DECOMOE: DECOUPLING VISUAL PROPAGATION AND EXPERT COMPUTATION FOR EFFICIENT MULTIMODAL MOE INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

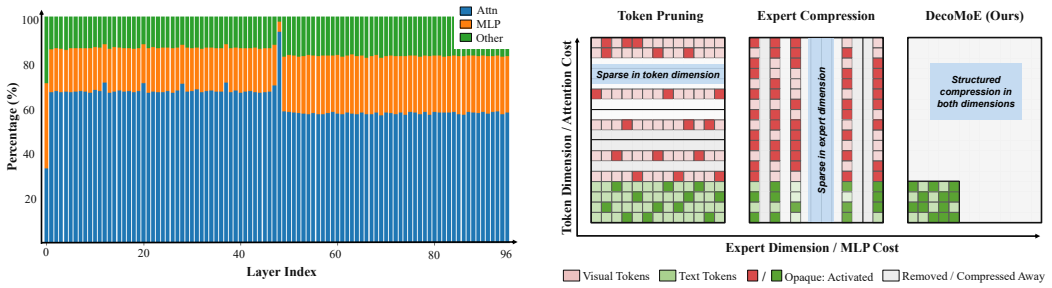
Multimodal mixture-of-experts (MoE) models combine sparse expert activation with visual-language capabilities, yet their inference remains costly because long visual-token sequences repeatedly incur attention, routing, dispatch, and expert-MLP computation. Existing methods typically compress either the token or expert dimension, leaving redundancy along the other. Our analysis reveals two complementary regularities: the depth required for visual propagation varies across inputs, while text-token routing exhibits concentrated and recurrent expert-importance patterns. Based on these observations, we propose DecoMoE, a two-dimensional structured compression framework that decouples visual propagation from expert computation. The Sample-Adaptive Visual Boundary (SAVB) predicts an input-dependent visual-exit layer at which the visual-token block is removed. The Routing-Calibrated Expert Prefix (RCEP) reorders experts offline using text-token routed mass and, from this predicted exit layer onward, retains at each MoE layer the shortest contiguous prefix covering a target routed-mass fraction. We evaluate DecoMoE on Qwen3-VL-MoE and InternVL3.5-30B-A3B across six benchmarks. On Qwen3-VL-MoE, DecoMoE retains 97.91% of dense-baseline performance while reducing computation from 27.06 to 16.73 TFLOPs and latency from 0.44 to 0.26 seconds, yielding a 1.69× speedup.

1 INTRODUCTION

Multimodal large language models (MLLMs) extend language models to visual understanding and reasoning Liu et al. (2023); Wang et al. (2024); Chen et al. (2024b), while mixture-of-experts (MoE) architectures scale model capacity through sparse token-wise expert activation Jiang et al. (2024); Wu et al. (2024). However, sparse activation alone does not make multimodal MoE inference inexpensive: long visual-token sequences may propagate through many decoder layers, repeatedly incurring attention, routing, dispatch, and expert-MLP computation. Multimodal MoE inference therefore presents a two-dimensional compression problem: the token dimension determines the sequence length and its associated attention cost, whereas the expert dimension determines the expert span available to routing and the corresponding dispatch and MLP costs. Compressing only one dimension leaves substantial redundancy along the other.

Figure 1(b) places existing approaches in this two-dimensional computation space. Visual-token pruning methods such as FastV and SparseVLM sparsify the token dimension within the language model Chen et al. (2024a); Zhang et al. (2024), while whole-block methods such as VTW and DyVTE remove the complete visual-token block at a selected layer Lin et al. (2025); Wu et al. (2026). Both reduce attention-side computation but retain the original expert span. Conversely, expert pruning and skipping methods reduce the expert footprint or activated expert computation Lu et al. (2024); Huang et al. (2026), while typically preserving the full multimodal sequence. Recent multimodal MoE methods combine token reduction with routing-aware expert activation or modality-specific expert skipping Xia et al. (2025); Huang et al. (2025b), suggesting that both dimensions contain reducible redundancy. Yet reducing both dimensions is not sufficient by itself: the token-side decision should reflect when visual propagation can stop for the current input, while expert-side compression should organize routing decisions into a structure that supports efficient

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) Per-layer latency composition **(b) Compression across token and expert dimensions**

Figure 1: Computation characteristics and structured compression of multimodal MoE inference. (a) Per-layer latency composition across one prefill pass and one decoding pass, shown as two consecutive 48-layer segments. Except for a small number of boundary-layer outliers, Attention and the expert MLP jointly account for more than 80% of the measured inference latency. (b) Comparison of compression across the token and expert dimensions, corresponding to attention and expert-MLP costs. Token pruning and expert compression sparsify one dimension, whereas DecoMoE removes the visual-token block and retains a contiguous prefix of experts. Opaque cells denote routed activations; gray regions denote removed computation.

execution. Existing approaches do not explicitly use a sample-dependent visual boundary to initiate text-calibrated, physically contiguous expert-prefix execution.

Our diagnostics reveal two complementary regularities (Figure 3). First, the earliest layer at which the complete visual-token block can be removed while preserving the answer varies across inputs, and later exits typically remain in a stable low-perplexity region. This motivates controlling visual propagation at the sample level rather than with one global layer. Second, text-token routing exhibits higher routed-mass concentration and larger cross-case expert overlap than visual-token routing. This suggests that expert importance for the subsequent text-only computation should be calibrated from text-aligned routing statistics rather than from mixed multimodal routing.

Motivated by these findings, we propose **DecoMoE**, a routing-aware framework that decouples the sample-dependent depth of visual propagation from the recurring expert structure used for subsequent text-only computation. DecoMoE contains two complementary components. The **Sample-Adaptive Visual Boundary (SAVB)** evaluates a lightweight gate from the hidden states entering sparse candidate layers. If the gate triggers at layer l, SAVB removes the complete visual-token block before self-attention in that layer. The gate is supervised by forced-exit correctness and first-token negative log-likelihood while the multimodal backbone remains frozen. Starting from the MoE block of the same layer, the **Routing-Calibrated Expert Prefix (RCEP)** follows a layer-wise Mean-Mass order calibrated offline from text-token routed mass, physically reorders the corresponding router outputs and expert parameters, and dynamically retains the shortest prefix covering a target fraction of the current activated routed mass. Native token-wise Top-k routing and expert-MLP execution are then restricted to the prefix selected independently at layer l and each subsequent layer.

We evaluate DecoMoE on Qwen3-VL-MoE and InternVL3.5-30B-A3B across six benchmarks. DecoMoE retains 97.91% and 97.87% of baseline performance while reducing computation from 27.06 to 16.73 and 9.86 to 5.97 TFLOPs, respectively, with measured speedups of 1.69× and 1.41×. Additional analyses examine both decoupling mechanisms: SAVB identifies sample-dependent visual boundaries after which PPL remains comparable to full visual propagation, while the Expert-Importance Decoupling Analysis shows that RCEP better identifies experts important for text-token computation, supporting its use for offline expert reordering. Our contributions are summarized as follows:

- We identify two complementary regularities in multimodal MoE inference: an input-dependent visual-propagation boundary and a more concentrated, cross-case recurrent expert structure for text-token routing. These observations motivate separate treatment of visual propagation and expert importance.

- We introduce DecoMoE, a two-dimensional structured compression framework that decouples visual propagation from expert computation, using a sample-adaptive boundary for contiguous visual-token removal and text-routing calibration to organize experts by importance into a contiguous prefix for structured expert compression.
- We evaluate DecoMoE on Qwen3-VL-MoE and InternVL3.5-30B-A3B across six benchmarks. On Qwen3-VL-MoE, DecoMoE achieves a 1.69× measured speedup with a 2.09% average performance reduction relative to the dense baseline.

2 Related Work

2.1 Multimodal Mixture-of-Experts Models

MLLMs align visual encoders with causal language models through projectors and visual instruction tuning Liu et al. (2023); Bai et al. (2023); Wang et al. (2024). In parallel, sparse MoE models such as Mixtral and DeepSeekMoE expand model capacity through token-wise conditional routing Jiang et al. (2024); Dai et al. (2024). This architecture has been extended to multimodal reasoning by MoE-LLaVA, DeepSeek-VL2, Kimi-VL, Qwen3-VL, and InternVL3.5 Lin et al. (2026); Wu et al. (2024); Team et al. (2025); Bai et al. (2025); Wang et al. (2025). Compared with dense MLLMs, their inference cost is jointly determined by sequence length and routed expert computation, while visual and textual tokens may induce different expert loads. Although sparse routing limits the experts activated by each token, long visual sequences still traverse attention, routing, dispatch, and expert MLPs at every retained layer.

