MAMBURA: A MAMBA-TRANSFORMER HYBRID FOR UNIFIED MULTIMODAL UNDERSTANDING AND GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unified models for image understanding and generation face competing task objectives and the quadratic cost of long-context attention. We introduce **Mambura**, a state-space model (SSM) architecture that, to our knowledge, is the first Mamba–Transformer hybrid to unify these tasks. Its core **Modality-aware Mixture-of-Mamba-Experts ($\mathbf{M^3}$)** layer combines modality-specific expert projections with a shared state-space core, selecting one expert per token within its modality's expert pool. A small number of attention layers provide global token interactions. Its principal advantage is inference efficiency. Against Qwen3.5-2B, the only baseline that still runs at a 1M-token context, Mambura prefills $2.3\times$ faster at 128K and $15.8\times$ faster at 1M, decodes $4.6\times$ faster at 128K and $1.7\times$ faster at 1M, and holds 26% lower peak memory at 1M. The full-attention unified models exhaust the device much earlier. Activating only 0.39B of its 1.7B parameters per token, Mambura nonetheless outperforms dense Transformer baselines that activate seven to eight billion: it exceeds Chameleon, UniToken and Emu3 both on GenEval, where it reaches 0.70, and on MMMU, and it outperforms Show-o on three of their four shared understanding benchmarks. These results demonstrate the potential of SSM-based hybrid architectures for efficient unified multimodal modeling.

1 INTRODUCTION

Image understanding and image generation are converging on a single model. Chameleon (Chameleon Team, 2024) and Show-o (Xie et al., 2025a) place both tasks on one Transformer, Janus-Pro (Chen et al., 2025) and BAGEL (Deng et al., 2025) scale the idea with task-specialized encoders and experts, and the 2026 systems continue it with mixture-of-transformers, decoupled diffusion decoders and dedicated generation heads (Diao et al., 2026; Wang et al., 2026; Tian et al., 2026). A model that understands, reasons over and generates images in one context is more than a convenience: the two abilities share the same visual vocabulary, and serving them from one backbone is what makes interleaved image-text interaction possible.

That capability is expensive to extend to long contexts, and the cost is architectural. Every one of these models is built on self-attention, and images are what make sequences long: a single image contributes hundreds to thousands of tokens, and interleaved image-text contexts multiply them. Three costs then grow together. Prefill is quadratic in sequence length, so time to first token rises sharply. The KV cache is linear in length, so peak memory becomes the binding constraint on a single device. Each decoding step attends over a history that keeps growing, so throughput falls as the context fills. Inference cost is therefore a major bottleneck for long-context unified modeling.

Recurrent sequence models answer all three directly: prefill is linear in length, the recurrent state is fixed in size, and each decoded token costs a constant state update (Gu & Dao, 2023; Dao & Gu, 2024). A purely recurrent backbone, however, compresses the whole history into that state and cannot recover arbitrary token-to-token interaction, which multimodal reasoning needs. We therefore build **Mambura** on a hybrid backbone: recurrent layers carry the bulk of the sequence and a small number of attention layers supply direct global interaction. A shared backbone alone is not enough, because text and image tokens are processed differently even when they occupy one sequence. Our $\mathbf{M^3}$ (**M**odality-aware **M**ixture-of-**M**amba-Experts) layer therefore keeps the recurrent computation

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.

shared while giving each modality its own expert projections, and Mambura is trained natively on both tasks from the first multimodal step rather than through an alignment stage.

Our contributions are:

1) A Mamba–Transformer hybrid for unified multimodal modeling. We introduce Mambura, to our knowledge the first Mamba–Transformer hybrid to unify image understanding and generation within a single backbone. We demonstrate that this architecture supports both tasks with only 0.39B activated parameters per token, achieving a GenEval score of 0.70 and outperforming Show-o on three of their four shared understanding benchmarks. On a single GPU it decodes $7.7\times$ faster than the fastest full-attention unified baseline at 128K, and at 1M it prefills $15.8\times$ faster and holds 26% lower peak memory than Qwen3.5-2B, the only baseline that still runs there, while the full-attention unified models exhaust the device much earlier.

