MAES: MODALITY-AWARE EXPERT SLIMMING FOR MULTIMODAL MIXTURE-OF-EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

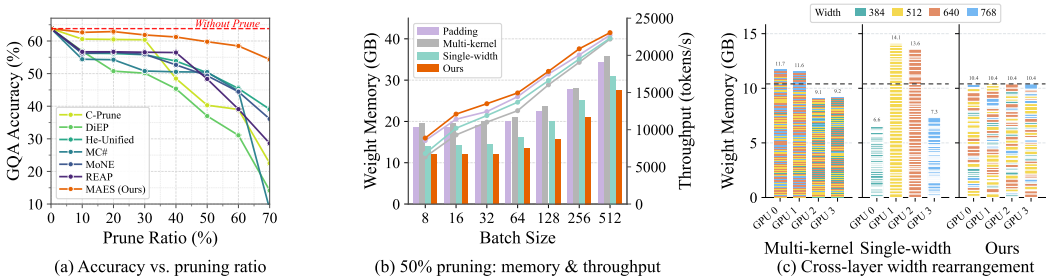
The rapid adoption of Mixture-of-Experts (MoE) in multimodal large language models has enabled remarkable capabilities, yet at the cost of critical bottlenecks in storage, memory, and inference speed. Structural pruning offers fine-grained compression that permanently reduces memory and computation, but existing methods treat all tokens uniformly and ignore the fundamental heterogeneity between modalities. Through systematic analysis, we uncover two key phenomena in multimodal MoEs: (*i*) experts exhibit pronounced *modality affinity*, inducing modality-specific redundancy; and (*ii*) different modalities exhibit different activation magnitude distributions, rendering modality-agnostic channel scoring unreliable. These findings motivate our **M**odality-**A**ware **E**xpert **S**limming (MAES), a training-free structural pruning and inference framework that proposes Expert Modality Affinity (EMA)-guided budget allocation according to each modality's contribution. To rank expert redundancy, we further introduce an exact second-order expert-importance criterion, replacing noisy proxies used in prior work. For deployment, we rearrange experts jointly across layers to balance weight-memory load across GPUs, which seamlessly integrates with modern inference frameworks and preserves efficient expert parallelism. Across 12 image and video benchmarks and 3 model families, MAES consistently outperforms existing MoE pruning methods, retaining over 98% of original performance at 30% channel pruning while attaining lower memory usage and higher throughput, achieving the state-of-the-art accuracy–efficiency trade-off. Our code is released at `https://anonymous.4open.science/r/MAES`.

1 INTRODUCTION

Multimodal large language models increasingly employ Mixture-of-Experts (MoE) architectures as their backbone to scale model capacity with a lower computational burden [34; 7; 17; 32; 40; 38]. Although only a subset of experts is activated per token, deploying an MoE model still requires loading the full parameter set into memory, incurring substantial memory overhead. In contrast to token dropping or expert skipping that only reduce runtime computation, model compression is favorable in permanently removing redundant parameters and directly shrinks the storage footprint.

However, existing large model compression methods typically focus on text-only MoEs [26; 13; 3] or dense LLMs and VLMs [36; 2; 27; 37], while barely exploring model compression tailored for large multimodal MoEs. To fill this gap, we begin by dedicated pilot experiments and comprehensively investigating model affinity and redundancy in multimodal MoEs, identifying three unique challenges for effective compression of multimodal MoEs: (i) *Experts' routing preference can mismatch their activation intensity on different modalities.* We observe that 79% of experts route predominantly either vision or text tokens, yet some of them show a smaller activation magnitude on the very modality for which they process more tokens, which misleads the selection of informative channels whenever the two signals are conflated. (ii) *Expert sensitivity measurements either inevitably mix cross-terms between experts or are prohibitively time-consuming.* Loss- or gradient-based proxies react unstably to the calibration data, and there is no efficient, exact expert-importance estimation method that can be computed in parallel. (iii) *The resulting pruned experts have arbitrary widths, which are not natively supported by modern inference frameworks.* Padding all experts to the maximum width before running the FusedMoE paradigm nullifies the deployment gains of static pruning.

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



(a) Accuracy vs. pruning ratio (b) 50% pruning: memory & throughput (c) Cross-layer width rearrangement

Figure 1: (a) Accuracy Pareto frontier; (b) peak memory and throughput compared to naive inference strategies (Details about each setting are described in Section 5.2); (c) an intuitive example of the rearrangement result, which balances per-GPU computation and memory load.

To address these three identified challenges, we propose Modality-Aware Expert Slimming (**MAES**), a training-free structural MoE pruning and inference framework. Corresponding to each challenge, MAES consists of three components: (i) *Modality-Affinity Channel Selection* addresses the mismatch between routing preference and activation intensity by scoring channels separately for each modality and selecting them under the constraint of Expert Modality Affinity. (ii) *Exact Second-Order Expert Sensitivity* is, to our knowledge, the first exact expert-importance criterion for multimodal MoEs, avoiding both the instability of other criteria and the cost of true ablation sweeps. (iii) *Cross-Layer Expert Rearrangement* reshapes arbitrary pruning results into deployment-friendly widths, enabling a single fused MoE kernel with balanced load, without padding or launching multiple kernels.

We evaluate MAES across 3 model families and 12 image and video benchmarks, compared against representative compression baselines. MAES consistently dominates in accuracy under various pruning ratios by large margins. On Qwen3-VL-MoE-30B-A3B, MAES retains 98.9% of the unpruned baseline's on average at 30% pruning, and retains 95.9% performance at 50% pruning. Ablations confirm that both modality affinity and exact second-order estimation are helpful indicators of expert specialization and inner redundancy. Figure 1(a) shows the pruning-ratio–accuracy Pareto frontier, where our MAES always stay in front. Meanwhile, our adaptation of shape regularization and load balancing strategies seamlessly integrate into modern inference paradigms for MoEs. Figure 1(b) shows a 19.7–41.1% lower peak memory than original padding at comparable or higher throughput under 50% pruning, and Figure 1(c) shows that the rearrangement equalizes the per-GPU load.

In aggregation, our contributions can be summarized as follows:

- We observe that multimodal MoEs show consistent specialization: 79.3% of experts are clearly dominated by a single modality, leaving room for compression.
- We propose MAES, a training-free structural MoE pruning framework that preserves the parameters each expert's preferred modality relies on, and estimates expert importance with an exact criterion that, to our knowledge, has not yet been investigated before and is more precise than any router-based indicator.
- We also make the heterogeneous pruning result integrate seamlessly with modern MoE inference paradigms through expert width regularization and cross-layer rearrangement.
- Across 3 multimodal MoEs and 12 image and video benchmarks, MAES consistently outperforms representative pruning algorithms and inference strategies.

2 Related Work

We review here the related topics to our paper. An extended related work can be found in Appendix A.

Multimodal MoE compression. Prior compression techniques for MoEs and dense models operate at coarser granularities: expert skipping and token compression [26; 13; 14; 42] reduce runtime computation but leave model storage unchanged; expert pruning [3; 19] removes whole experts, shrinking memory but accuracy drops; and dense-model techniques such as unstructured [36] or structured pruning [2; 27] ignore MoE-specific redundancy. Studies on multimodality such as Q-VLM [37], MC# [12], MoDES [14], and FastMMoE [42] show that vision-language models require modality-aware designs to better preserve performance, but none of them are aware of the expert-level modality specialization we identify, nor do they operate at the channel level.

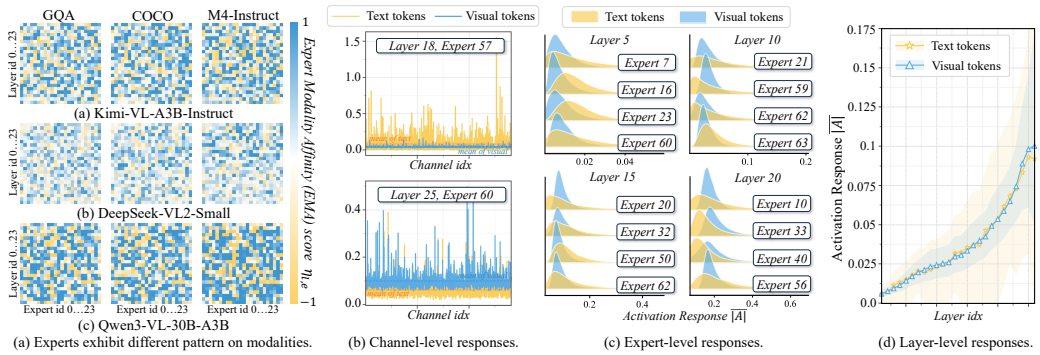


Figure 2: Modality-affinity behavior in multimodal MoEs. Expert Modality Affinity (EMA) heatmaps (a) reveal consistent modality specialization across layers and experts. Vision and text tokens induce distinct activation response distributions at the channel (b), expert (c), and layer (d) levels.

MoE expert parallelism and inference systems. Deploying large MoEs typically relies on expert parallelism (EP), which shards experts across devices [21; 33; 16] and, processes them through fused MoE kernels that require a shared per-rank width [18; 47; 1]. Channel-level pruning produces heterogeneous expert widths, forcing either padding or multiple kernel invocations that erode the deployment gains of pruning.

3 Pilot Observations on Modality

To identify whether expert prefers one modality, and to understand how the two modalities activate channels, we take Kimi-VL as an example and measure the routing behavior together with the activation intensity of every layer, expert, and channel (Figure 2).

Observation 1: Experts exhibit clear modality affinity. To quantify the modality specialization of each expert, we define the *Expert Modality Affinity* (EMA) score as $\eta_{\ell,e} = \frac{\rho^V_{\ell,e} - \rho^T_{\ell,e}}{\rho^V_{\ell,e} + \rho^T_{\ell,e} + \varepsilon}$, where $\rho^M_{\ell,e} = N^M_{\ell,e} / \sum_{e'} N^M_{\ell,e'}$ is the conditional routing rate for modality $M \in \{V, T\}$ (vision and text, respectively), and $N^M_{\ell,e}$ counts modality-M tokens routed to expert e in layer ℓ. A value $\eta_{\ell,e} \to +1$ indicates a higher conditional routing rate for vision tokens, $\eta_{\ell,e} \to -1$ indicates a higher rate for text tokens, and $\eta_{\ell,e} \approx 0$ indicates similar conditional routing rates for both modalities.

Figure 2(a) visualizes EMA across layers and experts for different MoEs, revealing strongly polarized modality preferences. In Qwen3-VL-MoE-30B-A3B-Instruct, 79.3% of experts show a clear modality inclination with $|\eta_{\ell,e}| > 0.3$ under GQA calibration. Across the three models and calibration sources, 85.3%–98.7% of experts have the same dominant modality on different calibration dataset, with Pearson $r = 0.65$–0.72. This indicates that the modality affinity is largely an intrinsic characteristic.

Takeaway 1: (1) Experts exhibit pronounced modality preference. (2) These patterns are consistent across calibration sources but differ across model architectures.

Observation 2: Modalities induce scale heterogeneity in channel-level responses. Beyond routing preferences, vision and text tokens also produce markedly different activation magnitudes within the same expert. In Figure 2(b–d), we measure the mean absolute activation $|\bar{A}|$ for each modality at three granularities. At the *channel level* (b), per-channel responses within individual experts show that the two modalities activate different channels at different magnitudes, and the dominant modality varies across experts. At the *expert level* (c), the kernel density estimates of activation responses show shifted distributions, concentrating at differently scales for each modality. At the *layer level* (d), aggregating $|\bar{A}|$ over all experts and channels, we find that the scale mismatch persists across depth. Together, these results expose that vision and text tokens produce significantly heterogeneous activation responses across all granularities, directly using activation as pruning proxy can lead to suboptimal pruning decisions.

