DEEP AND COMPACT VISUAL EXPERTS FOR FAST ENCODER-FREE MLLM

Anonymous authors
Paper under double-blind review

ABSTRACT

Encoder-free Multimodal Large Language Models (MLLMs) process images and text within a single backbone, enabling early cross-modal fusion. In this paper, we study another benefit of these architectures compared to modular MLLMs like LLaVA: reducing prefill latency. This prefill advantage is particularly important for edge deployment and real-time interactions. Despite this advantage, encoder-free architectures remain understudied in latency-constrained settings, particularly with small, efficient backbones. We first examine the design space of encoder-free architectures to assess both multimodal performance and latency bottlenecks. Our analysis shows that the performance of existing encoder-free designs is limited by insufficient visual processing depth in small backbones. Building on this finding, we propose **DeCoVE**, a **De**ep and **Co**mpact **V**isual **E**xpert architecture that trades visual expert width for depth. DeCoVE adds no additional parameter count compared to modular MLLM, while significantly improving prefill latency. Moreover, DeCoVE can be initialized from parts of pretrained vision encoders, bringing together the training efficiency of pretrained models and the prefill advantage of encoder-free designs. Based on DeCoVE, we develop **DeCoVE-3B**, an encoder-free MLLM that achieves state-of-the-art performance among encoder-free baselines and is competitive with Qwen3.5-VL 2B, while reducing prefill latency by 41.5% across 13 benchmarks. DeCoVE-3B establishes a new Pareto frontier between performance and prefill latency for MLLMs.

1 INTRODUCTION

Multimodal large language models (MLLMs) (Liu et al., 2024; Dai et al., 2023) have significantly advanced intelligent systems beyond natural language processing. Most state-of-the-art MLLMs (Bai et al., 2025a; Wang et al., 2025b; Team et al., 2026) adopt a modular architecture that connects a pretrained vision encoder to a language-model backbone through a lightweight connector. Despite their strong performance, modular MLLMs are suboptimal from an efficiency perspective because the vision encoder and LLM backbone must be executed sequentially. This sequential dependency increases prefill latency (Luo et al., 2025) and complicates balanced sharding during training (Zhang et al., 2025).

Encoder-free MLLMs offer a way to reduce this sequential overhead by processing images and text within a single LLM backbone, as demonstrated by early models such as Fuyu (Bavishi et al., 2023). However, this simple design suffers from catastrophic forgetting and slow convergence, motivating subsequent works to introduce modality-specific parameterization (Luo et al., 2025; Chen et al., 2024) and distillation from pretrained vision encoders (Wang et al., 2025a; Diao et al., 2024). While these advances improve the performance and training efficiency of encoder-free MLLMs, their design remains underexplored in latency-constrained settings with small and hybrid LLM backbones. This regime is particularly relevant for fully realizing the prefill advantage of encoder-free MLLMs.

In this work, we investigate how to design encoder-free MLLMs with small hybrid LLM backbones for both strong multimodal performance and high prefill efficiency. We begin by systematically evaluating existing encoder-free designs under this setting. Consistent with prior works, we find that modality-specific parameters significantly accelerate training convergence compared to a dense baseline. However, none of the existing designs accommodate the reduced visual processing depth resulting from removing the vision encoder. We show that insufficient visual processing depth is

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.

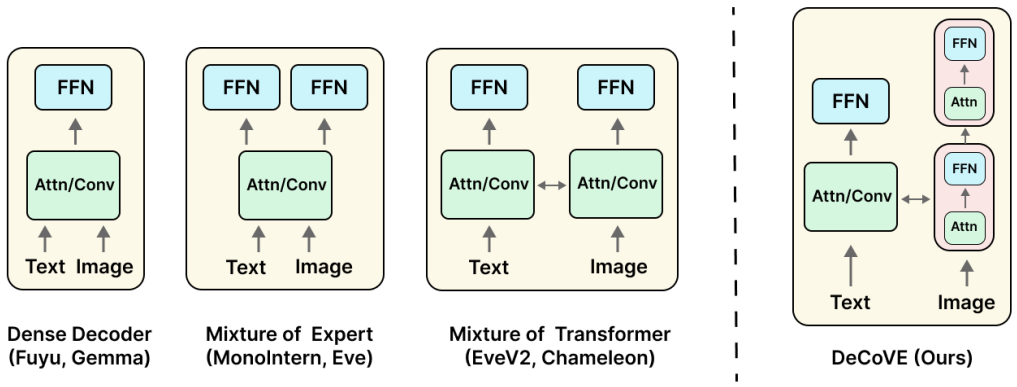


Figure 1: **Overview of encoder-free architectures.** Existing designs either share a single backbone across modalities or introduce modality-specific parameters that mirror the backbone architecture. In contrast, DeCoVE adopts a deeper and narrower modality-specific pathway, augmented with dedicated attention layers for visual tokens.

the main bottleneck for encoder-free MLLMs when the backbone is shallow, which is common in latency-constrained settings. To understand what architectural changes we can make with minimal impact on prefill speed, we progressively transform a modular MLLM into an encoder-free design and quantify the latency reduction from each change. This analysis shows that reducing visual tokens and avoiding sequential image–text computation account for much of the prefill advantage, while deeper modality-specific parameters incur only modest additional latency.

Based on these findings, we propose **DeCoVE**, a deep and compact visual expert architecture that trades width for visual processing depth, keeping the parameter budgets fixed. Unlike prior encoder-free designs that require large-scale multimodal pretraining or distillation, DeCoVE can be initialized from parts of a pretrained vision encoder to significantly accelerate training convergence. We further show that varying the number of visual layers and their arrangement within the backbone enables flexible trade-offs between performance and prefill speed.

Finally, we train **DeCoVE-3B**, a state-of-the-art encoder-free model built on a hybrid backbone that outperforms the strongest encoder-free baseline, NEO-ov, by 3.16 points on average across 13 benchmarks while achieving performance comparable to the modular Qwen3.5-VL 2B. More importantly, DeCoVE-3B is natively optimized for prefill efficiency, reducing average time to first token (TTFT) by 41.5% relative to Qwen3.5-VL 2B and 68.0% relative to NEO-ov across these benchmarks.

Our contributions are summarized as follows:

- We systematically study encoder-free MLLMs with small hybrid backbones and identify insufficient visual processing depth as a key bottleneck for their performance.
- We propose DeCoVE, a deep and compact visual expert that decouples visual processing depth from backbone depth by trading visual width for depth under a fixed parameter budget and supports initialization from pretrained vision encoders for efficient training.
- We develop DeCoVE-3B, an encoder-free hybrid MLLM that achieves state-of-the-art performance among encoder-free baselines and is competitive multimodal performance with modular MLLMs at substantially lower prefill latency.

