SCOPE: SELECTIVE CROSS-MODAL ORCHESTRATION OF VISUAL PERCEPTION EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-encoder vision-language models (VLMs) benefit from complementary visual representations, but activating every encoder increases inference cost and visual context length. We propose SCOPE, a Mixture-of-Encoders framework that keeps one shared encoder and selects one auxiliary encoder for each image–prompt pair. A lightweight router uses cross-attention between the text prompt and shared visual features, and is trained with dual entropy regularization and auxiliary losses to encourage confident instance-level decisions and diverse routing across samples. On the document-centric and OCR-heavy benchmarks studied here, SCOPE improves over the reported static single-auxiliary baselines and naive all-encoder concatenation on average. Relative to activating all auxiliary encoders, the tabulated vision-encoder and decoder FLOPs estimates decrease by 24–49%; text-conditioned routing also requires a separate text-encoder pass. These results support instance-adaptive selection under a one-auxiliary-stream deployment constraint, where the selected encoder determines the auxiliary token count.

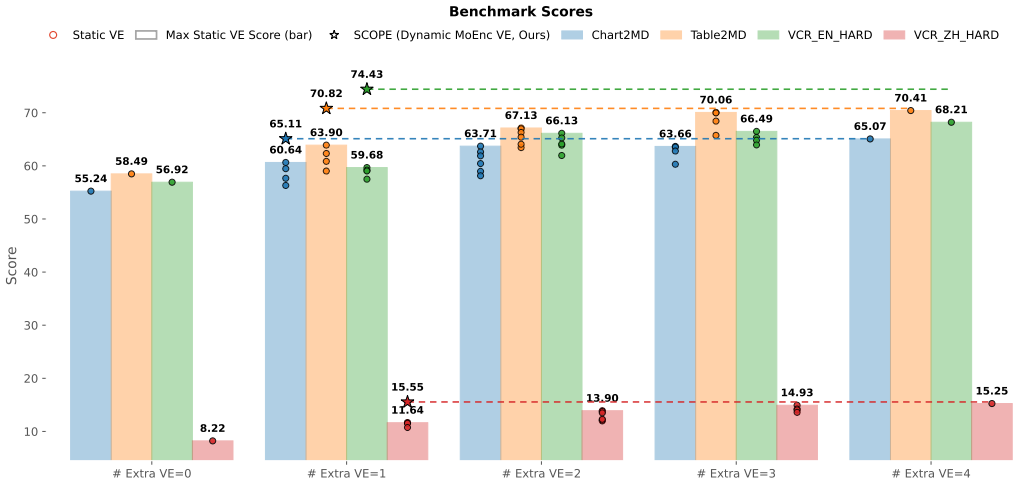

Figure 1: SCOPE (marked by ⋆) compared with static VLMs using zero to four extra vision encoders. SCOPE selects one auxiliary encoder per image–prompt pair and improves average performance over the reported single-auxiliary baseline and naive all-encoder concatenation. The one-auxiliary-stream constraint does not fix token count or FLOPs. See Table 2 for benchmark scores.

1 INTRODUCTION

Recent Vision Language Models (VLMs) have shown strong ability to jointly process visual and textual information (Hurst et al., 2024; Comanici et al., 2025; Anthropic, 2025; Bai et al., 2025; Zhu et al., 2025). One promising direction is to use multiple vision encoders to enrich the visual representations sent to the large language model (LLM) (Liu et al., 2025; Mao et al., 2025). Because

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.

different encoders are pre-trained on different datasets and architectures, they can capture complementary features, and prior work has reported gains across vision-language tasks (Fan et al., 2024; Kar et al., 2024; Liu et al., 2023b; Shi et al., 2025; Tong et al., 2024; Zong et al., 2024b).

However, activating all auxiliary encoders for every input becomes problematic when deployment allows only one auxiliary stream in the decoder. This is especially relevant for document-centric and OCR-heavy settings, where a single image already yields a long visual prefix and different encoders introduce different token costs. Concatenating multiple auxiliary streams increases compute and context length, while gains often saturate beyond one additional encoder (Mao et al., 2025). This motivates the question we study: *under a fixed one-auxiliary-stream budget, which auxiliary encoder should a document-centric VLM use for each image–prompt pair?* In this regime, the main comparison is against static single-auxiliary baselines, because both SCOPE and these baselines use one auxiliary stream at inference time.

To address this setting, we propose a dynamic Mixture-of-Encoders (MoEnc) framework, SCOPE[1]. Inspired by Mixture-of-Experts (MoE) (Jiang et al., 2024; Zhou et al., 2022), our method dynamically selects a *vision encoder* for each input. Unlike standard MoE routing, often token-level, our MoEnc operates at the *instance level*, conditioning the choice on both the image and the text prompt. We use a *shared vision encoder* that is always active and a pool of *routed vision encoders*. For each image–prompt pair, a lightweight router selects one routed encoder, without benchmark or task-family labels at deployment, and combines its representations with those of the shared encoder before passing them to the LLM. We choose instance-level routing because it preserves visual coherence and avoids expert switching across tokens.

A router uses cross-attention over *textual prompt embeddings* and *visual features* from the shared encoder to select the routed encoder for the current instance. This preserves encoder diversity while staying within a fixed one-auxiliary-stream budget. In the document-centric, OCR-heavy workloads studied here, a "1+1" configuration (one shared encoder plus one dynamically selected routed encoder) improves over the reported single-auxiliary baseline on average. This matched single-auxiliary comparison is the paper's main empirical result. As a secondary finding, the routed model can also outperform all-encoder concatenation while using fewer visual tokens.

Even when routing is driven only by visual features, the model remains competitive with the static multi-encoder baseline, suggesting much of the benefit can be captured from image-level cues. See Table 2 and Section 5.

Importantly, the "1+1" budget fixes the number of auxiliary streams, not auxiliary tokens. In our setup, the selected routed encoder contributes 192 to 1036 extra visual tokens for a 1024×768 image (Appendix D). The resulting comparison matches the number of auxiliary streams, while token counts and activated computation remain encoder-dependent.

Training the router requires confident decisions and diversity across samples. To encourage this, we use dual entropy regularization together with dual auxiliary losses. We treat this as a practical training recipe rather than a substitute for standard load-balancing objectives, and revisit this distinction in Section 2.3.

Our contributions are threefold:

- We study a deployment setting where one shared encoder plus one adaptive auxiliary encoder replaces fixed multi-encoder concatenation, and show that instance-adaptive selection improves average performance over the reported static single-auxiliary baseline on document-centric benchmarks.
- We propose a training recipe combining dual entropy regularization with auxiliary terms that stabilizes routing.
- We introduce a text- and vision-conditioned router that learns specialization across encoder types without task-family labels at test time, and show coherent family-level specialization when trained end-to-end.

[1]The name is inspired by how a microscope selects the appropriate lens for a specimen.