2) Modality-aware Mixture-of-Mamba-Experts ($\mathbf{M^3}$). We propose an SSM layer that assigns modality-specific experts to the input, gate and output projections while sharing the recurrent core, and routes tokens at two levels: modality awareness, a parameter-free mask, fixes the expert pool a token may reach, and content routing selects one expert within it. Each modality's expert pool is shared across understanding and generation, and expert selection requires modality indices but no task identifier. Controlled ablations examine the contribution of each of the two levels (Section 4.5).

3) A native multimodal training recipe. We train both tasks natively, from the first multimodal step, in place of the usual alignment stage followed by joint training. Gradients from both tasks are accumulated into every update, a bridge class of real images supplies supervision to both streams, and a long learning-rate warmup lets the newly initialized image-side components settle. Under this recipe both training losses decrease overall during joint optimization, and the recipe is not tied to our architecture: applied unchanged to dense Transformer backbones it reaches generation quality in the same range (Appendix A.5).

2 Related work

2.1 Unified multimodal models

A unified model serves two tasks from one backbone, and existing designs differ in where they separate the two. Chameleon (Chameleon Team, 2024) and Show-o (Xie et al., 2025a) keep a single discrete visual representation for both tasks, the former with one autoregressive objective and the latter pairing autoregressive text with discrete diffusion over image tokens. Janus and Janus-Pro (Wu et al., 2024; Chen et al., 2025) separate at the visual encoder, using a semantic encoder for understanding and a VQ tokenizer for generation. UniToken (Jiao et al., 2025) concatenates continuous and discrete encodings of the same image into one sequence. BAGEL (Deng et al., 2025) separates at the backbone, using separate Transformer experts for text and ViT tokens and for VAE tokens, coupled through joint self-attention. Recent systems keep separating at the backbone or the decoder: SenseNova-U1 (Diao et al., 2026) pairs a mixture-of-transformers with flow matching in pixel space, UniDDT (Wang et al., 2026) decouples a diffusion decoder from text decoding, and InternVL-U (Tian et al., 2026) attaches a dedicated generation head to a multimodal LLM. All of these are built on Transformers, so their self-attention computation scales quadratically with the total sequence length, making long interleaved image-text contexts costly.

2.2 Hybrid architectures with recurrent sequence models

A second line of work replaces full self-attention with recurrent sequence layers: Mamba (Gu & Dao, 2023) and Mamba-2 (Dao & Gu, 2024) through selective state spaces, Gated DeltaNet (Yang et al., 2025b) through a gated delta rule, and Kimi Delta Attention (KDA) (Kimi Team, 2025) through finer-grained gating of that rule. At fixed model and state sizes, these layers have linear computational cost in sequence length and constant state-update cost per generated token.

To complement recurrent state compression with direct token interactions, several large-scale models combine recurrent layers with attention. Jamba (Lieber et al., 2024), Zamba (Glorioso et al., 2024) and Samba (Ren et al., 2025) do so with Mamba, and Nemotron-H (NVIDIA, 2025) with Mamba-2. Qwen3-Next (Qwen Team, 2025) and Kimi Linear (Kimi Team, 2025) instead use Gated DeltaNet

