with low-attention vision tokens as a stable pruning signal, MAEP helps identify which experts can be removed with reduced risk of disproportionately degrading either modality. Experiments across multiple models and benchmarks show that this modality-aware approach retains near-original performance at 25% pruning and generally outperforms modality-agnostic baselines at 50% pruning, reducing the expert-weight footprint while preserving stronger performance than competing compression methods. This work inspires further research on modality-aware compression strategies for MoE-based MLLMs and enables deployment in resource-constrained settings.

Acknowledgments and Disclosure of Funding

The author thanks KyungHwan Shim, Huiseong Gim, and JinHyeok Park of the NetsPresso team for their help and valuable discussions. This work was supported by Nota Inc. and by the Technology Innovation Program (RS-2024-00468747, Development of AI and Lightweight Technology for Embedding Multisensory Intelligence Modules) funded by the Ministry of Trade Industry & Energy (MOTIE, Korea).

References

[1] Shuai Bai et al. Qwen3-VL technical report, 2025. URL `https://arxiv.org/abs/2511.21631`.

[2] Sikai Bai, Haoxi Li, Jie Zhang, Zicong Hong, and Song Guo. DiEP: Adaptive mixture-of-experts compression through differentiable expert pruning. In *Advances in Neural Information Processing Systems*, 2025. URL `https://nips.cc/virtual/2025/poster/116036`.

[3] Baidu ERNIE Team. ERNIE 4.5 technical report. Technical report, 2025. URL `https://ernie.baidu.com/blog/publication/ERNIE_Technical_Report.pdf`.

[4] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

[5] I-Chun Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, and Chun-Yi Lee. Retraining-free merging of sparse MoE via hierarchical clustering. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 8594–8620. PMLR, 2025. URL `https://proceedings.mlr.press/v267/chen25aq.html`.

[6] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, 2024.

[7] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. ShareGPT4V: Improving large multi-modal models with better captions, 2023. URL https://arxiv.org/abs/2311.12793.

[8] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models?, 2024. URL https://arxiv.org/abs/2403.20330.

[9] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

[10] OpenCompass Contributors. OpenCompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023.

[11] DeepSeek-AI et al. DeepSeek-V3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

[12] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. VLMEvalKit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11198–11201, 2024.

[13] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models, 2023. URL https://arxiv.org/abs/2306.13394.

[14] Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. Are we done with MMLU? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5069–5096, 2025.

[15] Gemma Team. Gemma 4 model card. Hugging Face model card, 2026. URL https://huggingface.co/google/gemma-4-26B-A4B-it.

[16] GLM-V Team, Wenyi Hong, Wenmeng Yu, et al. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2025. URL https://arxiv.org/abs/2507.01006.

[17] Hao Gu, Wei Li, Lujun Li, Zhu Qiyuan, Mark G. Lee, Shengjie Sun, Wei Xue, and Yike Guo. Delta decompression for MoE-based LLMs compression. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 20497–20514. PMLR, 2025. URL https://proceedings.mlr.press/v267/gu25c.html.

[18] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[19] Kai Huang, Hao Zou, Bochen Wang, Ye Xi, Zhen Xie, and Hao Wang. AirCache: Activating inter-modal relevancy KV cache compression for efficient large vision-language model inference, 2025. URL https://arxiv.org/abs/2503.23956.

[20] Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and Xiaojuan Qi. Mixture compressor for mixture-of-experts LLMs gains more, 2024. URL https://arxiv.org/abs/2410.06270.

[21] Weizhong Huang, Yuxin Zhang, Xiawu Zheng, Fei Chao, and Rongrong Ji. Determining layer-wise sparsity for large language models through a theoretical perspective. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 26175–26201. PMLR, 2025. URL https://proceedings.mlr.press/v267/huang25ax.html.

[22] Yushi Huang et al. MoDES: Accelerating mixture-of-experts multimodal large language models via dynamic expert skipping, 2025. URL https://arxiv.org/abs/2511.15690.

[23] Albert Q. Jiang et al. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

[24] Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models, 2025. URL https://arxiv.org/abs/2503.03321.

[25] Kimi Team et al. Kimi-VL technical report, 2025. URL https://arxiv.org/abs/2504.07491.

[26] Jovana Kondic, Pengyuan Li, Dhiraj Joshi, et al. ChartNet: A million-scale, high-quality multimodal dataset for robust chart understanding. *arXiv preprint arXiv:2603.27064*, 2026.

[27] Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. REAP the experts: Why pruning prevails for one-shot MoE compression. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ukGxWd2aDG.

[28] Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems 2*, pages 598–605, 1989. URL https://proceedings.neurips.cc/paper/1989/hash/6c9882bbac1c7093bd25041881277658-Abstract.html.

[29] Ke Li, Zheng Yang, Zhongbin Zhou, Feng Xue, Zhonglin Jiang, and Wenxiao Wang. HEAPr: Hessian-based efficient atomic expert pruning in output space, 2025. URL https://arxiv.org/abs/2509.22299.

[30] Lujun Li, Qiyuan Zhu, Jiacheng Wang, Wei Li, Hao Gu, Sirui Han, and Yike Guo. Sub-MoE: Efficient mixture-of-expert LLMs compression via subspace expert merging, 2025. URL https://arxiv.org/abs/2506.23266.

[31] Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. Merge, then compress: Demystify efficient SMoE with hints from its routing policy, 2024. URL https://arxiv.org/abs/2310.01334.

[32] Shiyao Li, Yingchun Hu, Xuefei Ning, Xihui Liu, Ke Hong, Xiaotao Jia, Xiuhong Li, Yaqi Yan, Pei Ran, Guohao Dai, et al. MBQ: Modality-balanced quantization for large vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4167–4177, 2025.

[33] Sijie Li, Biao Qian, and Jungong Han. Mostly text, smart visuals: Asymmetric text-visual pruning for large vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2603.16001.

[34] Wei Li, Lujun Li, Hao Gu, You-Liang Huang, Mark G. Lee, Shengjie Sun, Wei Xue, and Yike Guo. MoE-SVD: Structured mixture-of-experts LLMs compression via singular value decomposition. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 35209–35230. PMLR, 2025. URL https://proceedings.mlr.press/v267/li25az.html.

[35] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne 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*, 2023. URL https://arxiv.org/abs/2305.10355.

[36] Bin Lin et al. MoE-LLaVA: Mixture of experts for large vision-language models, 2024. URL https://arxiv.org/abs/2401.15947.

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

[38] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, 2024.

[39] Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture of experts large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://arxiv.org/abs/2402.14800.

[40] Bozhi Luan, Wengang Zhou, Hao Feng, Zhe Wang, Xiaosong Li, and Houqiang Li. Multi-cue adaptive visual token pruning for large vision-language models, 2025. URL https://arxiv.org/abs/2503.08019.

[41] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL*, 2022. URL https://aclanthology.org/2022.findings-acl.177/.

[42] Meta AI Team. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation. Blog post, 2025. URL https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

[43] Ruijie Miao, Yilun Yao, Zihan Wang, Zhiming Wang, Bairen Yi, LingJun Liu, Yikai Zhao, and Tong Yang. MergeMoE: Efficient compression of MoE models via expert output merging, 2025. URL https://arxiv.org/abs/2510.14436.

[44] Dung Viet Nguyen, Anh Nguyen Thi, Minh Hoang Nguyen, Luc Nguyen, Shiqi Jiang, Ethan Fetaya, Linh Duy Tran, Gal Chechik, and Tan Minh Nguyen. Expert merging in sparse mixture of experts with Nash bargaining. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=JLe9xfd0ln.

[45] Viet Dung Nguyen, Minh Nguyen Hoang, Luc Nguyen, Rachel Teo, Tan Minh Nguyen, and Linh Duy Tran. CAMEx: Curvature-aware merging of experts. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=nT2u0M0nf8.

[46] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. URL https://textvqa.org/assets/paper/TextVQA.pdf.

[47] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, 2024.

[48] Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. VL-Cache: Sparsity and modality-aware KV cache compression for vision-language model inference acceleration. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HMrcv7Q4Ub.

[49] Weiyun Wang et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025. URL https://arxiv.org/abs/2508.18265.

[50] Qiong Wu, Zhaoxi Ke, Yiyi Zhou, Xiaoshuai Sun, and Rongrong Ji. Routing experts: Learning to route dynamic experts in multi-modal large language models, 2024. URL https://arxiv.org/abs/2407.14093.