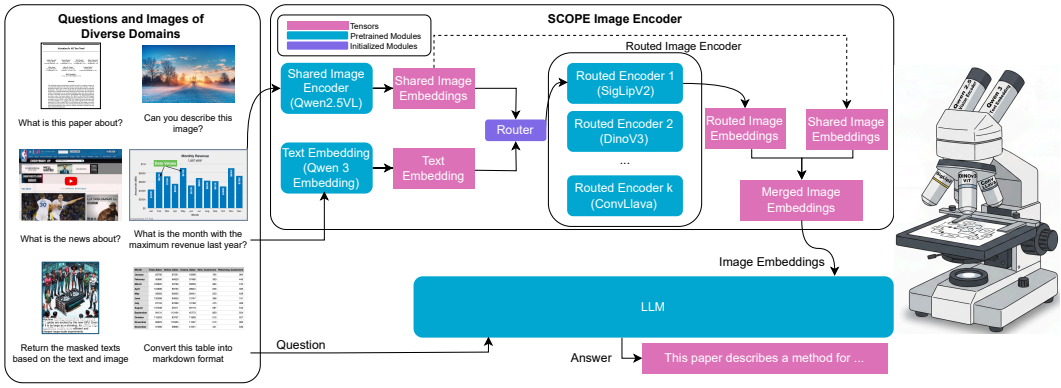

Figure 2: Overview of SCOPE. The shared image encoder and frozen text embedding model produce features for a router, which selects one auxiliary image encoder at inference. The selected features are combined with the shared visual representation and passed to the LLM. Cyan denotes pretrained modules, purple initialized modules, and pink tensors. Dashed paths denote routing; the microscope illustrates selecting an appropriate visual expert.

2 PROPOSED METHOD

Our proposed system introduces a dynamic vision-language processing pipeline that augments a standard VLM with expert selection and feature fusion. The architecture consists of four main stages: initial embedding, router-based expert selection, feature fusion, and final response generation, as shown in Figure 2.

In this section, we present the architecture and training methodology of our dynamic Mixture-of-Encoders framework. Our goal is to create a system that adaptively selects the most suitable vision encoder for a given context, thereby maximizing performance while minimizing computational overhead during inference.

2.1 NOTATION

Let I denote the input image, which becomes I' after preprocessing, and let P denote the input text prompt. A frozen text encoder E_T maps P to a representation $T \in \mathbb{R}^{N_T \times D_T}$, where N_T is the number of tokens and D_T the dimension of text embedding.

The SCOPE architecture employs a shared vision encoder E_S that produces an output representation $V_s \in \mathbb{R}^{N_S \times D_S}$ with N_S tokens and feature dimension D_S. In addition, SCOPE maintains a pool of K routed vision encoders, $\mathcal{E}_r = \{E_{r_1}, E_{r_2}, \ldots, E_{r_K}\}$, from which exactly one encoder is selected at inference time by a router network R. Each routed encoder E_{r_i} produces a representation $V_{r_i} \in \mathbb{R}^{N_r \times D_r}$, which is subsequently passed through a connector network C_i (typically a lightweight linear projection) to obtain an aligned representation $V'_{r_i} \in \mathbb{R}^{N_r \times D_S}$.

2.2 SCOPE ARCHITECTURE

Our framework consists of four main stages: (1) Initial Feature Extraction, (2) Dynamic Encoder Routing, (3) Representation Fusion, and (4) Alignment with the Large Language Model (LLM). The overall architecture is depicted in Figure 2.

Initial Feature Extraction Given an input image I, we first apply a dynamic resizing preprocessing step that preserves the aspect ratio while limiting the total number of pixels to meet the compute budget.

The processed image I' is then fed into the shared vision encoder E_S to obtain the shared visual representation $V_s = E_S(I')$. Simultaneously, the input text prompt P is encoded by the frozen text embedding model E_T to produce the query representation $T = E_T(P)$. The text encoder that we use in our experiments is *Qwen3-Embedding-0.6B*.

Dynamic Encoder Routing At the core of our method is the router module R, which dynamically selects a single encoder from the pool of routed encoders $\mathcal{E}_r$. The router leverages both visual and textual cues by employing a cross-attention mechanism in which the query is derived from the query representation T and the keys and values are derived from the shared vision representation V_s. The resulting cross-attention output is aggregated into a global representation, which is then passed through a linear layer to produce the routing logits $\mathbf{z} \in \mathbb{R}^K$:

$$\mathbf{z} = \text{Linear}(\text{CrossAttn}(Q = T, K = V_s, V = V_s)). \tag{1}$$

We also consider a variant that omits the text embedding and its corresponding query representation. In this case, the router reduces to a lightweight self-attention mechanism over the shared visual features, followed by a linear projection:

$\mathbf{z} = \text{Linear}(\text{SelfAtten}(V_s))$.

Representation Fusion Given the routing logits $\mathbf{z}$, the router performs a top-1 selection by activating only the encoder corresponding to the maximum logit value.

Formally, let the selected index be $k = \arg\max_i(z_i)$. At inference time, the preprocessed image is passed through the selected encoder E_{r_k}, producing features V_{r_k}. Then these are scaled by the corresponding weight z_k and mapped through the connector C_k, resulting in the routed representation: $V'_{r_k} = C_k(z_k V_{r_k})$. However, during training, the non-differentiability of $\arg\max$ prevents gradients from flowing directly. To address this challenge, we adopt straight-through estimator (STE) tricks to allow gradients to propagate through the discrete routing decision. Thus, the connector C_k does not receive V_{r_k} directly, instead

$$V'_{r_k} = C_k(\sum_{i=1}^{K} z_i V_{r_i} - \mathbf{sg}(\sum_{i=1}^{K} z_i V_{r_i} - z_k V_{r_k})) \tag{2}$$

where **sg** denotes the stop-gradient operator. In this formulation, the backward pass derives gradients from the soft combination $\sum_{i=1}^{K} z_i V_{r_i}$, while the forward pass uses the entries of V_{r_k}.

The final visual representation V_{final} is obtained by concatenating the shared encoder output with the confidence-weighted routed representation:

$$V_{\text{final}} = \text{Concat}\big(V_s; V'_{r_k}\big)$$

Alignment with LLM The fused visual representation V_{final} is projected into the word embedding space of the LLM, producing a sequence of visual tokens that are prepended to the text prompt embeddings. This visual prefix conditions the LLM on the image content and allows it to perform downstream vision–language understanding tasks.

2.3 Router training with dual regularization and dual auxiliary losses

In this subsection, we describe a practical router-training scheme for top-1 encoder selection. The objective combines dual entropy regularizers and complementary auxiliary losses to encourage diverse routing patterns across a batch together with confident decisions for each instance. We use these terms as optimization biases for stable hard routing in our setting; they are not claimed to be equivalent to standard marginal-usage load-balancing objectives.

A central difficulty during training is to avoid degenerate routing behavior while still making confident, instance-specific decisions. Let $\mathbf{Z} \in \mathbb{R}^{K \times B}$ denote the matrix of routing logits with entries $z_i^{(j)}$, where $i \in \{1, \ldots, K\}$ indexes routed encoders and $j \in \{1, \ldots, B\}$ indexes samples in a mini-batch.

Batch diversity via batch entropy and a batch auxiliary loss Our first objective is to encourage diversity in routing behavior across the batch. To this end, we introduce *batch entropy regularizer*. For each encoder i, we compute probabilities by normalizing along the batch dimension:

$$p_i^{(j)} = \frac{\exp\big(z_i^{(j)}\big)}{\sum_{j'=1}^{B} \exp\big(z_i^{(j')}\big)} = \text{softmax}_j\big(z_i^{(j)}\big), \tag{3}$$

and define the per-encoder batch entropy as $H_{\text{batch},i} = -\sum_{j=1}^{B} p_i^{(j)} \log p_i^{(j)}$. The total batch entropy is $H_{\text{batch}} = \mathbb{E}_{i\in\{1\cdots K\}} H_{\text{batch},i}$, which we *maximize*. In the loss, we divide it with a normalization factor, which appears as $\mathcal{L}_{\text{be}} = -\frac{H_{\text{batch}}}{\log B}$. Because the normalization is along the batch dimension, maximizing H_{batch} encourages each encoder's score mass to be spread across many samples rather than being dominated by only a few.