Takeaway 2: Vision and text tokens produce different activation distributions across channels, experts, and layers, thus naive modality-agnostic channel scoring introduces a bias toward the modality with larger responses.

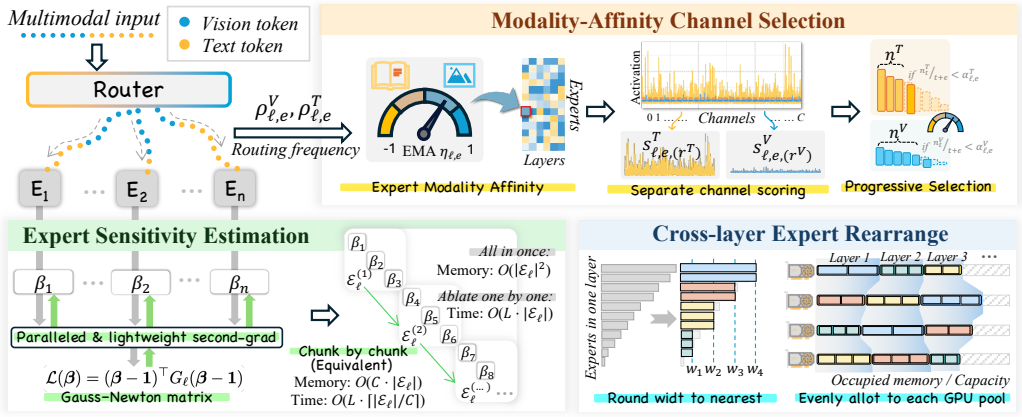


Figure 3: Illustration of our MAES. Yellow: Collect per-expert modality affinity and modality-separated channel scores, and expand each expert's channel budget under its affinity. Green: Estimate expert-wise sensitivity with an exact second-order criterion. Blue: Round experts to deployment-friendly widths and balance memory load on each rank for efficient parallelism.

Based on our observations, prior modality-agnostic methods may fail to identify multimodal redundancy: when an expert routes mostly vision tokens yet text tokens happen to activate it more strongly, a single pooled score may rank its text channels first, so the expert is slimmed according to the modality it does not specialize in. Therefore, we need a pruning framework that considers the specialty of each expert and allows it to keep the most relevant channels based on its routing behavior.

4 MAES: Modality-Aware Expert Slimming

In this section, we propose MAES, a training-free pruning and inference framework for multimodal MoEs. Within this framework, routing preference sets a per-modality channel quota for each expert (Section 4.1), while our Exact Second-Order Expert Sensitivity criterion (SOES, Section 4.2) precisely estimates expert-wise sensitivity via an exact Gauss-Newton matrix to allocate the pruning budget across experts. Finally, to make the pruned model deployable, we integrate it with modern expert-parallelism systems (Section 4.3) by regularizing expert widths and rearranging them across multi-GPU ranks for efficient parallelism. Figure 3 illustrates the overall pipeline.

4.1 Modality-Affinity Channel Selection

Motivated by the modality affinity phenomenon observed in Section 3, we decouple channel scoring across modalities and allocate the per-expert channel budget according to each expert's EMA score. This ensures that modality-dominant experts prioritize channels from their affinitive modality. We detail the procedure in three steps.

Step 1. Expert modality affinity. We convert the EMA score into expert modality-affinity weights to define the target channel quota for each modality. For $M \in \{V, T\}$:

$$\alpha^V_{\ell,e} = \frac{\eta_{\ell,e} + 1}{2}, \quad \alpha^T_{\ell,e} = 1 - \alpha^V_{\ell,e}, \quad \text{where } \sum_M \alpha^M_{\ell,e} = 1.$$

Step 2. Separate channel scoring. Given a calibration set containing both vision and text tokens, we compute per-channel activation magnitude scores separately for each modality. Channels within each expert are ranked separately by their per-modality scores in descending order, denoted $s^V_{\ell,e,(r^V)}$ and $s^T_{\ell,e,(r^T)}$, where (r^V) and (r^T) are indices of the ordered channels.

Step 3. Progressive selection to match affinity score. Since both modality-specific rankings are defined over the same channel set $\mathcal{C}_{\ell,e}$, independently selecting top-k channels from each ranking would introduce overlaps and fail to exactly match the prune budget. To avoid this, we construct the selected channel subset $\mathcal{S}_{\ell,e}$ by interleaving the selection of the two rankings, expanding the selected set that currently lags behind its affinity ratio iteratively, until the budget is exactly met.

Let $c^M_{\ell,e,(r^M)}$ denote the channel ranked r^M-th for modality M, and n^M_t is the size of selected channels for modality M after the t-th selection. Initialized from $\mathcal{S}^{(0)}_{\ell,e} = \emptyset$, we select at each step by

$$\mathcal{S}^{(t)}_{\ell,e} = \begin{cases} \mathcal{S}^{(t-1)}_{\ell,e} \cup \{c^V_{\ell,e,(n^V_{t-1}+1)}\}, & \text{if } \frac{n^V_{t-1}}{t} < \alpha^V_{\ell,e}, \\ \mathcal{S}^{(t-1)}_{\ell,e} \cup \{c^T_{\ell,e,(n^T_{t-1}+1)}\}, & \text{otherwise,} \end{cases} \tag{1}$$

where $t = 1, 2, \ldots$ Since $U(t) = |\mathcal{S}^{(t)}_{\ell,e}|$ is monotone non-decreasing, we terminate the expansion at the minimum t^* satisfying $U(t^*) \geq B_{\ell,e}$, where $B_{\ell,e}$ is the per-expert channel budget derived via [6] but fed with our proposed second-order criterion in Section 4.2, which is more precise than previous indicators.

Algorithm 1 Algorithm of MAES.

Input: Model $\mathcal{M}$, calibration set $\mathcal{D}$, prune ratio p
Output: Pruned model $\mathcal{M}'$

for each layer $\ell = 1, \ldots, L$ **do**
  // Second-order importance (Sec. 4.2)
  Attach a scalar probe $\boldsymbol{\beta}_\ell = \mathbf{1}$ to the outputs $\{z_{\ell,e}\}_{e\in\mathcal{E}_\ell}$; calibrate on $\mathcal{D}$ for $\mathcal{L}(\boldsymbol{\beta}_\ell)$ (Eq. (2)).
  Compute $\{h_{\ell,e}\}_{e\in\mathcal{E}_\ell}$ (Eq. (14)), then get $s^{\text{expert}}_{\ell,e}$.
end for
Allocate per-expert prune budgets $\{B_{\ell,e}\}$ from $\{s^{\text{expert}}_{\ell,e}\}$ under p.
for each layer $\ell = 1, \ldots, L$ **do**
  // Modality-affinity selection (Sec. 4.1)
  Collect per-modality channel scores and routing frequencies on $\mathcal{D}$; compute $\eta_{\ell,e}$ (EMA)
  for $e \in \mathcal{E}_\ell$ **do**
    Compute affinity weights $\alpha^V_{\ell,e}$, $\alpha^T_{\ell,e}$ (Eq. (1)); init $\mathcal{S}^{(0)}_{\ell,e} \leftarrow \emptyset$, $t \leftarrow 0$.
    while $|\mathcal{S}^{(t)}_{\ell,e}| < B_{\ell,e}$ **do**
      Expand $\mathcal{S}^{(t+1)}_{\ell,e}$ (Eq. (1)); $t \leftarrow t+1$.
    end while
    $\mathcal{S}_{\ell,e} \leftarrow \mathcal{S}^{(t)}_{\ell,e}$.
  end for
end for
return $\mathcal{M}'$ pruned with $\{\mathcal{S}_{\ell,e}\}$.

4.2 Exact Second-Order Expert Sensitivity

Motivation. Prior work estimates expert importance via router logits in forward pass [31; 20; 26] or gradients in backward pass. These criteria are easily affected by loss definition and mutual effects among experts. We propose an exact Second-Order Expert-Sensitivity (SOES) criterion. To our knowledge, this is the first use of the exact Gauss-Newton matrix to measure expert importance.

For expert e in layer ℓ, let $z_{\ell,e} = f_{\ell,e}(h_\ell)$ be its output and $g_{\ell,e}$ be the router weight. We attach an identity scalar $\beta_e = 1$ to each routed expert contribution. The perturbed block output is $\hat{y}_\ell(h_\ell; \boldsymbol{\beta}) = y_\ell(h_\ell) + \sum_{e\in\mathcal{E}_\ell}(\beta_e - 1) g_{\ell,e}(h_\ell) z_{\ell,e}(h_\ell)$. We construct a blockwise MSE loss against the original output y_ℓ, $\mathcal{L}(\boldsymbol{\beta}) = \sum_{h_\ell\in\mathcal{D}} \|\hat{y}_\ell(h_\ell; \boldsymbol{\beta}) - y_\ell(h_\ell)\|_2^2$. Expanding $\mathcal{L}$ around $\boldsymbol{\beta} = \mathbf{1}$ gives

$$\mathcal{L}(\mathbf{1} + \Delta\boldsymbol{\beta}) = \underbrace{\mathcal{L}(\mathbf{1}) + \nabla\mathcal{L}(\mathbf{1})^\top \Delta\boldsymbol{\beta}}_{=0\ (\boldsymbol{\beta}=\mathbf{1} \text{ is the minimizer of } \mathcal{L})} + \underbrace{\tfrac{1}{2}\Delta\boldsymbol{\beta}^\top H_{\boldsymbol{\beta}} \Delta\boldsymbol{\beta}}_{\text{Hessian term, Eq. (3)}} + R_3, \tag{2}$$

Furthermore, the second-order chain rule decomposes this Hessian into two terms:

$$H_{\boldsymbol{\beta}} = \underbrace{J^\top \left(\nabla^2_{\hat{y}_\ell}\mathcal{L}\right) J}_{\text{generalized Gauss–Newton}} + \underbrace{\sum_i \frac{\partial \mathcal{L}}{\partial \hat{y}_{\ell,i}} \nabla^2_{\boldsymbol{\beta}} \hat{y}_{\ell,i}}_{\text{model curvature} = 0}, \qquad J = \frac{\partial \hat{y}_\ell}{\partial \boldsymbol{\beta}}. \tag{3}$$

There are two conditions acting on different terms. (i) The curvature term is zero and $H_{\boldsymbol{\beta}}$ is exact: since $\hat{y}_\ell$ is affine in $\boldsymbol{\beta}$, we have $\nabla^2_{\boldsymbol{\beta}} \hat{y}_{\ell,i} = 0$, so the model-curvature term vanishes and $H_{\boldsymbol{\beta}}$ is exact for any loss. (ii) The remainder is zero and the second-order expansion is exact: with the squared-error loss (e.g., MSE), affineness further makes $\mathcal{L}$ strictly quadratic in $\boldsymbol{\beta}$, with derivatives of order three and above equal to zero, i.e., $R_3 = 0$. Together they make the Taylor expansion in Eq. (2) equal the generalized Gauss-Newton matrix exactly. The detailed derivation is in Appendix C.

We compare different importance estimators in Table 5 and compare against the synchronous first-order term in Figure 4; both show that our Hessian term is a precise replacement of the true ablation loss at affordable time cost. A quadratic loss also results in a closed form over the forward pass, $\Delta\mathcal{L}^{\text{remove}}_{\ell,e} = \frac{1}{2} H_{ee} = \sum_{h_\ell\in\mathcal{D}} \|a_{\ell,e}(h_\ell)\|_2^2$, while non-quadratic losses require differentiating twice. We provide chunked Hessian-diagonal processing to strive a memory-time trade-off if using general non-quadratic losses. In Appendix E.1, we prove that computing the Hessian diagonal entries chunk by chunk yields exactly the same values as computing them jointly.

