ROUTEVLM-NEXT: MAKING VISION-LANGUAGE MODELS LOOK SMARTER AND FURTHER VIA HYBRID ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language Models (VLMs) are severely constrained by the quadratic complexity of self-attention, making the processing of high-resolution images and long videos computationally prohibitive. Existing approaches that uniformly compress or prune visual tokens often discard the very fine-grained details necessary for complex reasoning. To address this trade-off between efficiency and fidelity, we introduce RouteVLM-NEXT, a VLM architecture with a router-guided hybrid attention mechanism. At its core, a learnable router dynamically directs each visual token to one of two pathways: a standard full-attention branch for capturing intricate details in critical regions, or a highly efficient gated linear attention branch for processing background context. This mechanism, governed by a budget-aware training objective, allows the model to autonomously manage its computational resources, allocating high-fidelity computation only where it is most needed. Through extensive experiments, we show that RouteVLM-NEXT achieves strong performance on a wide range of benchmarks, particularly excelling on detail-oriented tasks like V* Bench, while exhibiting improved scalability. It handles sequences with up to 64k visual tokens under our evaluation, a regime where standard full attention fails on the same GPU, providing a practical direction for next-generation VLM design.

1 INTRODUCTION

Recent years have witnessed the rapid progress of Large Language Models (LLMs) (Radford et al., 2019; Touvron et al., 2023; DeepSeek-AI, 2025) and Vision-Language Models (VLMs) (Bai et al., 2023; OpenAI, 2024; Team, 2025a), which have demonstrated impressive capabilities in understanding and reasoning across modalities. As these models are increasingly applied to high-resolution imagery and fine-grained visual reasoning, the demand for processing larger-scale visual inputs continues to grow. Tasks such as recognizing small objects, reading embedded text, or reasoning over long-duration videos require handling tens of thousands of visual tokens, pushing current architectures to their computational limits.

However, existing MLP-style VLM architectures such as the LLaVA family (Liu et al., 2023; 2024a) encounter severe inefficiencies when processing ultra-long visual token sequences. The quadratic complexity of self-attention (Vaswani et al., 2017) causes GPU memory usage to increase dramatically, resulting in out-of-memory (OOM) errors or degraded performance. Existing approaches tend to mitigate this issue by pruning (Huang et al., 2024; Li et al., 2025; Ye et al., 2025a) or compressing visual tokens (Ye et al., 2025b; Yang et al., 2025b), but such strategies inevitably discard fine-grained information that is essential for detail-sensitive tasks. This raises an important question: ***How can we extend a model's ability to process long visual sequences without destroying the integrity of the original visual information?*** In other words, can we design a mechanism that allows a model to handle large-scale visual inputs more efficiently while simultaneously enhancing its ability to focus on fine visual details?

We argue that the fundamental challenge arises from the extreme length of visual token sequences produced by the visiontower module (Radford et al., 2021; Zhai et al., 2023), which makes dense full attention computationally expensive. Interestingly, the NLP community has recently made significant

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.

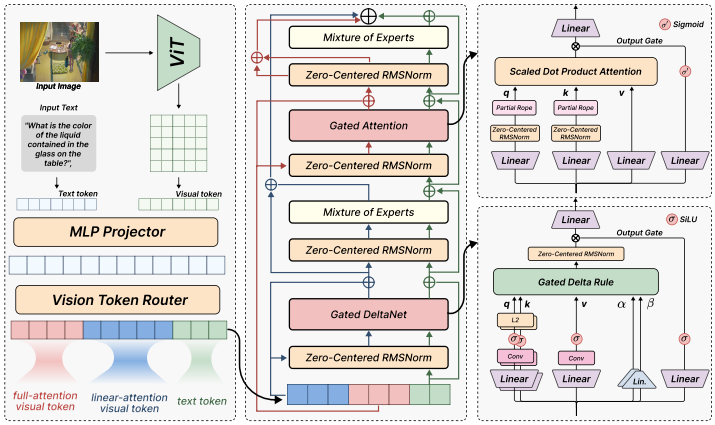


Figure 1: Overall pipeline of **RouteVLM-NEXT**. A Qwen3 ViT encodes the image into visual tokens, which are projected and concatenated with text tokens before entering the Vision Token Router. The router dynamically sends visual tokens to either the *Full Attention* or *Gated DeltaNet Linear* branch, while text tokens always use full attention. Each transformer block follows the Qwen3-Next-80B-A3B-Instruct design with two MoE layers surrounding a Hybrid Attention Block. The fused outputs enable efficient long-context processing while preserving fine-grained visual reasoning.

progress in overcoming similar challenges in long-text modeling through hybrid attention designs that combine full attention with more efficient mechanisms such as State-Space Models (SSMs) (Gu et al., 2022; Peng et al., 2023; Dao & Gu, 2024) or linear attention (Katharopoulos et al., 2020; Yang et al., 2025d). These approaches preserve the expressiveness of full attention while achieving linear or subquadratic scaling, offering a valuable blueprint for multimodal extensions. Inspired by this line of research, we argue that the most principled solution for long-context visual processing is a *hybrid attention* framework, guided by a mechanism that allows the model to "choose" what to focus on more carefully. By dynamically allocating high-fidelity computational resources to the most informative regions of the visual input, a model can simultaneously achieve efficiency, scalability, and strong semantic fidelity. To this end, we introduce **RouteVLM-NEXT**, a VLM with router-guided hybrid attention atop the **Qwen3-Next-80B-A3B-Instruct** (Yang et al., 2025a; Team, 2025b) LLM architecture. The centerpiece of our model is a learnable router that directs visual tokens to one of two pathways: a standard **full attention** branch for capturing intricate details in critical regions, and a highly efficient **gated linear attention** (Gated DeltaNet) branch for processing contextual or background information. This allows the model to manage its computational budget by controlling the ratio of tokens in each branch (Fig. 1). Our model is trained using public datasets and follows an LLaVA-style training phase.

Our comprehensive experiments validate the superiority of RouteVLM-NEXT. We demonstrate significant performance gains on challenging long-sequence benchmarks for video processing and general image understanding, including **MMMU** (Yue et al., 2024) and **LVBench** (Wang et al., 2025). Furthermore, our model excels at detail-oriented tasks, as shown by its strong performance on benchmarks like **V* Bench** (Wu & Xie, 2023), which require localizing and reasoning about fine visual cues. Critically, our analysis reveals that the router learns to intelligently allocate its attention, focusing the full-attention mechanism on semantically salient regions such as text, objects, and actions. This confirms that our model effectively utilizes its hybrid attention to "look" at the most critical parts of an image or video, striking an optimal balance between computational efficiency and excellent performance.