However, the term $\mathcal{L}_{\text{be}}$ alone is insufficient. Consider an extreme case with $K = B = 5$, where for every instance j the router produces a nearly uniform distribution with a fixed small bias, e.g. $[\,0.2 + 4\varepsilon, 0.2 - \varepsilon, 0.2 - \varepsilon, 0.2 - \varepsilon, 0.2 - \varepsilon\,]$ with $\varepsilon > 0$ tiny. Although H_{batch} remains high, top-1 routing (we pick $\arg\max_i$ instead of sampling) would still *always* select the same encoder. This illustrates that H_{batch} captures cross-sample spread in the batch-normalized distributions, not a guarantee of balanced marginal top-1 usage. We therefore add a *batch auxiliary loss*, inspired by balance auxiliaries in MoE: for each encoder i, we form the vector $p_i = [p_i^{(1)}, \ldots, p_i^{(B)}]^\top$ and a one-hot mask $F_i \in \{0,1\}^B$ with its 1 at $\arg\max_j p_i^{(j)}$. We treat F_i with a stop-gradient operator $\mathrm{sg}(\cdot)$ so it is a constant w.r.t. backpropagation. The auxiliary loss is then

$$\mathcal{L}_{\text{ba}} = \sum_{i=1}^{K} \mathrm{sg}(F_i)^\top p_i = \sum_{i=1}^{K} \max_j p_i^{(j)}, \tag{4}$$

which explicitly *minimizes* the largest across-batch probability for each encoder. This discourages any single sample from monopolizing one encoder's batch-normalized mass, but it still does not directly optimize marginal expert usage or remove all collapse modes. We emphasize that $\mathcal{L}_{\text{ba}}$ alone is also inadequate, as it penalizes only the largest entry of each distribution and leaves the rest unconstrained. Thus, we retain $\mathcal{L}_{\text{be}}$ and $\mathcal{L}_{\text{ba}}$ as complementary batch-side heuristics rather than as guarantees.

Instance confidence via instance entropy and an instance auxiliary loss Maximizing H_{batch} can push the router toward a trivial solution with *uniform* predictions for every instance, indicating that the router has learned little about the instance-specific context. We therefore introduce an *instance entropy regularizer* that acts across the encoder dimension to encourage confident decisions per instance. Define

$$q_i^{(j)} = \frac{\exp\left(z_i^{(j)}\right)}{\sum_{i'=1}^{K} \exp\left(z_{i'}^{(j)}\right)} = \mathrm{softmax}_i\left(z_i^{(j)}\right), \tag{5}$$

and the per-instance entropy $H_{\text{instance},j} = -\sum_{i=1}^{K} q_i^{(j)} \log q_i^{(j)}$. We *minimize* a normalized $H_{\text{instance}} = \mathbb{E}_{j\in\{1\cdots B\}} H_{\text{instance},j}$ through $\mathcal{L}_{\text{ie}} = \frac{H_{\text{instance}}}{\log K}$, which drives the per-instance distribution over encoders to be sharp.

To further ease optimization, we pair this with an *instance auxiliary loss* that directly rewards the top-1 probability per instance. Let $q^{(j)} = [q_1^{(j)}, \ldots, q_K^{(j)}]^\top$ and $G_j \in \{0,1\}^K$ be a one-hot vector with 1 at $\arg\max_i q_i^{(j)}$, again wrapped by $\mathrm{sg}(\cdot)$. We then define

$$\mathcal{L}_{\text{ia}} = -\sum_{j=1}^{B} \mathrm{sg}(G_j)^\top q^{(j)} = -\sum_{j=1}^{B} \max_i q_i^{(j)}, \tag{6}$$

so minimizing $\mathcal{L}_{\text{ia}}$ maximizes the instance-wise top-1 confidence.

Combined router objective Putting everything together, the router objective combines the language modeling loss with the two entropy regularizers and the two auxiliary terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{lm} + \lambda_{\text{ba}}\mathcal{L}_{\text{ba}} + \lambda_{\text{be}}\mathcal{L}_{\text{be}} + \lambda_{\text{ie}}\mathcal{L}_{\text{ie}} + \lambda_{\text{ia}}\mathcal{L}_{\text{ia}}, \tag{7}$$

with nonnegative coefficients. Note that $\mathcal{L}_{lm}$ is the cross-entropy language model loss. The batch-side terms and instance-side terms are complementary: $\mathcal{L}_{\text{ie}}$ and $\mathcal{L}_{\text{ia}}$ encourage sharp per-instance decisions, while $\mathcal{L}_{\text{be}}$ and $\mathcal{L}_{\text{ba}}$ encourage diverse routing patterns across samples. We do not claim these objectives provide collapse-proof guarantees; their effectiveness is established empirically in our setting. The ablations below therefore support their utility in this setting, and a stronger direct comparison to alternative balancing objectives remains future work.

The two entropies act along different axes (samples versus encoders); there is no constant γ such that $H_{\text{instance}} \equiv \gamma H_{\text{batch}}$ for all routing logits.

3 Implementation Details

We use the **Qwen-2.5-VL** vision encoder as the shared encoder because it natively supports variable spatial resolutions and aspect ratios. This encoder has 0.67B parameters. The text embedding model is **Qwen3-Embedding-0.6B**, and the LLM decoder is Qwen-2.5-7B. The routed encoder pool includes:

- **SigLIP 2 (ViT)** (**Tschannen et al., 2025**): An enhanced SigLIP model trained with sigmoid-based language–image alignment and location-aware captioning loss (LocCa (Wan et al., 2024)) on multilingual data, making it suitable for tasks requiring dense visual features such as document understanding. Size: 1.14B.
- **DINOv3 ViT** (**Siméoni et al., 2025**): A self-supervised vision transformer trained using self-distillation. Its features are effective for localization and semantic segmentation. Size: 0.3B.
- **DINOv3 ConvNeXt** (**Siméoni et al., 2025**): A ConvNeXt variant distilled from DINOv3 ViT, combining convolutional inductive biases and efficiency with strong semantic representations. It performs better than ViT on high-resolution images. Size: 0.3B.
- **ConvLLaVA** (**Ge et al., 2024**): A hierarchical ConvNeXt vision encoder that progressively reduces the token count of high-resolution images into fine-grained representations. Replacing attention with convolutions gives linear scaling. Size: 0.3B.

Our configuration ensures both supervisory and architectural diversity. The routed pool contains two language-supervised models (SigLIP 2 (ViT), ConvLLaVA) and two self-supervised models (DINOv3 ViT, DINOv3 ConvNeXt). It also balances ViT and ConvNeXt backbones, providing complementary strengths.

The total parameter count is approximately 11B, with 8B to 10B active depending on the routed encoder. The reported 9–27% reduction in active parameters concerns inference; it does not establish a corresponding reduction in resident memory. The STE soft combination evaluates the routed branches during training, so sparse inference does not imply equally sparse training.

For a 1024×768 image, a 64-token text prompt, and a 256-token answer, Table 1 gives analytic vision-encoder and decoder FLOPs estimates, with reductions of 24–49% relative to all-encoder activation. Scope-CA additionally runs the frozen Qwen3-Embedding-0.6B text encoder, whereas Scope-SA and the static baselines do not. The table has no separate allowance for this text-encoder pass or router overhead; its savings should therefore be read for the tabulated components, not as an end-to-end runtime measurement. Appendix D gives the calculation assumptions, and Appendix B.5 discusses their scope.

Table 1: Analytic inference FLOPs for the visual encoders and LLM (TFLOPs). Lower is better for TFLOPs; higher is better for *Compute Saving*. Text-encoder/router overhead is not separately included. These estimates are not measured latency, throughput, or memory.