[51] Zhiyu Wu et al. DeepSeek-VL2: Mixture-of-experts vision-language models for advanced multimodal understanding, 2024. URL https://arxiv.org/abs/2412.10302.

[52] xAI Corp. Grok-1.5 vision preview: Connecting the digital and physical worlds with our first multimodal model. xAI news post, 2024. URL https://x.ai/news/grok-1.5v.

[53] Guoyang Xia, Yifeng Ding, Fengfa Li, Lei Ren, Wei Chen, Fangxiang Feng, and Xiaojie Wang. FastMMoE: Accelerating multimodal large language models through dynamic expert activation and routing-aware token pruning, 2025. URL https://arxiv.org/abs/2511.17885.

[54] Guoyang Xia, Yifeng Ding, Fengfa Li, Lei Ren, Wei Chen, Fangxiang Feng, and Xiaojie Wang. SMAR: Soft modality-aware routing strategy for MoE-based multimodal large language models preserving language capabilities, 2025. URL https://arxiv.org/abs/2506.06406.

[55] Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. MoE-Pruner: Pruning mixture-of-experts large language model using the hints from its router, 2024. URL https://arxiv.org/abs/2410.12013.

[56] Haolei Xu, Haiwen Hong, Hongxing Li, Rui Zhou, Yang Zhang, Longtao Huang, Hui Xue, Yongliang Shen, Weiming Lu, and Yueting Zhuang. Seeing but not thinking: Routing distraction in multimodal mixture-of-experts, 2026. URL https://arxiv.org/abs/2604.08541.

[57] An Yang et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[58] Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Yuanlin Duan, Wenqi Jia, Miao Yin, Yu Cheng, and Bo Yuan. MoE-I^2: Compressing mixture of experts models through inter-expert pruning and intra-expert low-rank decomposition. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10456–10466, 2024. URL https://aclanthology.org/2024.findings-emnlp.612/.

[59] Jiangyong Yu, Sifan Zhou, Dawei Yang, Shuo Wang, Shuoyu Li, Xing Hu, Chen Xu, Zukang Xu, Changyong Shu, and Zhihang Yuan. MQuant: Unleashing the inference potential of multimodal large language models via full static quantization, 2025. URL https://arxiv.org/abs/2502.00425.

[60] Geng Zhang, Yuxuan Han, Yuxuan Lou, Yiqi Zhang, Wangbo Zhao, and Yang You. MoNE: Replacing redundant experts with lightweight novices for structured pruning of MoE, 2025. URL https://arxiv.org/abs/2507.00390.

[61] Yang Zhang, Yawei Li, Xinpeng Wang, Qianli Shen, Barbara Plank, Bernd Bischl, Mina Rezaei, and Kenji Kawaguchi. FinerCut: Finer-grained interpretable layer pruning for large language models. In *Compression Workshop at NeurIPS*, 2024. URL https://openreview.net/forum?id=jrSWzgno4W.

[62] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference, 2024. URL https://arxiv.org/abs/2410.04417.

[63] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

[64] Yixiao Zhou, Ziyu Zhao, Dongzhou Cheng, Zhiliang Wu, Jie Gui, Yi Yang, Fei Wu, Yu Cheng, and Hehe Fan. Dropping experts, recombining neurons: Retraining-free pruning for sparse mixture-of-experts LLMs, 2025. URL https://arxiv.org/abs/2509.10377.

A Limitations

MAEP applies a fixed prune mask per layer, without modeling the joint removal of experts that back each other up or input-conditional sparsity. Moreover, the adaptive α rule is empirically validated rather than derived from a theoretical optimum. Finally, MAEP also targets the training-free setting; fine-tuning (e.g., LoRA [18]) is orthogonal and could recover residual gaps.

Pairwise redundancy and one-shot scoring. ΔH is not a fully independent score: its redistribution counterfactual (Section 5.1) measures the residual output change after the remaining co-activated experts absorb the removed expert's routing mass, so first-order redundancy among co-activated experts is captured implicitly. What it does not model is the joint removal of experts that back each other up. To quantify how much this matters at our operating ratio, we compared the one-shot 50% prune mask on InternVL3.5 against a two-stage variant that prunes 25%, re-scores ΔH on the pruned model, and then prunes to 50% in total. The two masks differ substantially (Jaccard 0.77; 800 of the 3,472 experts in their union differ), but the two-stage variant is strictly worse (one-shot $\rightarrow$ two-stage: MME $2160.3 \rightarrow 1956.7$, MMStar $63.3 \rightarrow 53.9$, ChartQA $76.7 \rightarrow 54.3$, TextVQA $75.6 \rightarrow 64.9$). We attribute this to re-scoring on an already-redistributed model: ΔH re-measured after partial pruning no longer reflects the original model's redundancy structure, whereas the one-shot score models that redistribution analytically without altering the model it measures. This experiment therefore does not isolate interaction effects, and we leave them, as well as input-adaptive or iterative pruning at more aggressive ratios, open.

Calibration-domain sensitivity. As with other calibration-based post-training compression methods, MAEP's prune decisions depend on the calibration distribution: domains that the calibration set under-represents can lead to under-scored experts (Section 6.3.2). This sensitivity is confined to the text-side preservation signal, whereas the low-attention vision signal is stable across calibration datasets (Appendix G.3); it is therefore controllable rather than intrinsic, and we do not claim domain invariance. We recommend (i) a diverse calibration mixture as the default; (ii) augmenting it with in-domain data when the deployment domain is known (Table 2(b)); and (iii) when the domain is unknown, broadening the mixture with chart-, document-, and OCR-style data as an inexpensive safeguard against under-coverage.

Promising future directions include layer-wise α adaptation, calibration-domain mixing strategies for deployment specialization, and stacking with orthogonal compression families (quantization, intra-expert decomposition).

B Proof of the Motivational Bound (Proposition 1)

Proposition. *Let $\Phi_l : \mathbb{R}^{T\times d} \rightarrow \mathbb{R}^{T\times|V|}$ map the layer-l MoE outputs of a T-token sequence to the pre-softmax logits, with Lipschitz constant L_{Φ_l} (Frobenius norm) on the activation domain induced by RMSNorm/LayerNorm. With $\hat{h}_t := H^{(l)}_{MoE}(h_t)$ and $\hat{h}^{(-j)}_t := H^{(l,-j)}_{MoE}(h_t)$ the MoE outputs at position t before/after removing expert j, $\hat{H} := (\hat{h}_1, \ldots, \hat{h}_T)$ and $\hat{H}^{(-j)}$ the corresponding sequences, and $p_t := \mathrm{softmax}(\Phi_l(\hat{H})_t)$, $p^{(-j)}_t := \mathrm{softmax}(\Phi_l(\hat{H}^{(-j)})_t)$,*

$$\frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}\big(p_t \,\|\, p^{(-j)}_t\big) \leq \tfrac{L^2_{\Phi_l}}{4}\,\frac{1}{T}\sum_{t=1}^{T} \|\hat{h}_t - \hat{h}^{(-j)}_t\|_2^2.$$

Proof. Set $\Delta\hat{H} := \hat{H} - \hat{H}^{(-j)} \in \mathbb{R}^{T\times d}$, whose rows are $\hat{h}_t - \hat{h}^{(-j)}_t$, and $\Delta Z := \Phi_l(\hat{H}) - \Phi_l(\hat{H}^{(-j)}) \in \mathbb{R}^{T\times|V|}$, whose rows Δz_t are the logit perturbations at position t.

Step 1 (logit perturbation). Self-attention couples tokens within a sequence, so we view Φ_l as a sequence-level map $\mathbb{R}^{T\times d} \rightarrow \mathbb{R}^{T\times|V|}$ and bound $\|\Delta Z\|_F$ in terms of $\|\Delta\hat{H}\|_F$. Each subsequent residual block $f_i(X) = X + g_i(X)$ is $(1 + L_{g_i})$-Lipschitz in the Frobenius norm on the bounded activation domain, where L_{g_i} is the local Lipschitz constant of the attention+FFN branch g_i as a map on the whole sequence; the final RMSNorm/LayerNorm before the LM head acts row-wise and is locally L_{norm}-Lipschitz there, and the LM head acts row-wise with constant $\|W\|_{\mathrm{op}}$ (for row-wise

maps the per-row ℓ_2 Lipschitz constant is also the Frobenius constant); composing,

$$\|\Delta Z\|_F \;\le\; \|W\|_{\mathrm{op}} \cdot L_{\mathrm{norm}} \prod_{i=l+1}^{L} (1 + L_{g_i})\, \|\Delta \hat{H}\|_F \;=:\; L_{\Phi_l}\, \|\Delta \hat{H}\|_F, \tag{5}$$

where $\|\Delta Z\|_F^2 = \sum_t \|\Delta z_t\|_2^2$ and $\|\Delta \hat{H}\|_F^2 = \sum_t \|\hat{h}_t - \hat{h}_t^{(-j)}\|_2^2$.