2 Related Work

The Evolution of Vision-Language Models. The landscape of AI (LeCun et al., 2015) has been reshaped by the success of LLMs (Zhao et al., 2025; Minaee et al., 2025), which are predominantly based on the Transformer architecture (Vaswani et al., 2017). This success has naturally been extended into the multimodal domain, giving rise to powerful VLMs (Zhang et al., 2024; Yin et al., 2024). The dominant paradigm, popularized by seminal works like the LLaVA family (Liu et al., 2023), involves augmenting a pre-trained LLM with a visual perception backbone (Bertasius et al., 2021; Chen et al., 2024a), typically a Vision Transformer (ViT) (Dosovitskiy et al., 2021) from

models like CLIP (Radford et al., 2021). Visual features are extracted, projected into the LLM's embedding space via a simple MLP, and treated as a sequence of "visual tokens." This approach has proven remarkably effective, leading to models with impressive capabilities in visual question answering, image captioning, and fine-grained reasoning. However, as these models are applied to higher-resolution images or longer videos, this simple projection strategy generates excessively long visual token sequences, directly exposing the quadratic computational bottleneck of the underlying Transformer architecture, a core challenge our work aims to resolve.

Strategies for Handling Long Visual Sequences. The challenge of processing long visual token sequences has been widely recognized, leading to a variety of mitigation strategies. A primary line of work focuses on reducing the number of tokens before they enter the LLM. These methods can be broadly categorized into pruning and compression. *Pruning*-based approaches (Alvar et al., 2025; Zhang et al., 2025) aim to discard less informative tokens, using techniques like learnable samplers or attention-based scoring to select a salient subset of the original visual tokens. While effective at reducing length, these methods risk irreversibly discarding crucial fine-grained information. *Compression*-based approaches (Ye et al., 2025b; Wang et al., 2024), on the other hand, seek to merge multiple visual tokens into a more compact representation, often employing techniques like resampling, learnable pooling layers, or cross-attention mechanisms. This also reduces sequence length but can blur or average out important local details.

Efficient Attention and Sequence Modeling Architectures. Concurrent to efforts in token reduction, another significant research thrust has focused on fundamentally improving the efficiency of the attention mechanism itself. *Linear Attention* methods (Katharopoulos et al., 2020; Yang et al., 2025d; Dong et al., 2024) represent a major category, which reformulate attention to avoid the explicit computation of the quadratic-cost attention matrix, achieving linear time and constant memory complexity. While early variants often lagged behind standard softmax attention in performance, recent innovations such as Gated DeltaNet (Yang et al., 2025c), have substantially closed this performance gap, making them viable and powerful alternatives. In a parallel vein, *State Space Models (SSMs)* like Mamba (Gu & Dao, 2024) have emerged as a highly competitive alternative to attention, also offering linear-time scaling by formulating sequence modeling through a recurrent state. Although Gated DeltaNet is an SSM-style recurrent model, its linear-time complexity lets it play the role of linear attention in hybrid architectures. Our work builds upon the advancements in linear attention, leveraging its efficiency for one of the pathways.

3 METHODOLOGY

Built upon the Qwen3-Next-80B-A3B-Instruct backbone, our RouteVLM-NEXT integrates a dynamic token routing mechanism within its core transformer blocks to allocate computational resources effectively. This approach preserves fine-grained details for critical regions while maintaining near-linear complexity for long-context modeling.

3.1 OVERALL ARCHITECTURE

Our RouteVLM-NEXT consists of three main components: (1) a visual encoder, (2) a learnable Vision Token Router, and (3) a decoder composed of blocks with parallel processing pathways. The architecture strictly adheres to a Pre-LayerNorm (Pre-LN) paradigm. Within each decoder block, the Vision Token Router directs visual tokens to one of two parallel streams for processing:

- One stream processes "full-attention" visual tokens, first through a **Gated Attention** sub-layer, and then through a **Mixture of Experts (MoE)** sub-layer.
- The other stream concurrently processes "linear-attention" visual tokens, first through a **Gated DeltaNet** sub-layer, and subsequently through a **Mixture of Experts (MoE)** sub-layer.

Each of these sub-layers (Gated Attention, Gated DeltaNet, and MoE) is encapsulated within its own Pre-LN residual connection. For any given sub-layer, the computation follows the form: $\mathbf{X}_{\text{out}} = \mathbf{X}_{\text{in}} + \text{SubLayer}(\text{Zero-Centered RMSNorm}(\mathbf{X}_{\text{in}}))$. The outputs from the parallel streams are then combined to be fed into the next decoder block.

3.2 VISUAL TOKENIZATION

To process visual inputs, we follow the methodology of LLaVA-NEXT. We employ a pre-trained ViT from Qwen3-VL-30B-A3B as our vision tower to extract a sequence of patch-level feature vectors, $\mathbf{Z}_{\text{vision}}$. These features are then projected into the LLM's embedding dimension, d, using a two-layer MLP with a GELU activation function.

$$\mathbf{X}_{\text{vision}} = \text{MLP}(\mathbf{Z}_{\text{vision}}) = W_2\,(\text{GELU}(W_1\mathbf{Z}_{\text{vision}} + b_1)) + b_2. \tag{1}$$

The resulting sequence of visual tokens is concatenated with the text token embeddings to form the initial input $\mathbf{X}$ for the transformer backbone.

3.3 VISION TOKEN ROUTER

For each token in the normalized input sequence $\mathbf{H} = \text{LayerNorm}(\mathbf{X})$, the router computes a gating probability p_i for its assignment to the Full Attention branch:

$$p_i = \sigma\left(W_2\,\text{SiLU}\big(W_1\mathbf{h}_i + b_1\big) + b_2\right), \tag{2}$$

where $\mathbf{h}_i$ is the normalized token feature. During training, this probability is converted into a binary mask $m_i \in \{0, 1\}$ using a Gumbel-Sigmoid estimator, and we deterministically set $m_i = 1$ for all text tokens. At inference time, we drop the Gumbel noise and use a simple thresholding rule $m_i = \mathbb{I}[p_i \geq \tau]$ with $\tau = 0.5$ for vision tokens, while text tokens always take $m_i = 1$. We use a single router MLP whose parameters are shared across all transformer blocks. Only the block-specific normalized inputs $H^{(\ell)}$ differ. Hence, routing decisions remain layer-dependent even with shared weights.