	+ SigLIP2	+ DINOv3 ViT	+ DINOv3 ConvNeXt	+ ConvLLaVA	+ All 4 Encoders
Shared Enc.	2.24	2.24	2.24	2.24	2.24
Extra Enc.	2.77	1.40	1.08	1.08	6.33
LLM prefill	8.26	7.22	7.22	4.99	9.07
LLM decoding	1.52	1.50	1.50	1.47	1.70
Total	14.79	12.36	12.04	9.78	19.34
Compute Saving	0.24	0.36	0.38	0.49	—

4 Experiments

4.1 Model performance and compute analysis

We train our model in two stages with AdamW (learning rate $1e{-5}$, $\beta_1 = 0.95$, $\beta_2 = 0.999$, weight decay $1e{-6}$, $\epsilon = 1e{-8}$) and a cosine scheduler. In the first stage, we update only the MLP connectors and the router using 150K samples from the LLaVA-Pretrain dataset (Liu et al., 2023a). In the second stage, we include the 50K samples from the Arxiv-OCR (nz, 2024) dataset, 30K samples from Chart2MD, Table2Markdown split in BigDocs (Rodriguez et al., 2024), 50K training samples

from VCR (Zhang et al., 2024b), 70K training samples from the DocVQA, InfoVQA, TextVQA and ChartVQA split from DocDownstream (Hu et al., 2024). Among them, Arxiv-OCR is a pure-OCR dataset collected from arXiv.

Chart2Markdown and Table2Markdown is to reconstruct the chart or table's data as a Markdown table. VCR is a task where models must restore partially occluded text in images using pixel-level visual cues and context. DocVQA focus on VQA task on scanned documents that usually require OCR ability. InfoVQA combines reading embedded text with interpreting icons, charts, and diagrammatic elements to answer questions. TextVQA is a VQA task on natural images where answering hinges on detecting and understanding scene text within the image. ChartQA demands extracting plotted values and reasoning over the chart's structure to answer questions. In total, we trained on 200K training samples in the second stage. We trained on the connectors, router and all vision encoders in the second stage. The static baselines use the same data budget and broad two-stage VLM adaptation pipeline as SCOPE. The comparison therefore evaluates the complete MoEnc recipe under a shared deployment interface, while component-level attribution requires further ablations (Appendix B.7). We evaluate the following settings:

- **SCOPE-CA**: the router is a cross-attention mechanism receiving both shared vision encoder representation V_s and text embedding T;
- **SCOPE-SA**: the router is a self-attention mechanism receiving only the shared vision encoder representation V_s;
- **SCOPE-MLP**: the router is an MLP receiving only the shared vision encoder representation V_s;
- **baseline-0**: with no extra vision encoder; the connector is reinitialized for fair comparison;
- **baseline-1**: with one extra vision encoder; we train all four single-auxiliary variants, select one configuration by average score on a held-out validation split, and evaluate that fixed configuration across all test benchmarks;
- **baseline-2**: with two extra vision encoders; we instantiate this baseline for each of the $\binom{4}{2} = 6$ encoder pairs and report the maximum performance across these variants;
- **baseline-3**: with three extra vision encoders; we instantiate this baseline for each of the $\binom{4}{3} = 4$ encoder triplets and report the maximum performance across these variants;
- **baseline-4**: with four extra vision encoders; this is the single configuration with all four encoders active.

The performance of these models is listed in Table 2.

Table 2: Performance comparison of model components across eight benchmarks (higher is better). Bold indicates the best score per column.

Model	Table2MD	Chart2MD	$\text{VCR}_{\text{EN, HARD}}$	$\text{VCR}_{\text{ZH, HARD}}$	DocVQA	InfoVQA	TextVQA	ChartQA	Avg
SCOPE-CA	**70.8**	**65.1**	74.4	**15.6**	**73.1**	**63.4**	**67.9**	68.3	**62.3**
SCOPE-SA	68.4	63.9	**75.2**	14.1	70.1	59.9	65.6	**68.6**	60.6
SCOPE-MLP	67.9	62.2	70.1	13.8	69.9	60.2	64.7	67.1	59.5
baseline-0	58.5	55.2	56.9	8.2	65.5	52.4	61.8	64.0	53.1
baseline-1	63.9	60.6	59.7	11.6	67.1	56.5	64.4	65.9	56.2
baseline-2	67.1	63.7	66.1	13.9	68.3	57.5	65.0	66.8	58.6
baseline-3	70.0	63.7	66.5	14.9	69.8	59.4	65.6	67.9	59.7
baseline-4	70.4	**65.1**	68.2	15.3	69.0	60.1	65.2	67.4	60.1

Gains across benchmarks SCOPE-CA improves over baseline-1 on all eight reported benchmarks, with gains from 2.4 to 14.7 points and an average gain of 6.1 points. English VCR gives the largest individual improvement, but the advantage is not confined to that benchmark: excluding $\text{VCR}_{\text{EN,HARD}}$, the mean score is 60.6 versus 55.7, a 4.9-point gain. Appendix A.4 gives the full breakdown. These comparisons describe the reported scores; they do not establish statistical significance without variance estimates.

4.2 Hyper-parameter selection

The loss-weight sweep in Appendix A.1 compares equal weights, stronger instance-side weighting, removal of either side, and removal of all routing regularization. It evaluates task performance and the selection-frequency gap between the most-used and least-used routed encoders. The best average score uses $\lambda_{\mathrm{be}} = \lambda_{\mathrm{ba}} = 0.2$ and $\lambda_{\mathrm{ie}} = \lambda_{\mathrm{ia}} = 0.6$. Removing all router regularization lowers the average score from 62.3 to 55.8 and increases the frequency gap from 18.9 to 98.1 (Table 4).

More balanced encoder use does not necessarily yield better task performance. Setting all four weights to 0.5 gives the smallest frequency gap, 9.4, but an average score of 59.9, below the best configuration's 62.3 with a gap of 18.9. Thus, the sweep supports a trade-off between diverse encoder use and task performance, rather than uniform usage as a goal in itself. Since entropy and auxiliary weights are tied within each side in every reported configuration, these results do not independently isolate the two types of losses.

5 Discussions

Our experiments use subsets of the available datasets because of compute constraints. All compared models share the same data budget and broad two-stage training pipeline; the results characterize this controlled setting rather than full-data performance ceilings. The English OCR-oriented training component improves downstream performance in our experiments, which we attribute to improved use of dense, fine-grained visual evidence. Appendix B.1 discusses baseline coverage and comparison fairness.

When text-conditioned routing helps Scope-CA improves over Scope-SA on DocVQA, InfoVQA, and TextVQA, where questions can emphasize different visual regions and attributes. Table2MD and Chart2MD instead use semantically similar reconstruction prompts, while VCR uses a fixed query template. Visual-only routing remains competitive in these settings, although the benefit of text varies by benchmark: Scope-CA is stronger on both reconstruction tasks, and Scope-SA is stronger on English VCR and ChartQA. We interpret query diversity as a possible explanation, rather than a complete account of the ranking.

Routing specialization Table 3 summarizes routing across five task families, combining the two DINOv3 encoders. Family labels are used only for post-hoc analysis; the router receives an image–prompt pair at deployment.

Table 3: Encoder selection frequency (%) by task family. The DINOv3 column combines ViT and ConvNeXt; each row sums to 100%.