Step 2 (logit-to-KL, per position). Let $A(z) := \log \sum_i e^{z_i}$, $p := \mathrm{softmax}(z)$, $q := \mathrm{softmax}(z')$. Then $\nabla A(z) = p$ and $\nabla^2 A(z) = \mathrm{diag}(p) - pp^\top$. For any unit v, Popoviciu's inequality gives $\mathrm{Var}_p[v] \le \frac{1}{4}(\max v - \min v)^2$, and $(\max v - \min v)^2 \le 2(a^2 + b^2) \le 2\|v\|_2^2 = 2$ ($a := \max v$, $b := \min v$; the first step uses $(a-b)^2 \le 2(a^2+b^2)$ and the second that a, b are two coordinates of the unit vector v), so A is $\frac{1}{2}$-smooth. By Bregman/KL duality [4] and the descent lemma,

$$D_{\mathrm{KL}}(p\|q) = A(z') - A(z) - \nabla A(z)^\top (z' - z) \le \tfrac{1}{4}\|\Delta z\|_2^2. \tag{6}$$

Applied at position t with $z = \Phi_l(\hat{H})_t$ and $z' = \Phi_l(\hat{H}^{(-j)})_t$, this gives $D_{\mathrm{KL}}(p_t \| p_t^{(-j)}) \le \frac{1}{4}\|\Delta z_t\|_2^2$ for every t.

Step 3 (combine). Summing the per-position bound of Step 2 over the T positions and applying (5),

$$\sum_{t=1}^{T} D_{\mathrm{KL}}\big(p_t \,\|\, p_t^{(-j)}\big) \;\le\; \tfrac{1}{4}\|\Delta Z\|_F^2 \;\le\; \tfrac{L_{\Phi_l}^2}{4}\, \|\Delta \hat{H}\|_F^2 \;=\; \tfrac{L_{\Phi_l}^2}{4} \sum_{t=1}^{T} \|\hat{h}_t - \hat{h}_t^{(-j)}\|_2^2,$$

which, divided by T, is the proposition. Averaging over the calibration distribution (samples $x \sim \mathcal{D}$ with $|x|$ tokens $h \in x$), for each expert j,

$$\mathbb{E}_{x\sim\mathcal{D}}\Big[\tfrac{1}{|x|}\sum_{h\in x} D_{\mathrm{KL}}\Big] \;\le\; \tfrac{L_{\Phi_l}^2}{4}\, \mathbb{E}_{x\sim\mathcal{D}}\Big[\tfrac{1}{|x|}\sum_{h\in x} \|\hat{h} - \hat{h}^{(-j)}\|_2^2\Big] \;=\; \tfrac{L_{\Phi_l}^2}{4}\, \Delta H^{(l,j)}, \tag{7}$$

where $L_{\Phi_l} := \sup_x L_{\Phi_l(\,\cdot\,;x)} < \infty$ on the bounded activation domain. □

Remark (layer-depth effect). L_{Φ_l} contains $\prod_{i=l+1}^{L}(1+L_{g_i})$, which contracts as $l \to L$ (with $L_{\Phi_L} = \|W\|_{\mathrm{op}} L_{\mathrm{norm}}$); the bound tightens for deeper layers and the spectral-norm bound $\lambda_{\max}(\mathrm{diag}(p) - pp^\top) \le 1/2$ is loose for sharp vocabulary distributions, both consistent with the empirical findings of Section 5.1.1 and Appendix I.

C Model Configurations and Memory Analysis

Table 4: MoE model configurations and memory footprint under expert pruning (BF16).

	Configuration						Memory (GB)		
Model	Total	Expert	Layers	E/L	Top-k	Router	BF16	25%	50%
InternVL3.5	30.8B	29.0B (94%)	48	128	8	softmax	57.5	44.0 (-23%)	30.5 (-47%)
Qwen3-VL	31.1B	29.0B (93%)	48	128	8	softmax	57.9	44.4 (-23%)	30.9 (-47%)
Gemma4-26B-A4B	25.8B	22.8B (88%)	30	128	8	softmax	48.1	37.4 (-22%)	26.8 (-44%)

In Table 4, memory columns are reported in GB (BF16 weights). Expert parameters constitute the dominant share of total model size across all three models. For InternVL3.5 and Qwen3-VL, 50% pruning reduces the BF16 weight footprint from ∼58 GB to ∼31 GB, placing model weights within the nominal capacity of a single 40 GB GPU. Actual deployment capacity additionally depends on KV-cache size, the vision tower and projector, activations, framework overhead, context length, and batch size. Gemma4-26B-A4B shares a similar MoE configuration (128 experts, top-8) with a shallower backbone (30 layers), providing architectural diversity for validating MAEP across different model scales.

C.1 Measured Deployment Metrics

Table 5 reports end-to-end measurements for InternVL3.5-30B-A3B on a single NVIDIA A100 80GB (HuggingFace Transformers with `accelerate`; greedy decoding). Since top-k is unchanged by pruning, per-token FLOPs and single-sequence latency are invariant by design, and we confirm this empirically: the $B{=}1$ decode-throughput ratio (50%-pruned / unpruned) is at parity within run-to-run noise. The memory saving translates into throughput in two serving regimes. (a) *Memory-capped serving*: under a 40 GB device-map cap, the unpruned model must offload $\sim$33% of its parameters to CPU, whereas the 50%-pruned model fits fully resident and decodes $3.9\times$ faster. (b) *Large-batch serving*: MoE decode is bandwidth-bound, and the per-step cost tracks the number of unique activated experts, which grows with the batch size and is bounded by the size of the expert pool; halving the pool lowers this ceiling, and the decode-throughput ratio rises from parity at $B{=}1$ to $1.37\times$ at $B{=}64$ with diverse-prompt batches (identical-prompt batches keep the per-step expert union at top-k and show parity, consistent with the mechanism). These measurements use HuggingFace's non-fused MoE kernels; fused grouped-GEMM engines are more strictly bandwidth-bound and would likely amplify effect (b). Loading time is dominated by weight I/O and shrinks with the serialized checkpoint, which is 27 GB smaller at 50% pruning.

Table 5: Measured deployment metrics for InternVL3.5-30B-A3B (A100 80GB; HuggingFace Transformers + `accelerate`). Protocol: greedy decoding of a fixed short prompt ($\sim$25 tokens after the chat template), generating 128 new tokens when resident and 32 when offloaded (tok/s is rate-normalized; the wall clock includes the single prefill pass); the batch sweep generates 64 new tokens per sequence; warm-up runs excluded, CUDA-synchronized timing, averaged over 2–3 runs.

Metric	Measured
Resident weight memory at 0 / 25% / 50% pruning	57.5 / 44.0 / 30.5 GB (-47% at 50%)
$B{=}1$ decode-throughput ratio (50%-pruned / unpruned)	0.96–1.02 (parity within run-to-run noise)
Decode under a 40 GB memory cap	unpruned ($\sim$33% CPU-offloaded): 1.52 tok/s; 50%-pruned (fully resident): 5.88 tok/s ($3.9\times$)
Large-batch decode-throughput ratio (50%-pruned / unpruned)	1.13 ($B{=}8$) / 1.29 ($B{=}16$, $B{=}32$) / 1.37 ($B{=}64$)
Unique activated experts per layer per decode step at $B{=}64$	$\sim$55 (unpruned) vs. $\sim$39 (50%-pruned)

D Experimental Settings

D.1 Implementation Details

Hardware. All experiments are conducted on a single NVIDIA A100 80GB GPU. Calibration requires a single forward pass over the 1K calibration samples per model (no back-propagation or optimizer state); the remaining compute is benchmark inference.