Complexity analysis. Let L denote the total sequence length (text + vision) and let $B \in (0, 1]$ be the per-block budget on the fraction of tokens routed to the full-attention branch. In each hybrid block, the Gated Full Attention branch computes attention for at most $|I_{\text{full}}| \leq BL$ query tokens over the full key/value context of length L. Ignoring constant factors and the number of heads, the time and memory costs of this branch are

$$\text{Time}_{\text{full}} = O(BL^2 d_k) + O(Ld_{\text{model}}^2), \tag{3}$$

$$\text{Mem}_{\text{full}} = O(BL^2), \tag{4}$$

where the first term comes from the $|I_{\text{full}}| \times L$ attention matrix and the second term corresponds to the linear projections and MLP layers.

For the Gated DeltaNet branch, the convolutional feature extractor and recurrent update scale linearly with L:

$$\text{Time}_{\text{lin}} = O(Ld_{\text{lin}}^2) + O(Ld_{\text{lin}}d_k), \tag{5}$$

$$\text{Mem}_{\text{lin}} = O(Ld_{\text{lin}}^2), \tag{6}$$

where d_{lin} is the hidden dimension used in the DeltaNet path. The overall hybrid block therefore has

$$\text{Time}_{\text{hybrid}} = O(BL^2 d_k) + O(Ld_{\text{lin}}^2), \tag{7}$$

$$\text{Mem}_{\text{hybrid}} = O(BL^2) + O(Ld_{\text{lin}}^2). \tag{8}$$

In our experiments we fix B to a small constant (e.g., $B = 0.2$) in all layers and employ fused, chunked attention kernels from the Qwen3 implementation (similar to FlashAttention-style kernels). Under this fixed-budget setting, the empirical peak memory and latency in Fig. 2 grow approximately linearly with L over the range $L \in [2\text{k}, 64\text{k}]$. However, the asymptotic complexity still contains the $O(BL^2)$ term, so our method should be understood as *substantially mitigating* rather than completely eliminating the quadratic cost of full attention.

3.4 HYBRID ATTENTION BLOCK

Given the routing mask $m \in \{0, 1\}^L$, we first apply zero-centered RMSNorm to obtain the shared normalized input

$$H = \text{ZRMS}(X). \tag{9}$$

All computations in both sub-layers (Gated Attention and Gated DeltaNet) operate on this shared H. The visual tokens are then processed in two parallel branches.

Gated Full Attention Branch. Given the shared normalized input H, the full-attention branch applies linear projections to obtain

$$\mathbf{Q} = \mathbf{H}W_Q, \quad \mathbf{K} = \mathbf{H}W_K, \quad \mathbf{V} = \mathbf{H}W_V. \tag{10}$$

Let $I_{\text{full}} = \{i \mid m_i = 1\}$ be the index set of tokens served by the full-attention branch. We apply partial RoPE on the first d_{rope} dimensions of Q and K to obtain Q^ρ, K^ρ, and compute standard scaled dot-product attention over the full key/value context:

$$Y_{\text{full}} = \text{softmax}\left(\frac{Q^\rho[I_{\text{full}}](K^\rho)^\top}{\sqrt{d_k}} + M\right)V. \tag{11}$$

A learnable output gate is then applied element-wise:

$$G_{\text{full}} = \sigma(W_g^{\text{full}} H[I_{\text{full}}]), \quad Y_{\text{full}} \leftarrow G_{\text{full}} \odot Y_{\text{full}}. \tag{12}$$

Gated DeltaNet Linear Branch. Following Gated DeltaNet (Yang et al., 2025c), we build a lightweight feature extractor and a gated Δ-rule state update. From the same normalized input $\mathbf{H}$ as in Eq. (11), a depthwise short 1D convolution with SiLU is applied to produce preliminary features $(\tilde{\mathbf{q}}, \tilde{\mathbf{k}}, \tilde{\mathbf{v}}) = \Psi(\mathbf{H})$. We L2-normalize queries and keys:

$$\mathbf{q}_t = \tilde{\mathbf{q}}_t/\|\tilde{\mathbf{q}}_t\|_2, \quad \mathbf{k}_t = \tilde{\mathbf{k}}_t/\|\tilde{\mathbf{k}}_t\|_2, \quad \mathbf{v}_t = \tilde{\mathbf{v}}_t. \tag{13}$$

Two gates $\alpha_t, \beta_t \in (0, 1)$ are produced by a small network (e.g., linear + sigmoid) from $\mathbf{h}_t$:

$$\alpha_t = \sigma(W_\alpha \mathbf{h}_t + b_\alpha), \qquad \beta_t = \sigma(W_\beta \mathbf{h}_t + b_\beta). \tag{14}$$

For each head, the recurrent *gated-Δ update* maintains a state $\mathbf{S}_t \in \mathbb{R}^{d_v^{\text{lin}} \times d_k^{\text{lin}}}$:

$$S_t = S_{t-1}\Big(\alpha_t(I - \beta_t\, k_t k_t^\top)\Big) + \beta_t\, v_t k_t^\top, o_t = S_t\, q_t \tag{15}$$

and we *only collect* outputs at $t \in I_{\text{lin}} = \{i \mid m_i{=}0\}$: $\mathbf{Y}_{\text{lin}}[t] = \mathbf{o}_t$ if $t \in I_{\text{lin}}$ else 0. A final output gate is applied (SiLU or sigmoid):

$$\mathbf{G}^{\text{lin}} = \phi(W_g^{\text{lin}}\, \mathbf{h}_t), \qquad \mathbf{Y}_{\text{lin}} \leftarrow \mathbf{G}^{\text{lin}} \odot \mathbf{Y}_{\text{lin}}. \tag{16}$$

Notes. (i) Both sub-layers operate on the same normalized hidden states H, but are parameterized differently: the Gated Attention sub-layer uses linear projections (W_Q, W_K, W_V), whereas the Gated DeltaNet sub-layer uses the convolutional extractor $\Psi(\cdot)$ to obtain (q_t, k_t, v_t). (ii) Training can be parallelized over chunks using the UT/WY-style matrix form of Gated DeltaNet, without changing the recurrent inference rule above.