Task family	SigLIP 2 (ViT)	DINOv3 (combined)	ConvLLaVA
Document VQA	72.1	18.7	9.2
Scene and OCR VQA	64.6	19.8	15.6
VCR	2.9	94.5	2.6
Charts and Infographics	11.1	83.7	5.2
Table Reconstruction	18.1	21.5	60.4

SigLIP 2 receives 72.1% of Document VQA selections and 64.6% of Scene and OCR VQA selections, consistent with language-aligned features serving text-rich inputs. DINOv3 receives 94.5% on VCR and 83.7% on Charts and Infographics, while ConvLLaVA leads Table Reconstruction at 60.4%, consistent with useful local structural features. These are descriptive associations; secondary allocations show that selection is not a deterministic family-to-encoder map. Appendix A.3 discusses training-mixture effects and routing counterfactuals.

Interpreting the all-encoder comparison Scope-CA exceeds baseline-4 on average (62.3 versus 60.1), while tying its Chart2MD score. A shared Qwen-2.5-VL branch already contributes about 1,036 visual tokens for a 1024×768 image. Concatenating every auxiliary stream lengthens this prefix further, which may make relevant evidence harder to use. This interpretation is consistent with baseline-3 exceeding baseline-4 on DocVQA and TextVQA, but is not a direct measurement

of attention dilution. Learned compression or other cross-encoder fusion methods could change the comparison (Appendix B.3).

Router design: attention vs. MLP The visual-only MLP router underperforms self-attention on average (59.5 versus 60.6), even though MLP gates are common in MoE. This comparison supports attention-based routing in the evaluated setting.

VCR and language coverage The low Chinese VCR scores reflect the English-heavy training mixture. Improvements on both VCR languages suggest useful fine-grained visual selection, but do not establish broad multilingual capability; Appendix B.6 gives the detailed interpretation.

Batch size of one The tile-based pseudo-batch variant attains an average score of 61.4 and a selection-frequency gap of 22.3. Its setup is described in Appendix A.2.

6 Related Work

Vision encoders in VLMs. Modern VLMs differ mainly in the backbone and pretraining objective of their vision encoders. Contrastive models (CLIP (Radford et al., 2021), SigLIP/SigLIP2 (Zhai et al., 2023; Tschannen et al., 2025)) pair ViT/ResNet backbones with image–text alignment losses. Self-supervised families (DINO/DINOv2/DINOv3 (Caron et al., 2021; Oquab et al., 2024; Siméoni et al., 2025)) scale ViT/ConvNeXt backbones with improved data and training recipes. Task-specialized encoders (e.g., SAM (Kirillov et al., 2023; Ravi et al., 2024) for promptable segmentation; MAE (He et al., 2022) for masked image modeling) provide strong features but with different inductive biases. InternViT (Chen et al., 2024b;a; Zhu et al., 2025) exemplifies a ViT coupled to an LLM via cross-attention, trained with contrastive objectives. Finally, ConvNeXt backbones (ConvLLaVA (Ge et al., 2024)) trade some global context for stronger local spatial modeling and efficient high-resolution processing.

VLMs with multiple encoders. Recent systems combine complementary encoders to balance global semantics and fine detail. Two-encoder designs (Janus (Wu et al., 2024), Mini-Gemini (Li et al., 2024), Ferret (Zhang et al., 2024a)) pair low- and high-resolution branches, fusing features via interpolation / concatenation, patch-level refinement, or layer-wise cross-attention. Larger mixtures (SPHINX (Lin et al., 2023), Cambrian-1 (Tong et al., 2024)) concatenate or aggregate multigranular features (e.g., with learnable queries in an SVA). LEO (Azadani et al., 2025) also studies mixtures of vision encoders. Fusion simplicity often suffices: MouSi (Fan et al., 2024) finds MLP projection competitive with Q-Former, and Eagle (Shi et al., 2025) reports that straightforward concatenation of complementary features can match more complex schemes. MoAI (Lee et al., 2024) and MoVA (Zong et al., 2024a) introduce routing logic to their mixture of module architectures. MoAI precalculates visual, auxiliary, and language features for a learnable router to select; MoVA's routing technique is not based on learnable routers. Instead, it relies on LLM to classify the task and send the image to the corresponding specialized encoder. The Scope router is a learnable module for selecting encoders to process the image features, which is fundamentally different from them.

7 Limitations and Conclusions

Scope combines instance-level encoder selection, text- and vision-conditioned routing, and a practical dual-regularization objective. On the evaluated document-centric and OCR-heavy workloads, it improves average performance over the reported static baselines while using one auxiliary stream at inference. This deployment constraint fixes stream count, not token count: the selected encoder contributes 192 to 1036 auxiliary tokens in the representative setup. Training evaluates the routed branches for the STE, and Scope-CA also requires a frozen 0.6B text encoder. Our subsetted, English-heavy experiments and analytic component-level compute estimates leave broader task coverage, multilingual training, equal-token or equal-FLOPs comparisons, stronger fusion baselines, and measured serving efficiency for future work. The reported comparisons also leave controls using random, family-level static, or oracle routing, per-seed uncertainty, and independent training-component ablations unresolved (Appendices A.3 and B.7).

AI USE STATEMENT

We used LLMs for language editing, comparison and consolidation of manuscript versions, and LaTeX formatting and debugging.

REFERENCES

Anthropic. The claude 3 model family: Opus, sonnet, haiku. *https://www.anthropic.com*, 2025. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

Mozhgan Nasr Azadani, James Riddell, Sean Sedwards, and Krzysztof Czarnecki. Leo: Boosting mixture of vision encoders for multimodal large language models. *arXiv preprint arXiv: 2501.06986*, 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv: 2104.14294*, 2021.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiao wen Dong, Hang Yan, Hewei Guo, Conghui He, Zhenjiang Jin, Chaochao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, and Yu Qiao. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 2024a. doi: 10.48550/arXiv.2404.16821.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *CVPR*, 2024b.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Xiaoran Fan, Tao Ji, Changhao Jiang, Shuo Li, Senjie Jin, Sirui Song, Junke Wang, Boyang Hong, Lu Chen, Guodong Zheng, et al. Mousi: Poly-visual-expert vision-language models. *arXiv preprint arXiv:2401.17221*, 2024.

Chunjiang Ge, Sijie Cheng, Ziming Wang, Jiale Yuan, Yuan Gao, Jun Song, Shiji Song, Gao Huang, and Bo Zheng. Convllava: Hierarchical backbones as visual encoder for large multimodal models. *arXiv preprint arXiv: 2405.15738*, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, et al. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *arXiv preprint arXiv:2403.12895*, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

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.

Oğuzhan Fatih Kar, Alessio Tonioni, Petra Poklukar, Achin Kulshrestha, Amir Zamir, and Federico Tombari. Brave: Broadening the visual encoding of vision-language models. In *European Conference on Computer Vision*, pp. 113–132. Springer, 2024.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Byung-Kwan Lee, Beomchan Park, Chae Won Kim, and Yong Man Ro. *MoAI: Mixture of All Intelligence for Large Language and Vision Models*, pp. 273–302. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72967-6_16. URL https://link.springer.com/content/pdf/10.1007/978-3-031-72967-6_16.

Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024.

Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023a.

Shikun Liu, Linxi Fan, Edward Johns, Zhiding Yu, Chaowei Xiao, and Anima Anandkumar. Prismer: A vision-language model with multi-task experts. *arXiv preprint arXiv:2303.02506*, 2023b.

Yuchen Liu, Yaoming Wang, Bowen Shi, Xiaopeng Zhang, Wenrui Dai, Chenglin Li, Hongkai Xiong, and Qi Tian. Meteor: Multi-encoder collaborative token pruning for efficient vision language models. *arXiv preprint arXiv:2507.20842*, 2025.