2.2 Efficient Multimodal Mixture-of-Experts Models

Most MLLM acceleration methods reduce the visual-token sequence. Front-end approaches compress visual features before language decoding Li et al. (2025); Shang et al. (2024); Yang et al. (2025b), whereas decoder-side methods prune or merge tokens using attention, statistical, visual, optimization, or information-preservation criteria Chen et al. (2024a); Ye et al. (2025); Zhang et al. (2024); Xing et al. (2025); Zhang et al. (2025a); Yang et al. (2025a); Tan et al. (2025). Dynamic-LLaVA, LLaVA-Mini, and MustDrop learn or schedule compression across inference stages Huang et al. (2025a); Zhang et al. (2025b); Liu et al. (2024). Closest to whole-block removal, VTW withdraws visual tokens at a calibrated layer, while DyVTE predicts an input-dependent exit depth Lin et al. (2025); Wu et al. (2026). These studies show that visual redundancy varies across processing stages, but their compression decisions are independent of MoE routing behavior. They therefore primarily target dense backbones or leave original expert pool unchanged after token reduction.

A complementary line compresses MoE experts through pruning, decomposition, quantization, or contribution estimation Lu et al. (2024); Yang et al. (2024); Xie et al. (2024); Huang et al. (2024; 2026). More recent methods directly address multimodal MoE inference: FastMMoE combines visual-token pruning with reduced expert activation, MoDES performs layer- and modality-aware expert skipping, and MACS mitigates modality-induced load imbalance under expert parallelism Xia et al. (2025); Huang et al. (2025b); Li et al. (2026). Unlike permanent expert compression, these approaches adapt the executed computation or capacity while retaining the pretrained model structure. Routing analyses further reveal distinct visual and textual expert behaviors Xu et al. (2026). DecoMoE differs by using an input-dependent visual-propagation boundary to determine when computation becomes text-dominated, and cross-case text-routing regularity to determine which physically contiguous expert prefix is retained at each subsequent MoE layer. The two decisions are combined during inference but treated separately in their formulation and calibration.

3 Methodology

DecoMoE is a routing-aware, two-dimensional structured compression framework for multimodal MoE inference. It decouples the boundary of visual propagation from the text-aligned expert structure used thereafter. As shown in Figure 2, SAVB determines when the sequence transitions to text-only computation, while RCEP selects a contiguous prefix of reordered experts at each MoE layer thereafter. The visual boundary and layer-wise prefix lengths remain input-dependent, while their execution is structurally regular along the token and expert dimensions.

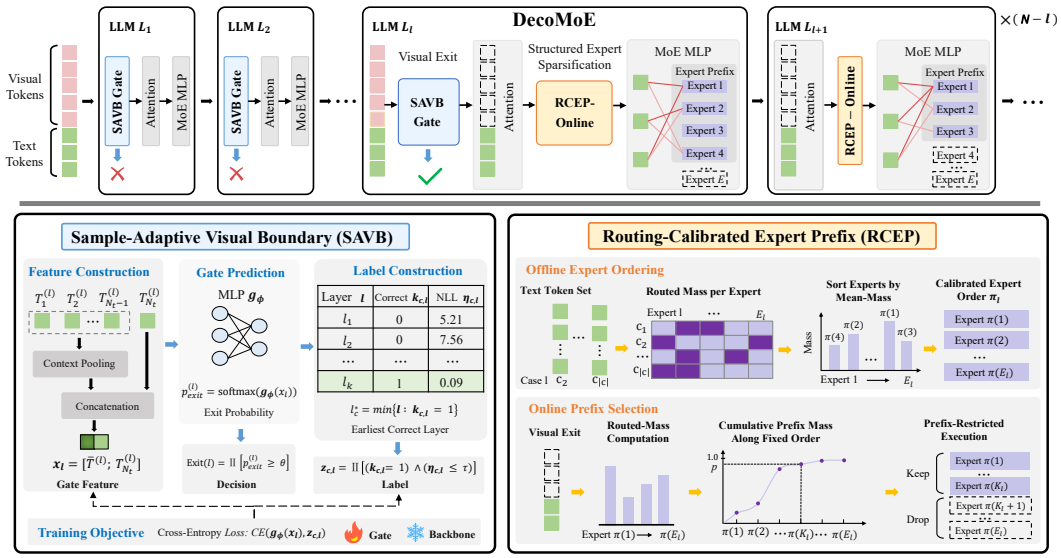


Figure 2: Overview of DecoMoE. **Top:** At each candidate-layer entrance, SAVB determines whether to remove the complete visual-token block before self-attention. Once triggered at layer l, RCEP is activated in the MoE block of the same layer and independently restricts the triggering layer and each subsequent MoE layer to a mass-sufficient contiguous prefix of offline-reordered experts. **Bottom left:** SAVB constructs its gate feature from the pooled preceding text states and the final text state, and is trained with labels derived from forced-exit correctness and first-token NLL while the backbone remains frozen. **Bottom right:** Offline, RCEP derives a layer-wise Mean-Mass expert order from cross-case text-token routing. Online, it aggregates the activated routed mass over tokens in each MoE invocation and retains the shortest prefix covering the target mass fraction p.

3.1 Principle

Consider a multimodal MoE decoder with N layers, indexed by $l \in \{1, \dots, N\}$. Let the multimodal input sequence be $X = [X_v; X_t]$, where X_v is a contiguous visual-token block and X_t is the text-token sequence. At the entrance to decoder layer l, the hidden sequence is $H_l = [V^{(l)}; T^{(l)}]$. Let $\mathcal{E}_l$ denote its expert pool and $E_l = |\mathcal{E}_l|$ the number of experts. For token position i, the MoE router produces logits $\mathbf{a}_{l,i}$ and assigns probability

$$\rho_{l,i,e} = \frac{\exp(a_{l,i,e})}{\sum_{e' \in \mathcal{E}_l} \exp(a_{l,i,e'})}, \qquad e \in \mathcal{E}_l. \tag{1}$$

These quantities characterize two coupled dimensions of MoE computation: visual-token propagation determines how long $V^{(l)}$ remains in the sequence, while $\boldsymbol{\rho}_{l,i}$ determines how computation is allocated across experts.

Figure 3 motivates decoupling the two dimensions. Figure 3(a) shows that the earliest prefill layer preserving answer correctness after visual-token removal varies across cases, while perplexity (PPL) often decreases after this first-good layer and remains below that of earlier exits. Figures 3(b) and 3(c) show that text-token routing exhibits higher routed-mass concentration and larger cross-case proportions of repeated experts than visual-token routing, suggesting more recurrent expert-importance patterns for text tokens. These observations motivate visual-propagation decoupling, where SAVB estimates an input-dependent visual-exit boundary rather than using a fixed propagation depth, and text-aligned expert-importance decoupling, where RCEP estimates expert importance from cross-case text-token routing statistics, constructs a layer-wise expert order, and selects an input-dependent expert prefix during online inference.

3.2 Sample-Adaptive Visual-Propagation Decoupling

The Sample-Adaptive Visual Boundary (SAVB) module predicts an input-dependent visual-exit layer under a calibrated confidence criterion. To limit overhead, SAVB is evaluated only at a sparse candidate set $\mathcal{G} \subseteq \{1, \dots, N\}$, with candidates placed at intervals of four decoder layers.

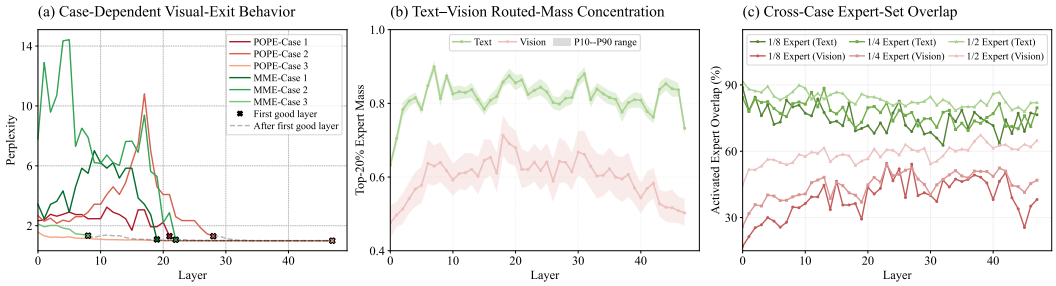


Figure 3: Behavioral diagnostics for decoupled compression. (a) Forced visual-exit perplexity trajectories for different cases. Each curve reports the perplexity of the reference answer when visual tokens are removed after different prefill layers, and each marker indicates the earliest prefill layer at which forced visual exit still produces the correct answer. (b) Dataset-level fractions of routed mass covered by the top 20% of experts for visual and text tokens across layers. (c) Cross-case proportions of repeated experts for visual and text tokens across layers under fixed expert-retention ratios of $1/8$, $1/4$, and $1/2$.