2 Related Work

Modular MLLMs. Popularized by LLaVA (Liu et al., 2024) and BLIP (Li et al., 2023b), modular MLLMs combine powerful pretrained LLM backbones (Brown et al., 2020; Grattafiori et al., 2024) with vision encoders (Zhai et al., 2023; Radford et al., 2021) to acquire multimodal capabilities without requiring large-scale multimodal pretraining. This design has since be the default for most

of the SoTA MLLMs (Bai et al., 2025a; Team et al., 2026; Wang et al., 2025b). However, the separate vision tower introduces additional computation and a sequential visual-processing stage, limiting inference efficiency (Luo et al., 2025). Our work explores encoder-free architectures to reduce this sequential processing overhead and enable faster prefill, while retaining the performance and training efficiency of a modular MLLM.

Encoder-free MLLMs. Early encoder-free models like Fuyu (Bavishi et al., 2023) and EVE (Diao et al., 2024) directly model multimodal input through a single dense backbone. While the performance was nontrivial, these models still had large performance gap with the modular MLLMs. Recent models like Gemma 4 (Team, 2026) and Inkling (Thinking Machines Lab, 2026) have shown that scaling model parameter and dataset size can effectively bridge this gap. Alternatively, other works (Luo et al., 2025; Diao et al., 2025) show that modality-specific parameters can improve both performance and training efficiency by reducing cross-modality interference. Common designs duplicate either the feed-forward networks (Luo et al., 2025) or entire transformer blocks (Liang et al., 2024; Diao et al., 2025). To accelerate visual representation learning for encoder-free models, some methods (Diao et al., 2024; Wang et al., 2025a; Li et al., 2025b) distill knowledge from pretrained vision encoders. We instead directly initialize our modality-specific parameters from parts of a pretrained vision encoder, further improving training efficiency.

Efficient MLLMs. ViT-based vision encoders such as CLIP (Radford et al., 2021) and SigLIP (Zhai et al., 2023) produce large number of visual tokens, which increasing prefill and decoding latency for the downstream MLLM. Prior works reduce the visual overhead through visual-token compression (Bai et al., 2025b; Dai et al., 2023; Cha et al., 2024), more efficient vision encoders (Vasu et al., 2025; 2024), and encoder-free design (Luo et al., 2025; Lac & Wu, 2026). Complementary efforts improve backbone efficiency with smaller (Marafioti et al., 2025) and hybrid (Amini et al., 2025) LLM backbones. Our work targets both sources of inefficiency by combining a token-efficient encoder-free design with a hybrid LLM backbone.

Table 1: **Comparison of Encoder-free visual architectures with hybrid backbone.**

Architecture	BLINK	RWQA	MMStar	ChartQA	MathVista	RefCOCO	Avg
SigLIP 2	50.2	62.6	50.4	68.3	32.5	79.3	57.2
VoRA	38.0	49.3	31.7	13.7	38.1	21.7	32.1
MoE	38.8	49.8	39.0	42.6	40.1	44.8	42.5
MoT	42.1	53.2	39.1	50.8	43.8	54.3	47.2
Attn-SwiGLU	39.7	51.4	40.5	49.9	43.6	57.1	47.0
DeCoVE	46.5	57.3	42.5	50.9	44.9	65.6	51.3

3 Encoder-Free Architectures

We evaluate existing encoder-free architectures on the attention-convolution hybrid LFM2.5-VL 1.6B backbone under a fixed visual parameter budget of $\sim$400M, matching SigLIP 2 (Tschannen et al., 2025). Their underperformance relative to the SigLIP 2 baseline motivates two hypotheses about architectural limitations of the small hybrid backbone, which we test with targeted experiments. Implementation details are in Appendix A.2.

Dense Models The simplest encoder-free architecture uses a single LLM backbone with a lightweight image embedding layer (Bavishi et al., 2023) to jointly process image and text. VoRA (Wang et al., 2025a) builds on this design by freezing the backbone and training only a vision-specific LoRA adapter, which is later merged back to the backbone. We adopt VoRA as the dense baseline because of its stable training dynamics compared to training the whole backbone.

Modality-Specific Parameters We evaluate Mixture-of-Experts (MoE) (Luo et al., 2025), which uses modality-specific feed-forward networks, and Mixture-of-Transformers (MoT) (Liang et al.,

2024; Diao et al., 2025), which additionally uses modality-specific QKV projections and normalization. To remain within parameter budget, we introduce modality-specific parameters only in the first 8 of the 16 backbone layers and the remaining 8 layers share parameters across modalities.

Sources of Prefill Efficiency Before proposing new encoder-free architectures, we examine the main sources of prefill latency. Figure 2 illustrates the prefill speed for AMD MI325x GPU as we progressively transform the modular LFM2.5-VL 3B model towards an encoder-free design (See Appendix A.3 for more details). First, reducing the number of visual patches entering the encoder by a factor of four with pixel shuffling, rather than applying pixel shuffling after the encoder, lowers TTFT from 72.2 to 55.4 ms. Next, reusing the encoder computation in place of the first eight backbone layers for visual tokens reduces TTFT to 50.4 ms. Finally, overlapping visual and textual processing lowers TTFT to 46.8 ms. The 27-layer visual parameters are not fully overlapped with the backbone computation due to different model shapes. This leaves a 4.7 ms gap compared to processing all tokens through the dense model, which serves as the lower bound for encoder-free TTFT. Overall, the largest prefill gains come from reducing the number of visual tokens and eliminating visual processing that must occur sequentially before the LLM backbone. In contrast, partial overlap between visual and backbone computation introduces only a modest latency overhead, suggesting that richer modality-specific visual processing can be added without sacrificing most of the encoder-free prefill advantage. These observations motivate the following modifications to visual token mixing and processing depth.

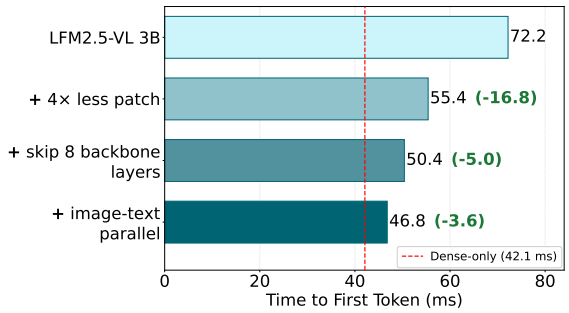


Figure 2: **TTFT breakdown.** Prefill advantage from each design choice of encoder-free MLLM

Attn-SwiGLU A potential limitation of hybrid backbones is insufficient global interaction among visual tokens. In the first 8 layers, only two use full attention, while the rest rely on 1-D convolutions (Appendix A.1). To test whether this limits performance, we replace the vision-specific convolutional mixers in MoT with bidirectional attention over visual tokens, while the text tokens remain to use 1-D convolutions.