4.3 Cross-Layer Expert Rearrangement for Parallelism

Motivation. After pruning, experts end up with arbitrary widths. Modern serving frameworks such as vLLM and SGLang use FusedMoE backends whose kernel interfaces and packed weight layouts require experts processed together to have the same intermediate dimension [18; 47]. Experts would be padded to the widest width before FusedMoE paradigm, which nullifies the benefits of structural pruning. Figure 1(b) and Table 3 show this padding overhead on pruned MoEs.

Algorithm 2 Cross-Layer Expert Placement.

Input: Width levels $\mathcal{W}$; expert counts n_k^ℓ assigned to level w_k; EP size m; refinement operator $\textsc{Refine}(\cdot)$
Output: Placement $\{G_{\ell,k}\}$
1: **for** $\ell \in \{1, \dots, L\}$ **do**
2: Round to nearest level; $\phi_k^\ell \leftarrow w_k n_k^\ell$.
3: **end for**
4: $q \leftarrow$ the atomic width, the largest integer dividing every w_k; $\Phi_{\text{tot}} \leftarrow \sum_{\ell,k} \phi_k^\ell$.
5: $\tau_0 \leftarrow 0$ **if** $m \mid (\Phi_{\text{tot}}/q)$ **else** q. {Prop. 1}
6: $\Phi_u \leftarrow 0$ for all u.
7: **for** ℓ in descending order of $\max_k \phi_k^\ell - \min_k \phi_k^\ell$ **do**
8: $\kappa \leftarrow$ sort ϕ_k^ℓ ($\gg$); $\sigma \leftarrow$ sort Φ_u ($\ll$).
9: $G_{\ell,\kappa(j)} \leftarrow \sigma(j)$; $\Phi_{G_{\ell,k}} \mathrel{+}= \phi_k^\ell$.
10: **end for**
11: $G \leftarrow \textsc{Refine}(G, \{\phi_k^\ell\}, m)$.
12: **return** $\{G_{\ell,k}\}$

We therefore propose Cross-Layer Expert Rearrangement (CLER), which assigns a single width to each EP rank at every layer. It thereby balances the weight memory load across all layers, while invoking FusedMoE backend only once for each layer on each GPU rank. As shown in Figure 3 (bottom right), we first round the arbitrary per-expert widths to a small set of width levels $\mathcal{W} = \{w_1, \dots, w_m\}$ via a classic greedy marginal-cost procedure [5] (lines 1–3 in Algorithm 2). For each $k \in \{1, \dots, m\}$, let n_k^ℓ denote the number of experts at layer ℓ assigned to level w_k, so the static load of level k can be computed as $\phi_k^\ell = w_k n_k^\ell$.

Perfect balance is attainable only when the total load divides evenly among the ranks; otherwise, only a lower bound is achievable, which can be computed directly and used as an early-stopping criterion. Write q for the atomic width, the largest integer with $w_k/q \in \mathbb{Z}$ for all $w_k \in \mathcal{W}$. Let Φ_u denote the weight memory that rank u accumulates across all layers, and $\Delta\Phi = \max_u \Phi_u - \min_u \Phi_u$ the spread between the most and least loaded rank.

Proposition 1. *Every placement satisfies $\Delta\Phi \geq \tau_0$, where $\tau_0 = 0$ if m divides Φ_{tot}/q, and $\tau_0 = q$ otherwise.*

The above proposition means every rank load, as well as $\Delta\Phi$ itself, is a sum of multiples of q. If $\Delta\Phi = 0$, every rank carries exactly the same load Φ_{tot}/m. Otherwise the smallest non-zero multiple of q (q itself) is the lower bound of the load spread. Appendix D.2 gives full proof.

Algorithm 2 shows the complete placement steps in our implementation. Lines 7–10 provides an initial placement G by sorting each layer's levels and pairing the heaviest with the emptiest rank. REFINE (line 11) is the optimization, which can be instantiated with classic algorithms. We compare seven alternatives in Section 5.4, including local and global search methods. Applying REFINE reduces the distance from the initial 1.67% down to under 0.1% (Section 5.4).

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models and Benchmarks. We evaluate on multimodal MoEs spanning diverse backbone architectures and scales: Kimi-VL-A3B-Instruct [17], Qwen3-VL-MoE-30B-A3B-Instruct [32], and InternVL-3.5-30B-A3B-HF [38]. We evaluate on 12 multimodal benchmarks. For brevity, we use abbreviated names: *For image*: GQA [15], TxVQA (TextVQA) [35], CQA (ChartQA) [29], MMS (MMStar) [4], MMB (MMBench) [25], MMV (MMVet) [43], MME [8], and RWQA (RealWorldQA) [41]. *For video*: MVB (MVBench) [23], VMME (VideoMME) [9], LVB (LongVideoBench) [39], and EgoS (EgoSchema) [28]. All benchmarks are evaluated by lmms-eval [46].

Baselines. We compare against representative MoE or multimodal LLM compression methods: C-Prune [10], He-Unified [11], DiEP [3], REAP [19], MoNE [44], ACP [6], covering both channel-level and expert-level pruning. Details are provided in Appendix D.3.

Implementation. We use 1024 samples with 2048 tokens per sample for calibration. Analysis of data mixture are provided in Section 5.4. Our experiments are conducted on 4×H20 GPUs with

Table 1: Comparisons on Kimi-VL-A3B-Instruct under 30% and 50% compression ratios.

p	Method	General Image						Text-rich Image		Video				Avg
		GQA	MMS	MMB	MMV	MME	RWQA	TxVQA	CQA	MVB	VMME	LVB	EgoS	
0	Baseline	62.67	65.25	83.16	68.90	2195.45	65.88	88.64	89.36	60.18	63.41	60.88	78.5	72.10
0.3	DiEP [3]	55.32	47.38	66.32	47.23	1933.71	63.92	76.61	64.20	56.43	52.11	52.67	69.8	60.09
	C-Prune [10]	57.42	42.41	74.61	44.95	1892.60	58.82	79.99	73.89	54.98	57.22	51.82	65.4	60.76
	He-Unified [11]	61.56	48.70	73.58	53.28	2132.97	63.66	82.86	70.28	58.65	58.70	54.63	**73.6**	64.64
	MoNE [44]	56.50	50.89	69.95	35.23	1923.78	59.87	76.77	75.04	40.95	26.65	24.18	60.5	53.77
	ACP [6]	61.95	61.26	79.86	57.66	2020.70	66.06	84.36	85.84	59.21	61.78	57.29	69.6	68.09
	REAP [19]	61.39	50.90	76.17	55.09	2044.41	65.49	83.70	77.01	61.35	61.67	54.77	71.7	66.02
	MAES *(Ours)*	**62.04**	**62.55**	**82.22**	**64.86**	**2155.07**	**66.14**	**86.35**	**87.42**	**61.68**	**62.33**	**60.51**	73.0	**70.51**
0.5	DiEP [3]	44.43	47.60	61.14	32.47	1692.07	55.95	55.77	55.20	52.58	45.50	46.50	68.2	52.15
	C-Prune [10]	43.64	35.21	36.53	32.19	940.30	45.10	66.02	57.98	50.04	51.91	46.67	45.2	45.34
	He-Unified [11]	55.97	43.52	65.54	40.58	2129.56	60.39	70.02	54.82	55.28	52.71	51.29	70.4	58.05
	MoNE [44]	51.56	49.18	62.18	26.81	2126.15	57.65	65.24	50.67	36.11	21.10	22.60	51.4	47.54
	ACP [6]	59.09	56.79	76.87	37.89	2072.30	63.71	79.12	75.88	56.89	56.52	57.41	65.2	63.28
	REAP [19]	**61.53**	43.78	70.98	41.86	**2240.59**	64.84	77.35	63.12	58.95	59.22	53.39	69.6	62.05
	MAES *(Ours)*	60.68	**57.46**	**80.76**	**51.74**	2110.28	**65.36**	**82.11**	**81.24**	**59.74**	**59.04**	**59.69**	**72.4**	**67.13**

Table 2: Detailed benchmark breakdown on Qwen3-VL-30B-A3B-Instruct across pruning ratios.

p	Method	General Image						Text-rich Image		Video				Avg
		GQA	MMS	MMB	MMV	MME	RWQA	TxVQA	CQA	MVB	VMME	LVB	EgoS	
0	Baseline	62.98	70.68	86.94	74.36	2498.64	72.03	83.45	85.00	65.61	68.85	65.52	71.2	74.65
0.3	DiEP [3]	50.57	42.02	64.51	49.86	2019.07	63.66	74.28	66.92	51.31	55.11	52.21	65.2	58.98
	C-Prune [10]	21.87	49.52	73.58	62.71	1829.40	69.28	78.72	75.79	55.51	59.67	56.53	68.8	61.44
	He-Unified [11]	53.45	36.11	66.84	57.29	2097.42	66.01	78.84	67.97	57.98	56.82	53.79	60.2	60.85
	MoNE [44]	24.68	52.10	75.39	61.08	2073.10	70.46	80.14	66.29	60.37	60.92	58.08	70.2	62.81
	ACP [6]	60.82	65.77	84.86	68.90	2207.20	68.23	81.77	82.56	61.42	69.85	61.13	64.8	70.74
	REAP [19]	37.37	50.38	73.32	60.55	2125.58	71.90	78.75	62.03	60.45	61.17	59.14	69.8	63.40
	MAES *(Ours)*	**63.13**	**68.94**	**85.99**	**69.18**	**2347.41**	**71.92**	**82.89**	**86.64**	**65.92**	**73.11**	**63.58**	**71.2**	**73.86**
0.5	DiEP [3]	29.12	29.69	30.57	27.25	1639.30	49.54	62.65	43.11	37.02	36.52	40.78	63.0	42.31
	C-Prune [10]	46.03	27.82	58.29	46.73	1166.71	62.88	31.39	64.28	39.85	55.29	34.67	47.8	46.39
	He-Unified [11]	48.29	29.28	49.48	45.64	1738.41	44.58	68.94	38.43	41.72	46.13	41.09	60.8	48.04
	MoNE [44]	62.56	40.82	67.10	48.72	2006.96	68.47	69.79	38.73	56.25	58.89	55.67	70.6	59.11
	ACP [6]	58.61	60.19	81.86	61.93	2031.60	70.76	79.76	81.21	58.58	66.07	62.48	68.0	68.50
	REAP [19]	**63.27**	47.31	70.98	49.42	1868.75	67.84	74.83	49.98	59.41	61.11	58.21	69.6	61.56
	MAES *(Ours)*	61.00	**66.15**	**82.99**	**63.76**	**2177.06**	**71.63**	**82.33**	**83.68**	**63.61**	**69.26**	**63.35**	**73.8**	**71.61**

expert parallelism supported by vLLM [18]. For simplicity, we set $m = 4$ width levels which 1-to-1 to the ranks. And we sweep m in Section 5.4 and layer amount in Appendix D.2.

5.2 Main Results

Accuracy. We evaluate MAES against representative pruning baselines on three large multimodal MoEs at 30% and 50% channel pruning ratios, with results summarized in Table 1 and Table 2, and, for InternVL3.5-30B-A3B-HF, Table 10 in Appendix E. MAES achieves the best average score and has the most robust performance across tasks and pruning ratios of 30% and 50%, outperforming the strongest baseline by a wide margin, up to 3.40-7.20% on individual tasks. On Qwen3-VL-MoE-30B-A3B-Instruct, MAES retains 98.9% of the unpruned baseline's average performance at 30% pruning, which is 73.86 vs. 74.65 of unpruned counterpart, and 95.9% at 50% pruning, which is 71.61 vs. 74.65 of unpruned one, while substantially outperforming the strongest pruning baseline at both ratios. We claim that channel-level pruning can more precisely identify and remove redundant capacity within each expert, preserving model performance under the same compression ratio.