At candidate layer $l \in \mathcal{G}$, let $T_j^{(l)} \in \mathbb{R}^d$ denote the incoming hidden state of the j-th text token, where $j \in \{1, \ldots, n\}$. Following the feature form used by DyVTE Wu et al. (2026), SAVB combines the final text state with the pooled preceding context:

$$\bar{T}^{(l)} = \frac{1}{n-1} \sum_{j=1}^{n-1} T_j^{(l)}, \qquad x_l = [\bar{T}^{(l)}; T_n^{(l)}]. \tag{2}$$

Here, $x_l \in \mathbb{R}^{2d}$ concatenates the pooled context with the final text state. A two-class MLP g_ϕ produces the visual-exit probability and decision:

$$p_{\text{exit}}^{(l)} = \text{softmax}(g_\phi(x_l))_1, \qquad \text{Exit}(l) = \mathbb{I}[p_{\text{exit}}^{(l)} \geq \theta]. \tag{3}$$

Here, θ is the online gate threshold. If SAVB triggers at the entrance to layer l, the complete visual-token block is removed before self-attention in that layer. Consequently, both attention and MoE computation in layer l operate on the shortened text-only sequence. RCEP is enabled for the MoE block of layer l and remains active in every subsequent MoE layer $m \in \{l, \ldots, N\}$.

Gate supervision is constructed by first generating a no-exit reference answer and then forcing visual exit at every candidate layer. For calibration case c, let $\kappa_{c,l} \in \{0, 1\}$ indicate whether the forced-exit output is correct under the dataset-specific evaluation rule, and let $\eta_{c,l}$ denote the negative log-likelihood (NLL) of the reference answer's first token. The earliest correct candidate layer is

$$l_c^\star = \min\{l \in \mathcal{G} : \kappa_{c,l} = 1\}. \tag{4}$$

The corresponding NLL values define a percentile-based confidence cutoff:

$$\tau(P_{\text{NLL}}) = \text{Quantile}_{P_{\text{NLL}}} \left(\{\eta_{c,l_c^\star} \mid l_c^\star \text{ exists}\} \right), \tag{5}$$

where $P_{\text{NLL}} \in (0, 1)$ is the selected percentile level. Each case–layer pair is then labeled independently:

$$z_{c,l} = \mathbb{I}[\kappa_{c,l} = 1 \ \wedge \ \eta_{c,l} \leq \tau(P_{\text{NLL}})], \qquad l \in \mathcal{G}. \tag{6}$$

The first-good layer is used only to calibrate the cutoff and does not impose a monotonic labeling rule on later layers. SAVB is trained with cross-entropy on $(x_{c,l}, z_{c,l})$ while the backbone multimodal MoE remains frozen.

3.3 Text-Aligned Expert-Importance Decoupling with Routing-Calibrated Prefixes

The Routing-Calibrated Expert Prefix (RCEP) module operates in the MoE block of the layer where SAVB triggers and in all subsequent MoE layers. It derives a layer-wise expert order from text-token routing statistics and materializes the calibrated order as a contiguous expert prefix for online inference. The order is fixed after offline calibration, whereas the retained prefix length is selected dynamically for every layer invocation from the visual-exit layer onward.

Using the router probabilities defined in Eq. equation 1, let

$$S_{c,l,i} = \mathrm{TopK}(\boldsymbol{\rho}_{c,l,i}, k_{\mathrm{moe}}) \tag{7}$$

denote the k_{moe} experts selected for text token i of calibration case c at layer l. For the text-token set $\mathcal{T}_{c,l}$, the routed mass assigned to expert e is

$$M_{c,l,e}(\mathcal{T}_{c,l}) = \frac{1}{|\mathcal{T}_{c,l}|} \sum_{i \in \mathcal{T}_{c,l}} \frac{\mathbb{I}[e \in S_{c,l,i}] \rho_{c,l,i,e}}{\sum_{e' \in S_{c,l,i}} \rho_{c,l,i,e'}}. \tag{8}$$

Let $\mathcal{C}$ denote the calibration cases used for offline expert ordering. Figure 3(c) shows substantial overlap among text-routed experts, suggesting a recurrent expert-importance structure across cases. We therefore calibrate a shared layer-wise expert order from mean routed mass:

$$\pi_l = \mathrm{argsort}_{e \in \mathcal{E}_l} \left(\frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} M_{c,l,e}(\mathcal{T}_{c,l}); \mathrm{desc} \right). \tag{9}$$

After calibration, the permutation π_l is applied consistently to the router output dimensions and corresponding expert parameters. The reordered parameters are loaded before inference, placing experts with larger mean text-token routed mass at earlier physical positions so that each leading segment forms a contiguous prefix. This changes only the parameter order, not the learned values.

During inference, let $m \in \{l, \ldots, N\}$ denote the current MoE layer after SAVB triggers at layer l. RCEP determines its prefix length from the aggregate activated mass of tokens in the current invocation. Let $q_{m,i,e}$ denote the softmax probability over all E_m reordered experts, and define

$$R_{m,i}(x) = \mathrm{TopK}(\mathbf{q}_{m,i}, k_{\mathrm{moe}}). \tag{10}$$

The activated mass assigned to expert e is

$$s_{m,e}(x) = \sum_{i \in \mathcal{I}_m(x)} \frac{\mathbb{I}[e \in R_{m,i}(x)] q_{m,i,e}}{\sum_{e' \in R_{m,i}(x)} q_{m,i,e'}}, \tag{11}$$

where $\mathcal{I}_m(x)$ denotes the token positions processed in this invocation. Since experts are reordered offline, no input-specific reordering occurs online.

Given a target routed-mass fraction $p \in (0, 1]$, the prefix length at layer m is

$$\begin{aligned} \widehat{K}_m(x; p) &= \min \left\{ K \in \{1, \ldots, E_m\} \;\middle|\; \frac{\sum_{e=1}^{K} s_{m,e}(x)}{\sum_{e=1}^{E_m} s_{m,e}(x)} \geq p \right\}, \\ K_m(x; p) &= \max\{k_{\mathrm{moe}}, \widehat{K}_m(x; p)\}. \end{aligned} \tag{12}$$

Thus, p controls the aggregate activated mass covered by the prefix, and K_m may vary across inputs, layers, and invocations. Once determined, RCEP restricts the reordered router logits to the first K_m experts, recomputes softmax and Top-k_{moe} selection within the prefix, and renormalizes the selected weights. Since $K_m \geq k_{\mathrm{moe}}$, each token remains dispatched to k_{moe} experts, with experts outside the prefix excluded from routing and MLP evaluation. Router logits are computed over all E_m reordered experts to estimate K_m; final routing uses only the first K_m entries.

4 Experiments

4.1 Experimental Setup

We evaluate DecoMoE on Qwen3-VL-30B-A3B-Instruct (Qwen3-VL-MoE) and InternVL3.5-30B-A3B across MMBench, ScienceQA, AI2D, AOKVQA, MMMU, and HallusionBench, using their accuracy-like metrics. We compare with the dense baseline, FastV, SparseVLM, DyVTE, and Fast-MMoE under the same generation and scoring protocol within each backbone. Unless stated otherwise, SAVB evaluates hidden states every four decoder layers and is supervised on a random 50% subset of POPE and MME, while RCEP uses $p = 0.70$ and calibrates its layer-wise expert order on a

Method	MMBench	ScienceQA	AI2D	AOKVQA	MMMU	HallusionBench	Avg. Ret.	Avg. TFLOPs	Latency (s)
Qwen3-VL-30B-A3B									
Baseline	88.98	90.90	87.21	87.60	58.26	74.97	100.00%	27.06	0.44 (1.00×)
FastV	87.74	87.31	82.93	86.11	**57.21**	71.67	96.32%	16.65	0.36 (0.82×)
SparseVLM	87.59	88.51	83.78	86.29	56.35	69.63	96.02%	16.21	0.37 (0.84×)
DyVTE	60.97	87.97	82.74	87.07	52.53	**74.21**	91.63%	18.29	0.37 (0.84×)
FastMMoE	82.18	85.33	74.61	80.17	54.35	67.09	90.35%	18.88	1.56 (3.56×)
DecoMoE	**87.75**	**90.80**	**85.62**	**87.51**	56.16	71.92	**97.91%**	16.73	0.26 (0.59×)
InternVL3.5-30B-A3B									
Baseline	88.85	94.08	86.79	87.95	59.60	69.70	100.00%	9.86	1.50 (1.00×)
FastV	87.55	**93.28**	84.53	86.70	**58.02**	**66.98**	97.32%	5.86	1.77 (1.18×)
SparseVLM	87.70	93.19	84.83	87.31	57.24	65.04	97.44%	5.03	1.85 (1.23×)
FastMMoE	85.67	92.27	78.28	84.86	57.23	63.42	95.05%	8.02	1.46 (0.97×)
DecoMoE	**88.19**	**93.28**	**86.59**	**87.94**	57.78	65.51	**97.87%**	5.97	1.06 (0.71×)

Table 1: Accuracy–efficiency comparison on Qwen3-VL-MoE and InternVL3.5-30B-A3B. Avg. Ret. is the mean performance retention across six benchmarks; Avg. TFLOPs is the theoretical computation per profiled case; Latency is mean profiler CUDA time. Parentheses report latency relative to the corresponding baseline.