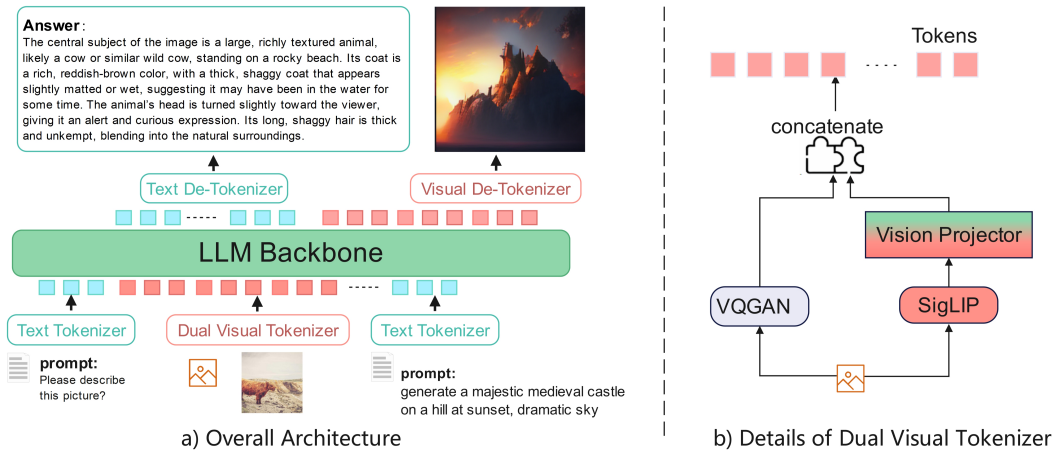

Figure 1: Overall architecture of Mambura. **(a)** Image and text tokens enter the hybrid Mamba-Transformer backbone and are detokenized back to text or images. **(b)** The dual visual tokenizer merges VQGAN discrete tokens and projected SigLIP features into one sequence.

and KDA respectively, both at a 3:1 recurrent-to-attention ratio. Qwen3.5 (Qwen Team, 2026) carries the same Gated DeltaNet hybrid into a native vision-language model, which reads images but does not generate them. Bae et al. (2025) report a favorable quality-efficiency trade-off at approximately a 1:5 attention-to-SSM ratio in their text-model experiments.

Sparsity inside these layers has also been explored, by replicating projections per modality (Liang et al., 2025) or routing projection experts by content (Zhan et al., 2025); Section 3.2.2 compares both with M^3. On the visual side, Cobra (Zhao et al., 2025) and VL-Mamba (Qiao et al., 2024) use a Mamba language backbone in a LLaVA-style visual-understanding framework, and AiM (Li et al., 2024) generates images autoregressively over VQ tokens with Mamba, each for a single task. OmniMamba (Zou et al., 2025) unifies both tasks on a pure Mamba-2 backbone, separating them by task through LoRA adapters selected by task identity and decoupled output vocabularies, with the backbone frozen while those modules are pretrained.

3 Proposed Method

This section introduces the state-space preliminaries, the Mambura architecture built around the M^3 (Modality-aware Mixture-of-Mamba-Experts) layer, and the training recipe.

3.1 Preliminaries of Mamba

Mamba-2 (Dao & Gu, 2024) is built on selective state-space models. Each head maintains a recurrent state h_t updated as

$$h_t = e^{\Delta_t A}\, h_{t-1} + \Delta_t B_t\, x_t, \qquad y_t = C_t\, h_t, \tag{1}$$

where A is a learned per-head scalar decay and B_t, C_t and the time step $\Delta_t = \mathrm{softplus}(\tilde{\Delta}_t + b_\Delta)$ are produced from the current input by linear projections, so the dynamics depend on the input and the model is *selective*. The update is computed in the state-space dual (SSD) form, a chunked matrix formulation that connects SSMs to variants of attention; the continuous-time system and zero-order-hold discretization from which it follows are given in Appendix A.1. An M^3 layer shares this recurrence across modalities, carrying the interleaved image-text sequence in one state per layer.

3.2 Model Architecture

Mambura (Figure 1) has three components: a text tokenizer, a dual visual tokenizer combining VQGAN and SigLIP, and a hybrid Mamba-Transformer backbone built on M^3 layers. Both tasks are represented as sequences processed by one backbone: understanding consumes VQ and SigLIP representations before predicting text, whereas generation conditions on text and predicts VQ tokens.