DeCoVE Another potential limitation is insufficient visual processing depth. While this is less concerning for encoder-free MLLMs with 48 (Team, 2026) or 66 (Thinking Machines Lab, 2026) layers, this can be problematic for small backbones. In our setting, visual tokens pass through only 16 backbone layers, compared to 27 SigLIP 2 layers plus 16 backbone layers in the modular baseline. Prior work also highlights the importance of depth in encoder-free MLLMs and Transformers more broadly (Luo et al., 2025; Giannou et al., 2023). Motivated by this, we introduce DeCoVE, an asymmetric design that trades visual width for depth by halving the FFN intermediate dimension and replacing each Attn-SwiGLU block with two smaller layers (Figure 1). Applying DeCoVE to the first 6 backbone layers yields 12 layers of visual processing at the same parameter budget.

Performance Comparison Table 1 compares encoder-free architectures on the same hybrid backbone. MoE and MoT substantially outperform VoRA, consistent with prior findings that modality-specific parameters improve performance. The large gap further suggests that removing the vision encoder can be especially detrimental to small MLLMs, where it represents a larger fraction of the total model parameter. While Attn-SwiGLU achieves performance comparable to MoT despite its attention layers being initialized from scratch, the absence of a clear improvement over MoT suggests that global interaction may not be the dominant bottleneck in this setting. Interestingly, DeCoVE outperforms both MoT and Attn-SwiGLU by 4% with both the FFN and attention being initialized from scratch. This result identifies insufficient processing depth as the dominant bottleneck for encoder-free MLLMs, particularly in small models with inherently shallow backbones. The detailed training setup can be found in Appendix B.1. In the following sections, we will leverage DeCoVE to build efficient encoder-free MLLMs and explore its performance-latency tradeoffs.

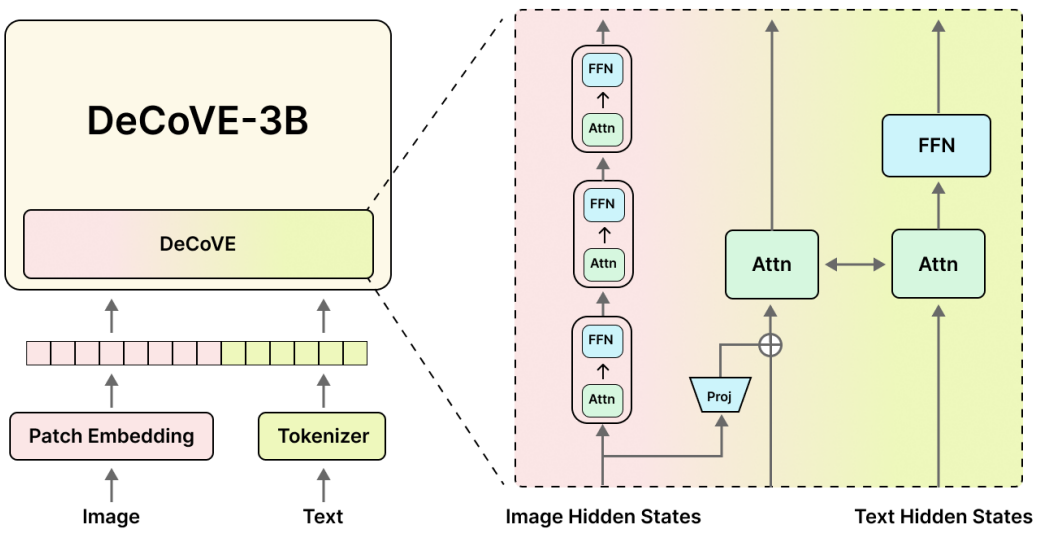


Figure 3: **Overview of DeCoVE.** Visual tokens are processed by a compact, deep expert that operates in parallel with the text backbone, while a residual stream retains the backbone hidden dimension. At global attention layers, pathway features are projected to the backbone dimension and added to the residual stream, which text tokens attend to for cross-modal fusion.

4 METHOD

4.1 MODEL ARCHITECTURE

Figure 3 illustrates the architecture of our DeCoVE-3B model, which consists of a lightweight image embedding followed by a hybrid LLM backbone with DeCoVE in the first 8 layers. Multimodal tokens are joint back and process together by shared weights after 8th layer.

Image Patch Embedding We adopt a simple 2-layer convolution with GELU as nonlinearity to embed the image into visual tokens:

$$\mathbf{x} = \mathrm{Conv}_2(\mathrm{GELU}(\mathrm{Conv}_1(\mathbf{I}))) .$$

Here, $\mathbf{I} \in \mathbb{R}^{C\times H\times W}$ denotes the input image. Conv_1 uses a 16×16 kernel with stride 16, and Conv_2 uses a 32×32 kernel with stride 32. We set the output width of Conv_2 to the LLM hidden dimension d_{LLM} and flatten its spatial dimensions to obtain $\mathbf{x} \in \mathbb{R}^{n_{\mathrm{visual}}\times d_{\mathrm{LLM}}}$.

DeCoVE Configuration Figure 2 shows that even the extreme case of using a full vision encoder as the modality-specific visual expert adds only modest prefill latency. Motivated by this observation, we design DeCoVE to match the architecture of Qwen-ViT. Its hidden dimension $d_{\mathrm{V}} = 1280$ is substantially smaller than the backbone dimension $d_{\mathrm{LLM}} = 2048$, naturally follows our design principle of trading width for greater visual processing depth. Moreover, unlike prior approaches that transfer knowledge from pretrained vision encoders through distillation (Diao et al., 2024; Wang et al., 2025a; Li et al., 2025b), this architectural flexibility allows DeCoVE to directly leverage any pretrained weights.

Concretely, we initialize DeCoVE with the first 24 layers of Qwen-ViT and distribute them evenly across the first 8 LLM layers, assigning 3 visual layers to each LLM layer. Input visual tokens $\mathbf{x}$ from the embedding layer are projected to visual hidden $\mathbf{h}_{\mathcal{V}}^{(0)} \in \mathbb{R}^{n_{\mathrm{visual}}\times d_{\mathrm{V}}}$. Then for layer $\ell \in \{1, \ldots, 8\}$, the text and visual hidden states, $\mathbf{h}_{\mathcal{T}}^{(\ell)}$ and $\mathbf{h}_{\mathcal{V}}^{(\ell)}$, are processed as

Text	Image
$\mathbf{h}_{\mathcal{T}}^{(\ell+1)} = \mathcal{B}_\ell\left(\mathbf{h}_{\mathcal{T}}^{(\ell)}\right)$	$\mathbf{h}_{\mathcal{V}}^{(\ell+1)} = (\mathcal{V}_{\ell,3} \circ \mathcal{V}_{\ell,2} \circ \mathcal{V}_{\ell,1})\left(\mathbf{h}_{\mathcal{V}}^{(\ell)}\right),$

where $\mathcal{B}_\ell$ is the LLM block and $\mathcal{V}_{\ell,j}$ is the j-th DeCoVE block assigned to LLM layer ℓ. Within DeCoVE, tokens interact through bidirectional self-attention.