Calibration Data. Following [32], which shows that image captioning data provides balanced coverage of both visual and linguistic processing, we construct the calibration set from ShareGPT4V [7]. Samples are sorted in descending order of text length; the top 1K samples form the **calibration set** used for modality-decomposed importance scoring. The set does not overlap with the downstream evaluation benchmarks. Longer text sequences provide richer token-level statistics for expert importance estimation, as they activate a broader range of expert routing patterns during the prefill stage. We report cross-calibration-source variants in Section 6.3.2.

HC-SMoE Configuration. We use HC-SMoE with hierarchical agglomerative clustering on expert-output similarity, average linkage, and frequency-weighted merging (`similarity-base=expert-output`, `cluster-method=hierarchical`, `linkage=average`, `merge-method=freq`). This corresponds to the default configuration in [5], applied to MoE-MLLM backbones in our setting.

REAP Configuration. We follow the original REAP [27] formulation, sharing the same calibration set as MAEP, and apply uniform per-layer pruning at the target ratio.

DiEP Configuration (Appendix E.2). DiEP [2] is reimplemented in our pipeline; on synthetic MoE blocks, the reimplementation reproduces the reference code's MoE forward outputs and router logits bit-exactly and selects the same prune map for a given budget. We use DiEP's published hyperparameters (128 calibration sequences drawn from the same ShareGPT4V calibration set as MAEP, 10 epochs, learning rate $5{\times}10^{-3}$ with a cosine schedule, sparsity weight $\lambda{=}0.01$) and the identical evaluation pipeline and judge. DiEP's differentiable search is ratio-independent, so a single

search yields both the 25% and 50% masks; every layer keeps at least top-k experts. Text-only benchmarks were not run for DiEP.

D.2 Evaluation Details

We evaluate on 11 benchmarks spanning two categories:

Multimodal Benchmarks.

- **ChartQA** [41]: Chart understanding with extractive and reasoning questions. Metric: relaxed accuracy.
- **TextVQA** [46]: Visual question answering requiring OCR of scene text. Metric: VQA accuracy.
- **MMStar** [8]: Hand-curated, vision-indispensable multi-discipline reasoning samples with minimal data leakage. Metric: accuracy.
- **MMBench** [38]: Comprehensive multimodal evaluation across 20 ability dimensions. We use the MMBENCH_DEV_EN_V11 split (English dev set, version 1.1) Metric: accuracy.
- **RealWorldQA** [52]: Real-world spatial understanding and reasoning from natural images. Metric: accuracy.
- **MME** [13]: Perception and cognition evaluation via yes/no questions. Metric: perception + reasoning score.
- **CV-Bench** [47]: Vision-centric perception benchmark with 2D (object counting, relation, spatial relations) and 3D (depth, distance) splits; we report the average of the 2D and 3D scores. Metric: accuracy.
- **POPE** [35]: Object hallucination evaluation via polling-based binary questions. Metric: F1 score.

Text-Only Benchmarks. We additionally include text-only benchmarks to measure the effect of expert pruning guided by multimodal signals on language-only capabilities.

- **MMLU-Redux** [14]: A manually re-annotated subset of MMLU across 30 subjects with corrected or removed erroneous ground-truth labels, yielding a cleaner multi-domain knowledge benchmark. We use the corrected subset and follow the OpenCompass 0-shot no-CoT protocol. Metric: accuracy.
- **IFEval** [63]: Instruction-following benchmark composed of prompts with verifiable constraints. Metric: prompt-strict accuracy.
- **ARC Challenge** [9]: Science exam questions requiring multi-step reasoning beyond simple retrieval. Metric: accuracy.

Evaluation Pipeline. All benchmarks are evaluated using VLMEvalKit [12], with the OpenCompass [10] protocol for text-only benchmarks. For MCQ benchmarks (MMBench, MMStar, RealWorldQA, CV-Bench, MMLU-Redux, ARC), we use GPT-3.5-TURBO-0125 as the answer-extraction judge. We evaluate the standard INSTRUCT variants of InternVL3.5-30B-A3B and Qwen3-VL-30B-A3B (the explicit THINKING variants are *not* used), so no model-level chain-of-thought reasoning is engaged. All pruned models—MAEP, HC-SMoE, and REAP—share the same calibration set, and all models, including the unpruned baselines, are evaluated under identical protocols.

E Additional Results

E.1 Separate Multimodal and Text-Only Averages

Table 6 splits the "Avg." column of Table 1 into the mean retention over the eight multimodal benchmarks and over the three text-only benchmarks. At 50%, MAEP attains the best multimodal average on all three models; on text-only benchmarks it leads on InternVL3.5 and Gemma4 and trails REAP on Qwen3-VL. At 25%, MAEP again has the best multimodal average on all three models; where its text-only average trails, the gap is at most 1.7 points, and its 11-benchmark average remains best or tied on every model.

Table 6: Mean performance retention (%) over the eight multimodal benchmarks (MM), the three text-only benchmarks (Text), and all eleven (All; the "Avg." column of Table 1). Bold marks the best method per model, ratio, and column.

Model	Ratio	Method	MM	Text	All
InternVL3.5-30B-A3B	25%	HC-SMoE	97.2	77.6	91.9
		REAP	97.9	90.5	95.9
		MAEP	**98.6**	**95.3**	**97.7**
	50%	HC-SMoE	86.4	41.4	74.1
		REAP	91.6	80.3	88.5
		MAEP	**94.0**	**83.8**	**91.2**
Qwen3-VL-30B-A3B	25%	HC-SMoE	96.2	**95.4**	96.0
		REAP	94.7	95.1	94.8
		MAEP	**97.5**	94.0	**96.6**
	50%	HC-SMoE	89.6	74.7	85.5
		REAP	88.4	**81.8**	86.6
		MAEP	**92.2**	77.9	**88.3**
Gemma4-26B-A4B	25%	HC-SMoE	74.7	83.8	77.2
		REAP	96.9	**93.9**	**96.0**
		MAEP	**97.4**	92.2	**96.0**
	50%	HC-SMoE	38.9	40.6	39.3
		REAP	85.2	69.3	80.9
		MAEP	**89.4**	**79.6**	**86.7**

E.2 Comparison with DiEP

DiEP [2] learns a non-uniform, layer-wise expert budget by differentiable search while scoring experts in a modality-agnostic way; it therefore shares MAEP's allocation axis (global, non-uniform) but not its scoring. Table 7 compares the two on InternVL3.5-30B-A3B under the identical calibration set, pipeline, and total budget (configuration in Appendix D). MAEP leads at both ratios and on nearly every benchmark, and the gap widens with sparsity; DiEP's one win is MME at 50% (the two are essentially tied on MME at 25%, and POPE separates no method). DiEP's largest deficit is ChartQA, the benchmark most sensitive to which experts survive (Section 6.3.2). DiEP's learned allocation is far more depth-skewed than MAEP's (averaged within depth quartiles it prunes 11/11/50/56 experts per layer at 25%, removing up to 69 of 128 experts from a single layer, and leaves as few as 37 experts in a layer at 50%), yet under the same budget its retention falls short. Combined with the decomposition in Appendix L, this indicates that the operative factor is which experts are selected rather than the non-uniform budget itself. DiEP additionally requires gradient-based search over its relaxation parameters, whereas every MAEP score comes from a single forward pass.

Table 7: DiEP vs. MAEP on InternVL3.5-30B-A3B (each cell: 25% / 50% expert pruning; identical calibration set, pipeline, and judge). MM-Ret. is the mean retention over these eight multimodal benchmarks. HC-SMoE and REAP are reported in Table 1.

Benchmark	DiEP	MAEP (ours)
MME	2364.3 / **2195.6**	**2364.9** / 2160.3
MMBench	79.2 / 66.7	**83.2 / 80.7**
MMStar	61.8 / 49.7	**69.3 / 63.3**
ChartQA	54.0 / 40.4	**87.0 / 76.7**
TextVQA	74.6 / 67.1	**78.9 / 75.6**
RealWorldQA	70.6 / 64.7	**71.6 / 71.6**
CV-Bench	81.3 / 69.4	**84.3 / 83.3**
POPE	89.2 / 89.5	**89.7 / 89.7**
MM-Ret. (%)	90.6 / 79.9	**98.6 / 94.0**
Parameter optimization	gradient search	none (single forward pass)

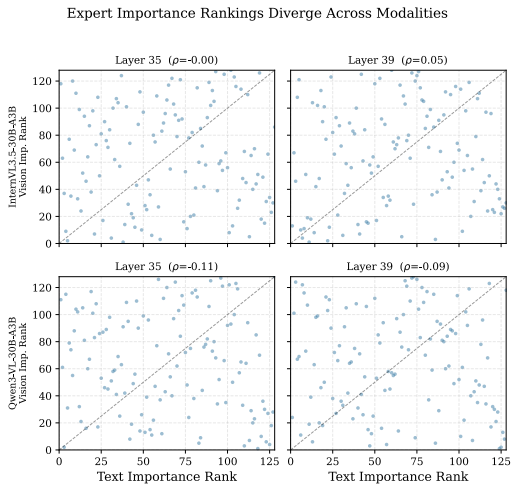


Figure 3: Text vs. vision expert importance ranking (ΔH) for InternVL3.5-30B-A3B and Qwen3-VL-30B-A3B at layers 35 and 39 (selected for their near-zero or negative Spearman ρ). Each point is one expert; the dashed line indicates perfect agreement.