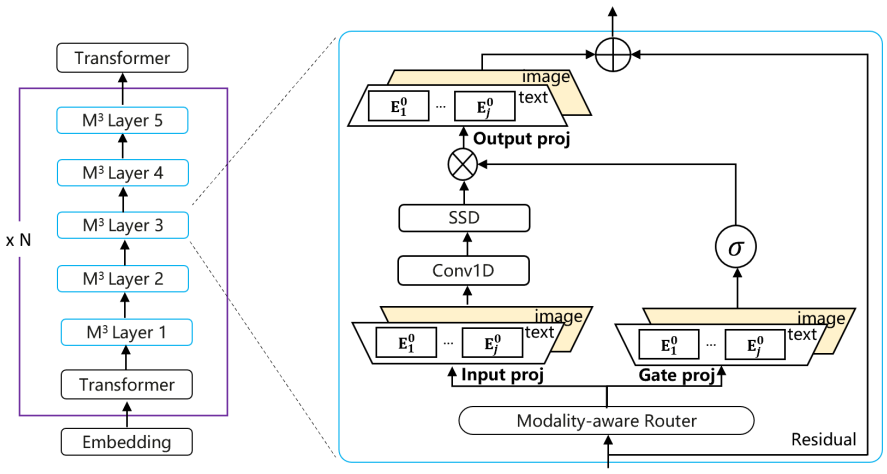

Figure 2: Mambura backbone (left) and M^3 layer (right). M^3 wraps a shared SSM core (Conv1D + SSD) with private per-modality expert projections; σ denotes SiLU. For readability the drawing omits the pre-norm and the gating branch that the input projection emits alongside the sequence branch.

3.2.1 TOKENIZER

Dual visual tokenizer. VQGAN (Esser et al., 2021) quantizes each image into 256 discrete codebook IDs that capture low-level spatial structure (object boundaries, layout, texture), and SigLIP (Zhai et al., 2023) extracts 729 continuous features rich in high-level semantics (object identity, scene category, visual relationships), projected to the model dimension by a small MLP. Both encoders are frozen. Understanding consumes both representations and generation targets the VQ tokens, so the discrete codes are the shared currency of the two tasks. In the understanding input the VQ tokens precede the SigLIP features:

$$[\text{VQ}_{1:256}]\ [\text{SigLIP}_{1:729}]\ [\text{Text}_{1:n}].$$

Under our initialization and training setup this ordering trained stably, whereas placing SigLIP features first caused the recurrent state to diverge; encoder configurations and our working explanation of the ordering effect are in Appendix A.2.

Text tokenizer. We use the Qwen3 tokenizer (Yang et al., 2025a), vocabulary 151,936.

3.2.2 LLM BACKBONE

As shown in Figure 2, the backbone stacks N identical blocks, each one Transformer layer (T) followed by five M^3 layers, a 1:5 attention-to-SSM ratio within each block inspired by Bae et al. (2025), and appends a final Transformer layer so that a global attention pass precedes the output heads. The full model therefore has $N{+}1$ Transformer and $5N$ M^3 layers. Transformer layers use RoPE (Su et al., 2021), SwiGLU (Shazeer, 2020) and Flash Attention (Dao et al., 2022); M^3 layers are built on the Mamba-2 SSD primitive.

Modality-aware Mixture-of-Mamba-Experts (M^3). The M^3 layer combines *shared computation* for cross-modal transfer with *modality-specific expert projections* for specialized processing: each token selects an expert through modality-aware routing, then passes through private projections interleaved with a shared SSM core. It differs from the two related Mamba-side designs of Section 2.2 in what it makes private and in how it selects: against Mixture-of-Mamba (Liang et al., 2025), which gives each modality one fixed set of projections, M^3 allocates several experts per modality and selects among them by content, while keeping the SSM-parameter projection shared; against Routing Mamba (Zhan et al., 2025), which routes by content alone, it adds a parameter-free modality mask before content routing.