Efficiency. Beyond accuracy, we evaluate the deployment efficiency of MAES by integrating into modern inference paradigm of expert parallelism with vLLM. We test 5 models across 4 model families at model scales of 16B–235B. Table 3 compares four inference strategies: (i) Padding, the native behavior of modern inference frameworks, keeps high throughput but occupies more memory compared to our implementation. At 50% pruning, MAES reduces peak non-KV memory relative to Padding by 19.7–41.1%. (ii) Multi-kernel launches a separate kernel per width level on each rank, which can be much slower due to multiple kernel launches. (iii) Single-width fixes one width to each rank across all layers, which has faster computation but risking imbalanced load. (iv) Our CLER

Table 3: Deployment efficiency of MAES at pruning ratios $p = 0.3$ and $p = 0.5$ across MoE families and sizes under expert parallelism (rank = 4). "Mem" denotes peak non-KV GPU memory in GiB; "Pref" and "Dec" denote prefill and decode throughput, in $\times 10^3$ and $\times 10^2$ tokens/s respectively.

p	Imple.	Kimi-VL			Qwen3-VL-30B			InternVL3.5			Mistral-119B			Qwen3-VL-235B		
		Mem ↓	Pref ↑	Dec ↑	Mem ↓	Pref ↑	Dec ↑	Mem ↓	Pref ↑	Dec ↑	Mem ↓	Pref ↑	Dec ↑	Mem ↓	Pref ↑	Dec ↑
0.3	Padding	14.35	52.01	14.87	19.61	43.34	13.65	19.68	43.83	15.21	34.48	32.87	9.01	63.23	10.61	5.96
	Multi-kernel	17.59	39.42	7.62	25.52	26.66	6.39	25.73	27.02	6.71	46.35	24.03	4.85	86.54	6.11	2.70
	Single-width	14.35	**59.32**	**15.09**	19.61	**49.90**	13.71	19.69	**49.65**	14.85	34.48	**36.59**	**9.02**	63.23	**11.58**	5.89
	CLER *(Ours)*	**12.85**	57.32	14.92	**15.55**	47.10	**13.92**	**15.62**	46.65	**15.65**	**26.70**	35.13	8.86	**47.95**	10.96	**5.99**
0.5	Padding	14.35	50.95	14.97	19.61	42.52	13.85	19.68	44.41	15.47	34.11	32.51	8.56	63.23	10.30	**5.87**
	Multi-kernel	15.01	42.16	7.74	20.13	28.20	6.29	20.33	28.59	6.61	35.60	24.25	4.77	65.37	6.33	2.71
	Single-width	14.35	59.32	14.91	19.61	50.25	13.84	19.68	49.62	15.45	34.11	35.70	8.90	63.23	11.58	5.86
	CLER *(Ours)*	**11.53**	**63.93**	**15.23**	**13.14**	**52.26**	**13.89**	**13.21**	**53.29**	**15.64**	**20.94**	**36.90**	**8.97**	**37.27**	**12.35**	5.83

Table 5: Comparison of expert importance methods. RMSE and Pearson are compared with $\Delta\mathcal{L}_{\ell,e}^{\text{true}}$.

Method	RMSE↓	Pearson↑	GQA	MMB	TxVQA	Avg
$\Delta\mathcal{L}_{\ell,e}^{\text{true}}$	0.0000	1.0000	60.65	80.65	79.27	69.57
Uniform	0.1083	N/A[b]	58.95	78.39	77.56	67.48
Token count	0.0625	0.8109	59.59	78.25	76.71	67.22
Router logits	0.1063	0.1111	59.48	80.19	78.02	68.06
$\lvert g_{\ell,e}\rvert$	0.1446	0.6939	59.57	79.87	79.09	68.68
$h_{\ell,e}$ (l_{kl})	0.1074	0.1505	60.70	80.37	79.21	69.19
$h_{\ell,e}$ (Rel l_2)	7.73×10^{-5}	1.0000	59.82	81.99	81.60	69.10
$h_{\ell,e}$ (l_2)	0.0000	1.0000	60.65	80.65	79.27	69.57

[a]Results for all tasks are put in Table 12 in Appendix E.

[b]Pearson is undefined because min-max normalization turns its constant score vector into a zero-variance vector.

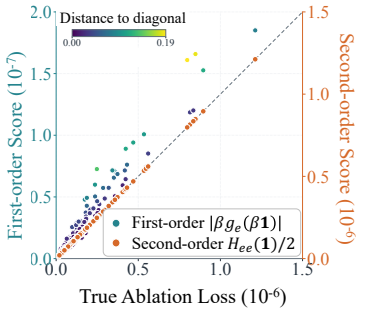


Figure 4: Our Hessian scores match individual L2 ablation losses.

achieves the lowest memory and the highest throughput in most cases. Prefll throughput improves more than decoding, by 3.3–10.2% at 30% pruning and 13.5–25.5% at 50% pruning.

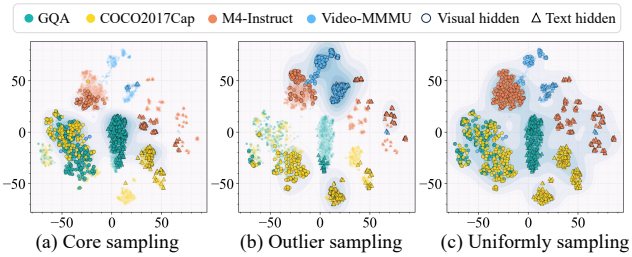
5.3 Ablation Studies

Table 4: Component ablation on Kimi-VL.

Method	General Image	Text-rich Image	Video	Overall
Compression Ratio $p = 0.3$				
Baseline	66.49	85.10	61.97	68.09
MACS	68.81$_{\uparrow 2.32}$	84.77$_{\downarrow 0.33}$	64.16$_{\uparrow 2.19}$	69.92$_{\uparrow 1.83}$
MACS + SOES	68.58$_{\uparrow 2.09}$	86.67$_{\uparrow 1.57}$	64.48$_{\uparrow 2.51}$	70.23$_{\uparrow 2.14}$
MACS + SOES + CLER	69.13$_{\uparrow 2.64}$	86.89$_{\uparrow 1.79}$	64.38$_{\uparrow 2.41}$	70.51$_{\uparrow 2.42}$
Compression Ratio $p = 0.5$				
Baseline	61.39	77.50	59.00	63.28
MACS	64.06$_{\uparrow 2.67}$	78.91$_{\uparrow 1.41}$	61.07$_{\uparrow 2.07}$	65.54$_{\uparrow 2.26}$
MACS + SOES	64.12$_{\uparrow 2.73}$	80.53$_{\uparrow 3.03}$	62.61$_{\uparrow 3.61}$	66.35$_{\uparrow 3.07}$
MACS + SOES + CLER	65.23$_{\uparrow 3.84}$	81.67$_{\uparrow 4.17}$	62.72$_{\uparrow 3.72}$	67.13$_{\uparrow 3.85}$

Component ablation. Table 4 summarizes the component ablation on Kimi-VL-A3B-Instruct at 30% and 50% channel pruning; the full per-task results are reported in Appendix E, Table 11. MACS (Section 4.1) decouples channel scoring across modalities. Hessian-based scores proposed in SOES (Section 4.2) determine the per-expert channel budget $B_{\ell,e}$ consumed by MACS. Both of the two methods help improve the performance of pruned MoEs under downstream tasks. SOES improves the overall average from 69.92 to 70.23 at p=0.3 and from 65.54 to 66.35 at $p = 0.5$. CLER (Section 4.3) rounds each expert to one of m deployment-friendly width levels and balances weight-memory load across ranks for efficient parallelism. This width rounding further improves the overall average from 70.23 to 70.51 at p=0.3 and from 66.35 to 67.13 at p=0.5, possibly because trimming the extremely small, low-information experts frees their capacity for more important ones; we leave a full explanation to future work.

Effect of second-order expert importance estimation. Table 5 compares expert-wise importance estimation against the per-expert ablated loss $\Delta\mathcal{L}_{\ell,e}^{\text{true}}$ in downstream accuracy. Routing-based proxies (token count, router logits) and the first-order gradient $|g_{\ell,e}|$ exhibit weak correlation with the true loss. In contrast, our second-order estimate $h_{\ell,e}$ under the ℓ_2 loss achieves an exact match with $\Delta\mathcal{L}_{\ell,e}^{\text{true}}$, and yields the best downstream accuracy. We additionally evaluate two alternative loss variants, the relative ℓ_2 loss and a non-quadratic KL-divergence loss, both of which also perform well. Figure 4



(a) Core sampling (b) Outlier sampling (c) Uniformly sampling

Figure 6: t-SNE projections of hidden states from vision and text tokens. Overlaid KDE density contours highlight regions of higher feature concentration.

Table 6: Comparisons on data sources and sampling strategies[a].

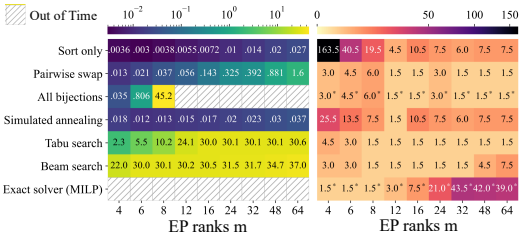
Calib. Data	MMB	GQA	Avg
Single-Source Data Sampling			
GQA	82.84	61.03	64.03
COCO	82.17	59.16	62.59
M4-Instruct	84.59	60.54	64.39
Mixed-Source Data Sampling			
Core	83.78	59.22	64.18
Outlier	83.39	59.33	64.47
Uniform	80.76	60.68	**71.85**

[a]Complete results: Table 13.

shows first-order estimates $|\beta g_e(\beta \mathbf{1})|$ (left axis) and our second-order estimates $H_{ee}(\mathbf{1})/2$ (right axis) against true ablation loss in layer 0 of Qwen3-VL. First-order points scatter on both sides of the equality diagonal, showing no strong relationship with the independent expert ablation loss.

5.4 Analysis

Refine operators for cross-layer rearrangement. To further balance the load produced by initialization in Algorithm 2 (lines 7–10), we apply a Refine operator implemented with classic algorithms. Figure 5 evaluates seven Refine operators and compares them against the bound of Proposition 1, sweeping the EP rank m from 4 to 64 on the full 48-layer model at $p = 0.3$. Since the pruned channel count does not divide evenly (m does not divide Φ_{tot}/q), the lower bound is $\tau_0 = 1.5\,\text{MiB}$. We also provide $p = 0.5$ counterpart in Figure 7 in Appendix D.2, where $\tau_0 = 0$ and an exact balance is attainable. As shown in Figure 5(a) and (b), most search algorithms stop within 10^2 s and reach the theoretical lower bound, except for All bijections (whose neighbourhood has size $m!$) and the exact solver (MILP); for these two, we record the result at the time limit and mark it with $*$ in (b). Both figures show that when $m \leq 16$, the choice of Refine barely matters, but for large m, Simulated annealing, or simply Pairwise swap, reaches the lower bound faster.



(a) Time cost (s, log scale) (b) Load spread $\Delta\Phi(G)$(MiB)

Figure 5: (a) Solving time cost of different Refine operators and (b) the resulting load spread $\Delta\Phi(G)$, as a function of the EP rank m, on the full 48-layer, p=0.3 instance, where τ_0=1.5 MiB.