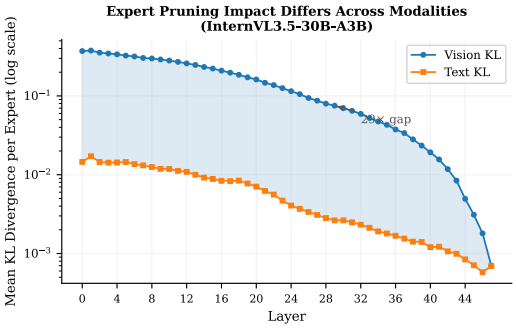


Figure 4: Mean per-expert KL divergence at the LM-head output, on vision-token (blue) vs. text-token (orange) positions, per layer for InternVL3.5-30B-A3B. Each curve averages over experts removed one at a time within the layer; the gap reaches up to $29\times$, indicating that expert removal perturbs vision-position outputs substantially more than text.

F Modality Gap in Expert Importance

We provide empirical evidence that expert importance is modality-dependent. Figure 3 plots text-side vs. vision-side ΔH for each expert at layers 35 and 39. If experts served both modalities equally, all points would lie on the diagonal; the observed scatter indicates that experts tend to specialize by modality to varying degrees.

Figure 4 further quantifies this gap by measuring the actual KL divergence at the LM head when each expert is removed. The vision-side KL (blue) consistently exceeds the text-side KL (orange) across all layers, with the gap reaching $29\times$ in shallow layers.

The per-model heterogeneity in Table 8 reflects differences in modality-decomposed importance structure: Qwen3-VL exhibits near-zero $\bar{\rho}_{\text{t,v}}$ (rankings essentially independent across modalities), while Gemma4 shows high $\bar{\rho}_{\text{v,la}}$ (vision and low-attention vision rankings are strongly aligned), driving its small α.

Table 8: Layer-averaged Spearman correlation between expert importance rankings under three modality views (text, full vision, low-attention vision) on the ShareGPT4V calibration set, computed per layer and averaged across all MoE layers. The closed-form expression $\alpha = a/(a+b)$ with $a := \bar{\rho}_{\text{t,v}} - \bar{\rho}_{\text{t,la}}$ and $b := \bar{\rho}_{\text{v,la}}$ recovers the per-model values used in the main results.

Model	$\bar{\rho}_{\text{t,v}}$	$\bar{\rho}_{\text{t,la}}$	$\bar{\rho}_{\text{v,la}}$	closed-form α
InternVL3.5-30B-A3B	$+0.38$	-0.27	$+0.22$	0.75
Qwen3-VL-30B-A3B-Instruct	$+0.01$	-0.22	$+0.46$	0.33
Gemma4-26B-A4B-Instruct	$+0.63$	$+0.55$	$+0.91$	0.08

This modality gap motivates decomposing importance by modality in our combined scoring (Eq. 4), where text-side ΔH acts as a preservation signal and low-attention vision-side ΔH acts as a pruning signal (Section 5.2; the sign convention is verified in Appendix I.1).

G Cross-Dataset Stability of Expert Rankings

A key design choice in MAEP is to use low-attention vision tokens rather than all vision tokens or high-attention ones. Figure 5 validates this choice by measuring the cross-dataset stability of expert importance rankings: we compute ΔH-based rankings using ShareGPT4V and ChartNet as separate calibration sets and report the Spearman ρ between them. The chart-domain source (ChartNet) is disjoint from the evaluation suite and matches the setup used in Section 6.3.2 and Figure 2.

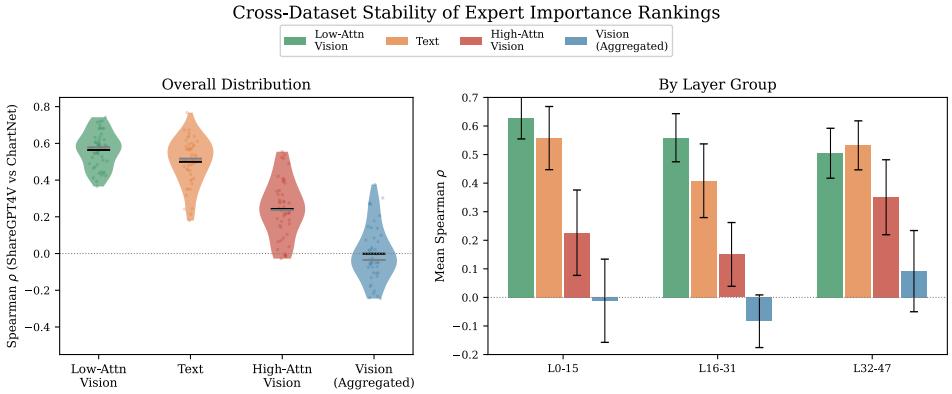


Figure 5: Cross-dataset stability of expert importance rankings (Spearman ρ between ShareGPT4V and ChartNet calibration sets) for InternVL3.5-30B-A3B. Left: overall distribution of per-layer ρ by token type (violins with mean and median markers). Right: mean ρ by layer group (L0–15, L16–31, L32–47). Low-attention vision tokens yield the most stable rankings across calibration sources ($\bar{\rho}\approx 0.56$), substantially exceeding text (≈ 0.50), high-attention vision (≈ 0.24 at $q{=}20\%$; the matched-size $q{=}2\%$ value reported in §6.3.1 is higher), and full vision (≈ 0); this supports their use as the vision-side pruning signal.

G.1 Alternative Vision-Token Selectors

Since expert pruning operates in the MoE blocks, attention is not the only conceivable signal for selecting the visual calibration tokens. Table 9 screens seven selectors on the two properties a calibration signal needs: stability of the induced expert ranking across calibration datasets (ρ_{stab}; ShareGPT4V vs. ChartQA images, the latter used only as a held-out image set and not for calibrating any pruned model in this paper) and distinctness from the text signal (ρ_{dist}; lower is more distinct). Besides the three attention-based signals, we include representation drift (one minus the cosine similarity between a token's input and output representations across the MoE layers; keeping the highest-drift tokens is also the selection rule of Li et al. [33]), diversity (farthest-point sampling), and random selection. For tractability the screening uses routing-mass importance rather than the exact ΔH; it reproduces the ΔH-based ordering of Figure 5 for the attention-based signals, so only the relative comparison is used.

Low-attention is the only selector that is both the most stable and the most distinct from the text signal; full-vision and random are unstable, drift-low is only weakly stable, and high-attention, drift-high, and diversity produce text-like importance. Run end-to-end through the full ΔH calibration (InternVL3.5, 50%; only the vision-token subset changed), the drift-high rule selects a very different expert set (Jaccard 0.33 against MAEP's) and loses on all four ablation benchmarks (MAEP $\rightarrow$ drift-high: MME $2160 \rightarrow 1490$, MMStar $63.3 \rightarrow 49.9$, TextVQA $75.6 \rightarrow 41.9$, ChartQA $76.7 \rightarrow 71.5$). Attention received from the answer-generating text stream measures whether a token's content reaches the output, which is what pruning must decide; drift and diversity instead measure how strongly a token is transformed or how it spreads in representation space.

G.2 Spatial Location of Low-Attention Tokens

To verify that bottom-2% low-attention tokens correspond to non-salient content, we mapped them onto the whole-image thumbnail grid of InternVL3.5-30B-A3B (ShareGPT4V calibration images; 60 images $\times$ 48 layers; per-layer bottom 2%) and compared them with a spectral-residual saliency

Table 9: Vision-token selector screening on InternVL3.5-30B-A3B (routing-mass importance; ShareGPT4V vs. ChartQA images as a held-out set, 60 samples each; every subset selector keeps 2% of the vision tokens). Higher ρ_{stab} is better; lower ρ_{dist} means more distinct from the text signal.