Modality-aware routing. Given a per-sample input $\mathbf{H} \in \mathbb{R}^{L\times d}$, let $\hat{\mathbf{H}} = \mathrm{RMSNorm}(\mathbf{H})$ denote its pre-normed form and $\hat{\mathbf{h}}_t \in \mathbb{R}^d$ its t-th row; with modal indices $\mathbf{m} \in \{0,\dots,K-1\}^L$, modality-aware routing operates at two levels: modality awareness fixes the pool of experts a token may reach, and content routing selects one within it.

Table 1: Shared-private decomposition of the M^3 layer.

Operation	Description
$(\mathbf{u}, \mathbf{z}) = \text{MoE}_{\text{in}}(\hat{\mathbf{H}}; e)$	Private: input projection, $d \to 2d_{\text{inner}}$
$\mathbf{x} = \text{SiLU}(\text{CausalConv1D}(\mathbf{u}))$	Shared: depthwise causal convolution
$(\tilde{\Delta}_t, B_t, C_t) = W_x\, \mathbf{x}_t$	Shared: SSM parameters, $\Delta_t = \text{softplus}(\tilde{\Delta}_t + b_\Delta)$
$\mathbf{y} = \big[\text{SSD}(\mathbf{x}; A, \Delta, B, C) + D \odot \mathbf{x}\big] \odot \text{SiLU}(\mathbf{z})$	Shared: Mamba-2 SSD with skip D, gated by $\mathbf{z}$
$\hat{\mathbf{g}} = \text{SiLU}(\text{MoE}_{\text{gate}}(\hat{\mathbf{H}}; e))$	Private: gate projection, $d \to d_{\text{inner}}$
$\mathbf{H}' = \text{MoE}_{\text{out}}(\mathbf{y} \odot \hat{\mathbf{g}}; e) + \mathbf{H}$	Private: output projection, $d_{\text{inner}} \to d$

Level 1: modality-aware (hard constraint). A hard $-\infty$ mask constrains each token to the experts designated for its modality. For token t with modality m_t, modality k's expert range starts at s_k and spans n_k experts:

$$\text{mask}_{t,j} = \begin{cases} 0 & \text{if } s_{m_t} \le j < s_{m_t} + n_{m_t} \\ -\infty & \text{otherwise.} \end{cases} \tag{2}$$

Level 2: content routing (top-1). Within the modality's expert pool, a router computes logits from content features and a modal embedding, and hard-routes the token to the highest-scoring expert:

$$\mathbf{g}_t = W_{\text{router}}\big(\hat{\mathbf{h}}_t + \text{Emb}_{\text{modal}}(m_t)\big) + \text{mask}_t, \qquad e_t = \arg\max_j\, g_{t,j}. \tag{3}$$

With unit-weight hard top-1 routing and no straight-through estimator, the router receives no task-loss gradients; the auxiliary load-balancing objective that could train it is disabled in our main model (Appendix A.3). The two levels are therefore complementary: modality awareness guarantees that tokens of different modalities reach disjoint expert pools, while content routing makes the assignment within a pool depend on the token.

Shared-private computation. Each M^3 layer has three private projection groups (*input*, *gate*, *output*), each an expert set, around a shared core of a causal Conv1D followed by Mamba-2 SSD; the routing decision e_t is reused across all three groups. Table 1 gives the forward pass. The input projection is $d \to 2d_{\text{inner}}$ and splits into a sequence branch $\mathbf{u}$ that enters the shared core and a gating branch $\mathbf{z}$ that multiplies the core's output; the input-dependent SSM parameters come from a shared projection W_x of the convolved features, and the per-head decay A and skip weight D are shared learned parameters. Following Routing Mamba (Zhan et al., 2025), a second gate $\hat{\mathbf{g}}$ from its own expert projection is applied in the expanded d_{inner} space before the output projection returns to d.

3.3 Training recipe

Our pretraining consists of text pretraining followed directly by joint multimodal training, without a separate frozen-backbone alignment stage. Instruction tuning is applied afterwards and is described in Section 4.6.