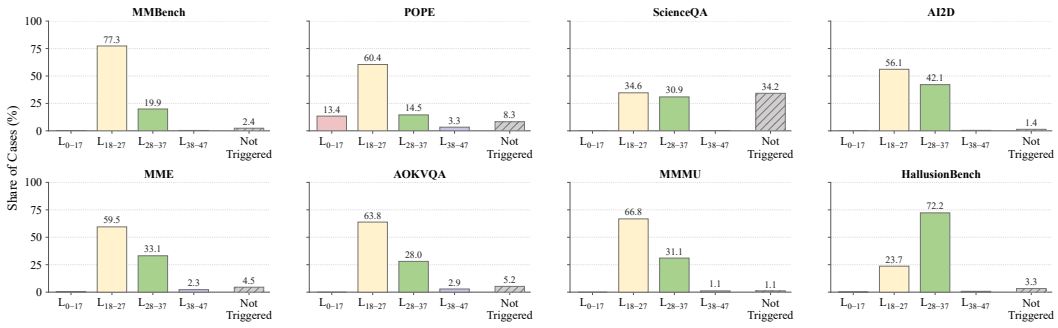


Figure 4: Dataset-wise SAVB exit distributions on Qwen3-VL-MoE. Triggered cases are grouped by zero-based layer intervals; Cases labeled Not Triggered retain visual tokens through all layers.

random 10% subset of POPE and MME. POPE and MME are therefore excluded from the main evaluation to avoid overlap between calibration data and reported evaluation benchmarks. The backbone remains frozen throughout, with construction details provided in the appendix. All experiments use NVIDIA H200 GPUs. Latency is measured as CUDA time with `torch.profiler`, while FLOPs are theoretical estimates detailed in the appendix. Avg. Ret. is the unweighted mean of per-dataset retention relative to the corresponding dense baseline across the six evaluation benchmarks.

4.2 Main Results

On Qwen3-VL-MoE, DecoMoE retains 97.91% of baseline while reducing computation from 27.06 to 16.73 TFLOPs and latency from 0.44 to 0.26 seconds, corresponding to a 1.69× speedup. At similar FLOPs, its retention is 1.59 and 1.89 percentage points higher than FastV and SparseVLM, respectively. This trend also holds on InternVL3.5-30B-A3B, where DecoMoE retains 97.87%, reduces computation from 9.86 to 5.97 TFLOPs, and lowers latency from 1.50 to 1.06 seconds (1.41×). FastV and SparseVLM attain lower TFLOPs but exceed the dense latency, indicating that lower FLOPs do not necessarily translate into lower measured latency on this backbone.

4.3 Component-wise Ablation Study

Table 2 isolates visual-boundary and expert-prefix designs on POPE, HallusionBench, and AI2D. Replacing SAVB with a fixed exit at L_{28} yields 96.18% retention, whereas SAVB reaches 97.41% while reducing computation from 27.04 to 17.85 TFLOPs. This indicates that visual-propagation depth is not well represented by a global exit layer. With SAVB fixed, the dynamic RCEP prefix

Method	POPE	HallusionBench	AI2D	Avg. Ret.	Avg. Exit Layer	Avg. Prefix Size	Avg. TFLOPs	Latency (s)
Baseline	89.82	74.97	87.21	100.00%	–	128	27.04	0.37 (1.00×)
RCEP only	88.52	69.51	84.84	96.18%	28.00	34.86	16.69	0.28 (0.76×)
SAVB + Fixed Prefix	89.08	71.40	85.49	97.48%	28.07	50	17.17	0.29 (0.79×)
SAVB only	89.06	70.95	85.85	97.41%	28.08	128	17.85	0.30 (0.81×)
Full DecoMoE	89.27	71.92	85.62	97.83%	28.07	35.01	17.07	0.26 (0.69×)

Table 2: Component ablation on Qwen3-VL-MoE over POPE, HallusionBench, and AI2D. RCEP only uses a fixed visual exit at layer L_{28}; SAVB + Fixed Prefix retains 50 experts after exit; SAVB only keeps all 128 experts.

(a) Expert-order comparison

Order Mode	Keep Ratio 10%	20%	30%	40%	50%
Native	87.3%	90.9%	97.0%	96.7%	97.4%
Random	85.4%	89.8%	94.9%	95.5%	95.9%
RCEP	**96.9%**	**97.4%**	**97.7%**	**97.9%**	**97.5%**

(b) Routed-mass target sensitivity

Metric	Mass Target p 0.1	0.3	0.5	0.7	0.9	1.0
Avg. Ret.	96.7%	96.7%	97.2%	**97.8%**	97.6%	97.3%
Avg. Prefix	8.0	8.7	17.0	35.0	67.0	128.0

Table 3: RCEP analysis on Qwen3-VL-MoE over POPE, HallusionBench, and AI2D. (a) Performance retention under different expert orders and fixed keep ratios. (b) Sensitivity to the routed-mass target p, where Avg. Prefix denotes the mean retained expert-prefix length.

averages 35.01 experts and reaches 97.83% retention, compared with 97.48% for a fixed 50-expert prefix. Full DecoMoE also achieves the lowest latency (0.69× baseline), showing that sample-adaptive visual exit and mass-adaptive expert selection contribute complementary reductions.

4.4 Visual-Propagation Decoupling Analysis

4.4.1 Dataset-wise Visual-Exit Behavior

Figure 4 reports DecoMoE's visual-exit results. On MMBench, 77.3% of cases exit in layers 18–27; on HallusionBench, 72.2% exit in layers 28–37; and POPE exhibits a broader distribution across multiple intervals. These dataset-dependent distributions indicate that SAVB adapts its exit decisions instead of relying on a fixed global boundary. In addition, 34.2% of ScienceQA cases do not trigger, indicating that visual tokens are retained when the gate criterion is not satisfied.

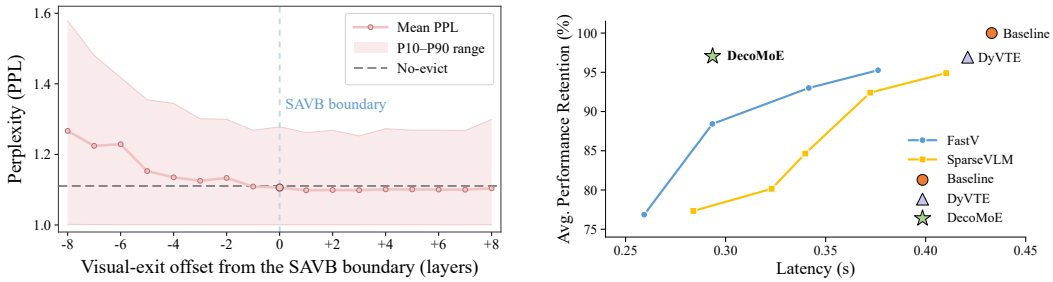
4.4.2 Visual-Propagation Decoupling Validation

Figure 5(a) examines whether SAVB identifies when visual propagation can be terminated. Exiting before the predicted boundary increases perplexity, reaching 1.2666 at offset -8. At the predicted boundary, perplexity is 1.1060, close to the no-exit result of 1.1105, while later offsets remain within 1.0983–1.1038. These results support the view that the SAVB boundary marks a point after which visual-token removal yields PPL comparable to full visual propagation.

4.5 Expert-Importance Decoupling Analysis

4.5.1 Mass–Performance Validation of Expert Reordering

We evaluate whether routed mass provides a criterion for offline expert reordering by comparing the native order, a random permutation, and the RCEP order under fixed expert keep ratios. RCEP achieves the highest retention at every tested budget: at a 10% keep ratio, it retains 96.9% of baseline, compared with 87.3% for Native and 85.4% for Random. Averaged across the five budgets, RCEP reaches 97.5%, versus 93.9% and 92.3%, respectively. These results indicate that Mean-Mass better distinguishes experts that are important for text-token computation and organizes them into a compact prefix, supporting the offline reordering used for expert-importance decoupling.



(a) Visual-propagation boundary validation **(b) Latency–performance trade-off**

Figure 5: Analysis of DecoMoE's visual boundary and realized acceleration. (a) Validation of the visual-propagation boundary predicted by SAVB, where visual exit is forced at offsets ranging from 8 layers before to 8 layers after the predicted boundary. (b) Latency–performance trade-off among representative efficient inference methods on Qwen3-VL-MoE over AI2D and HallusionBench; higher retention and lower latency lie toward the upper-left region.