Cross-Modal Fusion To enable early fusion despite the difference between hidden dimensions d_{LLM} and d_{V}, we maintain a residual stream $\mathbf{r}^{(\ell)}$ with dimension d_{LLM} for each visual token. At each global-attention layer in the range of DeCoVE, we project the corresponding DeCoVE hidden states to d_{LLM} and aggregate to $\mathbf{r}^{(\ell)}$. Concretely:

$$\mathbf{r}^{(\ell)} = \begin{cases} \mathbf{r}^{(\ell-1)} + \text{Linear}_\ell\left(\mathbf{h}_{\mathcal{V}}^{(\ell)}\right), & \ell \in \mathcal{G}, \\ \mathbf{r}^{(\ell-1)}, & \ell \notin \mathcal{G}, \end{cases}$$

where $\ell \in \mathcal{G}$ denotes a global-attention layer; for our backbone, $\mathcal{G} = \{2, 5\}$. After that, causal attention is performed between text and visual tokens using $\mathbf{r}^{(\ell)}$ and the text hidden states $\mathbf{h}_{\mathcal{T}}^{(\ell)}$.

Joint Multimodal Processing At the end of layer 8, we combine the residual stream with the projected output of DeCoVE:

$$\mathbf{h}_{\mathcal{V}}^{(9)} = \mathbf{r}^{(9)} + \text{Linear}_{\text{out}}\left(\mathbf{h}_{\mathcal{V}}^{(9)}\right).$$

The text and visual hidden states are then concatenated along the sequence dimension and processed by the remaining layers of the shared LLM backbone to enable joint modeling of vision and text.

5 Experimental Setup

5.1 Training Setup

DeCoVE-3B is built on the LFM2.5-2.6B hybrid backbone, with newly initialized image patch embeddings and linear projections for cross-modal fusion. Training consists of multimodal pretraining followed by instruction tuning. All model parameters are trainable in both stages. We use AdamW (Loshchilov & Hutter, 2019) with a cosine learning-rate schedule and sequence packing, with a global batch size of 4M tokens across four 8-GPU nodes.

Multimodal Pretraining During multimodal pretraining, the model acquires broad visual perception and multimodal understanding by aligning visual and linguistic representations through an autoregressive objective. Training uses approximately 153M publicly available samples, including approximately 106M (Gadre et al., 2023) captions recaptioned with off-the-shelf models, 3.8M VQA samples, 1.4M OCR samples, and 42M text-only samples to preserve the language capabilities of the pretrained LLM. Multimodal pretraining runs for 25k steps with a base learning rate of 5e-5 and a higher learning rate of 4e-8 for the newly initialized patch embedding and projection layers to accelerate convergence.

Instruction Tuning During instruction tuning, the model is further finetuned to improve instruction-following and alignment with downstream applications. We use 15M high-quality instruction-tuning samples curated from publicly available sources, covering visual reasoning, OCR, visual grounding, and captioning. During this stage, all model parameters are optimized with the same learning rate of 5e-5 and is trained for 12k steps.

5.2 Evaluation Setup

We evaluate DeCoVE-3B and baseline MLLMs on general, document understanding, and grounding tasks. General includes MMMU-PRO (Yue et al., 2025), RealworldQA (xAI, 2024), BLINK (Fu et al., 2024), SEEDBench (Li et al., 2023a), CountBench (Paiss et al., 2023), MathVista (Lu et al., 2024), POPE (Li et al., 2023c). Document understanding includes ChartQA (Masry et al., 2022), CharXiv (Wang et al., 2024), DocVQA (Mathew et al., 2021). Grounding includes Refcoco (Yu et al., 2016), Refcoco-m (Reid, 2026), Screenspot-V2 (Wu et al., 2024). Evaluations are conducted using VLMEval-kit (Duan et al., 2024).

Table 2: **Main results on general and reasoning benchmarks.** TTFT denotes the average time-to-first-token across benchmarks on AMD MI325X GPU, measured in milliseconds. Among encoder-free MLLMs, **bold** indicates the best result and underlined indicates the second-best result.

Model	TTFT ↓	MMMU-PRO	RWQA	BLINK	SEED	Count	MathVista	POPE	Avg
Modular MLLMs									
InternVL3.5 2B	53.1	21.27	61.57	51.97	75.36	70.43	56.80	88.03	60.78
Qwen3.5-VL 2B	44.6	24.86	65.10	48.61	75.78	83.78	48.70	88.59	62.20
Qwen3.5-VL 4B	55.0	35.95	67.06	58.65	76.13	86.65	63.60	86.01	67.72
Gemma-4-E4B	51.2	32.60	64.31	52.18	75.27	80.49	45.20	86.92	62.42
LFM 2.5-VL 3B	33.8	30.50	73.10	61.50	77.70	87.30	68.50	88.70	69.61
Encoder-Free MLLMs (∼3B)									
Mono-InternVL 1.5	38.6	18.32	50.98	40.24	66.58	67.97	46.30	85.13	53.65
HoVLE	37.8	17.69	55.56	41.19	70.16	63.86	46.20	85.44	54.30
OneCAT	71.8	19.42	65.75	44.71	70.66	29.98	58.80	**88.36**	53.95
NEO	60.8	30.64	62.75	50.50	74.10	72.07	56.30	87.79	62.02
NEO-ov	70.8	**32.49**	60.10	53.90	**75.30**	78.23	59.10	81.80	62.99
DeCoVE-3B	**26.2**	26.42	**67.19**	**54.87**	73.90	**84.19**	**63.30**	87.42	**65.33**

Table 3: **Main results on document understanding and grounding benchmarks.** For encoder-free MLLMs, SS-v2 is the unweighted mean of the Desktop, Mobile, and Web splits; split scores and evaluation notes are in Table 7.

Model	TTFT (ms) ↓	CharXiv	DocVQA	ChartQA	RefCOCO	RefCOCO-m	SS-v2	Avg
Modular MLLMs								
InternVL3.5 2B	54.6	44.25	88.40	81.72	82.92	90.71	82.01	78.34
Qwen3.5-VL 2B	52.82	71.30	92.59	78.40	78.54	91.14	66.45	79.74
Qwen3.5-VL 4B	67.41	86.48	94.75	84.16	86.64	93.19	78.53	87.29
Gemma-4-E4B	48.5	70.38	87.43	42.12	72.14	82.48	51.23	67.63
LFM 2.5-VL 3B	39.1	70.53	91.10	81.30	87.90	94.30	80.70	84.31
Encoder-Free MLLMs (∼3B)								
Mono-InternVL 1.5	36.5	45.88	80.03	72.56	77.17	85.21	0.40	60.21
HoVLE	35.9	50.70	84.60	74.44	78.55	88.51	0.00	62.80
OneCAT	80.2	57.35	89.03	75.92	74.90	84.76	2.28	64.04
NEO	80.0	60.33	88.91	79.9	78.83	85.87	31.81	70.94
NEO-ov	109.1	63.25	**90.42**	**82.0**	81.07	87.62	47.26	75.27
DeCoVE-3B	**30.8**	**64.43**	88.00	78.1	**84.30**	**91.60**	**69.95**	**79.40**

For comparsion, we use SoTA modular MLLMs (InternVL3.5 2B (Wang et al., 2025b), Qwen3.5-VL 2B (Qwen Team, 2026), Qwen3.5-VL 4B (Qwen Team, 2026), Gemma-4-E4B (Team, 2026), LFM2.5-VL 3B (AI, 2026)) and encoder-free MLLMs (Mono-InternVL 1.5 (Luo et al., 2025), HoVLE (Tao et al., 2025), OneCAT (Li et al., 2025a), NEO (Diao et al., 2026a), NEO-OV (Diao et al., 2026b)) as our baselines.