Song Mao, Yang Chen, Pinglong Cai, Ding Wang, Guohang Yan, Zhi Yu, and Botian Shi. Investigating redundancy in multimodal large language models with multiple vision encoders. *arXiv preprint arXiv:2507.03262*, 2025.

nz. nz/arxiv-ocr-v0.2: Synthetic ocr dataset from arxiv pdfs. https://huggingface.co/datasets/nz/arxiv-ocr-v0.2, 2024. Accessed: 2025-07-06.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.

Juan Rodriguez, Xiangru Jian, Siba Smarak Panigrahi, Tianyu Zhang, Aarash Feizi, Abhay Puri, Akshay Kalkunte, François Savard, Ahmed Masry, Shravan Nayak, Rabiul Awal, Mahsa Massoud, Amirhossein Abaskohi, Zichao Li, Suyuchen Wang, Pierre-André Noël, Mats Leon Richter, Saverio Vadacchino, Shubham Agarwal, Sanket Biswas, Sara Shanian, Ying Zhang, Noah Bolger, Kurt MacDonald, Simon Fauvel, Sathwik Tejaswi, Srinivas Sunkara, Joao Monteiro, Krishnamurthy DJ Dvijotham, Torsten Scholak, Nicolas Chapados, Sepideh Kharagani, Sean Hughes, M. Özsu, Siva Reddy, Marco Pedersoli, Yoshua Bengio, Christopher Pal, Issam Laradji, Spandana Gella, Perouz

Taslakian, David Vazquez, and Sai Rajeswar. Bigdocs: An open dataset for training multimodal models on document and code tasks. *arXiv preprint arXiv: 2412.04626*, 2024.

Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, Yilin Zhao, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, Bryan Catanzaro, Andrew Tao, Jan Kautz, Zhiding Yu, and Guilin Liu. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *International Conference on Learning Representations*, 2025.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.

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

Bo Wan, Michael Tschannen, Yongqin Xian, Filip Pavetic, Ibrahim Alabdulmohsin, Xiao Wang, André Susano Pinto, Andreas Steiner, Lucas Beyer, and Xiaohua Zhai. Locca: Visual pretraining with location-aware captioners, 2024. URL https://arxiv.org/abs/2403.19596.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv: 2410.13848*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *IEEE International Conference on Computer Vision*, 2023. doi: 10.1109/ICCV51070.2023.01100.

Haotian Zhang, Haoxuan You, Philipp Dufter, Bowen Zhang, Chen Chen, Hong-You Chen, Tsu-Jui Fu, William Yang Wang, Shih-Fu Chang, Zhe Gan, and Yinfei Yang. Ferret-v2: An improved baseline for referring and grounding with large language models. *arXiv preprint arXiv: 2404.07973*, 2024a.

Tianyu Zhang, Suyuchen Wang, Lu Li, Ge Zhang, Perouz Taslakian, Sai Rajeswar, Jie Fu, Bang Liu, and Y. Bengio. Vcr: A task for pixel-level complex reasoning in vision language models via restoring occluded text. *International Conference on Learning Representations*, 2024b.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv: 2504.10479*, 2025.

Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu. Mova: Adapting mixture of vision experts to multimodal context. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on*

Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/bb0fea29f7aa6ede17e906ac6a225f34-Abstract-Conference.html.

Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu. Mova: Adapting mixture of vision experts to multimodal context. *Advances in Neural Information Processing Systems*, 37:103305–103333, 2024b.

A Additional Ablations and Routing Analysis

A.1 Loss-weight selection

Design of the sweep. The configurations are structured around relative weighting and removal of the two sides of the routing objective. They include equal batch- and instance-side weights, increased instance-side weights, removal of the batch-side pair, removal of the instance-side pair, and removal of all four routing terms. This design tests whether the two sides are useful together and how their weighting affects task performance and selection-frequency balance.

Table 4 reports the loss-weight configurations for Scope-CA. The subscripts b/i denote batch/instance terms, and a/e denote auxiliary/entropy terms. The best average score in this sweep uses $(\lambda_{\mathrm{be}}, \lambda_{\mathrm{ie}}, \lambda_{\mathrm{ba}}, \lambda_{\mathrm{ia}}) = (0.2, 0.6, 0.2, 0.6)$. The three tested settings with an instance-to-batch weight ratio of $3:1$ attain average scores from 61.9 to 62.3. Their selection-frequency gaps differ, illustrating the empirical trade-off between task score and encoder-use balance. The equal-weight setting of 0.5 gives the smallest gap in this table, but lower task performance. These observations support the practical role of the objective without implying that it guarantees uniform top-1 usage.

Table 4: Scope-CA performance across loss-weight configurations. **Avg** is the average task score (higher is better). **Range** is the selection-frequency gap between the most-used and least-used routed encoders (lower is better).

λ_{be}	λ_{ie}	λ_{ba}	λ_{ia}	Avg	Range
0.3	0.9	0.3	0.9	61.9	14.2
0.2	0.2	0.2	0.2	61.1	25.8
0.2	0.4	0.2	0.4	62.0	16.8
0.2	0.6	0.2	0.6	**62.3**	18.9
0.1	0.3	0.1	0.3	62.1	21.5
0.5	0.5	0.5	0.5	59.9	**9.4**
0.2	0.0	0.2	0.0	61.5	29.4
0.0	0.6	0.0	0.6	60.2	33.8
0.0	0.0	0.0	0.0	55.8	98.1

A.2 Batch size of one

When the batch size is 1, we emulate batch diversity by tiling each image with a LLaVA-Next–style preprocessor: the input is split into 336×336 patches, which we treat as a pseudo-batch. The router operates per tile, so the dual-entropy/auxiliary objectives are computed across tiles instead of across different images, encouraging diversity across tiles during training. The same tiling is used at inference in this variant. Using this scheme, SCOPE-CA attains an average score of 61.4, with a selection-frequency gap of 22.3, indicating a minor impact on overall performance.

A.3 Additional Routing Analysis

Encoder preferences by task family. Table 3 in Section 5 reports encoder selection frequency for five coarse task families. These labels are used only for post-hoc analysis; the router is trained and deployed without hard task-family labels. The two DINOv3 encoders are aggregated in this table, so a dominant DINOv3 share describes the family rather than either individual encoder.

SigLIP 2 receives 72.1% of selections on Document VQA and 64.6% on Scene and OCR VQA, consistent with the usefulness of language-aligned representations for text-rich inputs. The DINOv3 family receives 94.5% on VCR and 83.7% on Charts and Infographics. ConvLLaVA is selected most often for Table Reconstruction (60.4%), consistent with an advantage from local geometry and structural detail. These interpretations are hypotheses about complementary representations, rather than causal conclusions from frequency alone. The secondary allocations show that selection is not a deterministic family-to-encoder map.

Why not a family-level static selector? A family-level heuristic could capture part of these aggregate preferences. At deployment, however, the input consists of an image–prompt pair without a

benchmark or family label. Such a policy would require an additional taxonomy and classification rule. Within-family variability also remains: document questions can emphasize dense reading, layout, or small local evidence, while chart and table inputs vary in semantic and geometric demands. SCOPE learns its decisions from the input pair. The frequency table motivates this distinction, but does not replace a direct accuracy and cost comparison with a family-static selector; that comparison remains open.

Remaining analysis. The task-family breakdown addresses coarse specialization. Analysis by prompt semantics, text density, image resolution, and failure mode would further clarify when the router chooses useful experts. Because the DINOv3 counts are pooled here, separate ViT and ConvNeXt routing statistics would also be informative.

Training mixture and routing frequencies. The training mixture can influence the router's preferences. Task-family frequencies are post-hoc summaries and do not independently separate useful responses to visual or query requirements from correlations with dataset composition. Conditional routing statistics within a family, separate counts for the two DINOv3 encoders, and experiments that reweight the training mixture would help distinguish these effects. The current frequencies should therefore not be interpreted as showing that routing is invariant to the task distribution.