Selector	ρ_{stab} (stability)	ρ_{dist} (vs. text; low = distinct)
Low-attention (ours)	**0.51**	$\mathbf{-0.12}$
High-attention	0.30	0.47
Full vision (no subset)	-0.23	0.30
Drift-low	0.19	0.01
Drift-high	0.29	0.34
Diversity (farthest-point)	0.36	0.36
Random (3 seeds)	-0.23	0.30
Text signal (reference)	0.30	—

map of each image (Table 10). Low-attention tokens fall overwhelmingly outside salient regions: a bottom-2% token is about $2.9\times$ less likely than an average vision token to lie in the top-50% saliency-mass region, and about $2.8\times$ less likely to lie in the Otsu-thresholded foreground. A purely geometric border measure is uninformative here, because the image border attracts the *highest*-attention tokens (top-2% tokens are about $1.8\times$ over-represented on the border ring, a fixed-position pattern consistent with attention-sink behavior), whereas bottom-2% tokens are under-represented there ($0.69\times$ the area expectation).

Table 10: Saliency statistics of bottom-2% low-attention vision tokens (InternVL3.5-30B-A3B, ShareGPT4V calibration; 60 images $\times$ 48 layers). Base rate: the same statistic over all vision tokens.

Statistic	Low-attention tokens	Base rate
Fraction outside the top-50% saliency mass	92.0%	76.5%
Fraction in the saliency-map background (Otsu threshold)	89.5%	70.1%

G.3 Which Signal Carries Calibration-Domain Sensitivity

Table 2(b) shows that changing the calibration domain changes the prune decisions. To locate where this sensitivity enters, we swapped the calibration data of each signal separately on Qwen3-VL at 50% pruning and compared the resulting prune map with the original ShareGPT4V map (Table 11). Swapping only the text-side calibration to the ChartNet mixture moves the prune map about $3\times$ more than swapping only the vision-side signal (as the ratio of decision-flip rates) and reproduces most of the mask movement of the full mixture: of the experts whose prune/keep decision flips under the full mixture, 82% also flip under the text-only swap, vs. 13% under the vision-only swap. Domain composition is thus carried by the text-side preservation signal, while the low-attention vision signal stays stable, consistent with Figure 5; the practical guidance that follows is given in Appendix A.

Table 11: 2×2 calibration swap on Qwen3-VL-30B-A3B (50% pruning): Jaccard similarity of the resulting prune map to the original ShareGPT4V map, and the number of prune/keep decisions that flip (of 6,144 experts; each rescued expert pairs with a newly pruned one).

Calibration swapped to the ChartNet mixture	Jaccard vs. original	Decisions flipped
Vision (low-attention) side only	0.922	248
Text side only	0.776	776
Both (full mixture; Table 2(b))	0.780	760

H Validation of Fixed Top-k Counterfactual

Our importance metric (Eq. 2) uses a fixed top-k counterfactual that does not evaluate the promoted $(k{+}1)$-th expert, whose output is set to zero. An alternative is to evaluate the promoted $(k{+}1)$-th expert so that exactly k experts remain active, matching the router's actual behavior in the pruned

model. We compare the two counterfactuals on InternVL3.5-30B-A3B by computing ΔH under each variant for every expert in every layer.

Table 12 shows that the two counterfactuals produce nearly identical rankings and pruning decisions across all modalities. We therefore adopt the simpler fixed top-k variant, which avoids evaluating the promoted $(k{+}1)$-th expert at negligible cost in ranking fidelity or prune-mask selection.

Table 12: Agreement between the fixed top-k and promote-$(k{+}1)$ counterfactuals on InternVL3.5-30B-A3B. Spearman ρ is the mean across 48 layers; Jaccard is computed at 50% per-layer pruning of the lowest-score experts.

Modality	Spearman ρ	Jaccard @ 50%
Text	0.999	0.97
Low-attn vision	0.987	0.92
High-attn vision	0.994	0.93
Full vision	0.981	0.89

I Validating ΔH as an Importance Proxy

To validate ΔH as an importance proxy, we compare it against two baselines. The first is the **routing-KL** when expert j is masked—a principled distributional distance that quantifies how much the router distribution shifts when expert j is removed:

$$\Delta\mathrm{KL}^{(l,j)}_{\text{routing}} = \mathbb{E}_x\Big[D_{\mathrm{KL}}\big(p^{(l)}(x)\,\|\,p^{(l,-j)}(x)\big)\Big], \tag{8}$$

where $p^{(l,-j)}$ is obtained by zeroing p_j and renormalizing the remaining entries, matching the redistribution used in ΔH (we additionally apply Laplace smoothing with $\epsilon_{\mathrm{KL}}{=}10^{-12}$ to prevent numerical infinities without affecting rank ordering). A larger $\Delta\mathrm{KL}_{\text{routing}}$ indicates that expert j carries a more distinctive routing pattern. The second is **REAP**'s [27] router-weighted activation norm, which captures each expert's magnitude contribution:

$$s^{(l,j)}_{\text{REAP}} = \mathbb{E}_x\big[\,p^{(l)}_j(x)\cdot\|E^{(l)}_j(x)\|_2\,\big], \tag{9}$$

where $p^{(l)}_j(x)$ is the post-softmax router gate value (defined in Section 3.1) and $E^{(l)}_j(x)$ is expert j's output. Figure 6 reports the per-layer Spearman rank correlation between each of these proxies and the actual KL divergence at the LM head caused by expert removal. For computational tractability, the ground-truth KL is averaged over $n{=}10$ calibration samples per (layer, expert) pair, since exact computation requires re-running the post-MoE network once per ablated expert.

I.1 Causal Check of the Vision-Side Sign Convention

Section 5.2 treats a high ΔH_{la} as evidence for removal. To verify this sign direction at the model output, we isolated the low-attention signal ($\alpha{=}1$, 25% pruning, InternVL3.5-30B-A3B), ranked experts by ΔH_{la}, and compared three removal orders, measuring the per-token LM-head KL divergence each induces against the unpruned model (same protocol as above; $n{=}10$ calibration samples, 3,948 text and 17,152 vision positions). Table 13 shows that removing the highest-ΔH_{la} experts first perturbs the output-generating text positions least, random removal is intermediate, and the inverted order perturbs them most, which confirms that high-ΔH_{la} experts contribute least to the output distribution.

J Cross-Layer Magnitude Variation of ΔH

Raw ΔH values exhibit two properties that necessitate the log-transform and layer-wise z-score normalization described in Section 5.3.

Cross-layer magnitude variation. Figure 7 shows the mean ΔH per layer for InternVL3.5-30B-A3B and Qwen3-VL-30B-A3B on the ShareGPT4V calibration set. The deepest layer exceeds the shallowest by $\sim$1,900$\times$ (InternVL) and $\sim$9,200$\times$ (Qwen), spanning nearly four orders of magnitude. Without layer-wise normalization, global ranking would be dominated by deep-layer experts regardless of their relative importance within each layer.

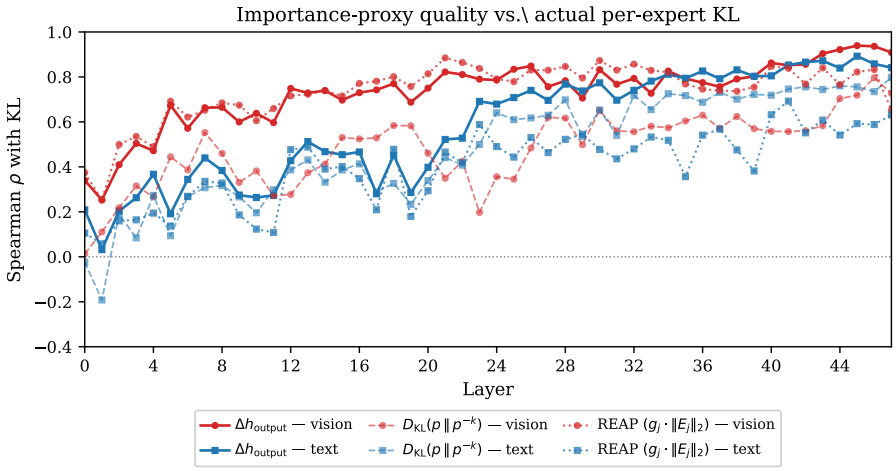


Figure 6: Per-layer Spearman ρ between importance proxies and actual KL divergence for InternVL3.5-30B-A3B. ΔH (solid) is the strongest proxy on text tokens, substantially exceeding both routing-KL (dashed) and REAP's activation-norm product (dotted); on vision tokens, ΔH and REAP are comparable, while routing-KL trails both. For vision tokens, ΔH correlation exceeds 0.8 in deep layers (L36–L47); for text tokens, ΔH rises from $\sim$0.05 in shallow layers to $\sim$0.89 in deep layers, suggesting that ΔH is a tighter proxy where it matters most.