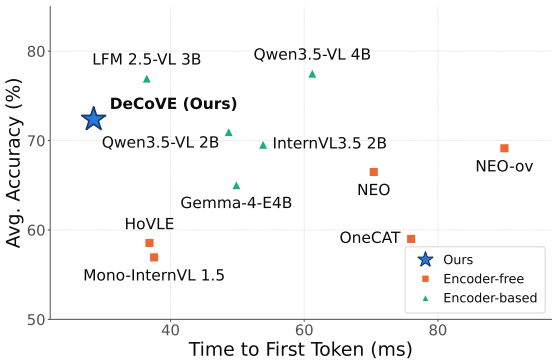


Figure 4: **Performance–latency trade-off**

5.3 Main Results

Table 2 presents results on general visual understanding benchmarks. DeCoVE outperforms the strongest encoder-free baseline, Neo-ov (Diao et al., 2026b), by 2.4% on average while reducing time to first token (TTFT) by 63%. Neo-ov incurs substantially higher latency due to its design of stacking additional LLM layers to the backbone for early cross-model fusion and specialized attention kernels for combining bidirectional

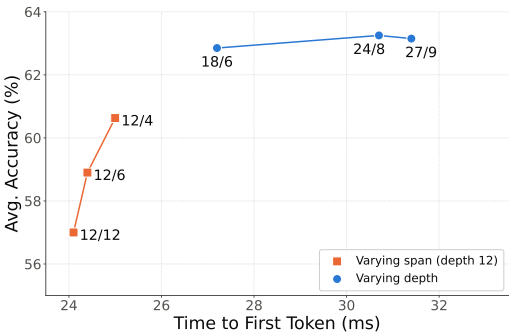


Figure 5: **Effect of visual depth and layer placement on the accuracy–latency trade-off.** Labels denote the number of visual layers / backbone layers spanned.

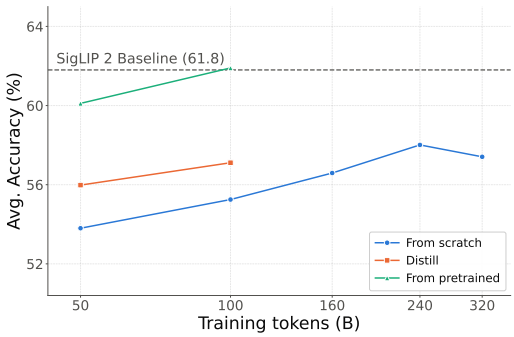


Figure 6: **Accuracy versus training-token budget.** The token budget denotes the pretraining budget, and all models undergo the same Instruction tuning afterward.

and casual attention. In contrast, DeCoVE increases visual processing depth through a deep and compact visual experts to keep the computational cost low, and uses dedicated attention layers for visual tokens without modifying the attention mask pattern of the language backbone. Compared with modular MLLMs, DeCoVE achieves competitive average performance while delivering the lowest TTFT. In particular, the TTFT comparison with LFM 2.5-VL 3B (AI, 2026) confirms that an encoder-free architecture can further reduce prefill latency when paired with an already efficient hybrid backbone. Notably, the modular MLLMs in Table 2 undergo additional post-training like reinforcement learning and model merging, whereas DeCoVE is evaluated after instruction tuning.

Table 3 presents results on document understanding and visual grounding benchmarks, where DeCoVE achieves competitive performance across a diverse set of tasks. The prefill advantage of encoder-free models is particularly pronounced in this setting, as document images and web screenshots are often processed at high resolutions and therefore incur substantial visual encoding overhead in modular MLLMs. Consistent with the observations in prior encoder-free models (Diao et al., 2026b;a; Luo et al., 2025), DeCoVE remains subpar on fine-grained OCR and information extraction compared to modular MLLMs. We expect that increasing the amount and diversity of OCR-centric training data could further close the performance gap on these tasks. Overall, these results establish DeCoVE as the pareto frontier model for prefill latency and performance (see Figure 4) among both modular and encoder-free MLLMs in the efficent model setting.

5.4 Ablations

For all ablation studies, we use the smaller LFM 2.5 1.2B model as the hybrid LLM backbone to enable efficient experimentation. Unless otherwise specified, all models are pretrained on 100B tokens and instruction-tuned on 50B tokens using the same data mixture as DeCoVE-3B.

Layer Placements within Backbone The placement of DeCoVE within the backbone introduces a trade-off between accuracy and prefill latency. At each cross-fusion layer, the visual and language streams must synchronize. If the computation of DeCoVE takes longer than that of the corresponding backbone layers, the visual stream becomes the bottleneck and the excess computation directly increases end-to-end latency. From a performance perspective, however, we find it beneficial to concentrate more visual processing in earlier layers. Figure 5 studies this trade-off by fixing the total number of visual layers to 12 and distributing them uniformly across the first 4, 6, or 12 backbone layers. Concentrating the visual layers earlier consistently improves downstream performance, at the cost of a modest increase in prefill latency.

We next study the effect of the total depth of DeCoVE. Fixing the allocation to three visual layers per backbone layer, we vary the total number of visual layers. Figure 5 shows increasing visual depth consistently improves downstream performance, while incurring additional prefill latency. Based on this, we use 24 visual layers for DeCoVE-3B, which provides the best performance while still having the fastest prefill speed compared among all baselines.

Table 4: **Ablation on early cross-model fusion.**

Model	BLINK	RWQA	MMStar	ChartQA	MathVista	RefCOCO	Avg
No fusion	51.55	58.95	45.13	67.00	54.60	76.70	58.99
Cross-fusion	53.97	59.22	48.73	69.28	56.10	79.70	61.17

Table 5: **Ablation on vision encoder initialization.**

Teacher	BLINK	RWQA	MMStar	ChartQA	MathVista	RefCOCO	Avg
SigLIP2	50.50	60.78	50.47	70.20	58.80	80.70	61.91
MoonViT	52.50	61.57	50.90	73.04	58.60	81.86	63.08
QwenViT	53.39	61.57	51.47	72.88	58.60	81.90	63.30

Training Efficiency Figure 6 compares training from scratch, distillation from a pretrained vision encoder (see Appendix B.2), and direct initialization from pretrained vision weights. Both distillation and initialization substantially accelerate training convergence, measured by performance after instruction tuning. Remarkably, direct initialization matching the SigLIP 2 baseline using only 100B training tokens and outperfoms dislliation by a large margin. In contrast, training from scratch remains below the baseline after 320B tokens. These results demonstrate that leveraging pretrained visual representations can substantially improve the training efficiency of encoder-free models.