Counterfactual routing controls. Random routing would test whether some benefits arise from training with access to several encoders rather than from input-dependent selection. Replacing a trained router only at inference and training a model with random assignments answer different questions; a controlled comparison must specify the randomization stage and distribution. A family-static policy would choose one encoder for each task family, with the choice fixed using validation data. An oracle family label provides privileged information, while a predicted label introduces classifier errors and additional cost. Either can be a useful diagnostic when its assumptions are stated. An instance-level oracle that uses reference answers to choose among encoder-specific outputs would measure available selection headroom under that evaluation procedure, but is not a deployable policy. The reported tables do not provide quantitative results for these controls, so the routing-frequency analysis does not resolve their relative accuracy.

A.4 PER-BENCHMARK GAINS AND SENSITIVITY TO ENGLISH VCR

Table 5 reorganizes the SCOPE-CA and baseline-1 scores from Table 2 and subtracts them benchmark by benchmark. All eight differences are positive. The largest gain occurs on English VCR, but excluding that benchmark still leaves a mean gain of approximately 4.9 points. This calculation addresses the contribution of one benchmark to the headline average; it does not change the training or evaluation results.

Table 5: Descriptive score differences computed from the reported values in Table 2. Δ is SCOPE-CA minus baseline-1, in score points. Means and differences are rounded to one decimal place.

Benchmark	SCOPE-CA	baseline-1	Δ
Table2MD	70.8	63.9	+6.9
Chart2MD	65.1	60.6	+4.5
$\text{VCR}_{\text{EN,HARD}}$	74.4	59.7	+14.7
$\text{VCR}_{\text{ZH,HARD}}$	15.6	11.6	+4.0
DocVQA	73.1	67.1	+6.0
InfoVQA	63.4	56.5	+6.9
TextVQA	67.9	64.4	+3.5
ChartQA	68.3	65.9	+2.4
Mean over all eight	62.3	56.2	+6.1
Mean excluding English VCR	60.6	55.7	+4.9

Uncertainty and interpretation. The differences use rounded benchmark scores and are descriptive comparisons. Aggregate scores alone do not provide per-seed variance or paired sample-level uncertainty, so these differences do not establish statistical significance. Per-seed scores or suitable paired uncertainty estimates would be needed to assess the stability of the improvements.

B Evaluation Scope, Baseline Availability, and Fairness of Comparison

B.1 Baseline Coverage and the Controlled Comparison

Some multi-encoder systems do not provide the implementation details, checkpoints, training recipes, or data mixtures needed for faithful reproduction. We therefore focus on a controlled family of baselines rather than speculative reconstructions of such systems. Baseline-0 through baseline-4 activate increasing numbers of auxiliary encoders; Scope learns which one to activate for an image–prompt pair. The compared models share the LLM and shared-encoder families, candidate encoder pool, training-data budget, and optimization setup. This controls several sources of variation across systems, while differences in routing, auxiliary token counts, and text-encoder use remain explicit parts of the design.

Shared training pipeline and baseline selection. The static baselines and Scope use the same data budget and broad two-stage VLM adaptation pipeline. The first stage aligns newly introduced modules; the second adapts vision-side components to the downstream mixture. This common pipeline prevents the comparison from being interpreted as two-stage adaptation for Scope versus a deliberately weaker training schedule for the baselines. It does not make the objectives, active components, or training FLOPs identical across architectures.

For baseline-1, all four single-auxiliary variants are trained. One configuration is chosen by its average score on a held-out validation split and then used across all test benchmarks. This is a globally fixed auxiliary encoder, rather than a separate test-set choice for each benchmark.

The principal comparison is with the reported single-auxiliary baseline, since it uses the same number of auxiliary streams at inference. The all-encoder baseline answers a complementary question about static aggregation. These comparisons study how to use a fixed encoder pool under the reported setup; they do not exhaust prior multi-encoder architectures or establish a ranking among systems trained with different data and recipes.

B.2 Empirical Scope and Training Data

The benchmarks emphasize document understanding, OCR-relevant visual perception, chart and table reconstruction, and recovery of fine-grained visual evidence. They require different combinations of reading, localization, structure understanding, and visual reasoning. The results support dynamic selection in this regime. Broader natural-image understanding, open-ended captioning, video, and other modalities require separate evaluation.

All methods use the same subsets of the available training corpora because of compute constraints. This supports relative comparisons under the reported data and optimization budgets; it does not identify the full-data performance ceiling. Matching these budgets limits data-access confounds, but does not establish that the ranking would be unchanged under larger-scale training. Scaling both static and routed methods remains a useful follow-up.

Scope of the encoder pool. The pool varies language-supervised versus self-supervised pretraining and ViT versus ConvNeXt architectures. It does not include dedicated detection or segmentation backbones. Those experts could offer other kinds of visual evidence, but their value under the same deployment constraint is not evaluated here.

B.3 The All-Encoder Baseline and Budget Matching

Naive all-encoder concatenation is a transparent static counterpart to routing: it exposes the decoder to every available auxiliary stream. Its weaker average result in this setup supports the value of selecting visual features, while leaving open the potential of learned token compression, cross-encoder aggregation, or top-k merging. It represents one fusion strategy, not the whole multi-encoder design space.

A common encoder pool and training-data budget do not imply identical inference FLOPs, token counts, or capacity. Even the "1+1" configurations differ in auxiliary prefix length: the representative 1024×768 setup yields 192 to 1036 auxiliary tokens, depending on the selected encoder. The

comparisons therefore address the performance–compute trade-off under a stream-count constraint. Equal-token and equal-FLOPs experiments would isolate additional factors and are not supplied by the current results.

B.4 Routing and the Role of the Training Objective

Appendix A.3 reports task-family routing statistics and discusses family-static selection, the unresolved influence of the training mixture, and counterfactual routing comparisons needed for stronger attribution. The reported frequencies describe coarse preferences and within-family variation; they are not a complete interpretability study.

The two batch-side terms distribute each encoder's batch-normalized score mass across samples, while the two instance-side terms sharpen decisions across encoders. As explained in Section 2.3, these are distinct normalization axes. The batch-side terms do not directly optimize marginal top-1 usage and do not guarantee freedom from collapse. The loss-weight sweep in Table 4 supports the objective as a practical training recipe. Direct comparisons with alternative load-balancing objectives, including simpler formulations, remain open.

B.5 Efficiency Metrics and Deployment Costs

The compute analysis reports analytic FLOPs and active-parameter estimates. These quantities describe activated computation under stated architectural assumptions; they do not measure latency, throughput, peak memory, or the cost of a serving system. In particular, inactive weights may still occupy device memory. Kernel efficiency, batching, data movement, and encoder loading can affect realized speed and memory use.

The rows of Table 1 account for shared and auxiliary visual encoders, LLM prefill, and LLM decoding. They do not separately include the text-encoder pass required by Scope-CA or router overhead. Scope-SA avoids the additional text encoder. Moreover, the STE soft combination requires evaluating routed branches during training, so the one-auxiliary-stream inference saving should not be interpreted as a training-cost reduction. End-to-end runtime profiling is needed to assess deployment savings beyond the tabulated estimates.