Method	Baseline	FastV	SparseVLM	DyVTE	FastMMoE	DecoMoE
TTFT (ms)	370.80	320.77	339.13	358.93	1515.57	**256.53**
TPOT (ms/token)	135.80	117.48	124.20	131.45	555.05	**93.95**

Table 4: TTFT and TPOT comparison on Qwen3-VL-MoE. Lower is better.

4.5.2 Top-p Sensitivity and Dynamic Prefix Selection

Table 3(a) shows that retention changes little once the keep ratio reaches 30%, indicating that a fixed large expert budget may be unnecessary. Meanwhile, Figures 3(b) and 3(c) show that routed-mass concentration and expert overlap vary across layers, making a fixed prefix size potentially suboptimal across layer invocations. RCEP therefore retains the shortest prefix covering target mass p rather than a fixed number of experts. Table 3(b) shows that $p = 0.70$ achieves the highest retention of 97.8% with an average prefix of 35.0 experts.

4.6 Structured Execution for Realized Acceleration

Figure 5(b) reports a focused latency sweep on AI2D and HallusionBench, where DecoMoE achieves 97.05% retention at 0.29 seconds. Compared with the similar-FLOPs results in Table 1, competing methods exhibit lower performance retention at latency settings closer to that of DecoMoE. Table 4 further shows that DecoMoE reduces TTFT from 370.80 to 256.53 ms and TPOT from 135.80 to 93.95 ms/token relative to the dense baseline, corresponding to reductions of 30.8% in both metrics, while also achieving lower TTFT and TPOT than the compared efficient inference methods. These gains are consistent with DecoMoE's structured execution: SAVB removes a contiguous visual-token block, while RCEP restricts expert dispatch and MLP computation to a contiguous expert prefix. Such structured reductions reduce irregular selection and fragmented expert dispatch, helping theoretical computation savings translate into lower realized inference latency.

5 Conclusion

We present DecoMoE, a two-dimensional structured compression framework for multimodal MoE inference. Its central principle is to separate two decisions that are coupled in standard execution but exhibit different regularities. Visual-propagation decoupling identifies an input-dependent boundary after which visual-token removal yields PPL comparable to full visual propagation. Expert-importance decoupling separates expert importance for subsequent text-token computation from the original physical expert order, representing recurrent routing structure as a compact contiguous prefix. Together, the two mechanisms provide structured reductions along the token and expert dimensions. On Qwen3-VL-MoE, DecoMoE achieves a 1.69× measured speedup with a 2.09% average performance reduction relative to the dense baseline.

AI USE STATEMENT

Generative AI tools were used primarily to assist with manuscript writing and language refinement. All AI-assisted text was reviewed and revised by the authors to ensure technical accuracy and consistency with the underlying methods and experimental results. The authors take full responsibility for the final content of this work, including text produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on improving the inference efficiency of existing multimodal MoE models and does not involve human-subject studies or the collection of new user data. We use publicly available models and established benchmarks and do not identify additional ethical concerns beyond those associated with the underlying models and datasets.

REPRODUCIBILITY STATEMENT

We provide the model configurations, evaluation benchmarks, calibration protocol, and implementation settings in the main paper. The appendix further provides the exact DecoMoE algorithm, analytical FLOPs accounting, SAVB supervision and RCEP calibration procedures, data-isolation protocol, complete sensitivity results, and reference implementation. Together, these materials specify the methodological and experimental details needed to reproduce the reported results.

REFERENCES

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. URL https://arxiv.org/abs/2308.12966.

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

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*, pp. 19–35. Springer, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024b.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1280–1297, 2024.

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. *arXiv preprint arXiv:2410.06270*, 2024.

Weizhong Huang, Yuxin Zhang, Xiawu Zheng, Fei Chao, Rongrong Ji, and Liujuan Cao. Discovering important experts for mixture-of-experts models pruning through a theoretical perspective. *Advances in Neural Information Processing Systems*, 38:135973–136003, 2026.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, and Shaohui Lin. Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification. In *International Conference on Learning Representations*, volume 2025, pp. 69927–69955, 2025a.

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. *arXiv preprint arXiv:2511.15690*, 2025b.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Bo Li, Chuan Wu, and Shaolin Zhu. Macs: Modality-aware capacity scaling for efficient multimodal moe inference. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22134–22148, 2026.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, 133(10):6794–6812, 2025.

Bin Lin, Zhenyu Tang, Yang Ye, Jinfa Huang, Junwu Zhang, Yatian Pang, Peng Jin, Munan Ning, Jiebo Luo, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *IEEE Transactions on Multimedia*, 2026.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5334–5342, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. Multi-stage vision token dropping: Towards efficient multimodal large language model. *arXiv preprint arXiv:2411.10803*, 2024.

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 (Volume 1: Long Papers)*, pp. 6159–6172, 2024.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024.

Xudong Tan, Peng Ye, Chongjun Tu, Jianjian Cao, Yaoxin Yang, Lin Zhang, Dongzhan Zhou, and Tao Chen. Tokencarve: Information-preserving visual token compression in multimodal large language models. *arXiv preprint arXiv:2503.10501*, 2025.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Qiong Wu, Wenhao Lin, Yiyi Zhou, Weihao Ye, Zhanpeng Zeng, Xiaoshuai Sun, and Rongrong Ji. Accelerating multimodal large language models via dynamic visual-token exit and the empirical findings. *Advances in Neural Information Processing Systems*, 38:168378–168403, 2026.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024.

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. *arXiv preprint arXiv:2511.17885*, 2025.

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. *arXiv preprint arXiv:2410.12013*, 2024.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14593–14603, 2025.

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. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31164–31178, 2026.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Yuanlin Duan, Wenqi Jia, Miao Yin, Yu Cheng, and Bo Yuan. Moe-i2: 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*, pp. 10456–10466, 2024.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, et al. Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19803–19813, 2025a.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025b.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22128–22136, 2025.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20857–20867, 2025a.

Shaolei Zhang, Qingkai Fang, Yang Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. In *International Conference on Learning Representations*, volume 2025, pp. 53285–53310, 2025b.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

A Appendix

This appendix provides implementation-level details and additional evidence for the two decoupled components introduced in the main paper: the Sample-Adaptive Visual Boundary (SAVB) and the Routing-Calibrated Expert Prefix (RCEP). We first profile the inference bottleneck, then specify the exact offline and online algorithms, reproduce the analytical FLOPs accounting used in our experiments, document the calibration protocol, and finally present additional routing diagnostics and case studies.

Algorithm 1 Offline SAVB and RCEP calibration

Require: Frozen multimodal MoE f, SAVB calibration cases $\mathcal{C}_S$, RCEP calibration cases $\mathcal{C}_R$, candidate layers $\mathcal{G}$, NLL percentile P_{NLL}
Ensure: Gate parameters ϕ and layer-wise expert permutations $\{\pi_l\}_{l=1}^{L}$
for each case $c \in \mathcal{C}_S$ **do**
  Generate the no-exit reference answer.
  for each $l \in \mathcal{G}$ **do**
    Force visual exit after layer l without expert slicing.
    Record correctness $\kappa_{c,l}$, first-token NLL $\eta_{c,l}$, and feature $x_{c,l} = [\bar{T}^{(l)}; T_n^{(l)}]$.
  end for
  Set $l_c^\star = \min\{l \in \mathcal{G} : \kappa_{c,l} = 1\}$ when it exists.
end for
Compute $\tau(P_{\mathrm{NLL}})$ from $\{\eta_{c,l_c^\star}\}$ and set $z_{c,l} = \mathbb{I}[\kappa_{c,l} = 1 \wedge \eta_{c,l} \le \tau(P_{\mathrm{NLL}})]$.
Train the two-class MLP g_ϕ on $(x_{c,l}, z_{c,l})$.
for each case $c \in \mathcal{C}_R$ and decoder layer l **do**
  Collect text-token routed mass $M_{c,l,e}$ after the calibrated visual-exit condition.
end for
for each decoder layer l **do**
  Sort experts by descending case-mean routed mass to obtain π_l.
  Apply π_l to router rows, gate/up expert tensors, and down expert tensors using the same permutation.
end for

A.1 Inference Bottleneck Analysis

Figure 1(a) reports a layer-wise end-to-end latency breakdown. Attention is the largest component in most layers, typically contributing approximately 57–68% of the measured time, while the expert MLP contributes another approximately 18–26%. Their combined contribution therefore generally exceeds 80% after excluding startup and boundary effects. This profile motivates optimizing both axes rather than treating visual-token pruning and MoE computation as independent alternatives.

High-resolution images introduce a long visual prefix. For a prefill sequence of length S, self-attention contains an $O(S^2)$ interaction term, so retaining all visual tokens through all decoder layers repeatedly pays the quadratic attention cost. The same visual tokens also enter each MoE block. As shown later in Appendix A.6, their routing mass is spread across a large fraction of the expert pool, which forces the dense batched expert implementation to maintain a large effective expert set. SAVB shortens the sequence after the visual information has been transferred to the text states; RCEP then reduces the physically evaluated expert prefix for the remaining text-only computation.