Stage I: text pretraining. We pretrain the base model on SlimPajama (SlimPajama, 2023), covering code, Q&A, documents, and web text. During text pretraining, all tokens are routed within the text expert group, while the image-specific components remain untrained.

Stage II: native multimodal joint training. Starting from the Stage I checkpoint, we train the backbone, the token embeddings and output heads, and the vision projector jointly on both tasks, while the VQGAN and SigLIP encoders stay frozen. The understanding input follows the format above, and the generation input is

$$[\text{Caption}]\ [\text{BOI}]\ [\text{VQ}_{1:256}],$$

where BOI is a beginning-of-image token. Each optimizer step draws one batch from an understanding dataloader and one from a generation dataloader, computes the understanding loss $\mathcal{L}_u$ (cross-entropy on the text targets) and the generation loss $\mathcal{L}_g$ (cross-entropy on the 256 VQ tokens), and accumulates both gradients into a single update of $\mathcal{L}_u + \lambda_g \mathcal{L}_g$ with $\lambda_g = 1$ (Appendix A.4). During training the caption is replaced by a null condition with probability 0.1, which enables classifier-free guidance (Ho & Salimans, 2022) at inference. On the generation pathway, a mean-pooled embedding of the caption, scaled by 0.5, is added to the hidden states after every M^3 layer to counter the decay of the condition along the recurrent sequence (Appendix A.4). The image-side components are newly initialized, and

a long learning-rate warmup limits their early updates (Appendix A.5). The Stage II data combines understanding-only, generation-only, and a *bridge* class of captioned real images that is read by both dataloaders, so the same image serves as understanding input and as generation target over the course of training; the two dataloaders sample independently, so a given image need not appear in both batches of the same step. Data composition is in Appendix A.5.

4 EXPERIMENTS

4.1 SETUP

Model. The Mambura backbone has $N = 4$ blocks (5 Transformer layers + 20 M^3 layers, 25 in total) with hidden dimension $d = 1024$. Each Transformer layer uses 16 heads (d_{head}=64) and a SwiGLU intermediate dimension of 2816; each M^3 layer uses Mamba-2 (SSD) with state dimension 128, expansion factor 2 (d_{inner}=2048), 32 SSM heads, and a causal convolution of kernel size 4. With $K = 2$ modalities and $n_k = 4$ experts per modality, the model has 1.7B parameters in total and 0.39B activated per token. The frozen VQGAN and SigLIP encoders are configured as in Appendix A.2.

Training. Stage I pretrains on 142B SlimPajama tokens (sequence length 4096, 540K steps, peak LR 3×10^{-4}, AdamW with weight decay 0.1). Stage II trains jointly for 440K steps at a global batch of 256 samples per task pass (16 per GPU on 16 H200 GPUs): a 20K-step warmup to 2×10^{-4}, followed by 380K steps at that learning rate and a 40K-step cosine decay to 2×10^{-5} (AdamW β_1=0.9, β_2=0.95, weight decay 0.01, gradient clipping 1.0, bf16 mixed precision).

Evaluation. Understanding benchmarks are run with `lmms-eval` (Zhang et al., 2024) at batch size 1 under each task's default prompt; MM-Vet is graded by an open-weight judge (gemma-4-31B-it) rather than a commercial API, so its score is not directly interchangeable with published numbers. Generation follows the official GenEval protocol (Ghosh et al., 2023): 553 prompts with four images each, scored by the official object detector and CLIP color classifier, with Overall the macro-average over the six task accuracies. Images are sampled autoregressively over the 256 VQ tokens with classifier-free guidance scale 5.0, temperature 1.0, and nucleus sampling with $p = 0.95$.