Mixture of calibration data sources. Diverse data mixtures have been widely shown to improve the generalization of multimodal LLMs [30; 45; 24; 22; 40]. We also observe that more diverse calibration data leads to better performance. Figure 6 visualizes the 2D t-SNE projection of four datasets used for calibration: GQA, COCO2017Cap, M4-Instruct, and Video-MMMU. We define three sampling strategies based on the projected Euclidean distance $\mathbf{p}_i = (\mathbf{z}_i^V + \mathbf{z}_i^T)/2$ to the global centroid $\bar{\mathbf{p}}$, where $\mathbf{z}_i^V$ and $\mathbf{z}_i^T$ are the 2D t-SNE coordinates of the vision and text centroids of sample i. As shown in Table 6, uniform sampling achieves the best average score. This suggests that calibration sets with broader feature space provide more representative channel importance estimates, reducing overfitting and improving robustness.

6 Conclusion and Limitation

This paper introduces a training-free pruning framework, termed as MAES, that brings modality-aware design into multimodal MoE compression. This is achieved by allocating budget according to expert modality affinity, ranking experts with an exact second-order criterion, and cross-layer expert rearrangement, MAES makes principled pruning. Nevertheless, our study focuses on vision-language MoEs with gated-MLP experts. Our method has not been evaluated on omni-modal MoEs that jointly process additional modalities, which is our immediate-next goal in future work.

Reproducibility Statement

The exact-quadratic property underlying our second-order expert-importance criterion, together with its full derivation, is proved in Appendix C. For experimental settings, Section 5 and Appendix D.1 detail every calibration setting used to obtain our results, including the datasets, token budgets, and sampling strategies for each model and pruning ratio, as well as the implementation of all compared baselines. We will release the full MAES implementation together with the calibration manifests used in our experiments, so that all numbers reported in this paper can be reproduced from the released code and data.

AI Use Statement

We used generative AI tools to assist with polishing the writing of this paper, to assist with code review during the implementation of the proposed method, and to assist with producing several figures used in this paper. We did not use generative AI tools to generate research ideas, experimental results, or any content presented as an original finding. All AI-assisted writing, code, and figures were reviewed and verified by the authors, who take full responsibility for the final content of this work, including all text, claims, and artifacts.

References

[1] Osayamen Jonathan Aimuyo, Byungsoo Oh, and Rachee Singh. FlashMoE: Fast distributed MoE in a single kernel, 2025.

[2] Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns, 2024.

[3] 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.

[4] 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.

[5] George B. Dantzig. Discrete-variable extremum problems. *Operations Research*, 5(2):266–288, April 1957. ISSN 1526-5463. doi: 10.1287/opre.5.2.266. URL http://dx.doi.org/10.1287/opre.5.2.266.

[6] Yifu Ding, Jiacheng Wang, Ge Yang, Yongcheng Jing, Jinyang Guo, Xianglong Liu, and Dacheng Tao. Attribution-guided and coverage-maximized pruning for structural MoE compression. In *International Conference on Machine Learning (ICML)*, 2026.

[7] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *JMLR*, 23(1):1–39, 2022.

[8] Chaoyou Fu, Peixian Chen, Yunhang Shen, et al. MME: A comprehensive evaluation benchmark for multimodal large language models, 2023.

[9] Chaoyou Fu, Yuhan Dai, Yongdong Luo, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis, 2024.

[10] Hongcheng Guo, Juntao Yao, Boyang Wang, Junjia Du, Shaosheng Cao, Donglin Di, Shun Zhang, and Zhoujun Li. Cluster-driven expert pruning for mixture-of-experts large language models, 2025.

[11] Shwai He, Daize Dong, Liang Ding, and Ang Li. Towards efficient mixture of experts: A holistic study of compression techniques. *Transactions on Machine Learning Research*, 2025.

[12] Wei Huang, Yue Liao, Yukang Chen, Jianhui Liu, Haoru Tan, Si Liu, Shiming Zhang, Shuicheng Yan, and Xiaojuan Qi. MC#: Mixture compressor for mixture-of-experts large models, 2025.

[13] 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. In *International Conference on Learning Representations*, 2025.

[14] Yushi Huang, Zining Wang, Zhihang Yuan, Yifu Ding, Ruihao Gong, Jinyang Guo, Xianglong Liu, and Jun Zhang. MoDES: Accelerating mixture-of-experts multimodal large language models via dynamic expert skipping, 2026.

[15] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.

[16] Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, Joe Chau, Peng Cheng, Fan Yang, Mao Yang, and Yongqiang Xiong. Tutel: Adaptive mixture-of-experts at scale, 2022.

[17] Kimi Team. Kimi-VL technical report, 2025.

[18] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *ACM SIGOPS Symposium on Operating Systems Principles (SOSP)*, 2023.

[19] Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. REAP the experts: Why pruning prevails for one-shot MoE compression, 2025.

[20] Jaeseong Lee, Seung-won Hwang, Aurick Qiao, Daniel Campos, Zhewei Yao, and Yuxiong He. Stun: Structured-then-unstructured pruning for scalable moe pruning. *arXiv preprint arXiv:2409.06211*, abs/2409.06211, 2024.

[21] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding, 2020.

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

[23] Kunchang Li, Yali Wang, Yinan He, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024.

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

[25] Yuan Liu, Haodong Duan, Yuanhan Zhang, et al. MMBench: Is your multi-modal model an all-around player? In *ECCV*, 2024.

[26] 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 *Annual Meeting of the Association for Computational Linguistics*, 2024.

[27] Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the structural pruning of large language models, 2023.

[28] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023.

[29] Ahmed Masry, Do Xuan Long, Jia Qing Tan, et al. ChartQA: A benchmark for question answering about charts with visual and logical reasoning, 2022.

[30] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, Anton Belyi, Haotian Zhang, Karanjeet Singh, Doug Kang, Ankur Jain, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Guoli Yin, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

[31] Alexandre Muzio, Alex Sun, and Churan He. Seer-moe: Sparse expert efficiency through regularization for mixture-of-experts. In *arXiv.org*, 2024. doi: 10.48550/arXiv.2404.05089.

[32] Qwen Team. Qwen3-VL technical report, 2025. arXiv preprint.

[33] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale. In *International Conference on Machine Learning (ICML)*, 2022.

[34] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

[35] Amanpreet Singh, Vivek Natarajan, Meet Shah, et al. Towards VQA models that can read. In *CVPR*, 2019.

[36] Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models, 2023.

[37] Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Q-VLM: Post-training quantization for large vision-language models. In *Advances in Neural Information Processing Systems*, 2024.

[38] Weiyun Wang et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025.

[39] Haoning Wu, Dongxu Li, Bei Chen, et al. LongVideoBench: A benchmark for long-context interleaved video-language understanding, 2024.

[40] Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, Zhenda Xie, Yu Wu, Kai Hu, Jiawei Wang, Yaofeng Sun, Yukun Li, Yishi Piao, Kang Guan, Aixin Liu, Xin Xie, Yuxiang You, Kai Dong, Xingkai Yu, Haowei Zhang, Liang Zhao, Yisong Wang, and Chong Ruan. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024.

[41] xAI. RealWorldQA benchmark, 2024.

[42] 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.

[43] Weihao Yu, Zhengyuan Yang, Linjie Li, et al. MM-Vet: Evaluating large multimodal models for integrated capabilities. In *ICML*, 2024.

[44] 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.

[45] Haotian Zhang, Mingfei Gao, Zhe Gan, Philipp Dufter, Nina Wenzel, Forrest Huang, Dhruti Shah, Xianzhi Du, Bowen Zhang, Yanghao Li, Sam Dodge, Keen You, Zhen Yang, Aleksei Timofeev, Mingze Xu, Hong-You Chen, Jean-Philippe Fauconnier, Zhengfeng Lai, Haoxuan You, Zirui Wang, Afshin Dehghan, Peter Grasch, and Yinfei Yang. Mm1.5: Methods, analysis & insights from multimodal llm fine-tuning. *arXiv preprint arXiv:2409.20566*, 2024.

[46] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality check on the evaluation of large multimodal models, 2024.

[47] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, 2024.

A Extended Related Work

Multimodal MoE compression. Many prior pruning techniques were originally designed for dense LLMs or text-only MoEs, and operate at various granularities: (1) expert skipping [26; 13; 14] and token compression [42] reduce runtime computation and activation load but do not mitigate the heavy model storage; (2) expert pruning [3; 19] removes whole experts, which shrinks the memory footprint but operates at too coarse a granularity and usually degrades accuracy; (3) pruning for dense LLMs includes unstructured weight sparsification such as Wanda [36] and structured dimension removal [2; 27], but carries no notion of routing or experts and therefore ignores the redundancy that is specific to MoEs. These methods are agnostic to the unique expert specialization of multimodal MoEs. Multimodal-specific studies such as Q-VLM [37], MC# [12], MoDES [14], and FastMMoE [42] show that vision-language models require modality-aware designs to better preserve performance. To the best of our knowledge, MAES is the first multimodal MoE compression method that reduces model parameters at fine-grained channel level, recognizing inner redundancy within each expert with awareness of modality affinity.

MoE expert parallelism and inference systems. Deploying large MoEs in multi-GPU systems typically relies on expert parallelism (EP), which shards experts across devices and dispatches each token to its assigned rank [21; 33; 16]. To make runtime computation efficient, mainstream inference implementations use fused MoE backends with kernel interfaces and packed weight layouts that require experts processed together to share specific dimensions [18; 47]. Recent work further folds communication into persistent MoE kernels [1]. Under these concrete backend constraints, channel-level pruning produces heterogeneous widths that require either padding to the maximum supported width or multiple kernel invocations, reducing the deployment gains of pruning. Our MAES therefore regularizes expert widths and balances the load on each rank, making pruned models directly compatible with existing EP and fused-kernel systems.

MoE compression: detailed comparison. Table 9 provides a systematic comparison of existing MoE compression methods, categorizing them by: (a) compression granularity (expert-level vs. channel-level), (b) multimodal support, and (c) whether training is required.

B Notation Reference

The notation used throughout the main text is sorted and listed in Table 7 below.

C Exact Hessian, GGN, and Gram structure of the MSE probe

This appendix derives the complete curvature of the proposed probe and clarifies the scope of its exactness. the input hidden states is h_ℓ and the routed contribution of expert is e as $a_{\ell,e}(h_\ell) = g_{\ell,e}(h_\ell) z_{\ell,e}(h_\ell)$. Stacking these vectors as columns gives $A_\ell(h_\ell) = [a_{\ell,1}, \ldots, a_{\ell,E}]$. Because the identity scalars are inserted after the expert transformations and before aggregation, the perturbed output is affine in $\boldsymbol{\beta}$:

$$\hat{y}_\ell(h_\ell; \boldsymbol{\beta}) - y_\ell(h_\ell) = A_\ell(h_\ell)(\boldsymbol{\beta} - \mathbf{1}). \tag{4}$$

The Hessian is exact for any loss. The probe is not tied to one loss: any differentiable $\mathcal{L}(\boldsymbol{\beta}) = \sum_{h_\ell \in \mathcal{D}} \ell(\hat{y}_\ell(h_\ell; \boldsymbol{\beta}), y_\ell(h_\ell))$ may be used, for instance a squared error or a KL divergence. The standard second-order chain rule decomposes such a loss Hessian into a generalized Gauss–Newton term and a residual-weighted model-curvature term. By Eq. (4) the output is affine in $\boldsymbol{\beta}$, so $\nabla^2_{\boldsymbol{\beta}} \hat{y}_\ell = 0$

Table 7: Summary of notation used throughout the paper.