A.2 Exact DecoMoE Algorithm

A.2.1 Offline Calibration

Algorithm 1 expands the training description in the main paper. The backbone is frozen throughout. SAVB labels are obtained by forced visual exit at the sparse candidate layers $\mathcal{G} = \{3, 7, 11, \ldots, 47\}$ (zero-based implementation indices). RCEP is calibrated separately from text-token router statistics after visual exit. The resulting permutation is applied consistently to the router output rows and the corresponding expert tensors; therefore reordering changes only physical storage, not the function of the unsliced model.

A.2.2 Online Inference and Prefix-Restricted Routing

Algorithm 2 removes an ambiguity that is easy to miss in a high-level description of RCEP. At a text-only MoE layer, DecoMoE first computes router logits over all E_l reordered experts. These full logits are used only to estimate the required prefix length K_l. DecoMoE then slices the logits to the first K_l entries, applies a *new* softmax over this prefix, reruns native Top-k_{moe} inside the prefix, and renormalizes the selected weights. It does not intersect the full-pool Top-k set with the prefix. Because $K_l \ge k_{\mathrm{moe}}$, prefix routing always returns exactly k_{moe} experts per token.

Layer-index convention. The implementation records zero-based layer indices. A trigger recorded as l means that layer l itself is evaluated with the full multimodal sequence; the visual

Algorithm 2 DecoMoE online prefill and recurrent decoding

Require: Reordered model, gate g_ϕ, target routed-mass fraction p, native routing width k_{moe}

Set exited $\leftarrow$ false.
for $l = 1, \ldots, L$ **do**
  if exited $=$ false **then**
    Evaluate layer l on $H_l = [V^{(l)}; T^{(l)}]$ with the full expert pool.
    if $l \in \mathcal{G}$ and $\text{softmax}(g_\phi(x_l))_1 \geq \theta$ **then**
      Remove the complete visual block after layer l.
      Set exited $\leftarrow$ true.
    end if
  else
    Compute full reordered router logits $a_{i,1:E_l}$ and probabilities $q_{i,1:E_l}$.
    Compute full-pool native Top-k_{moe} and aggregate activated mass $s_{l,e}(x)$.
    Find the smallest prefix covering p of the total activated mass and set $K_l = \max(k_{\text{moe}}, \widehat{K}_l)$.
    Recompute $\widetilde{q}_{i,1:K_l} = \text{softmax}(a_{i,1:K_l})$.
    Rerun Top-k_{moe} on $\widetilde{q}_{i,1:K_l}$ and renormalize the selected weights.
    Evaluate and combine only the first K_l physically contiguous expert MLPs.
  end if
end for
Reuse the shortened text-only cache and repeat the same prefix-restricted routing during autoregressive decoding.

block is removed from the hidden state and KV cache before layer $l + 1$. Consequently, a reported trigger at layer 27 corresponds to preserving visual tokens through the 28th decoder block.

A.3 Analytical FLOPs Accounting

We report decoder-side theoretical FLOPs using the same case-level replay code as the main experiments. Let L be the number of decoder layers, E the full expert count, k_{moe} the native per-token routing width, d the hidden size, q the attention output size, k_v the concatenated key/value projection size used by the implementation, and m the intermediate width of one expert. A multiply-add is counted as two FLOPs. The per-token linear projection, router, and retained-expert costs are

$$C_{\text{proj}} = 2(dq + 2dk_v + qd), \tag{13}$$

$$C_{\text{router}} = 2dE, \tag{14}$$

$$C_{\text{expert}}(K) = 6Kdm. \tag{15}$$

The factor six in C_{expert} accounts for the gate, up, and down matrix multiplications. For a prefill sequence of length S, the cost of one decoder layer is

$$\begin{aligned} F^{(l)}_{\text{pre}}(S, K) = {} & SC_{\text{proj}} + 2qS(S+1) \\ & + S(C_{\text{router}} + C_{\text{expert}}(K)) \,. \end{aligned} \tag{16}$$

For one decode forward with KV-cache length C, the corresponding layer cost is

$$F^{(l)}_{\text{dec}}(C, K) = C_{\text{proj}} + C_{\text{router}} + C_{\text{expert}}(K) + 4qC. \tag{17}$$

For Qwen3-VL-MoE-30B-A3B, the replay uses $L = 48$, $E = 128$, $k_{\text{moe}} = 8$, $d = 2048$, $q = 4096$, $k_v = 512$, and $m = 768$. Let S_c, V_c, and $T_c = S_c - V_c$ denote the measured prefill, visual, and text-token counts for case c. Let G_c be its generated-token count. Prefill produces the first generated token, so the number of recurrent decode forwards is $D_c = \max(G_c - 1, 0)$.

If SAVB triggers after layer $l^\star_c$, the case-level DecoMoE cost is replayed as

$$F^c_{\text{pre}} = \sum_{l=1}^{l^\star_c+1} F^{(l)}_{\text{pre}}(S_c, E) + \sum_{l=l^\star_c+2}^{L} F^{(l)}_{\text{pre}}(T_c, K^c_l), \tag{18}$$

$$F^c_{\text{dec}} = \sum_{t=1}^{D_c} \sum_{l=1}^{L} F^{(l)}_{\text{dec}}(T_c + t - 1, K^c_{l,t}). \tag{19}$$

Here K_l^c and $K_{l,t}^c$ are read from the recorded prefill and decode RCEP traces. If a trace entry is unavailable, the replay uses the case-level mean retained expert count. If SAVB never triggers, all layers use (S_c, E).

The SAVB MLP uses an input of width $2d$, hidden width $h = 512$, and a two-class output. If it is evaluated n_c^{gate} times, its explicitly counted overhead is

$$F_{\text{gate}}^c = n_c^{\text{gate}} \left(2(2d)h + 2h \cdot 2\right). \tag{20}$$

The final case total is $F_c = F_{\text{pre}}^c + F_{\text{dec}}^c + F_{\text{gate}}^c$. Reported total saving is computed from sums over the same matched cases:

$$\text{Saving} = 1 - \frac{\sum_c F_c^{\text{method}}}{\sum_c F_c^{\text{baseline}}}. \tag{21}$$

Dense-equivalent expert convention. The evaluated Qwen inference path forms a dense batched BMM over every physically retained expert and applies sparse routing weights afterward. Accordingly, the baseline expert term uses $K = E = 128$, whereas RCEP uses the recorded prefix length K_l. This convention follows executed expert matrix multiplications rather than counting only the eight nonzero routing weights.

Scope and limitations. These equations count the language decoder backbone and the explicit SAVB MLP. They do not include the vision encoder, embeddings, normalization, LM head, sampling, Python control flow, softmax/Top-k/cumulative-sum kernels, memory movement, or kernel-launch overhead. Full router projection over all E experts is included even after RCEP activates. Therefore theoretical FLOPs and measured profiler CUDA time are complementary quantities; neither is inferred from the other.

A.4 Calibration Data and Data Isolation

A.4.1 SAVB Supervision

The default Qwen SAVB checkpoint is trained from direct-answer forced-exit labels collected on POPE and MME. We randomly sample 50% of each benchmark using a fixed seed and evaluate visual-only eviction with all 128 experts kept, so expert compression cannot contaminate the boundary labels. Candidate layers are $3, 7, 11, \ldots, 47$. The default $P95$ setting uses the 95th percentile of the global first-good-layer NLL distribution to construct the independent case–layer labels defined in the main paper. The default model uses an independent two-class MLP at each candidate position, with a 512-dimensional hidden layer and cross-entropy training on frozen features. The reported checkpoint is taken after epoch 2.

A.4.2 RCEP Calibration and Text-Routing Alignment

RCEP uses an independent calibration pass over a random 10% sample from POPE and MME with direct-answer prompts. The current scripts do not enforce this sample to be disjoint from the SAVB sample. During collection, visual tokens are removed after implementation layer 24 and no expert slicing is enabled. Routed mass is computed only from text-token positions. We average each expert's routed mass first within a case and then across cases, which prevents long examples from dominating the layer-wise order. The same permutation is applied to the router output rows and both expert parameter tensors before evaluation.

The fixed layer-24 collection condition is used to expose text-aligned routing, not to constrain online execution. RCEP reorders every calibrated MoE layer, including layers before 24, while online prefix slicing starts only after the sample-specific SAVB trigger. Figure 7 and Figure 8 verify that removing visual tokens does not materially change the remaining text-token routing distribution in deep layers, supporting transfer of the calibrated order to dynamic SAVB exits.