Choice of Vision Encoder to Initialize In Table 5, we examine the effect of initializing DeCoVE from different vision encoders. Notably, MoonViT and QwenViT perform slightly better compared to SigLIP 2, although all three models share the same architecture. We attribute the improvment on the native high resolution supported by MoonViT and QwenViT compared to SigLIP 2. More broadly, these results show that initialization from pretrained vision weights is beneficial across different vision-encoder families.

Early Fusion Improves Understanding Table 4 investigates the role of early cross-modal fusion in encoder-free architectures. We use 12 layers of DeCoVE, distributed across the first 6 backbone layers, with cross-modal fusion occurring at the global-attention layers 2 and 5. Enabling cross-modal fusion improves average downstream performance by 2%, highlighting the importance for effective early multimodal interaction through the global attention layers

6 Limitations

Despite its strong performance and prefill efficiency, DeCoVE has several limitations. First, post-training techniques such as reinforcement learning remain unexplored in this setting and may further close the gap with modular MLLMs. Second, its modality-specific parameters introduce additional parameter overhead similar to prior works, motivating future work on merging them back into the backbone. Finally, DeCoVE is currently limited to image inputs; extending encoder-free architectures to video and audio is an important direction toward omni encoder-free models.

7 Conclusion

We presented DeCoVE, an encoder-free hybrid MLLM architecture designed to jointly optimize for prefill speed and visual understanding. DeCoVE trades visual width for depth to increase the expressiveness of the model without sacrificing for prefill speed. Our results with DeCoVE-3B demonstrate that encoder-free MLLMs can achieve competitive performance with substantially faster prefill speed, offering a promising direction for efficient real-time multimodal systems.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing our method in code. We did not use generative AI tools for any other applicable tasks requiring disclosure. Additionally, we used generative AI tools to assist with manuscript writing, editing, and LaTeX formatting. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Liquid AI. Lfm2.5-vl-3b: A better and faster vision-language model for the edge. *Liquid AI Blog*, 2026. www.liquid.ai/blog/lfm2-5-vl-3b.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints, 2023. URL https://arxiv.org/abs/2305.13245.

Alexander Amini, Anna Banaszak, Harold Benoit, Arthur Böök, Tarek Dakhran, Song Duong, Alfred Eng, Fernando Fernandes, Marc Härkönen, Anne Harrington, Ramin Hasani, Saniya Karwa, Yuri Khrustalev, Maxime Labonne, Mathias Lechner, Valentine Lechner, Simon Lee, Zetian Li, Noel Loo, Jacob Marks, Edoardo Mosca, Samuel J. Paech, Paul Pak, Rom N. Parnichkun, Alex Quach, Ryan Rogers, Daniela Rus, Nayan Saxena, Bettina Schlager, Tim Seyde, Jimmy T. H. Smith, Aditya Tadimeti, and Neehal Tumma. Lfm2 technical report, 2025. URL https://arxiv.org/abs/2511.23404.

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

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, 2025b. URL https://arxiv.org/abs/2502.13923.

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. Introducing our multimodal models, 2023. URL https://www.adept.ai/blog/fuyu-8b.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm, 2024. URL https://arxiv.org/abs/2312.06742.

Junyi Chen, Longteng Guo, Jia Sun, Shuai Shao, Zehuan Yuan, Liang Lin, and Dongyu Zhang. Eve: Efficient vision-language pre-training with masked prediction and modality-aware moe, 2024. URL https://arxiv.org/abs/2308.11971.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Haiwen Diao, Yufeng Cui, Xiaotong Li, Yueze Wang, Huchuan Lu, and Xinlong Wang. Unveiling encoder-free vision-language models, 2024. URL https://arxiv.org/abs/2406.11832.

Haiwen Diao, Xiaotong Li, Yufeng Cui, Yueze Wang, Haoge Deng, Ting Pan, Wenxuan Wang, Huchuan Lu, and Xinlong Wang. Evev2: Improved baselines for encoder-free vision-language models, 2025. URL https://arxiv.org/abs/2502.06788.

Haiwen Diao, Mingxuan Li, Silei Wu, Linjun Dai, Xiaohua Wang, Hanming Deng, Lewei Lu, Dahua Lin, and Ziwei Liu. From pixels to words – towards native vision-language primitives at scale, 2026a. URL https://arxiv.org/abs/2510.14979.

Haiwen Diao, Jiahao Wang, Penghao Wu, Yuhao Dong, Yuwei Niu, Yue Zhu, Zhongang Cai, Weichen Fan, Linjun Dai, Silei Wu, Xuanyu Zheng, Mingxuan Li, Yuanhan Zhang, Bo Li, Hanming Deng, Huchuan Lu, Quan Wang, Lei Yang, Lewei Lu, Dahua Lin, and Ziwei Liu. From pixels to words – towards native one-vision models at scale, 2026b. URL https://arxiv.org/abs/2605.28820.

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

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive, 2024. URL https://arxiv.org/abs/2404.12390.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023.

Angeliki Giannou, Shashank Rajput, Jy yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers, 2023. URL https://arxiv.org/abs/2301.13196.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Aurélien Lac and Tony Wu. Neomme: A single-tower multimodal-native multilingual foundation encoder for efficient fine-tuning and inference, 2026. URL https://arxiv.org/abs/2609.01657.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension, 2023a. URL https://arxiv.org/abs/2307.16125.

Han Li, Xinyu Peng, Yaoming Wang, Zelin Peng, Xin Chen, Rongxiang Weng, Jingang Wang, Xunliang Cai, Wenrui Dai, and Hongkai Xiong. Onecat: Decoder-only auto-regressive model for unified understanding and generation, 2025a. URL https://arxiv.org/abs/2509.03498.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023b.

Tianle Li, Yongming Rao, Winston Hu, and Yu Cheng. Breen: Bridge data-efficient encoder-free multimodal learning with learnable queries, 2025b. URL https://arxiv.org/abs/2503.12446.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023c. URL https://arxiv.org/abs/2305.10355.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

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.

Gen Luo, Xue Yang, Wenhan Dou, Zhaokai Wang, Jiawen Liu, Jifeng Dai, Yu Qiao, and Xizhou Zhu. Mono-internvl: Pushing the boundaries of monolithic multimodal large language models with endogenous visual pre-training. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24960–24971. IEEE, 2025.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Smolvlm: Redefining small and efficient multimodal models, 2025. URL https://arxiv.org/abs/2504.05299.