Fusion. Let $\mathbf{X}''_{\text{full}}$ and $\mathbf{X}''_{\text{lin}}$ denote the outputs of the full-attention and DeltaNet streams after their respective MoE sub-layers and residual connections (the full per-block forward pass is given in Algorithm 1, Appendix A.4). We then merge the two processed streams back into the original token order by the router mask:

$$\mathbf{X}_{\text{out}}[i] = \begin{cases} \mathbf{X}''_{\text{full}}[\pi_{\text{full}}(i)], & \text{if } m_i = 1, \\ \mathbf{X}''_{\text{lin}}[\pi_{\text{lin}}(i)], & \text{if } m_i = 0, \end{cases} \tag{17}$$

where π_{full} and π_{lin} map the original token index i to its local index in the corresponding stream. This "scatter" operation completes the hybrid block and produces $\mathbf{X}_{\text{out}}$ for the next layer.

Mixture-of-Experts (MoE) FFN. We employ sparse MoE layers after each attention sub-layer in both streams. For a generic stream $b \in \{\text{full}, \text{lin}\}$, let $\mathbf{X}'_b$ be the input to its MoE block and $\mathbf{u}_b = \text{ZRMS}(\mathbf{X}'_b)$ the pre-activated hidden state. A router produces logits $a_e^{(b)} = W_r^{(b)}\mathbf{u}_b + b_r^{(b)}$ over E experts; we keep the Top-k (with optional one shared expert *within the stream*) and compute normalized gates on the selected set:

$$\mathcal{E}_k^{(b)} = \text{TopK}(a^{(b)}, k), \tag{18}$$

$$r_e^{(b)} = \begin{cases} \dfrac{\exp(a_e^{(b)})}{\sum_{j \in \mathcal{E}_k^{(b)}} \exp(a_j^{(b)})}, & \text{if } e \in \mathcal{E}_k^{(b)}, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

The MoE output for stream b is

$$\mathrm{MoE}_b(\mathbf{u}_b) = \sum_{e \in \mathcal{E}_k^{(b)}} r_e^{(b)} \, \mathrm{FFN}_e^{(b)}(\mathbf{u}_b), \tag{20}$$

$$\mathbf{Y}_{\text{moe}}^{(b)} = \mathrm{Drop}\big(\mathrm{MoE}_b(\mathbf{u}_b)\big), \tag{21}$$

$$\mathbf{X}_b'' = \mathbf{X}_b' + \mathbf{Y}_{\text{moe}}^{(b)}. \tag{22}$$

In our implementation, both streams use the same (E, k) configuration but have separate parameters $\{W_r^{(b)}, b_r^{(b)}, \mathrm{FFN}_e^{(b)}\}$, forming two parallel MoE modules.

Config. In Qwen3-Next-Instruct-80B-A3B, E=512, k=10, with 1 shared expert; only $\sim$3B parameters are active per step.

3.5 Training Objectives

We adopt a minimal two-term objective: a standard text modeling loss and a routing-usage regularizer that softly encourages visual tokens to prefer the linear branch while respecting a target full-attention budget.

Text loss. Let $\mathcal{T}$ be the set of indices of target text tokens in the output sequence. We use the standard autoregressive language-modeling objective over these tokens:

$$\mathcal{L}_{\text{text}} = -\frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \log p_\theta\big(y_i \mid y_{<i}, \mathbf{X}, \mathbf{V}\big), \tag{23}$$

where y_i is the i-th target token, $y_{<i}$ are all previous tokens, and $(\mathbf{X}, \mathbf{V})$ denote the input text and visual tokens. This is the standard cross-entropy loss used to train the underlying autoregressive VLM.

Routing-Usage Regularizer. For each transformer layer ℓ, let $\mathcal{V}_\ell$ be the set of visual-token indices (text tokens are excluded since they are deterministically routed to the full branch), and $p_i^{(\ell)} \in (0, 1)$ be the router's probability of sending token $i \in \mathcal{V}_\ell$ to the *full* branch. Define the layer-wise expected full-branch usage as

$$\bar{p}_\ell = \frac{1}{|\mathcal{V}_\ell|} \sum_{i \in \mathcal{V}_\ell} p_i^{(\ell)}. \tag{24}$$

Given a target budget $B \in [0, 1]$ (fraction of visual queries allowed to use full attention), we penalize only when usage exceeds B via a squared hinge:

$$\mathcal{L}_{\text{usage}} = \frac{1}{L_{\text{blk}}} \sum_{\ell=1}^{L_{\text{blk}}} \big[\bar{p}_\ell - B\big]_+^2, \quad \text{where} \quad [x]_+ = \max(x, 0). \tag{25}$$

Overall Objective. The final loss is

$$\mathcal{L} = \mathcal{L}_{\text{text}} \; + \; \lambda_{\text{usage}} \, \mathcal{L}_{\text{usage}}. \tag{26}$$

Practical Notes (not part of the loss). During training, we compute $\mathcal{L}_{\text{usage}}$ with probabilities $p_i^{(\ell)}$ (rather than sampled masks) to maintain differentiability. Budget B or the weight λ_{usage} can be scheduled/annealed to avoid early collapse and to match deployment targets. We linearly ramp λ_{usage} from 0 to $\lambda_{\max} = 0.1$ over the first 20% of training steps and keep it fixed afterwards.

4 Experiments

Experimental setup. To evaluate RouteVLM-NEXT, we compare against leading open-source and proprietary VLMs on 15 benchmarks grouped into **General VQA**, **Reasoning / STEM & Puzzle**, **OCR & Chart Understanding**, and **Others / Long Vision Tasks**. The suite includes MMStar (Chen et al., 2024b), MMBench (Liu et al., 2024b), ScienceQA (Lu et al., 2022), RealWorldQA (xAI Corp., 2024), MathVista (Lu et al., 2024), MMMU (Yue et al., 2024), MMMU-Pro (Yue et al., 2025), DocVQA (Mathew et al., 2021), ChartQA (Masry et al., 2022), OCRBench (Liu et al., 2024c),

Table 1: Comparative results across multimodal benchmarks. We include **Qwen3-Next-80B (Base)** as a controlled baseline trained on identical data without the hybrid routing mechanism. Bold values denote the best performance within each row. The final column (highlighted) corresponds to **RouteVLM-NEXT**.