A.4.3 Leakage Boundary and Reporting Protocol

The calibration procedures never update the multimodal backbone and never use examples, labels, or answers from MMBench, ScienceQA, AI2D, AOKVQA, MMMU, or HallusionBench. Results on these six datasets are therefore strict cross-dataset generalization tests. The current experiment

Order Mode	Keep Ratio 10%	15%	20%	25%	30%	40%	50%
Native	87.26%	86.60%	90.85%	95.39%	96.97%	96.67%	97.41%
Random	85.43%	94.77%	89.79%	91.57%	94.93%	95.50%	95.94%
RCEP	**96.87%**	**97.39%**	**97.41%**	**97.55%**	**97.71%**	**97.91%**	**97.47%**

Table 5: Complete mass–performance comparison of Native, Random, and RCEP expert orders on Qwen3-VL-MoE over POPE, HallusionBench, and AI2D. Entries report Avg. Ret. under fixed expert keep ratios.

Metric	Mass Target p 0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Avg. Ret.	96.7%	96.8%	96.7%	96.9%	97.2%	97.6%	**97.8%**	97.2%	97.6%	97.3%
Avg. Prefix	8.0	8.0	8.7	11.7	17.0	24.7	35.0	49.0	67.0	128.0

Table 6: Complete sensitivity sweep for the RCEP routed-mass target p on Qwen3-VL-MoE over POPE, HallusionBench, and AI2D. Avg. Prefix is the mean retained expert-prefix length.

scripts draw SAVB and RCEP calibration subsets from the POPE/MME benchmark pools themselves. Consequently, aggregate POPE/MME numbers that include those calibration IDs should be described as *transductive calibration results*, not as a strict held-out test.

For a fully inductive POPE/MME report, calibration IDs must be persisted before training and excluded from evaluation; only the complementary IDs may be scored. This distinction does not affect the six unseen datasets, but it is important for reproducibility and prevents calibration overlap from being mischaracterized as test-set generalization.

A.5 Complete RCEP Sensitivity Results

The main paper reports a compact subset of the RCEP sensitivity results to reduce space. Here we provide the complete expert-order comparison and routed-mass target sweep used in the original analysis.

A.5.1 Complete Expert-Order Comparison

Table 5 reports all seven evaluated expert keep ratios. RCEP achieves the highest retention at every tested budget. In particular, at a 10% keep ratio it retains 96.87% of baseline performance, compared with 87.26% for the native order and 85.43% for a random order. Across all seven budgets, the corresponding mean retentions are 97.47%, 93.02%, and 92.56% for RCEP, Native, and Random, respectively. These complete results support the same conclusion as the compact table in the main paper: text-routed mass provides a stable offline signal for organizing important experts into a contiguous prefix.

A.5.2 Complete Routed-Mass Target Sensitivity

Table 6 restores the full sweep from $p = 0.1$ to 1.0. Retention remains relatively stable across a broad range of mass targets, while the retained prefix grows monotonically as p increases. The default $p = 0.70$ gives the highest measured retention, 97.8%, with an average prefix of 35.0 experts, providing a favorable operating point between performance preservation and expert-prefix length.

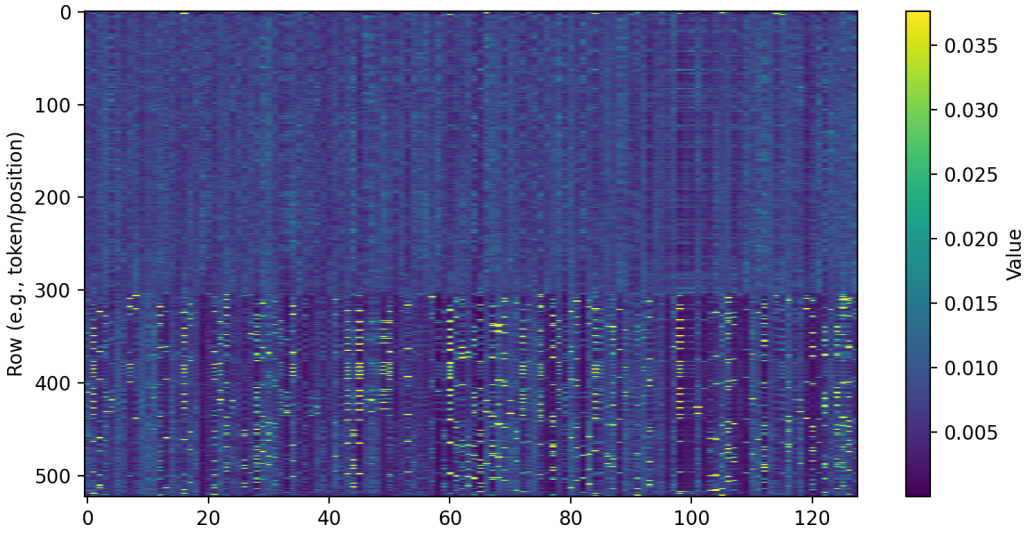


Figure 6: Token-to-expert routing map. Visual-token routing is diffuse and spans a broad expert set, whereas text-token routing forms a substantially more concentrated and recurrent structure.

A.6 Additional Routed-Mass Evidence

Figure 6 visualizes token-level expert activation. Visual tokens distribute routed mass across many experts, forcing a large effective expert set while they remain in the sequence. Text tokens form more concentrated vertical structures, indicating that a shared text-aligned expert order can capture recurring importance across cases.

Figure 7 compares the remaining text-token router maps with and without visual-token eviction. Across cases, the dominant experts and relative mass patterns remain visually aligned. This result is important for the SAVB–RCEP interface: expert importance calibrated in a text-only state remains representative after a sample-specific visual exit.

The mass-count scatter in the top panel of Figure 8 rules out a purely superficial coverage effect. At a routed-mass threshold of 0.5, visual tokens assign substantial probability to almost the entire 128-expert pool, with most points near 120–128 experts. Text tokens generally require only approximately 50–80 experts, with a small number of harder cases extending beyond this range. Thus visual propagation does not merely touch more experts with negligible weights; it preserves broad, nontrivial expert activity.

The bottom panel of Figure 8 quantifies the same phenomenon over deep layers. The router-distribution cosine similarity remains above 0.997, while the mean absolute error remains below 0.0003. Early visual eviction therefore does not redirect the remaining text tokens to a different family of experts; the original high-probability text experts remain stable. Together with the left panel, this observation explains why expert-prefix compression should be activated *after* the learned visual boundary rather than while visual routing is still diffuse.

A.7 Why the Visual Boundary Must Be Sample Adaptive

A.7.1 Case 1: Many Visual Tokens but an Early Boundary

Prompt.

> Task: Answer the following multiple-choice questions about the image, then answer the detailed questions.
>
> 1. What type of scene is shown? (A) Indoor (B) Residential outdoor (C) Commercial (D) Natural.
> 2. What season is depicted? (A) Spring (B) Summer (C) Autumn (D) Winter.
> 3. What recent weather event is evident? (A) Heavy rain (B) Strong wind (C) Snowfall (D)

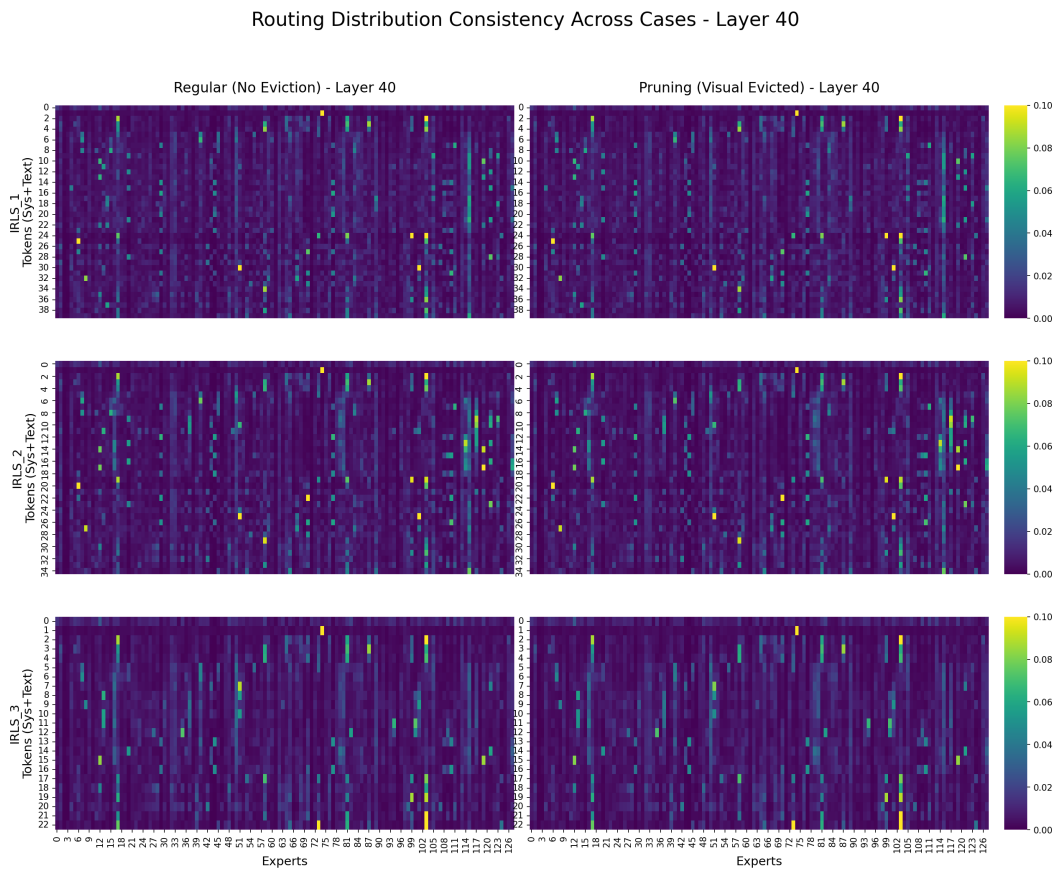


Figure 7: Text-token router distributions at layer 40 for three cases, before and after visual-token eviction. The two conditions retain nearly identical structures.