Symbol	Description
L	Number of MoE layers
E	Number of routed experts per layer
D_{ffn}	Number of channels (intermediate neurons) per expert
ℓ, e, c	Layer index, expert index, channel index
p	Global channel pruning ratio
$\mathcal{D}$	Calibration set
$\mathcal{E}_\ell$	Set of (active) experts at layer ℓ
$\mathcal{C}_{\ell,e}$	Channel set of expert (ℓ, e)
h_ℓ	Input hidden state to MoE layer ℓ
$z_{\ell,e}$	Output of expert e at layer ℓ
$g_{\ell,e}(h_\ell)$	Router gate value of expert e on input h_ℓ
y_ℓ, $\hat{y}_\ell$	Original / perturbed MoE block output (Eq. 4)
$\rho^V_{\ell,e}, \rho^T_{\ell,e}$	Routing frequency of visual / text tokens at expert (ℓ, e)
$\eta_{\ell,e}$	Expert Modality Affinity (EMA) score $\in [-1, +1]$
$\alpha^V_{\ell,e}, \alpha^T_{\ell,e}$	Per-modality affinity weights derived from $\eta_{\ell,e}$ (Eq. 1)
$s^M_{\ell,e,c}$	Channel importance score for modality $M \in \{V, T\}$
$s^M_{\ell,e,(r^M)}$	Channel ranked r^M-th under modality M
$B_{\ell,e}$	Per-expert channel budget after expert budget allocation
$\mathcal{S}_{\ell,e}$	Selected channels of expert (ℓ, e) after progressive expansion (Eq. 1)
β	Scalar identity factor on expert output ($\beta = 1$ at operating point)
$\mathcal{L}(\beta)$	Blockwise MSE calibration loss (Eq. 5)
$\Delta\mathcal{L}_{\ell,e}$	Loss change from removing expert e via second-order expansion
$h_{\ell,e}$	Diagonal Hessian entry for expert e at $\boldsymbol{\beta} = \mathbf{1}$ (Eq. 14)
$s^{\text{expert}}_{\ell,e}$	Exact single-expert block-MSE increase ($\frac{1}{2}h_{\ell,e}$)
C	Hessian-diagonal chunk size for chunked computation
m	Number of EP ranks, equal to the number of width levels
k, u	Width-level index, EP rank index
$\mathcal{M}$	Set of width levels $\{w_1, \ldots, w_m\}$; w_k is the width of level k
n^ℓ_k	Number of experts at layer ℓ assigned to level w_k
w^ℓ_e	Quantized width of expert e at layer ℓ
ϕ^ℓ_k	Static weight load of level k at layer ℓ ($\phi^\ell_k = w_k n^\ell_k$)
$G_{\ell,k}$	EP rank hosting level k of layer ℓ (Algorithm 2)
Φ_u	Cumulative static weight load of rank u over all layers
$\Delta\Phi$	Per-rank load deviation ($\max_u \Phi_u - \min_u \Phi_u$)
q	Atomic width, the largest integer dividing every $w_k \in \mathcal{W}$
Φ_{tot}	Total static weight load over all layers and levels
τ_0	Closed-form lower bound on $\Delta\Phi$ (Proposition 1)

everywhere and the model-curvature term vanishes whatever ℓ is; at $\boldsymbol{\beta} = \mathbf{1}$ the residual is zero as well, which makes it vanish for a second, compatible reason. Hence the complete Hessian is exactly the GGN, not an approximation to it. This step alone does not make the second-order expansion exact: it fixes $H_{\boldsymbol{\beta}}$, but the remainder R_3 of Eq. (2) is governed by third derivatives, which will become zero if using quadratic loss.

A quadratic-error loss removes the remainder. Substituting Eq. (4) into the reconstruction MSE gives

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{h_\ell \in \mathcal{D}} \|A_\ell(h_\ell)(\boldsymbol{\beta} - \mathbf{1})\|_2^2 = (\boldsymbol{\beta} - \mathbf{1})^\top G_\ell (\boldsymbol{\beta} - \mathbf{1}), \quad G_\ell = \sum_{h_\ell \in \mathcal{D}} A_\ell(h_\ell)^\top A_\ell(h_\ell), \tag{5}$$

a quadratic form in $\boldsymbol{\beta}$. Consequently,

$$\nabla_{\boldsymbol{\beta}} \mathcal{L} = 2G_\ell(\boldsymbol{\beta} - \mathbf{1}), \qquad \nabla^2_{\boldsymbol{\beta}} \mathcal{L} = 2G_\ell, \qquad \nabla^k_{\boldsymbol{\beta}} \mathcal{L} = 0 \ \ \forall k \geq 3, \tag{6}$$

so $R_3 = 0$ and the second-order expansion holds with equality. The Hessian is moreover a positive-semidefinite Gram matrix whose diagonal measures individual expert energy and whose off-diagonal entries measure expert-output alignment:

$$H_{ee} = 2\sum_h \|a_{\ell,e}(h)\|_2^2, \qquad H_{ef} = 2\sum_h a_{\ell,e}(h)^\top a_{\ell,f}(h). \tag{7}$$

The Hessian diagonal equals the single-expert removal score. At $\boldsymbol{\beta} = \mathbf{1}$ the perturbed output equals the original one, so $\boldsymbol{\beta} = \mathbf{1}$ is the minimizer of $\mathcal{L}$ and both $\mathcal{L}(\mathbf{1}) = 0$ and $\nabla\mathcal{L}(\mathbf{1}) = \mathbf{0}$. For single-expert removal, set $\Delta\boldsymbol{\beta} = -\mathbf{e}_e$; the constant, first-order and remainder terms of Eq. (2) all drop out, leaving the curvature term alone:

$$\mathcal{L}(\mathbf{1} - \mathbf{e}_e) - \mathcal{L}(\mathbf{1}) = \frac{1}{2}\mathbf{e}_e^\top H \mathbf{e}_e = \frac{1}{2}H_{ee} = \sum_h \|a_{\ell,e}(h)\|_2^2. \tag{8}$$

Thus the diagonal score exactly equals the single-expert removal loss under this blockwise MSE. For jointly removing a set S, the exact loss is $\frac{1}{2}\mathbf{1}_S^\top H \mathbf{1}_S$ and includes off-diagonal interactions H_{ef}. The diagonal method intentionally ignores these interactions and should not be described as an exact joint subset-selection objective.

Isolated versus synchronous perturbation. Writing $g_e = \partial\mathcal{L}/\partial\beta_e$ and $D_e = H_{ee}/2$, perturbing only expert e with all other scales fixed at 1 gives

$$g_e = (\beta_e - 1)H_{ee}, \qquad D_e = \frac{g_e}{2(\beta_e - 1)}, \quad \beta_e \neq 1. \tag{9}$$

This recovery uses the exact quadratic identity, rather than a linear Taylor extrapolation, and requires a separate perturbed evaluation for each expert. With synchronous scaling $\boldsymbol{\beta} = \beta\mathbf{1}$, the same conversion instead gives

$$\frac{g_e(\beta\mathbf{1})}{2(\beta - 1)} = \frac{1}{2}\sum_f H_{ef} = D_e + \frac{1}{2}\sum_{f\neq e} H_{ef}, \quad \beta \neq 1. \tag{10}$$

It recovers D_e only when the off-diagonal row sum vanishes. At $\boldsymbol{\beta} = \mathbf{1}$, all first-order gradients are zero, but the Hessian remains $2G_\ell$. Extracting its diagonal therefore recovers the individual removal costs without changing the operating point or including cross-expert terms.

Definition of synchronous first-order probes. Since we compare against synchronous first-order estimation in the main text, we provide its definition here. The first-order Taylor deletion score differs from the quadratic recovery formula above. At a synchronous operating point $\beta\mathbf{1}$, removing expert e changes its scale by $-\beta$, giving the signed linear term $-\beta g_e(\beta\mathbf{1})$ and sensitivity $|\beta g_e(\beta\mathbf{1})|$. The raw derivative $|g_e|$ omits this deletion step. At unit scale, the linear term reduces to the usual $-g_e$, consistent with gradient-activation multiplication. For the quadratic objective, the full change from the perturbed background is

$$\mathcal{L}(\beta\mathbf{1} - \beta\mathbf{e}_e) - \mathcal{L}(\beta\mathbf{1}) = -\beta g_e(\beta\mathbf{1}) + \tfrac{1}{2}\beta^2 H_{ee}. \tag{11}$$

Therefore, multiplying synchronous gradients by a shared positive β preserves their ranking and ratios, but scales absolute gaps and does not remove cross-expert interactions.

For this exact MSE objective, $H_{ee}/2$ can also be accumulated directly from routed expert-output energy without automatic differentiation. We retain HVPs as a common backend for non-quadratic and downstream-propagated loss variants, where this direct identity and exact agreement with removal loss need not hold.

Non-quadratic losses. In our experiment, we use quadratic losses, such as L2 loss. But for general losses such as cosine similarity, $\mathcal{L}(\beta)$ becomes a non-quadratic function of β due to nonlinear operations (e.g., softmax and logarithm). In this case,

$$\frac{\partial^k \mathcal{L}}{\partial \beta^k} \neq 0 \quad \text{for } k \geq 3, \tag{12}$$

and the Taylor expansion includes a non-zero remainder term:

$$\mathcal{L}(0) = \mathcal{L}(1) - \mathcal{L}'(1) + \frac{1}{2}\mathcal{L}''(1) - \frac{1}{6}\mathcal{L}^{(3)}(\xi), \quad \xi \in (0, 1), \tag{13}$$

so the second-order approximation is no longer exact.

D Implementation Details

D.1 Experiment Setting

All experiments run on NVIDIA H20 GPUs (96 GB HBM). Backend inference and evaluation use vLLM. All benchmarks are run through lmms-eval v0.7.1, following each benchmark's official

few-shot count, prompt template and metric. Judged benchmarks (e.g. MM-Vet) use a locally hosted Qwen2.5-32B-Instruct judge served via vLLM, not a remote GPT judge. We do not report COCO2017-Cap as an evaluation benchmark because the models do not release an official results.

D.2 Proof of the Placement Lower Bound

We restate and prove Proposition 1. Let $\mathcal{W} = \{w_1, \ldots, w_m\}$ be the width levels, n_k^ℓ the number of experts at layer ℓ carrying level w_k, and $\phi_k^\ell = w_k n_k^\ell$ the resulting static load. A placement G assigns each (ℓ, k) to one of m ranks; rank u accumulates $\Phi_u = \sum_\ell \phi_k^\ell \, [\, G_{\ell,k} = u \,]$, the total load is $\Phi_{\text{tot}} = \sum_u \Phi_u = \sum_{\ell,k} \phi_k^\ell$, and the objective is the spread $\Delta\Phi = \max_u \Phi_u - \min_u \Phi_u$. Write q for the atomic width, the largest integer dividing every $w_k \in \mathcal{M}$.

Proposition 1. *For every placement G, $\Delta\Phi(G) \geq \tau_0$, where $\tau_0 = 0$ when $m \mid (\Phi_{\text{tot}}/q)$ and $\tau_0 = q$ otherwise.*

Proof. The proof can be divided into three steps.

Step 1: every rank load is a multiple of q. By definition of the greatest common divisor, each w_k is an integer multiple of q, say $w_k = q c_k$ with $c_k \in \mathbb{Z}_{>0}$. Hence $\phi_k^\ell = w_k n_k^\ell = q\, c_k n_k^\ell$ is an integer multiple of q, and Φ_u, being a finite sum of such terms, is an integer multiple of q as well. Consequently $\Delta\Phi = \max_u \Phi_u - \min_u \Phi_u$ is a difference of two integer multiples of q and is therefore itself an integer multiple of q. In particular $\Delta\Phi \in \{0, q, 2q, \ldots\}$; it cannot take any value strictly between 0 and q.

Step 2: perfect balance forces divisibility. Suppose $\Delta\Phi(G) = 0$. Then $\max_u \Phi_u = \min_u \Phi_u$, so all m rank loads are equal to a common value Φ^*. Summing over ranks gives $m\Phi^* = \Phi_{\text{tot}}$, that is $\Phi^* = \Phi_{\text{tot}}/m$. By Step 1, Φ^* is an integer multiple of q, say $\Phi^* = qt$ with $t \in \mathbb{Z}_{\geq 0}$. Then $\Phi_{\text{tot}} = mqt$, and dividing by q yields $\Phi_{\text{tot}}/q = mt$, i.e. $m \mid (\Phi_{\text{tot}}/q)$.