	Benchmark	LLaVA-OV-1.5 8B	Qwen2.5-VL 7B	LLaVA-OV 7B	Qwen2.5-VL 72B	Qwen3-VL 30B-A3B	PruneVid (LLaVA-OV)	Qwen3-Next 80B	RouteVLM-NEXT
General VQA	MMStar	67.7	62.5	61.7	70.2	72.1	57.6	71.0	**73.9**
	MMBench	84.1	83.4	82.5	88.6	87.0	73.4	86.5	**89.2**
	ScienceQA	95.0	88.8	90.0	74.8	96.7	84.1	96.0	**98.2**
	RealWorldQA	68.1	68.5	66.3	75.7	73.7	65.2	71.2	**75.9**
	Avg.	78.7	75.8	75.1	77.3	82.4	70.0	81.2	**84.3**
Reasoning / STEM & Puzzle	MathVista_mini	69.6	68.6	58.5	74.8	80.1	52.9	78.5	**82.3**
	MMMU_val	55.4	51.3	48.8	70.2	74.2	42.7	72.8	**75.8**
	MMMU-Pro_standard	37.4	36.3	28.0	51.1	60.4	31.8	60.1	**62.2**
	Avg.	54.1	52.1	45.1	65.4	71.6	42.5	70.5	**73.4**
OCR & Chart Understanding	DocVQA	95.0	94.9	87.2	**96.4**	95.0	85.9	92.0	93.6
	ChartQA	**86.5**	84.1	80.0	79.8	85.5	73.5	82.1	83.8
	OCRBench	82.9	84.2	62.1	63.7	**90.3**	78.1	86.5	88.2
	Avg.	88.1	87.7	76.4	80.0	**90.3**	79.1	86.9	88.5
Others / Long Vision Tasks	V*	78.0	77.0	72.3	69.1	81.2	65.2	82.4	**86.6**
	CountBench	82.8	86.4	78.4	**93.6**	72.3	83.8	87.5	90.2
	LVBench	46.7	43.8	42.1	47.3	62.5	42.7	60.1	**68.9**
	VideoMME	72.1	69.8	62.3	73.3	74.5	58.6	73.2	**77.8**
	Avg.	69.9	69.2	63.8	70.8	77.0	62.6	75.8	**80.9**

V* Bench (Wu & Xie, 2023), CountBench (Paiss et al., 2023), LVBench (Wang et al., 2025), and VideoMME (Fu et al., 2025). We compare with small open-source baselines such as LLaVA-OV-1.5 8B (An et al., 2025), Qwen2.5-VL 7B (Bai et al., 2025), and LLaVA-OV 7B (Li et al., 2024); Qwen2.5-VL 72B (Bai et al., 2025); Qwen3-VL 30B-A3B (Bai et al., 2025; Team, 2025b); and PruneVid (Huang et al., 2024) with an LLaVA-OneVision backbone. To isolate the architectural contribution, we introduce **Qwen3-Next-80B (Base)**, which uses the identical pre-trained ViT, LLM backbone, data, and schedule as RouteVLM-NEXT. The only difference is that the Base model applies standard full attention to all visual tokens. Additional matched token-reduction baselines, dense-image evaluations, and a cross-backbone transfer check are reported in Appendix A. All experiments are conducted on NVIDIA A100.

Experimental results in Table 1 show that RouteVLM-NEXT achieves the highest average scores in General VQA (84.3), Reasoning (73.4), and Long Vision (80.9). In OCR & Chart Understanding, RouteVLM-NEXT remains competitive (88.5), although Qwen3-VL 30B-A3B reaches 90.3; we therefore rely on the controlled Qwen3-Next-80B (Base) and the matched token-reduction comparisons in Appendix A.1 for causal attribution. RouteVLM-NEXT consistently outperforms its direct parent baseline across all four categories, including General VQA (84.3 vs. 81.2), Reasoning (73.4 vs. 70.5), and Long Vision Tasks (80.9 vs. 75.8). This validates that the router does not merely compress information, but reallocates computation toward salient visual regions. For detailed router analysis on V*, see Sec. 5.

Additional validation. Appendix A further stress-tests baseline matching and generalization. Under the identical Qwen3-Next-80B setting, RouteVLM-NEXT outperforms SparseVLM, PruMerge+, and VisPruner across all four aggregate categories. It also improves over Qwen3-VL-30B-MoE on dense and high-resolution image settings, including VISUALOVERLOAD (68.3 vs. 65.8) and HRScene (55.7 vs. 52.3). Finally, a Nemotron-H-8B + SigLIP-2-g transfer check shows that the routing mechanism can be adapted to a Mamba2-based stream, while broader cross-backbone scaling remains future work.

5 Analysis of the Routing Policy

Efficiency and Scalability Analyses. We compare RouteVLM-NEXT with a standard full-attention baseline that forces all vision tokens through the full-attention branch. The experiment varies input length by changing the number of sampled video frames, producing visual token sequences of 2k, 4k, 8k, 16k, 32k, and 64k. For each configuration, we measure LVBench performance, peak GPU memory, end-to-end inference latency, and GFLOPs. Unless otherwise specified, latency is the average wall-clock time per LVBench example with batch size 1 on a NVIDIA A100 using bf16 mixed precision. Appendix A.2 further evaluates dense and high-resolution single-image settings to complement the video-frame scalability study.

Figure 2 illustrates the superior scalability of RouteVLM-NEXT. While the Full Attention baseline suffers from quadratic $\mathcal{O}(N^2)$ costs, our model achieves near-linear scaling. At 8k tokens, despite comparable accuracy, RouteVLM-NEXT reduces memory by $\sim 2\times$ and latency by over an order of magnitude. Crucially, our model extends effectively to 64k tokens—a regime where the baseline fails—sustaining high performance (68.9%) with only minor degradation after the 8k-16k peak.

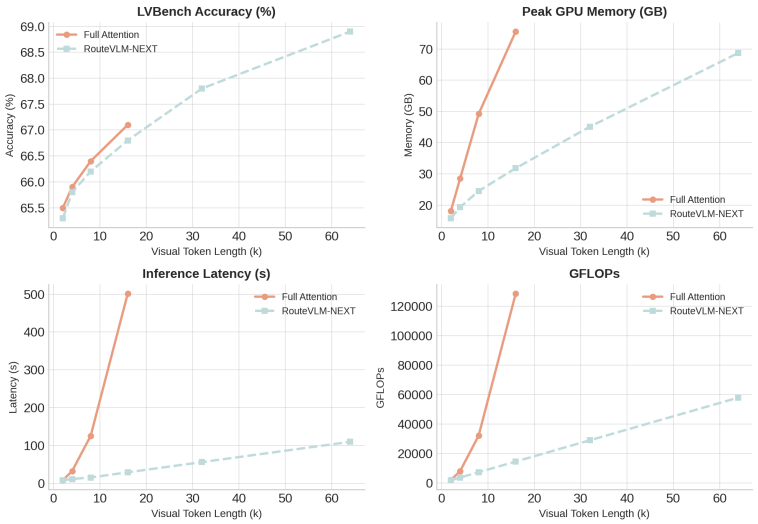


Figure 2: Direct scaling comparison between RouteVLM-NEXT and a Full Attention baseline. Visual token length is controlled by video frame sampling.