4.2 INFERENCE EFFICIENCY

Efficiency is measured in bf16 on one 143 GB H200 at batch size 1: each point is the mean of five trials that prefill L tokens and then decode, with prefill chunked at 4,096 tokens wherever the implementation allows; Appendix A.6 gives the decode measurement used for each model. Every baseline runs its released implementation unmodified: the same-scale unified peers Show-o-1.3B and Janus-Pro-1B, and two architectural references, Qwen3.5-2B (Qwen Team, 2026) for a contemporary hybrid of linear and full attention and Mamba2-1.3B for a pure SSM. Mambura runs with operator-level optimizations of the M^3 inference path that leave its outputs unchanged. Figure 3 reports peak memory, prefill latency and decode throughput at 128K, 512K and 1M tokens; measurement details and the baselines' failure modes are in Appendix A.6.

At 128K, the longest context every baseline completes, the ranking already follows the architecture. The two hybrids hold the lowest peak memory, within 3% of each other at about 8.5 GiB against about 28 GiB for the full-attention models. Mambura has the lowest prefill latency among the multimodal models and the highest decode throughput, 141 tok/s against 31 for Qwen3.5-2B and 18 for Show-o and Janus-Pro; only the text-only Mamba2 prefills faster.

Beyond 128K, the families separate by the cost that binds first. Show-o and Janus-Pro exhaust the device beyond 256K and Mamba2's kernels abort, so only the two hybrids reach 1M tokens. Between them, attention share alone does not decide the scaling: both confine attention to a minority of layers, five of 25 and six of 24, yet Qwen3.5-2B's prefill time still grows near-quadratically while Mambura's grows near-linearly, so Mambura's prefill advantage widens from 2.3× at 128K to 15.8× at 1M (21.2 s against 334.7 s) and its peak memory is 26% lower at 1M. Decode is the one metric on which its lead narrows, from 4.6× to 1.7×. One plausible contributor lies outside the hybrid layout: Mambura's multi-head attention stores 20 KiB of cache per token against 12 KiB for Qwen3.5-2B's grouped-query attention (Appendix A.6). Mambura therefore remains usable on one GPU at million-token context, where the other unified models do not run at all; at batch 8 it is the only model that still completes at 512K (Figure 5).