Accounting for the extra routing components. Let $C_{\mathrm{tab},j}$ be the visual-encoder and decoder estimate for auxiliary encoder j in Table 1. Extending that component accounting to include text-conditioned routing gives

$$\widehat{C}_{\mathrm{CA},j} = C_{\mathrm{tab},j} + C_{\mathrm{text}} + C_{\mathrm{router,CA}}, \qquad \widehat{C}_{\mathrm{SA},j} = C_{\mathrm{tab},j} + C_{\mathrm{router,SA}}.$$

The frozen text encoder is required by CA and not by SA or the static baselines. Under this accounting scope, the CA saving relative to the static all-encoder estimate C_{all} would be $1 - \widehat{C}_{\mathrm{CA},j}/C_{\mathrm{all}}$. Its evaluation requires consistent sequence lengths, model configurations, and multiply–add conventions for every term. The table does not quantify C_{text} or the router terms, and we do not assign a corrected total here. These identities specify additional computation to count; they are not latency measurements or a claim to cover every preprocessing and serving overhead.

Runtime evaluation. An empirical comparison should use common hardware and software settings and specify precision, batch size, image resolution, prompt length, and generation length. Timing should include the routing-related forwards and distinguish prefill from decoding. Memory measurements must state whether inactive encoder weights remain resident. Such measurements would complement the analytic estimates and are not supplied by the current results.

B.6 Language Coverage and VCR

Performance on $\mathrm{VCR}_{\mathrm{ZH,\ HARD}}$ is substantially lower than on the other benchmarks. The OCR-oriented training data are exclusively English, and the other training sets contain no Chinese. We retain Chinese VCR as a stress test of fine-grained visual evidence recovery under mismatched language supervision. Improvements on both English and Chinese VCR suggest that selecting complementary visual features can still help in this setting. The low absolute Chinese scores and restricted training coverage limit any broader multilingual claim.

B.7 Contribution and Interpretation of the Results

The contribution is the combination of instance-level encoder selection, routing conditioned on shared visual features and text, and a practical objective for training hard top-1 decisions. These ingredients draw on existing mixture-model and multimodal-routing ideas; the empirical contribution is their use for document-centric VLMs under a one-auxiliary-stream deployment constraint.

Taken together, the results support a useful average performance–compute trade-off in the controlled setting studied here. They do not establish universal superiority across multimodal workloads, all fusion designs, or fully scaled training regimes.

Limits of training-component attribution. The empirical comparison concerns the complete MoEnc training recipe, including connector alignment, routing losses, and second-stage vision-side finetuning. Shared data and a common broad training pipeline control some differences between methods, but do not isolate the causal contribution of every component. One-stage versus two-stage training, frozen versus unfrozen vision encoders, and connector/router-only second-stage updates would provide additional component controls. Those ablations are not reported here, so the current results should not be read as showing that the learned router alone explains every improvement.

C Pretrained Model List

In Table 6, we list the pretrained models used in this paper.

Table 6: Pretrained models used and their links.

Model	Link
Qwen-2.5-VL-7B-Instruct	huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct
Qwen3-Embedding-0.6B	huggingface.co/Qwen/Qwen3-Embedding-0.6B
SigLIP 2 (ViT)	huggingface.co/google/siglip2-so400m-patch14-384
DINOv3 ViT (vit-l/16, lvd1689m)	huggingface.co/facebook/dinov3-vitl16-pretrain-lvd1689m
DINOv3 ConvNeXt (large, lvd1689m)	huggingface.co/facebook/dinov3-convnext-large-pretrain-lvd1689m
ConvLLaVA-ConvNeXt-1536	huggingface.co/ConvLLaVA/ConvLLaVA-ConvNeXt-1536

D Vision Tokenization and LLM FLOPs

D.1 Assumptions

- Image size: 1024×768.
- Text prompt length: 64 tokens.
- LLM (Qwen-2.5 7B): $L = 28$ layers, hidden size $d = 3584$, FFN size $f = 18944$.
- Generation length: $T = 256$ tokens, with KV cache in decoding.
- Prefill length into the LLM: $S_0 =$ #vision tokens to LLM $+ 64$.

D.2 Vision Tokens Sent to the LLM (Prefill)

Below, "grid" denotes the spatial token grid before any optional merging; divisions are exact or use ceiling when noted.

(1) Qwen-2.5-VL native ViT (patch 14, with 2×2 merge)

$$\text{grid} = \left\lceil \frac{1024}{14} \right\rceil \times \left\lceil \frac{768}{14} \right\rceil = 74 \times 56, \quad \text{\#ViT tokens} = 74 \cdot 56 = 4144.$$

After 2×2 spatial merge:

$$\text{\#vision to LLM} = \frac{74}{2} \cdot \frac{56}{2} = 37 \times 28 = 1036, \quad S_0 = 1036 + 64 = 1100.$$

(2) ConvLLaVA–ConvNeXt-1536 (effective stride ≈ 64; no extra merge)

$$\text{grid} = \left\lceil \frac{1024}{64} \right\rceil \times \left\lceil \frac{768}{64} \right\rceil = 16 \times 12 = 192, \quad S_0 = 192 + 64 = 256.$$

(3) SigLIP2 so400m patch14 (ViT-like, with 2×2 merge)

$$\text{grid} = 74 \times 56 = 4144 \quad (\text{patch } 14), \qquad \text{after } 2 \times 2 \text{ merge} \Rightarrow 1036, \quad S_0 = 1036 + 64 = 1100.$$

(4) DINOv3–ConvNeXt-L (output stride 32; no extra merge)

$$\text{grid} = \frac{1024}{32} \times \frac{768}{32} = 32 \times 24 = 768, \quad S_0 = 768 + 64 = 832.$$

(5) DINOv3–ViT-L/16 (patch 16, with 2×2 merge)

$$\text{grid} = \frac{1024}{16} \times \frac{768}{16} = 64 \times 48 = 3072,$$
$$\text{after } 2 \times 2 \text{ merge} \Rightarrow 32 \times 24 = 768, \quad S_0 = 768 + 64 = 832.$$

Encoder	Vision tokens to LLM	S_0 (prefill length)
Qwen2.5–ViT (2×2 merge)	1036	1100
ConvLLaVA–ConvNeXt-1536 (stride ≈ 64)	192	256
SigLIP2 so400m patch14 (2×2 merge)	1036	1100
DINOv3–ConvNeXt-L (stride 32)	768	832
DINOv3–ViT-L/16 (2×2 merge)	768	832

D.3 LLM FLOPs Formulas

We use standard Transformer FLOPs approximations (FP32; one multiply–add $= 2$ FLOPs). For a sequence length S within a layer:

$$\text{Attn proj (Q,K,V,O)}: \ 4Sd^2, \qquad \text{Attn matmuls}: \ 2S^2d, \qquad \text{FFN}: \ 2Sdf.$$

Prefill FLOPs (sequence length S_0)

$$\text{FLOPs}_{\text{prefill}} = L\left(4S_0d^2 + 2S_0^2d + 2S_0df\right).$$

Decode FLOPs with KV cache (generate T tokens) At decoding step $t \in \{1, \ldots, T\}$ the seen length is $S_0 + t - 1$. Per step, per layer:

$$4d^2 + 2(S_0 + t - 1)d + 2df.$$

Summing over t and multiplying by L:

$$\text{FLOPs}_{\text{decode}} = L\left(4Td^2 + 2d\left(TS_0 + \frac{T(T-1)}{2}\right) + 2Tdf\right).$$

Total LLM FLOPs

$$\text{FLOPs}_{\text{LLM}} = \text{FLOPs}_{\text{prefill}} + \text{FLOPs}_{\text{decode}}.$$

D.4 Plug-in Values (for replication)

For Qwen-2.5 7B alignment used here:

$$L = 28, \quad d = 3584, \quad f = 18944, \quad T = 256,$$
$$S_0 \in \{1100,\ 256,\ 1100,\ 832,\ 832\} \text{ per the encoders above.}$$