Figure 3: Visualization of the token router's decisions for the V* Bench example. Regions processed by the "Full Attention Branch" are not masked and are clearly visible; the others are regions processed by the "Gated DeltaNet Branch".

Qualitative Analysis of the Token Router. To qualitatively understand *how* RouteVLM-NEXT maintains strong performance with high efficiency, we visualize its token-routing behavior on examples from V* Bench, which targets fine-grained visual reasoning. As shown in Fig. 3, regions routed to the computationally intensive **full-attention branch** remain visible, while the majority of regions—handled by the efficient **Gated DeltaNet Branch**—are faded out. The router adopts a clear, semantics-aware allocation strategy. For fine-grained "visual search" queries, it selectively routes small or informative regions—e.g., the doll's company logo, text on the black mug ("Rubbermaid", "SIGGRAPH"), the child's shoes, or the red book title—to the full-attention branch. Meanwhile, broad, low-value context such as walls, desks, and background objects is handled by the linear branch. This dynamic, content-aware routing demonstrates that the model is not merely approximating attention, but actively learning *what* to focus on.

6 Ablation Study

To isolate and verify the contributions of the core components within our RouteVLM-NEXT architecture, we conduct ablations on V* Bench, as its emphasis on fine-grained detail recognition and visual search makes it an ideal testbed for evaluating the effectiveness of our token routing mechanism. Our full model reaches 86.6% on V* Bench; results are summarized in Tables 2 and 3.

Impact of the Hybrid Attention Mechanism. Our first two ablations validate the fundamental premise of our hybrid design. By removing the router and forcing all tokens through **full attention**, we observe a marginal 0.4% performance increase to 87.0%, establishing a practical upper bound. However, this comes at the cost of a crippling 8.3× increase in latency. Conversely, using only **linear attention** results in a catastrophic 8.4% drop in accuracy, confirming its inadequacy for

Table 2: Ablation study on the V* Bench. We analyze the impact of the router and the usage regularizer. Our full model balances high accuracy and low latency.

Model Config	V* Acc. (%)	Latency (s)	Full Attn. (%)
RouteVLM-NEXT (Full)	**86.6**	**15.2**	**~18**
1. w/o Router (All Full Attn.)	87.0 (+0.4)	125.4 (8.3×)	100
2. w/o Router (All Linear)	78.2 (-8.4)	9.7 (0.6×)	0
3. w/o Usage Regularizer	84.9 (-1.7)	32.1 (2.1×)	61

Table 3: Ablation study of the full-attention budget (B) on V* Bench. Our default setting of $B = 0.20$ is validated as the optimal point for performance and efficiency.

Budget (B)	V* Bench Acc. (%)	Actual Full Attn. Usage (%)	Inference Latency (s)
0.05	82.1	4.9	13.0
0.10	85.4	9.8	14.0
0.20 (Default)	**86.6**	**18.0**	**15.2**
0.30	86.7	28.5	21.5
0.60	86.9	57.1	56.8
1.00 (All Full Attn.)	87.0	100.0	125.4

detail-sensitive tasks, despite being the fastest configuration. These results unequivocally demonstrate that our hybrid approach is critical, achieving performance nearly identical to a full-attention model while being over 8× faster.

Importance of the Usage Regularizer. To evaluate the role of the budget-enforcing loss term, we remove the usage regularizer $\mathcal{L}_{\text{usage}}$. In this case, the model partially collapses toward a trivial strategy of overusing the full-attention branch: the actual full-attention usage rises from ∼18% to 61%, and the latency more than doubles (15.2 s → 32.1 s), while accuracy drops by 1.7 points. This does not mean that full attention is harmful in isolation; rather, removing the constraint disrupts the intended division of labor between the full-attention and DeltaNet branches, causing many medium-value background tokens to consume the expensive branch and weakening specialization. The regularizer is therefore part of the routing objective, not an incidental hyperparameter.

Visual Token Budget. Table 3 varies B on V* Bench. The routing budget B controls the fraction of visual tokens sent to full attention. Our default setting of $B = 0.20$ strikes an optimal balance. While increasing the budget further yields marginal accuracy gains (e.g., from 86.6% to 87.0%), it comes at a disproportionately high computational cost, with latency increasing by over 8×. This confirms that $B = 0.20$ achieves nearly peak performance while preserving the core efficiency advantages of our hybrid architecture.

Training Details. We train only on public multimodal instruction data, using curated subsets from the LLaVA-OV-1.5 (An et al., 2025) and BLIP-3 (Xue et al., 2025) mixtures; no proprietary or private data is used. Following the LLaVA-OV paradigm, training has two stages. First, we freeze the LLM backbone and optimize the Qwen3-VL-30B-A3B ViT, two-layer MLP projector, and Vision Token Router for visual-language alignment. Second, after stable alignment, we unfreeze all components and perform end-to-end SFT on large-scale visual instruction data. Stage-wise details are provided in Appendix A.5.

7 Conclusion

We introduced RouteVLM-NEXT, a scalable VLM that uses router-guided hybrid attention to allocate full attention to information-dense visual tokens while processing broader context through an efficient linear pathway. Experiments across diverse benchmarks show strong performance and graceful scaling where full-attention baselines degrade.

Limitations and impact. RouteVLM-NEXT mitigates, but does not remove, the quadratic term of full attention because selected visual queries still attend to the full context. Our controlled scalability study covers up to 64k visual tokens, and the fixed default budget can be suboptimal for extremely dense documents or scenes. The primary positive impact is lower memory and latency for long-context multimodal inference; routing behavior should be audited before high-stakes deployment (Appendix A.6).

AI USE STATEMENT

Large language models were used as general-purpose assistants for polishing the writing of this manuscript and for LaTeX formatting (including adapting the paper to the ICLR template). They were not used to generate research ideas, design the method, run experiments, or produce the reported results. All AI-assisted text was reviewed and edited by the authors, who take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

Section 3 specifies the router, the hybrid attention block, the training objectives, and the budget schedule; Section 4 describes the public training mixtures, controlled baselines, benchmark groups, and latency protocol (NVIDIA A100, bf16, batch size 1); Appendix A.5 lists the routing configuration, and Appendix A.1 gives the matched token-reduction baselines. Algorithm 1 (Appendix A.4) gives the per-block forward pass. We will release code, training configurations, and model checkpoints upon publication.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models, 2025. URL https://arxiv.org/abs/2503.02175.