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.

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.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Ethan Reid. Moondream segmentation: From words to masks, 2026. URL https://arxiv.org/abs/2604.02593.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Chenxin Tao, Shiqian Su, Xizhou Zhu, Chenyu Zhang, Zhe Chen, Jiawen Liu, Wenhai Wang, Lewei Lu, Gao Huang, Yu Qiao, and Jifeng Dai. Hovle: Unleashing the power of monolithic vision-language models with holistic vision-language embedding, 2025. URL https://arxiv.org/abs/2412.16158.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Thinking Machines Lab. Inkling: Our open-weights model. https://thinkingmachines.ai/news/introducing-inkling/, July 2026. Accessed: 2026-09-02.

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, 2025. URL https://arxiv.org/abs/2502.14786.

Pavan Kumar Anasosalu Vasu, Hadi Pouransari, Fartash Faghri, Raviteja Vemulapalli, and Oncel Tuzel. Mobileclip: Fast image-text models through multi-modal reinforced training, 2024. URL https://arxiv.org/abs/2311.17049.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokul Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari. Fastvlm: Efficient vision encoding for vision language models, 2025. URL https://arxiv.org/abs/2412.13303.

Han Wang, Yongjie Ye, Bingru Li, Yuxiang Nie, Jinghui Lu, Jingqun Tang, Yanjie Wang, and Can Huang. Vision as lora, 2025a. URL https://arxiv.org/abs/2503.20680.

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

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, Alexis Chevalier, Sanjeev Arora, and Danqi Chen. Charxiv: Charting gaps in realistic chart understanding in multimodal llms, 2024. URL https://arxiv.org/abs/2406.18521.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*, 2024.

xAI. Realworldqa: A benchmark for spatial understanding. https://x.ai/blog/grok-1.5v, 2024.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions, 2016. URL https://arxiv.org/abs/1608.00272.

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. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Zili Zhang, Yinmin Zhong, Yimin Jiang, Hanpeng Hu, Jianjian Sun, Zheng Ge, Yibo Zhu, Daxin Jiang, and Xin Jin. Disttrain: Addressing model and data heterogeneity with disaggregated training for multimodal large language models. In *Proceedings of the ACM SIGCOMM 2025 Conference*, pp. 24–38, 2025.

A Architecture Detail

A.1 Hybrid Backbone Architecture

For all of our experiments, we use the LFM2.5 1.2B and LFM2.5 2.6B (Amini et al., 2025) as the hybrid LLM backbone. Their architecture consists a mix of Group-Query Attention and depthwise 1D convolution blocks.

LFM2.5 1.2B LFM2.5 1.2B consists of 16 layers with a hidden dimension of 2048, interleaving grouped-query attention (GQA) (Ainslie et al., 2023) and depthwise 1D convolution blocks. Full-attention layers are placed at layers $2, 5, 8, 10, 12, 14$ using zero-based indexing, while the remaining layers use convolution blocks. Each attention layer uses 32 query heads and 8 key-value heads, corresponding to a GQA ratio of 4:1. Rotary positional embeddings (RoPE) (Su et al., 2023) are applied with a base frequency of $\theta = 10^6$. Each layer uses a SwiGLU feed-forward network with an intermediate dimension of 12,288. The convolution blocks use depthwise 1D convolutions with a kernel size of 3 and no convolution bias.

LFM2.5 2.6B LFM2.5 2.6B consists of 30 layers with a hidden dimension of 2048, interleaving grouped-query attention (GQA) and depthwise 1D convolution blocks. Full-attention layers are placed at layers $2, 5, 9, 13, 17, 21, 24, 27$ using zero-based indexing, while the remaining layers use convolution blocks. Each attention layer uses 32 query heads and 8 key-value heads, corresponding to a GQA ratio of 4:1. Rotary positional embeddings (RoPE) are used with a base frequency of $\theta = 10^7$. Each layer uses a SwiGLU feed-forward network with an intermediate dimension of 10,752. The convolution blocks use depthwise 1D convolutions with a kernel size of 3 and no convolution bias.

A.2 Encoder-Free Architectures

This section describes the architectural formulations used in our comparison of encoder-free architectures.

VoRA For the modality specific LoRA, we apply LoRA rank 512 to all linear layers of the LFM2.5 1.2B backbone, resulting in 390M LoRA parameters. After training, we merge the LoRA back into the backbone, so this approach introduce no additional parameter.

MoE For the MoE setting, we duplicate all linear layers in the first 8 layers' FFN network, resulting in an additional 402M parameters.

MoT For the MoT setting, we duplicate all linear layers and normalization layers in the first 8 layers, resulting in an additional 423M parameters.

Attn-SwiGLU Building on top of MoT, we replace all deplicated 1D convolution layer with GQA initialized from scratch. Becuase the size of the GQA layer is slightly larger than the depthwise convolution, Attn-SwiGLU results in an additional 487M parameters, which is 64M larger than MoT.

DeCoVE Building on Attn-SwiGLU, DeCoVE halves the intermediate dimension of each vision-specific SwiGLU FFN and replaces each visual blocks with two compact GQA–SwiGLU layers. To preserve a fair comparison in model size, DoCoVE is applied to the first 6 layers of the model instead of 8, resulting in an additional 377M parameters. In total, DeCoVE allows 12 layers of visual processing depth in this setting, which is 4 layers deeper than other architectures.

A.3 From Modular to Encoder-Free MLLM

This section provides implementation details for the transformation from modular to encoder-free MLLMs discussed in Section 3.

Reducing Visual Tokens In modular MLLMs, pixel shuffling is typically applied after the vision encoder by concatenating each 2×2 group of spatially adjacent patch features along the channel dimension (Bai et al., 2025b). This operation reduces the number of visual tokens by a factor of four, but the vision encoder must still process all patch features before the reduction. An encoder-free model can instead apply the same rearrangement to raw image patches or using a larger patch size before backbone processing, so only one-quarter as many visual tokens enter the backbone (Diao et al., 2026a). To isolate the latency effect of this earlier token reduction, we apply pixel shuffling before the vision encoder in the modular baseline. This modification is used only for the latency decomposition and is not intended as an accuracy-preserving model variant.

Skipping the First Eight Backbone Layers In a standard modular MLLM, the vision encoder first processes the image, and the resulting visual tokens subsequently pass through the entire LLM backbone. For this latency decomposition, we instead treat the vision encoder as a visual expert that replaces the first eight backbone layers for visual tokens. We use eight layers because their parameter count is comparable to that of the vision encoder. Importantly, the visual and text streams are not processed in parallel. The vision encoder first processes the image, after which the backbone processes the text tokens through its first eight layers. The visual and text tokens are then combined

as input to the ninth layer and processed jointly by the remaining backbone layers. This design reduces FLOPs by avoiding the first eight backbone layers for visual tokens.