Table 13: Mean per-token LM-head KL divergence against the unpruned model at the output-generating (text) positions, by expert-removal order on InternVL3.5-30B-A3B (α=1; 25% of experts removed from the indicated end of the ΔH_{la} ranking; n=10). At vision positions, which generate no tokens and are perturbed strongly by any removal (Figure 4), the three orders give 1.24, 1.01, and 1.30; that our order exceeds random removal there shows that the removed experts are not inert.

Removal order	KL at text positions
Highest-ΔH_{la} experts first (our sign)	**0.30**
Randomly selected experts	0.48
Lowest-ΔH_{la} experts first (inverted)	1.13

Within-layer heavy-tailed distribution. Figure 8 shows the per-expert ΔH distribution within representative layers, sorted in descending order. In both models, a small number of experts account for a disproportionate share of the total importance (e.g., the top-10 experts contribute 35–70% of the layer's total ΔH). The log-transform compresses this heavy tail, ensuring that the z-score normalization produces well-separated scores across all experts rather than being dominated by a few outliers.

K Per-Layer Pruning Distribution under Global Pruning

We highlight three observations from Figure 9. (1) Global pruning distributes the budget adaptively—some layers contribute more pruning candidates than others, rather than enforcing a fixed quota. (2) Although our algorithm does not enforce a per-layer minimum, no layer is stripped below the top-k threshold (k=8) in practice: the minimum number of remaining experts across all layers is 56 for InternVL and 59 for Qwen at 50% pruning, ensuring that sparse routing remains meaningful. At higher sparsities where a layer might lose all experts, this would correspond to our score identifying that layer's MoE as redundant; the layer naturally falls back to its residual identity path (still a valid transformer block), rather than constituting a routing failure. (3) In Qwen3-VL, the final layers (L44–L47) are pruned less aggressively under 25% pruning, reflecting a more concentrated $\tilde{z}$ distribution in deep layers (§6.3.1).

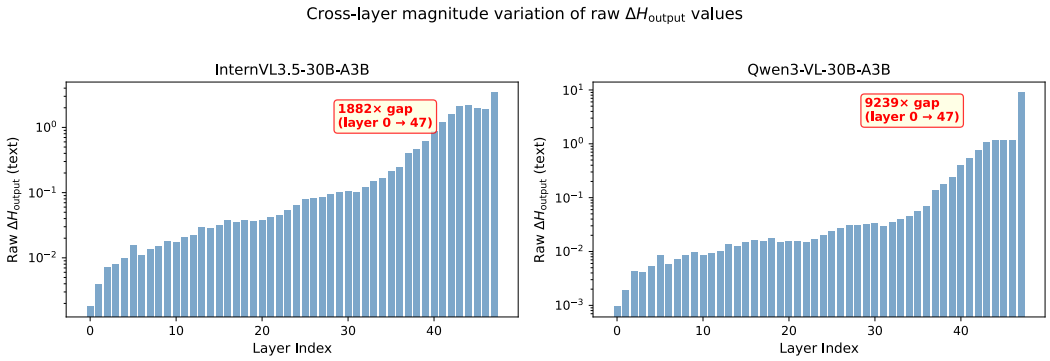


Figure 7: Mean ΔH (text tokens) per layer for InternVL3.5-30B-A3B and Qwen3-VL-30B-A3B. Y-axis is log-scaled. Deep layers produce values 10^3–$10^4\times$ larger than shallow layers, necessitating layer-wise normalization before cross-layer comparison.

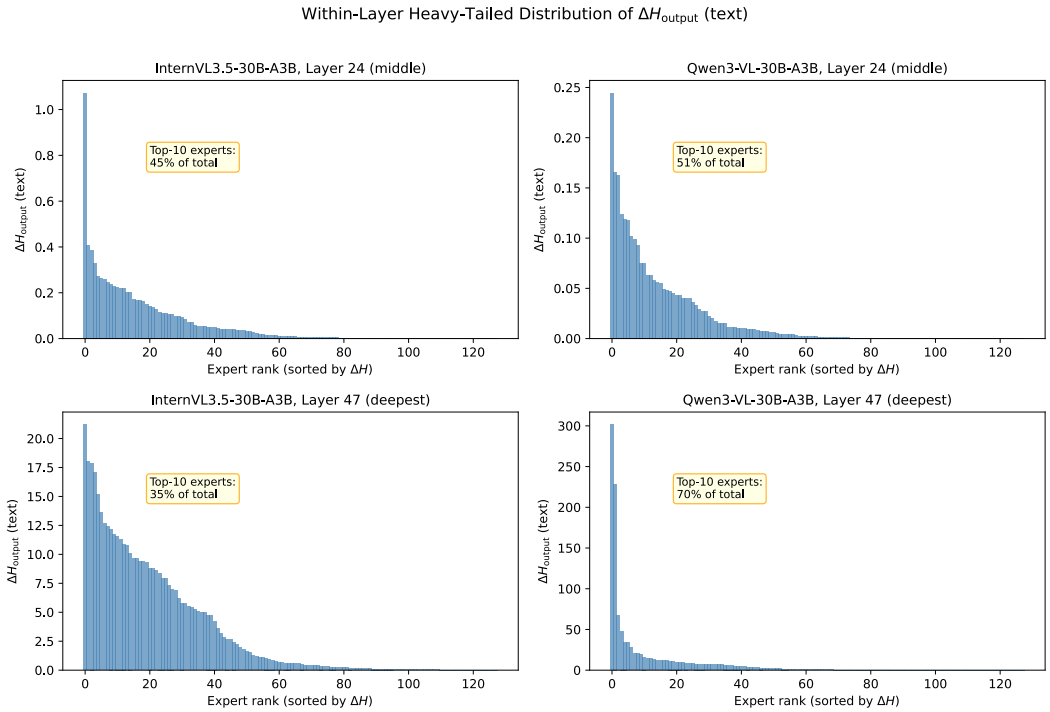


Figure 8: Within-layer distribution of ΔH (text tokens) for selected layers of InternVL3.5-30B-A3B and Qwen3-VL-30B-A3B. Expert importance follows a heavy-tailed distribution, with the top-10 experts contributing 35–70% of total ΔH per layer.

L Scoring vs. Allocation and α under a Uniform Budget

MAEP differs from the uniform baselines in Table 1 along two axes: the modality-aware score and the global, non-uniform allocation. Table 14 crosses the two on InternVL3.5-30B-A3B at 50% pruning. Applying REAP's scores under a global budget changes essentially nothing: after the same per-layer log+z-score normalization, its scores induce a near-uniform allocation (the global and uniform REAP masks agree at Jaccard 0.955; 140 of the 6,144 prune/keep decisions flip), and performance is unchanged within noise. Within the global column, the modality-aware score is ahead on all four benchmarks and decisively so on ChartQA. The DiEP comparison (Appendix E.2) holds the allocation axis fixed and varies only the scoring, so the two analyses isolate the scoring contribution from both directions.