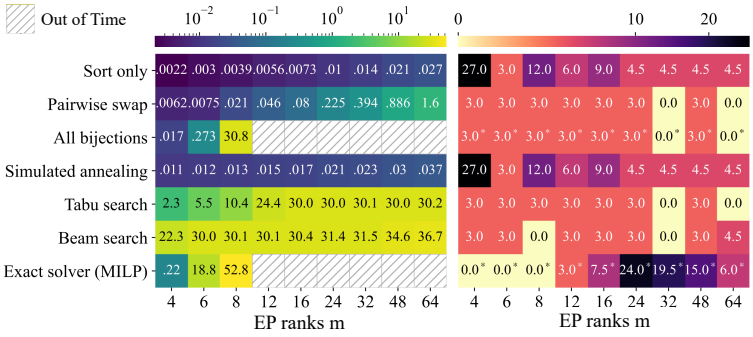
Step 3: combining the two. If $m \mid (\Phi_{\text{tot}}/q)$ the claim is the trivial bound $\Delta\Phi \geq 0$. If instead $m \nmid (\Phi_{\text{tot}}/q)$, the contrapositive of Step 2 gives $\Delta\Phi(G) \neq 0$; by Step 1, $\Delta\Phi(G)$ is a multiple of q, and the smallest positive such multiple is q, so $\Delta\Phi(G) \geq q$. □

The bound is directly computable, so we can regard it as an optimal target and easily measure one placement by its distance to the optimum. Any placement G must satisfies $\Delta\Phi(G) \geq \tau_0$, so $\Delta\Phi(G) - \tau_0$ is an upper bound before the refinement is conducted. Only when τ_0 is attainable, the algorithm can finally reach τ_0 after refinement. Table 8, Figures 5 and 7 show the exact solver empirically attains this bound in our tested cases, especially when the model has more layers, its easier to have a more balanced load placement. Any heuristic algorithm that improves the balance over the initial placement can be applied here. We therefore leave REFINE unspecified in Algorithm 2.

Table 8 gives the per-depth numbers across different layer counts and heuristic algorithms. We find that as depth increases, placements more readily approach the optimal balance. Figure 7 repeats the main text's m-sweep at $p = 0.5$, where $\tau_0 = 0$ for every tested m and the exact solver reaches this bound at $m \in \{4, 6, 8\}$ before timing out at larger ranks.

D.3 Compared Methods

We compare MAES with representative MoE compression baselines covering channel-level pruning, expert-level pruning, and multimodal MoE compression. **C-Prune** [10] clusters functionally similar experts within each MoE layer by parameter similarity and prunes redundant experts within each cluster. **He-Unified** [11] provides a unified MoE compression framework that combines expert-level trimming with intra-expert slimming, reducing both the number of active experts and the width of remaining experts. **DiEP** [3] formulates expert pruning as a differentiable optimization problem and learns non-uniform layer-wise expert pruning decisions instead of applying a fixed pruning ratio to all layers. **REAP** [19] estimates expert saliency from router gate values and expert activation norms, and performs one-shot router-weighted expert pruning. **MoNE** [44] reduces MoE redundancy by replacing low-importance experts with lightweight novice experts rather than directly removing them. **ACP** [6] prunes at the channel level inside experts, observes that expert information concentrates in a small subset of channels. It is training-free and shares our channel-level granularity, but targets unimodal LLM MoEs and assigns each expert a single budget without distinguishing which modality



(a) Time cost (s, log scale) (b) Load spread $\Delta\Phi(G)$(MiB)

Figure 7: (a) Solving time per REFINE operator and (b) computation load spread $\Delta\Phi(G)$. We sweep over EP rank m, on the full 48-layer, p=0.5 prune ratio. Under this setting, the perfect τ_0=0 MiB for every tested m.

Table 8: Distance from the bound of Proposition 1 across depths L at $m = 4$, for all seven REFINE operators and both pruning ratios. We report $\Delta\Phi$ (MiB) in black, and grey subscript denote $\Delta\Phi - \tau_0$). $\bar{\Phi}$ is the memory amount each rank holds.

Refine operator ($p = 0.3$)	L=4	8	12	16	24	32	48
Sort only	147.8 146.2	145.5 144.0	206.2 204.8	157.5 156.0	208.5 207.0	216.0 214.5	163.5 162.0
Pairwise swap	42.0 40.5	39.0 37.5	10.5 9.0	19.5 18.0	20.2 18.8	15.0 13.5	3.0 1.5
All bijections	45.0 43.5	39.0 37.5	15.0 13.5	19.5 18.0	7.5 6.0	3.0 1.5	3.0 1.5
Simulated annealing	40.5 39.0	29.2 27.8	17.2 15.8	10.5 9.0	18.8 17.2	10.5 9.0	25.5 24.0
Tabu search	42.0 40.5	14.2 12.8	3.8 2.2	4.5 3.0	6.0 4.5	1.5 0.0	4.5 3.0
Beam search	42.0 40.5	15.8 14.2	9.8 8.2	7.5 6.0	4.5 3.0	1.5 0.0	3.0 1.5
Exact solver (MILP)	30.0 28.5	1.5 0.0	1.5 0.0	1.5 0.0	1.5 0.0	1.5 0.0	1.5 0.0
τ_0	1.5	1.5	1.5	1.5	1.5	1.5	1.5
$\bar{\Phi}$ (GiB)	0.78	1.56	2.36	3.19	4.73	6.26	9.45
Refine operator ($p = 0.5$)	L=4	8	12	16	24	32	48
Sort only	102.8 101.2	76.5 75.0	73.5 72.0	64.5 63.0	71.2 69.8	66.0 66.0	27.0 27.0
Pairwise swap	69.0 67.5	31.5 30.0	14.2 12.8	12.0 10.5	17.2 15.8	9.0 9.0	3.0 3.0
All bijections	69.0 67.5	27.8 26.2	14.2 12.8	12.0 10.5	14.2 12.8	9.0 9.0	3.0 3.0
Simulated annealing	36.0 34.5	25.5 24.0	18.8 17.2	18.0 16.5	5.2 3.8	4.5 4.5	27.0 27.0
Tabu search	27.0 25.5	15.0 13.5	6.8 5.2	7.5 6.0	6.8 5.2	4.5 4.5	3.0 3.0
Beam search	43.5 42.0	21.8 20.2	10.5 9.0	7.5 6.0	3.0 1.5	3.0 3.0	3.0 3.0
Exact solver (MILP)	27.0 25.5	1.5 0.0	1.5 0.0	1.5 0.0	1.5 0.0	0.0 0.0	0.0 0.0
τ_0	1.5	1.5	1.5	1.5	1.5	0.0	0.0
$\bar{\Phi}$ (GiB)	0.56	1.11	1.67	2.27	3.38	4.44	6.75

its channels serve. All baselines are evaluated under comparable settings using the same calibration data when applicable.

D.4 LAYER AND CHANNEL SCORE COLLECTION

We compare different layer-level importance estimation methods in Section 5.3. Here, we provide the definition of each estimation metric. Common choices fall into three categories: (i) *Uniform* assigns equal importance to all experts and is used as a simple baseline. (ii) *Routing-based* scores use routing statistics such as token usage frequency or router logit magnitudes, which are cheap to collect but measure selection probability rather than actual contribution to the output. (iii) *Loss-based* scores estimate the loss change induced by removing an expert, either via a first-order Attribution-guided Loss Approximation (ALA), which approximates $\Delta\mathcal{L}^{(e)} \approx -(\partial\mathcal{L}_\ell/\partial y_\ell)^\top z_{\ell,e}$ in a single backward pass, or via the second-order estimate used in MAES (Eq. 14), which additionally captures curvature

Table 9: Comparison of MoE compression methods.

Method	Type	Multimodal	Training-free	Granularity
NAEE [26]	Expert prune/skip	No	Yes	Expert
MC-MoE [13]	Quant + dyn. prune	No	Yes	Expert
DiEP [3]	Expert prune	No	Yes	Expert
Q-VLM [37]	Quantization	Yes	Yes	Weight
MC# [12]	Quant + expert prune	Yes	Partial	Expert
MoDES [14]	Expert skip	Yes	Yes	Token
FastMMoE [42]	Expert skip + token	Yes	Yes	Expert + Token
ACP [6]	Channel prune	No	Yes	Channel
MAES (Ours)	Channel prune	Yes	Yes	Channel

Table 10: Comparisons on InternVL3_5-30B-A3B-HF under 30% and 50% compression ratios.

p	Method	General Image						Text-rich Image		Video				Avg
		GQA	MMS	MMB	MMV	MME	RWQA	TxVQA	CQA	MVB	VMME	LVB	EgoS	
0	Baseline	62.93	70.98	86.17	80.23	2446.25	69.28	78.28	88.32	72.16	60.52	58.79	80.40	74.62
0.3	DiEP [3]	43.52	58.72	77.98	52.39	2125.08	65.62	73.90	63.91	63.37	49.53	56.02	75.8	63.05
	C-Prune [10]	58.93	64.32	83.42	64.82	2244.11	66.27	72.86	71.72	69.51	59.61	55.72	74.8	68.51
	He-Unified [11]	54.47	57.40	78.24	59.31	2061.16	63.27	77.11	77.21	59.55	54.94	52.33	72.6	65.00
	MoNE [44]	63.25	62.82	83.42	63.90	2325.28	69.15	77.19	70.98	68.69	**62.60**	58.12	**82.4**	70.46
	ACP [6]	61.36	67.14	82.86	70.37	2346.40	67.09	76.87	82.04	69.79	58.44	57.69	80.8	71.52
	REAP [19]	62.91	61.80	84.72	60.83	2308.19	69.02	**77.41**	69.80	66.74	61.80	**58.52**	80.8	69.73
	MAES *(Ours)*	**63.87**	**70.79**	**84.88**	**70.83**	**2434.10**	**69.41**	77.27	**86.24**	**71.13**	58.56	58.27	79.2	**73.12**
0.5	DiEP [3]	37.32	45.30	53.11	29.08	1906.14	55.95	65.53	47.61	49.29	49.76	45.12	66.8	51.08
	C-Prune [10]	47.75	57.55	71.24	53.49	2087.56	53.07	66.62	60.50	61.95	**56.02**	49.08	70.2	60.17
	He-Unified [11]	35.77	40.15	47.93	15.14	2118.69	45.75	65.73	45.40	16.06	41.52	32.42	61.2	43.56
	MoNE [44]	62.25	52.92	79.02	48.85	2055.69	66.14	71.45	47.42	62.25	58.48	56.33	**80.6**	63.26
	ACP [6]	59.29	63.55	79.87	59.57	2134.80	65.32	70.11	**79.44**	64.26	54.22	56.81	72.0	66.72
	REAP [19]	**63.13**	56.90	81.35	46.56	2012.34	67.32	69.50	41.73	63.67	59.72	56.98	79.0	63.14
	MAES *(Ours)*	63.01	**66.84**	**83.33**	**62.16**	**2257.74**	**68.63**	**72.67**	65.92	**69.03**	54.22	**57.82**	76.0	**68.36**

and provides a more accurate proxy for removability. As for intra-expert channel scoring, we simply use activation magnitude following [6].

E Additional Results

Additional benchmark results on InternVL3.5-30B-A3B-HF. Table 10 reports the full per-task comparison on InternVL3.5-30B-A3B-HF at 30% and 50% channel pruning, complementing the Kimi-VL and Qwen3-VL-MoE results in Section 5.2. MAES retains 98.0% of the original average accuracy at 30% pruning (73.12 vs. 74.62) and 91.6% at 50% pruning (68.36 vs. 74.62), consistently outperforming all baselines at both ratios.