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Chunsheng Wu, Huajie Tan, Chunyuan Li, Jing Yang, Jie Yu, Xiyao Wang, Bin Qin, Yumeng Wang, Zizhen Yan, Ziyong Feng, Ziwei Liu, Bo Li, and Jiankang Deng. Llava-onevision-1.5: Fully open framework for democratized multimodal training, 2025. URL https://arxiv.org/abs/2509.23661.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. URL https://arxiv.org/abs/2308.12966.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding?, 2021. URL https://arxiv.org/abs/2102.05095.

Jieneng Chen, Qihang Yu, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Vitamin: Designing scalable vision models in the vision-language era, 2024a. URL https://arxiv.org/abs/2404.02132.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024b.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality, 2024. URL https://arxiv.org/abs/2405.21060.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex attention: A programming model for generating optimized attention kernels, 2024. URL https://arxiv.org/abs/2412.05496.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis, 2025. URL https://arxiv.org/abs/2405.21075.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022. URL https://arxiv.org/abs/2111.00396.

Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models, 2024. URL https://arxiv.org/abs/2412.16117.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention, 2020. URL https://arxiv.org/abs/2006.16236.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer, 2024. URL https://arxiv.org/abs/2408.03326.

Duo Li, Zuhao Yang, and Shijian Lu. Todre: Visual token pruning via diversity and task awareness for efficient large vision-language models, 2025. URL https://arxiv.org/abs/2505.18757.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL https://arxiv.org/abs/2304.08485.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024a. URL https://arxiv.org/abs/2310.03744.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player?, 2024b. URL https://arxiv.org/abs/2307.06281.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12), December 2024c. ISSN 1869-1919. doi: 10.1007/s11432-024-4235-6. URL http://dx.doi.org/10.1007/s11432-024-4235-6.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering, 2022. URL https://arxiv.org/abs/2209.09513.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts, 2024. URL https://arxiv.org/abs/2310.02255.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning, 2022. URL https://arxiv.org/abs/2203.10244.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images, 2021. URL https://arxiv.org/abs/2007.00398.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2025. URL https://arxiv.org/abs/2402.06196.

OpenAI. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten, 2023. URL https://arxiv.org/abs/2302.12066.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Jiaju Lin, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanislaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. Rwkv: Reinventing rnns for the transformer era, 2023. URL https://arxiv.org/abs/2305.13048.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI technical report, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Gemini Team. Gemini: A family of highly capable multimodal models, 2025a. URL https://arxiv.org/abs/2312.11805.

Qwen Team. Qwen3 technical report, 2025b. URL https://arxiv.org/abs/2505.09388.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Han Wang, Yuxiang Nie, Yongjie Ye, Deng GuanYu, Yanjie Wang, Shuai Li, Haiyang Yu, Jinghui Lu, and Can Huang. Dynamic-vlm: Simple dynamic visual token compression for videollm, 2024. URL https://arxiv.org/abs/2412.09530.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Lvbench: An extreme long video understanding benchmark, 2025. URL https://arxiv.org/abs/2406.08035.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. *arXiv preprint arXiv:2312.14135*, 2023.

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

Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, Shrikant Kendre, Jieyu Zhang, Shaoyen Tseng, Gustavo A Lujan-Moreno, Matthew L Olson, Musashi Hinck, David Cobbley, Vasudev Lal, Can Qin, Shu Zhang, Chia-Chih Chen, Ning Yu, Juntao Tan, Tulika Manoj Awalgaonkar, Shelby Heinecke, Huan Wang, Yejin Choi, Ludwig Schmidt, Zeyuan Chen, Silvio Savarese, Juan Carlos Niebles, Caiming Xiong, and Ran Xu. xgen-mm (blip-3): A family of open large multimodal models, 2025. URL https://arxiv.org/abs/2408.08872.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. Qwen2.5-1m technical report. *arXiv preprint arXiv:2501.15383*, 2025a.

Chenyu Yang, Xuan Dong, Xizhou Zhu, Weijie Su, Jiahao Wang, Hao Tian, Zhe Chen, Wenhai Wang, Lewei Lu, and Jifeng Dai. Pvc: Progressive visual token compression for unified image and video processing in large vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24939–24949, 2025b.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *The Thirteenth International Conference on Learning Representations*, 2025c. URL https://openreview.net/forum?id=r8H7xhYPwz.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length, 2025d. URL https://arxiv.org/abs/2406.06484.

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 Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025a. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i21.34366. URL https://doi.org/10.1609/aaai.v39i21.34366.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. Voco-llama: Towards vision compression with large language models, 2025b. URL https://arxiv.org/abs/2406.12275.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12), November 2024. ISSN 2053-714X. doi: 10.1093/nsr/nwae403. URL http://dx.doi.org/10.1093/nsr/nwae403.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024. URL https://arxiv.org/abs/2311.16502.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark, 2025. URL https://arxiv.org/abs/2409.02813.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. URL https://arxiv.org/abs/2303.15343.

Hao Zhang, Mengsi Lyu, Chenrui He, Yulong Ao, and Yonghua Lin. Trimtokenator: Towards adaptive visual token pruning for large multimodal models, 2025. URL https://arxiv.org/abs/2509.00320.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey, 2024. URL https://arxiv.org/abs/2304.00685.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2025. URL https://arxiv.org/abs/2303.18223.

A Additional Experiments and Analysis

A.1 Matched Token-Reduction Baselines

To make the baseline comparison more controlled, we adapt three recent token-reduction methods to the same Qwen3-Next-80B backbone, vision tower, data mixture, and training schedule used by RouteVLM-NEXT. Table 4 shows that RouteVLM-NEXT improves over these matched pruning/compression baselines in all four benchmark groups, with the largest gains on OCR & Chart Understanding and Long Vision Tasks.

Table 4: Strictly matched comparison against token-reduction baselines on Qwen3-Next-80B.

Model	Gen. VQA	Reasoning	OCR & Chart	Long Vision
SparseVLM	81.5	71.2	86.7	78.2
PruMerge+	82.3	71.5	86.5	79.2
VisPruner	81.8	71.2	86.3	78.5
RouteVLM-NEXT	**84.3**	**73.4**	**88.5**	**80.9**

A.2 Dense and High-Resolution Inputs

The main scalability experiment varies visual length through densely sampled video frames, which may contain temporal redundancy. We therefore also evaluate low-redundancy, information-dense image settings. RouteVLM-NEXT outperforms the comparable Qwen3-VL-30B-MoE baseline on both dense-scene and high-resolution single-image benchmarks, indicating that the routing benefit is not restricted to redundant video inputs.