Parallel Image and Text Processing We next execute the vision encoder in parallel with the first eight backbone layers applied to the text tokens. Although these two branches have similar parameter counts, their latencies are not equal. In our setting, the vision encoder remains on the critical path because it contains more global-attention layers and requires more kernel launches than the corresponding backbone layers. The text branch would become the bottleneck only for substantially longer text sequences. Our latency measurements use 2560 visual tokens and 1000 text tokens as the input sequence.

B Training Setup

B.1 Encoder-Free Comparison Training

During the multimodal pretraining stage, all encoder-free architectures are trained for 130k steps with a global batch size of 983k tokens across four 8-GPU nodes, corresponding to approximately 128B training tokens. We use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 10^{-6}, and gradient clipping at 1.0. The learning rate follows a cosine schedule with 1500 warmup steps, using a base learning rate of 8×10^{-4} for the language backbone and 1.6×10^{-3} for the vision-specific parameters and projector. Training uses a mixture of 60% caption, 30% text-only data, and 5% VQA data. For VoRA, only the LoRA is trainable, while every parameters are trainable in the first 8 layers are trainable for other architectures.

B.2 Scaling Ablation

For the scaling ablations, we train all models on the same dataset while varying the pretraining token budget. In all three settings, we use 12 DeCoVE layers distributed evenly across the first four backbone layers. For the *from-pretrained* setting, we initialize the DeCoVE layers from the first 12 layers of SigLIP 2. For the *Distill* model, we perform distillation between the intermediate hidden state at layer 12 of DeCoVE to the hidden state of SigLIP 2 on the same visual token after token shuffle to match to sequence length. The modular baseline is LFM2.5-VL 1.6B with a SigLIP 2 vision encoder. It is pretrained on 100B tokens and instruction-tuned on 50B tokens using the same datasets as the encoder-free variants. We use SigLIP 2 for both initialization and distillation because it is also the vision encoder of the modular baseline. This controls the source of visual representations and better isolates the architectural difference between modular and encoder-free models.

C Additional Results

C.1 Evaluation details

All models are evaluated with thinking-off to ensure a fair comparison, as the encoder-free models considered in this work do not support a thinking mode. We limit each model to a maximum of 2560 visual tokens for consistent comparisons of both performance and prefill latency. We use greedy decoding to obtain deterministic and reproducible results. When a response cannot be parsed by the standard answer extractor, we use an LLM judge to determine its score. The prefill latency is measured by taking the average over all the benchmark on pytorch implementation with AMD mI325x 256gb.

C.2 Training-Token Scaling Results

Table 6 provides the detailed results underlying Figure 6.

Training strategy	Tokens (B)	BLINK	RWQA	MMStar	ChartQA	MathVista	RefCOCO	Avg
SigLIP2	–	53.24	61.05	49.13	70.56	58.00	79.26	61.87
From scratch	50	49.40	57.78	44.20	51.92	52.10	67.47	53.81
–	100	50.13	59.48	43.80	59.64	52.40	66.04	55.25
–	160	50.55	59.08	46.60	61.76	53.00	68.54	56.59
–	240	51.45	62.35	48.67	63.24	54.10	68.26	58.01
–	320	49.97	61.18	45.47	63.56	55.30	69.01	57.41
Distill	50	49.08	57.91	44.80	59.88	48.30	75.92	55.98
–	100	51.08	57.65	44.47	61.76	53.10	74.60	57.11
From pretrained	50	49.50	58.82	50.67	67.44	53.80	80.45	60.11
–	100	50.50	60.78	50.47	70.20	58.80	80.58	61.89

Table 6: **Detailed results for the training-token scaling comparison in Figure 6.** Tokens denotes the multimodal pretraining budget in billions. All models undergo the same instruction-tuning stage afterward.

Model	Desktop	Mobile	Web	Mean
Modular MLLMs				
InternVL3.5 2B	79.94	86.23	79.86	82.01
Qwen3.5-VL 2B	63.77	69.66	65.90	66.45
Qwen3.5-VL 4B	76.35	81.44	77.80	78.53
Gemma-4-E4B	45.81	60.28	47.60	51.23
LFM 2.5-VL 3B	78.70	81.20	82.20	80.70
Encoder-Free MLLMs (∼3B)				
Mono-InternVL 1.5	1.20	0.00	0.00	0.40
HoVLE	0.00	0.00	0.00	0.00
OneCAT	1.80	4.59	0.46	2.28
NEO	27.55	45.91	21.97	31.81
NEO-ov	39.82	67.86	34.10	47.26
DeCoVE-3B	68.56	69.66	71.62	69.95

Table 7: **SS-v2 screenshot grounding results by split.** Scores are percentages. Modular-model means are reported evaluation averages; encoder-free means are calculated as the unweighted arithmetic mean of the three splits. These means appear in the SS-v2 column of Table 3. Model names follow Table 3.

C.3 SS-v2 screenshot grounding split results

In this section, we provide the full results on Screenspot-v2 benchmark on the Desktop, Mobile, and Web splits. As shown in Table 7, DeCoVE-3B consistently outperforms all encoder-free baselines

C.4 Visual Depth Result

This section provides detailed results for the visual depth comparison in Figure 5, which varies the total number of visual layers while fixing the allocation to three visual layers per backbone layer. Table 8 reports the per-benchmark results and TTFT.

C.5 Layer Placement Result

This section provides detailed results for the layer placement comparison in Figure 5, which distributes 12 visual layers uniformly across the first 4, 6, or 12 backbone layers. Table 9 reports the per-benchmark results and TTFT.

Layers / span	BLINK	RWQA	MMStar	ChartQA	MathVista	RefCOCO	Avg	TTFT (ms)
12 / 4	50.39	60.65	47.33	69.28	55.10	81.03	60.63	25.0
18 / 6	52.55	64.97	50.13	70.68	57.40	81.35	62.85	27.2
24 / 8	53.39	61.57	51.47	72.88	58.60	81.61	63.25	30.7
27 / 9	50.18	62.22	51.33	73.24	58.80	83.14	63.15	31.4

Table 8: **Full results for the visual depth comparison in Figure 5.** Each backbone layer contains three visual layers. Layers / span denotes the number of visual layers / backbone layers spanned. Dashes indicate results to be filled in.

Layers / span	BLINK	RWQA	MMStar	ChartQA	MathVista	RefCOCO	Avg	TTFT (ms)
12 / 4	50.39	60.65	47.33	69.28	55.10	81.03	60.63	25.0
12 / 6	51.55	58.95	45.13	67.00	54.60	76.70	58.99	24.4
12 / 12	49.80	55.60	46.00	62.10	55.10	73.40	57.00	24.1

Table 9: **Full results for the layer placement comparison in Figure 5.** The total visual depth is fixed at 12 layers. Layers / span denotes the number of visual layers / backbone layers spanned. Dashes indicate results to be filled in.