Full component-ablation results. Table 11 provides the complete per-task results underlying the compact component-ablation summary in Table 4. Tasks are partitioned into general image, text-rich image, and video categories: general image includes GQA, MMStar, MMBench, MMVet, MME, and RealWorldQA; text-rich image includes TextVQA and ChartQA; video includes MVBench, VideoMME, LongVideoBench, and EgoSchema. Before computing category and overall averages, raw MME scores are converted to the same 0–100 scale as the other benchmarks using $\mathrm{MME}/2800 \times 100$.

Comparison of intra-layer expert importance criteria. Table 12 extends the analysis in Section 5.3 by comparing the proposed second-order Hessian-diagonal expert score $h_{\ell,e}$ against representative intra-layer baselines, including uniform allocation, token-count, router-logit magnitude, and the first-order attribution score $|g_{\ell,e}|$. We also report three loss variants for the second-order estimator (vanilla ℓ_2, relative ℓ_2, and KL divergence) on Kimi-VL-A3B-Instruct at 50% channel pruning, with $\Delta\mathcal{L}_{\ell,e}^{\mathrm{true}}$ obtained by directly ablating each expert as the upper-bound reference. All three Hessian variants outperform routing- and first-order-based baselines in average accuracy, and the ℓ_2 variant attains 69.57 Avg, surpassing both the uniform (67.48) and first-order (68.68) baselines. This confirms that

Table 11: Full per-task ablation of MAES's components on Kimi-VL-A3B-Instruct at 30% and 50% channel pruning. Baseline is ACP [6] at the same pruning ratio. Overall Avg is the unweighted mean over available tasks, with raw MME scores converted to the 0–100 scale as $\mathrm{MME}/2800 \times 100$.

Method	General Image						Text-rich Image		Video				Overall Avg
	GQA	MMS	MMB	MMV	MME	RWQA	TxVQA	CQA	MVB	VMME	LVB	EgoS	
Compression Ratio $p = 0.3$													
$p = 0$	62.67	65.25	83.16	68.90	2195.45	65.88	88.64	89.36	60.18	63.41	60.88	78.5	72.10
Baseline	61.95	61.26	79.86	57.66	2020.70	66.06	84.36	85.84	59.21	61.78	57.29	69.6	68.09
MACS	61.77	62.09	81.86	59.68	2224.51	68.00	84.02	85.52	59.00	64.22	62.63	70.8	69.92
MACS + SOES	62.07	61.86	82.56	62.34	2145.66	66.01	86.38	86.96	60.42	63.48	60.81	73.2	70.23
MACS + SOES + CLER	62.04	62.55	82.22	64.86	2155.07	66.14	86.35	87.42	61.68	62.33	60.51	73.0	70.51
Compression Ratio $p = 0.5$													
Baseline	59.09	56.79	76.87	37.89	2072.30	63.71	79.12	75.88	56.89	56.52	57.40	65.2	63.28
MACS	60.06	58.51	79.03	44.40	2137.71	66.00	79.82	78.00	56.95	58.74	60.39	68.2	65.54
MACS + SOES	60.36	56.28	80.24	50.50	2085.24	62.88	81.43	79.64	59.58	60.04	60.21	70.6	66.35
MACS + SOES + CLER	60.68	57.46	80.76	51.74	2110.28	65.36	82.10	81.24	59.74	59.04	59.69	72.4	67.13

Table 12: Comparison of different intra-layer methods on multimodal benchmarks. For second-order Hessian criteria, we adopt three variants: L2 norm, relative L2 norm, and KL divergence. We use Kimi-VL-A3B-Instruct at 50% pruning ratio.

Method	GQA	TxVQA	CQA	MMS	MMB	MME	RWQA	Avg
$\Delta\mathcal{L}_{\ell,e}^{\mathrm{true}}$	60.65	79.27	76.47	61.99	80.65	2038.96	64.26	69.57
Uniform	58.95	77.56	73.26	56.06	78.39	2040.36	63.34	67.48
Token count	59.59	76.71	69.55	57.85	78.25	2076.48	62.78	67.22
Router logits	59.48	78.02	74.26	56.70	80.19	2054.08	62.59	68.06
$\lvert g_{\ell,e}\rvert$	59.57	79.09	75.88	56.79	79.87	2072.30	63.71	68.68
$h_{\ell,e}$ ($l_{\mathrm{kl_div}}$)	60.70	79.21	73.92	63.04	80.37	2035.04	63.71	69.19
$h_{\ell,e}$ (Rel l_2)	59.82	81.60	76.46	55.57	81.99	2010.40	64.44	69.10
$h_{\ell,e}$ (l_2)	60.65	79.27	76.47	61.99	80.65	2038.96	64.26	69.57

curvature-aware estimation provides a tighter proxy of the true ablated loss and translates to better downstream accuracy under the same calibration budget.

Per-task ablation of calibration data sources. Table 13 provides per-task results behind Table 6 in the main text, comparing single-source (GQA, COCO, M4-Instruct) and mixed-source calibration with three sampling strategies (Core, Outlier, Uniform). All settings use 1024 samples with 2048 tokens per sample on Kimi-VL-A3B-Instruct at 50% pruning. Uniform farthest-point sampling on the mixed corpus achieves the best average score (71.85), outperforming the strongest single-source setting (M4-Instruct, 64.39) and both Core/Outlier samplers (64.18/64.47). The improvement is consistent across diverse tasks (e.g., MMB 80.76 and RWQA 65.36), supporting the claim that broader feature-space coverage produces more representative channel-importance estimates and is more robust on cross-domain benchmarks.

E.1 Chunked Hessian-Diagonal Processing and Analysis

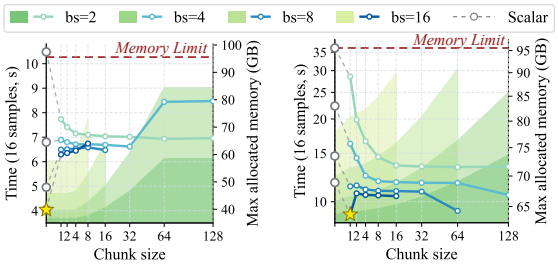
Chunked Hessian-diagonal processing. To keep the criterion available for any loss probe, we differentiate twice by default. However, materializing all $|\mathcal{E}_\ell| = 128$ experts' Hessian–vector products in one batched call may exhaust device memory. We partition the active experts $\mathcal{E}_\ell$ at layer ℓ into $\lceil |\mathcal{E}_\ell|/C \rceil$ chunks with chunk size C, and recover the Hessian diagonal chunkwise as

$$h_{\ell,e} \;=\; \mathbf{e}_e^\top \left(\nabla^2_{\boldsymbol{\beta}} \mathcal{L}\right)\Big|_{\boldsymbol{\beta}=\mathbf{1}} \mathbf{e}_e, \qquad e \in \mathcal{E}_\ell^{(b)}, \tag{14}$$

where $\mathbf{e}_e \in \mathbb{R}^{|\mathcal{E}_\ell|}$ is the unit vector along coordinate e. Each chunk is computed independently by one batched Hessian-vector product. Since each $\partial^2\mathcal{L}/\partial\beta_e^2$ is independently derived, it yields the same results as processing all at once, while reducing peak memory from $O(|\mathcal{E}_\ell|^2)$ to $O(C \cdot |\mathcal{E}_\ell|)$. Section E.1 sweeps the chunk size C and the calibration batch size across MoEs to map the speed–memory trade-off.

Table 13: Ablation over calibration corpora and mixing strategies on Kimi-VL-A3B-Instruct at 50% pruning ratio.

Source	Sample	Token/Sample	GQA	TxVQA	CQA	MMS	MMB	MME	RWQA	Avg
Single-Source Calibration Data										
GQA	1024	2048	61.03	78.87	71.04	46.67	82.84	1801.52	43.40	64.03
COCO	1024	2048	59.16	79.96	71.20	44.60	82.17	1767.36	37.91	62.59
M4-Instruct	1024	2048	60.54	81.07	70.52	44.40	84.59	1781.08	46.01	64.39
Mixed-Source Calibration Data										
Core Sampling	1024	2048	59.22	81.00	71.92	44.20	83.78	1730.12	47.32	64.18
Outlier Sampling	1024	2048	59.33	79.97	72.24	47.93	83.39	1719.20	47.06	64.47
Uniform Sampling	1024	2048	60.68	82.10	81.24	57.46	80.76	2110.28	65.36	71.85

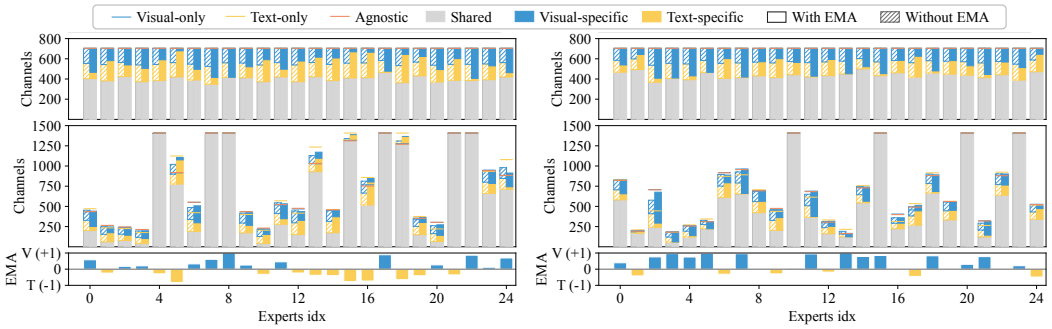


(a) Kimi-VL-A3B-Instruct (b) Qwen3-VL-30B-A3B-Instruct

Figure 8: Time cost of processing 16 samples (left y-axis) and memory requirement (right y-axis) of the chunked second-order estimation algorithm under different batch sizes and chunk sizes. Settings with the lowest time cost are marked with stars.

Time and memory cost of chunked Hessian-diagonal computation. Figure 8 sweeps batch and chunk sizes for our chunked vectorized estimator against a scalar baseline that processes one β at a time. The fastest setting is model dependent: vectorization substantially reduces wall-clock time on Qwen3-VL-MoE-30B-A3B-Instruct by amortizing the forward-backward overhead across experts, whereas Kimi-VL-A3B-Instruct, with fewer but heavier active experts per layer, is more efficient with scalar β. Chunk size controls a memory-time trade-off, where larger chunks have fewer iterations but raise peak memory. We therefore regard chunked computation and C as a tunable parameter depending on model and device resources.

Visualization of expert-wise channel allocation. Figure 9 visualizes the per-expert channel budgets produced by MAES across multiple MoE layers, complementing the single-layer view in the main paper (Section 4.1). For each layer we plot three reference allocations as flat lines—text-only ranking (yellow), vision-only ranking (blue), and modality-agnostic pruning (orange)—and overlay stacked bars showing MAES allocations with and without the modality-affinity weighting. Across layers, modality-affinity weighting redistributes channel budget toward each expert's affinitive modality: vision-dominated experts inherit a larger share of visually salient channels, while text-dominated experts inherit more text-salient channels. This shows that the EMA-driven expansion in Eq. (1) produces qualitatively different allocations from any single-modality or modality-agnostic baseline, and confirms that the budget shifts observed in the main paper persist across depth.



(a) Layer 15 (b) Layer 26

Figure 9: Expert-wise channel allocation at different layers. Flat lines are text-only (yellow), vision-only (blue), and modality-agnostic pruning results (orange); stacked bars compare allocation with and without modality-affinity weighting. Within each stacked bar, the grey portion ("Shared") counts channels selected by both modality rankings, while the blue and yellow portions count the vision- and text-specific channels contributed by each ranking.