Fog.
4. Which is clearly visible? (A) Vehicles (B) People (C) Buildings (D) Animals.
5. The tallest trees are located where relative to the buildings? (A) In front (B) Behind (C) On rooftops (D) Inside.
6. Describe the trees and how snow has accumulated on them.
7. Describe the background buildings, including facades, windows, balconies, and roofs.
8. Identify small or subtle infrastructure or activity.
9. Is there a traffic signal or parking-lot sign in the picture?

Answer at $K = 26$.

1. **(B) Residential outdoor**. 2. **(D) Winter**. 3. **(C) Snowfall**. 4. **(C) Buildings**. 5. **(A) In front**. The trees are tall evergreens and deciduous trees whose branches carry a thick layer of fresh snow. The background contains multi-story residential buildings with light-colored facades, repeated rows of windows and balconies, and snow-covered roofs. Smaller details include parked cars, fences, paths, and residential street infrastructure partly occluded by the trees and snow. **No traffic signal or parking-lot sign is visible.**

Despite the large visual-token count, removing the visual block after layer 26 preserves both fine visual details and the model's textual response ability. The case demonstrates that a long image prefix need not be propagated to a late layer once its information has been consolidated into text states.

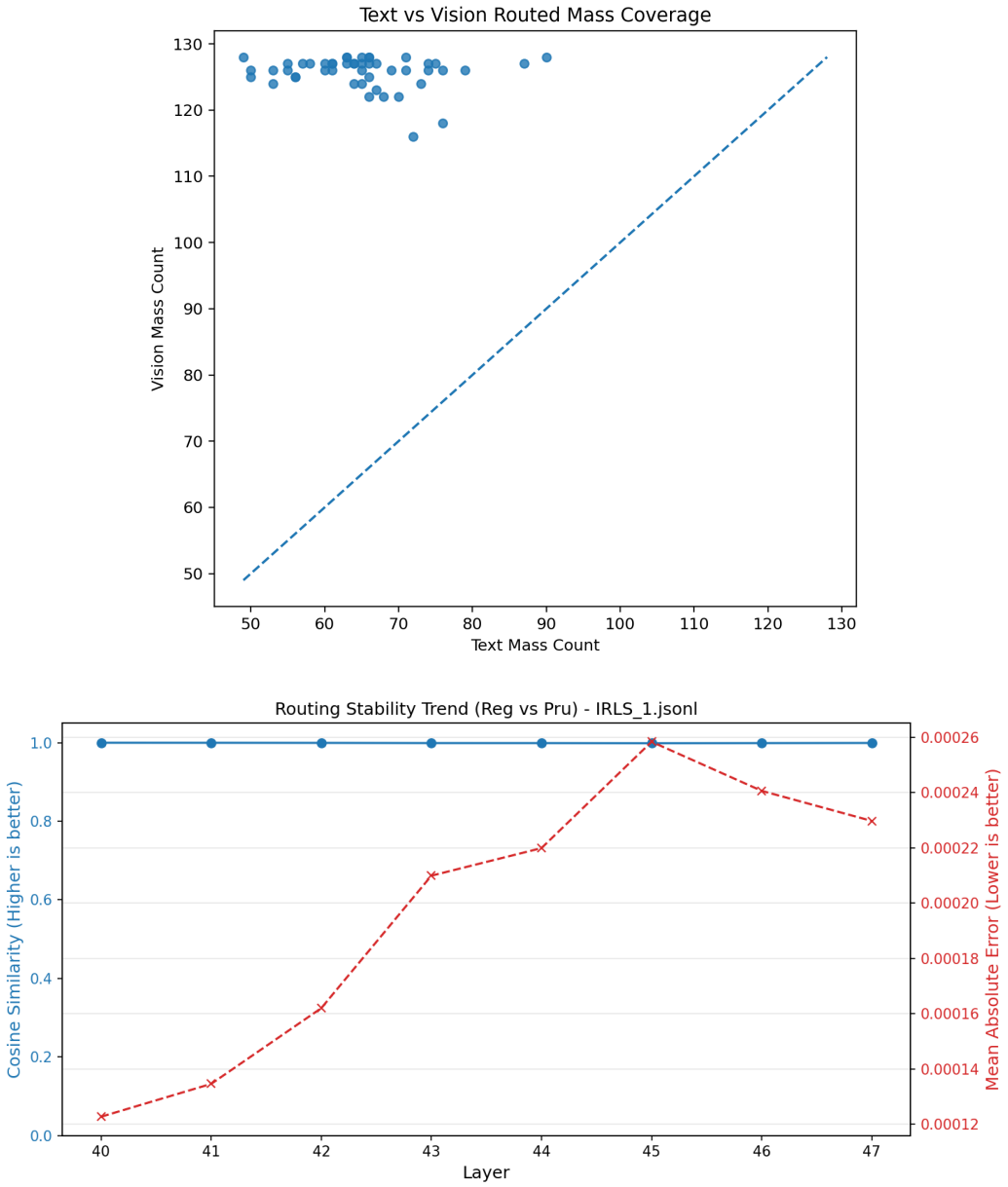


Figure 8: Complementary routed-mass diagnostics. Top: number of experts required to cover 50% routed mass; most text cases require roughly 50–80 experts, whereas visual tokens lie near 120–128 experts. Bottom: deep-layer text-routing stability after visual-token eviction; cosine similarity remains above 0.997 and MAE below 0.0003.

A.7.2 Case 2: Few Visual Tokens but a Late Boundary

Prompt.

> 1. What kind of image is this? 2. From which country is the text? 3. How many words or characters are shown? 4. What is the English translation? 5. What color is the text?

Answer with exit at $K = 42$.



Figure 9: The snowy residential scene used in Case 1. Its high-resolution input produces approximately 4,000 visual tokens.

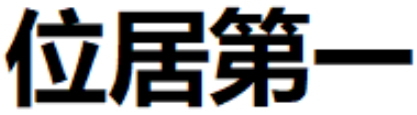


Figure 10: The compact OCR input used in Case 2, containing approximately 80 visual tokens.

> This is a digital image containing a four-character Chinese phrase. The phrase is read as *wei ju di yi* and means **"ranked first"** or **"in first place"**. The text is black on a white background.

Answer with exit at $K = 41$.

> The response misreads and reorders the four characters, gives an incorrect translation, and then enters a long repetitive explanation *(repetitive tail omitted)*.

Unlike Case 1, this image is visually small but semantically precise. Exiting one layer too early causes a discontinuous OCR failure, whereas exiting at layer 42 preserves the correct phrase and translation. Token count is therefore an insufficient boundary heuristic; the required depth depends on when the task-relevant visual semantics become available to the text stream.

A.7.3 Case 3: The Same Image Requires Different Depths by Task

Level 1: Seen ($K = 18$).



Figure 11: The fire-hydrant image shared by the three prompts in Case 3.

Prompt: Is there a blue and yellow fire hydrant in the image? Answer the question using a single word or phrase.
Answer: Yes.

Level 2: Inspect ($K = 22$).

Prompt: Is there a blue and yellow fire hydrant in the image? Why?
Answer: Yes. A fire hydrant is clearly visible beside the road; its body is blue and its top and side caps are yellow.

Level 3: Understand ($K = 42$).

Prompt: Identify the object and its colors, specify where it is located, describe its physical features and surrounding environment, and read its identification number.
Answer: The object is a fire hydrant with a blue body and yellow top and caps. It stands on the sidewalk beside a road. The hydrant has a cylindrical body, a domed top, side outlets, and metal fittings; the surrounding scene is an urban roadside area. Its identification number is **459775**.

All three prompts use identical visual evidence, yet their required cross-modal reasoning depths differ substantially. Object presence is available early, grounded color justification requires additional propagation, and fine-grained attribute extraction requires a much later boundary. This controlled case supports SAVB's central design choice: visual exit must depend on the joint image–prompt state rather than image resolution, visual-token count, or a globally fixed layer.