Table 5: Dense and high-resolution image evaluation.

Dataset	Scenario	Qwen3-VL-30B-MoE	RouteVLM-NEXT
VISUALOVERLOAD	Dense visual information	65.8	**68.3**
HRScene	High-resolution single image	52.3	**55.7**

A.3 Cross-Backbone Transfer Check

To test whether the routing mechanism is tied to Qwen3-Next and Gated DeltaNet, we conduct an additional portability check using Nemotron-H-8B-Base-8K with a SigLIP-2-g vision tower. Because this backbone uses Mamba2 rather than DeltaNet, we route linear visual tokens into the Mamba2 stream. This experiment is intended as an architecture-transfer sanity check rather than a fully tuned new model-family benchmark.

Table 6: Cross-backbone portability check with a different sequence module and vision tower.

Backbone	Vision Tower	Adaptation	MMMU (val)	LVBench
Nemotron-H-8B-Base-8K	SigLIP-2-g	Route linear visual tokens into Mamba2	45.2	38.7

A.4 Per-Block Forward Pass

Algorithm 1 summarizes the forward pass of a single RouteVLM-NEXT block, including the Pre-LN structure, the two routed streams, and the final scatter back to the original token order.

Algorithm 1 Forward Pass of a RouteVLM-NEXT Block

Require: Input visual tokens $\mathbf{X} \in \mathbb{R}^{L \times d}$
$\mathbf{H} \leftarrow \mathrm{ZeroCenteredRMSNorm}(\mathbf{X})$
$I_{\text{full}}, I_{\text{lin}} \leftarrow \mathrm{Router}(\mathbf{H})$ {Get indices for each branch}
$\mathbf{X}_{\text{full}}, \mathbf{X}_{\text{lin}} \leftarrow \mathbf{X}[I_{\text{full}}], \mathbf{X}[I_{\text{lin}}]$ {Split original input for residuals}
$\mathbf{H}_{\text{full}}, \mathbf{H}_{\text{lin}} \leftarrow \mathbf{H}[I_{\text{full}}], \mathbf{H}[I_{\text{lin}}]$ {Split normalized input for computation}

{**— Stream 1: Gated Attention → MoE —**}
$\mathbf{Y}_{\text{attn}} \leftarrow \mathrm{GatedAttention}(\mathbf{H}_{\text{full}})$ {Full attention mechanism; Using linear Q/K/V projections on H_full}
$\mathbf{X}'_{\text{full}} \leftarrow \mathbf{X}_{\text{full}} + \mathrm{Dropout}(\mathbf{Y}_{\text{attn}})$ {First residual connection}
$\mathbf{H}'_{\text{full}} \leftarrow \mathrm{ZeroCenteredRMSNorm}(\mathbf{X}'_{\text{full}})$ {Pre-Norm for MoE}
$\mathbf{U}_{\text{full}} \leftarrow \mathrm{MoE}_{\text{full}}(\mathbf{H}'_{\text{full}})$
$\mathbf{X}''_{\text{full}} \leftarrow \mathbf{X}'_{\text{full}} + \mathrm{Dropout}(\mathbf{U}_{\text{full}})$ {Second residual connection}
{**— Stream 2: Gated DeltaNet → MoE —**}
$\mathbf{Y}_{\text{delta}} \leftarrow \mathrm{GatedDeltaNet}(\mathbf{H}_{\text{lin}})$ {Linear attention mechanism; Using convolutional (q_t,k_t,v_t) derived from H_lin}
$\mathbf{X}'_{\text{lin}} \leftarrow \mathbf{X}_{\text{lin}} + \mathrm{Dropout}(\mathbf{Y}_{\text{delta}})$ {First residual connection}
$\mathbf{H}'_{\text{lin}} \leftarrow \mathrm{ZeroCenteredRMSNorm}(\mathbf{X}'_{\text{lin}})$ {Pre-Norm for MoE}
$\mathbf{U}_{\text{lin}} \leftarrow \mathrm{MoE}_{\text{lin}}(\mathbf{H}'_{\text{lin}})$
$\mathbf{X}''_{\text{lin}} \leftarrow \mathbf{X}'_{\text{lin}} + \mathrm{Dropout}(\mathbf{U}_{\text{lin}})$ {Second residual connection}

$\mathbf{X}_{\text{out}} \leftarrow$ torch.zeros_like($\mathbf{X}$) {Initialize output tensor}
$\mathbf{X}_{\text{out}}[I_{\text{full}}] \leftarrow \mathbf{X}''_{\text{full}}$ {Scatter full-attention results}
$\mathbf{X}_{\text{out}}[I_{\text{lin}}] \leftarrow \mathbf{X}''_{\text{lin}}$ {Scatter linear-attention results}
return $\mathbf{X}_{\text{out}}$

A.5 Routing Configuration and Regularizer Interpretation

For all reported experiments, we use $B = 0.20$ and $\tau = 0.5$ for visual tokens, while text tokens are deterministically routed to full attention. We linearly ramp λ_{usage} over the first 20% of training steps and keep it fixed afterwards. The router is a single shared MLP; routing remains layer-dependent because each block provides different normalized hidden states. The training pipeline first optimizes the ViT, MLP projector, and router, then performs end-to-end SFT. Latency is measured as end-to-end wall-clock time on A100 with bf16 and batch size 1.

The no-regularizer ablation in Table 2 is counterintuitive because routing 61% of visual tokens to full attention is slower and less accurate than the default 18–20% usage. Our interpretation is that the hybrid model is optimized to learn a selective division of labor: the full-attention branch handles sparse high-value visual evidence, while the DeltaNet branch absorbs broader context and background. Removing the usage regularizer changes this operating point. The router sends many medium-value tokens to the expensive branch, branch specialization weakens, and the model spends more compute without improving the evidence selection that matters for V* Bench.

A.6 Limitations and Responsible Use

The current default budget should be viewed as a practical operating point, not as a universal optimum for every input. Extremely information-dense documents, charts, or full-frame scenes may require adaptive or input-conditioned budgets. The reported long-context study is controlled up to 64k visual tokens; stability and quality beyond this range, such as 128k-token evaluation, remain open. Finally, learned routing may systematically ignore contextual regions that appear visually peripheral but are semantically important. Before deployment in sensitive domains, routing maps and failure cases should be audited across domains, languages, layouts, and visual groups.