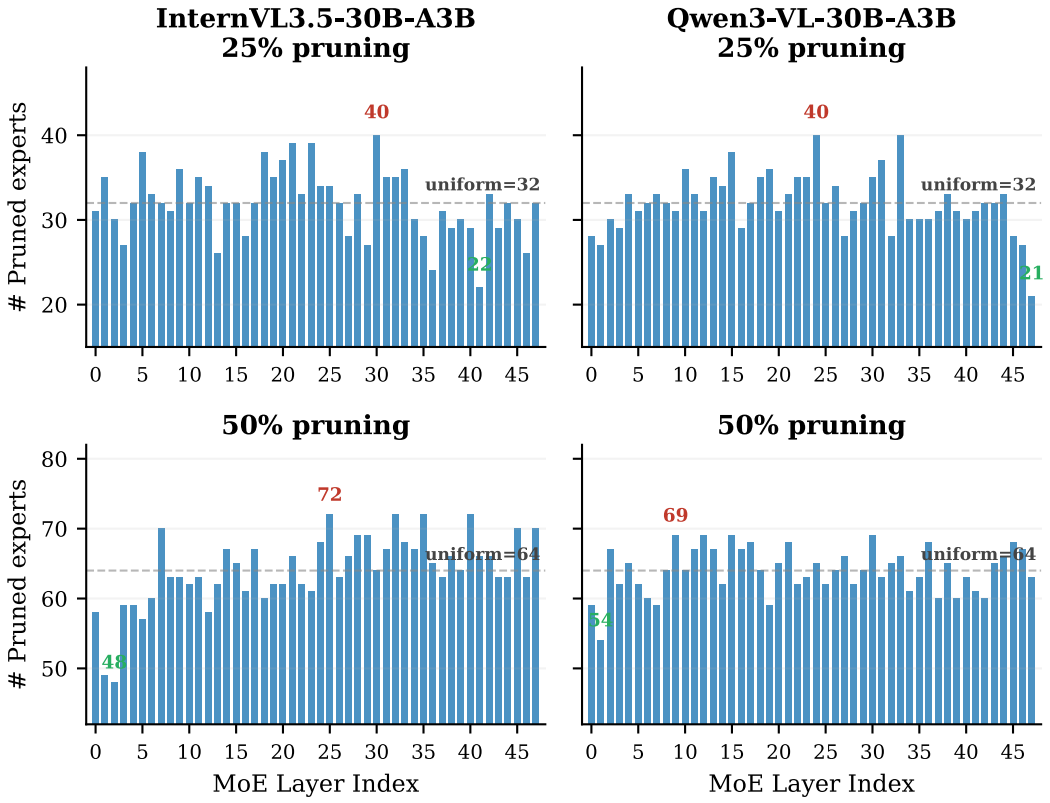


Figure 9: Per-layer pruned expert count under global pruning for InternVL3.5-30B-A3B and Qwen3-VL-30B-A3B at 25% and 50% pruning ratios. Gray dashed lines indicate the uniform per-layer baseline (32 and 64 experts, respectively); red and green annotations mark the layers with the maximum and minimum pruned-expert counts.

Why the uniform variant needs a different α. A uniform per-layer quota uses the composite score of Eq. 4 as a within-layer ordering. That score is a removability score whose terms are weighted for the global operating point: on InternVL3.5, text-side and low-attention importance are anti-correlated (Table 8), so removing the highest-ΔH_{la} experts already steers clear of text-critical ones, and the global cut stops early enough for this to hold. A forced per-layer 50% cut instead runs past the experts the low-attention signal clearly flags; from there, protection falls to the text term, whose weight $1-\alpha$ is only 0.2 at α=0.8, too little to keep text-critical experts out of the cut. The mechanism predicts that a uniform budget requires a smaller α, which Table 15 confirms: at α=0.5, uniform MAEP is at least competitive with uniform REAP on every benchmark (within about one point on MME and MMStar, ahead on ChartQA and TextVQA); TextVQA stays flat and high across the whole band $\alpha \leq 0.5$, ChartQA improves with more vision-side evidence up to α=0.5, and $\alpha \geq 0.7$ is unstable under a uniform quota. The recovery is not a convergence to REAP's selection: the re-tuned masks remain far from REAP's (Jaccard 0.47–0.49 across the band). We therefore report the re-tuned α=0.5 variant as the uniform baseline in Table 2(a). Global allocation still leads the re-tuned uniform variant clearly on MME and ChartQA (MMStar and TextVQA are within about one point either way), so we retain the global operating point.

Table 14: Scoring $\times$ allocation on InternVL3.5-30B-A3B at 50% pruning (MME / MMStar / ChartQA / TextVQA).

	Uniform per-layer budget	Global non-uniform budget
REAP scoring	2109.6 / 63.3 / 65.3 / 74.4 (Table 1)	2085.7 / 63.0 / 65.1 / 74.6
MAEP scoring	2103.9 / 62.2 / 72.1 / 76.5 (α=0.5; Table 2(a))	**2160.3** / **63.3** / **76.7** / 75.6 (Table 1)

Table 15: Uniform per-layer 50% pruning on InternVL3.5-30B-A3B with α re-tuned for the uniform budget. All rows were evaluated with the same, final version of our evaluation pipeline.

Score (uniform per-layer budget)	MME	MMStar	ChartQA	TextVQA
MAEP, α=0.0	2144.1	60.1	56.2	76.5
MAEP, α=0.1	2155.9	59.8	56.8	76.8
MAEP, α=0.2	2138.0	61.4	60.9	76.9
MAEP, α=0.3	2124.6	60.9	61.2	76.6
MAEP, α=0.5 (Table 2(a))	2103.9	62.2	72.1	76.5
MAEP, α=0.7	1675.4	49.3	44.8	58.2
MAEP, α=0.8 (global operating point)	1918.4	52.8	61.6	61.2
REAP (uniform; Table 1)	2109.6	63.3	65.3	74.4



Q: How many uncut fruits are in the image?

Unpruned: Four uncut fruits. A detailed breakdown: top-left, top-center, top-right, and bottom-left—each whole, dark-purple with green leaves; the remaining two are cut open and excluded.

HC-SMoE: In the image, there are five uncut fruits.

MAEP (50%): Three whole mangosteens with intact skins, plus one cut in half revealing the white flesh inside; brief reasoning over each fruit's state.



Q: Do you know who paint this?

Unpruned: The painting is the **Mona Lisa**, by **Leonardo da Vinci** (1452–1519). Discusses sfumato technique, the High-Renaissance period (1503–1517), and the work's location at the Louvre.

HC-SMoE: The painting is created by the artist known as "InternVL". This painting is a part of a multi-modal language model.

MAEP (50%): The painting is the **Mona Lisa**, by **Leonardo da Vinci**. Mentions sfumato (mis-spelled "sfonto"), the High Renaissance, and the Louvre Museum.

Figure 10: Qualitative comparisons on LLaVA-Bench (in-the-wild) for InternVL3.5-30B-A3B at 50% pruning. Selected as the two samples with the most pronounced output divergence across configurations. Red marks factually incorrect or hallucinated content; **bold** marks key factual content.

M Qualitative Examples

Figure 10 contrasts model outputs across three conditions on LLaVA-Bench (in-the-wild) [37]: the unpruned baseline, HC-SMoE at 50% pruning, and MAEP at 50% pruning. In the counting case (top), MAEP correctly identifies the fruit composition (three whole mangosteens and one halved fruit), whereas HC-SMoE substantially overcounts and even the unpruned baseline miscounts. In the Mona Lisa case (bottom), MAEP matches the unpruned baseline in correctly attributing the painting and its art-historical context, while HC-SMoE produces a self-referential hallucination of model identity.

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 contributions stated in the abstract and Section 1 are substantiated in Sections 5–6.3.

   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: Limitations are discussed in Appendix A.

   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: [Yes]

Justification: Proposition 1 (Section 5.1.1) states its assumptions, with the proof in the appendix.

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 method (Sections 5–5.3) and experimental setup (Section 6.1, Appendix D) provide the details needed for reproduction.

   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: Code is publicly released at `https://github.com/nota-github/MAEP`; all used models and datasets are publicly available.

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: Experimental settings are described in Section 6.1 and Appendix D.

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: [No]

Justification: Multi-seed runs across 30B-parameter MoE-MLLMs are computationally prohibitive; consistency is shown across multiple baselines, models, and benchmarks instead (Section 6.2).

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: Compute requirements (GPU type; calibration is a single forward pass over 1K samples, and the remaining compute is benchmark inference) are reported in Appendix D.

   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 conforms with the NeurIPS Code of Ethics; it uses only publicly released models and datasets and does not involve human subjects.

   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: [N/A]

    Justification: This is a foundational efficiency method applied to existing pre-trained models; it introduces no new capabilities and has no direct path to negative applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: No high-risk models or scraped datasets are released.

    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: All used models and datasets are properly cited (Sections 2, 6.1) and their licenses respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: No new datasets or pre-trained models are released.

    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 research does not involve crowdsourcing or human subjects.

    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 research does not involve human subjects, so 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: [N/A]

    Justification: LLMs were not used as a core component of the method; the MoE-MLLMs studied are objects of analysis, not tools used for the methodology.

    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.