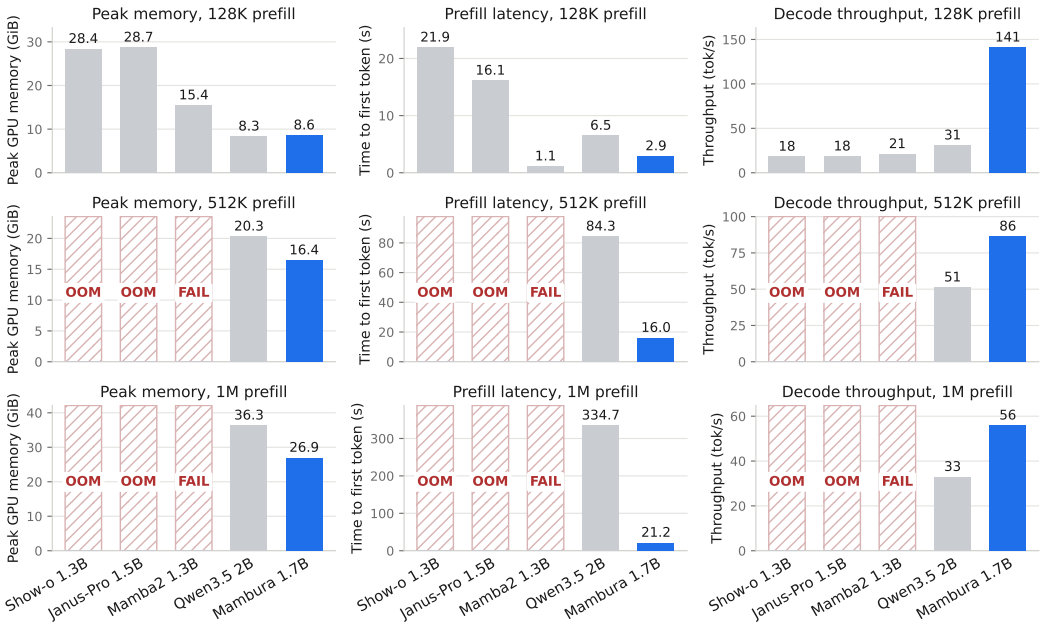

Figure 3: **Inference efficiency at** 128**K,** 512**K and** 1**M-token prefill** (batch 1, one H200, bf16; mean of five trials). Hatched bars mark runs that did not complete: OOM is an out-of-memory failure, FAIL is Mamba2's 32-bit indexing limit. Bar labels give released parameter counts, language backbone only for Show-o and Janus-Pro, as in Tables 2 and 3.

Table 2: Multimodal understanding benchmarks against published unified and understanding-only models. Parameters are total / activated; dense models activate all of theirs. Baseline scores are from the cited papers, except Chameleon's MMMU and MM-Vet (Wu et al., 2024) and Emu3-Chat's MME-P (Chen et al., 2025). Qwen3.5 is non-thinking mode, SenseNova-U1 thinking mode.

Type	Model	Params (total / act.)	POPE↑	MME-P↑	MMB↑	SEED↑	GQA↑	MMMU↑	MM-Vet↑
Und. Only	Emu3-Chat (Wang et al., 2024)	8B / 8B	85.2	1244	58.5	68.2	60.3	31.6	37.2
	InstructBLIP (Dai et al., 2023)	7B / 7B	–	–	36.0	53.4	49.2	–	26.2
	Qwen-VL-Chat (Bai et al., 2023)	7B / 7B	–	1487.5	60.6	58.2	57.5	–	–
	LLaVA-Phi (Zhu et al., 2024)	2.7B / 2.7B	85.0	1335.1	59.8	–	–	–	28.9
	Qwen3.5-2B (Qwen Team, 2026)	2B / 2B	–	–	81.3	–	–	64.2	–
	MobileVLM (Chu et al., 2023)	1.4B / 1.4B	84.5	1196.2	53.2	–	56.1	–	–
	MobileVLM-V2 (Chu et al., 2024)	1.4B / 1.4B	84.3	1302.8	57.7	–	59.3	–	–
	Qwen3.5-0.8B (Qwen Team, 2026)	0.8B / 0.8B	–	–	68.0	–	–	47.4	–
Und. and Gen.	SenseNova-U1 (Diao et al., 2026)	16.4B / 8.2B	–	–	90.3	–	–	74.8	–
	BAGEL (Deng et al., 2025)	14B / 7B	–	1687	85.0	–	–	55.3	67.2
	UniToken (Jiao et al., 2025)	7B / 7B	–	–	71.1	69.9	–	32.8	–
	Chameleon (Chameleon Team, 2024)	7B / 7B	–	–	–	–	–	22.4	8.3
	UniDDT (Wang et al., 2026)	5B / 5B	–	1699.5	82.2	76.5	62.3	52.6	–
	InternVL-U (Tian et al., 2026)	4B / 2B	–	1607.5	–	75.2	–	54.7	–
	D-DiT (Li et al., 2025)	2.0B / 2.0B	84.0	1124.7	–	–	59.2	–	–
	Janus (Wu et al., 2024)	1.5B / 1.5B	87.0	1338.0	69.4	63.7	59.1	30.5	34.3
	Janus-Pro-1B (Chen et al., 2025)	1.5B / 1.5B	86.2	1444.0	75.5	68.3	59.3	36.3	39.8
	Show-o2 (Xie et al., 2025b)	1.5B / 1.5B	–	1450.9	67.4	65.6	60.0	37.1	–
	Show-o (Xie et al., 2025a)	1.3B / 1.3B	80.0	1097.2	–	–	58.0	26.7	–
	Mambura	**1.7B / 0.39B**	**83.8**	**1107.9**	**49.3**	**46.2**	**54.5**	**33.7**	**23.3**

4.3 Understanding benchmarks

Against Show-o, a same-scale unified baseline with discrete visual tokenization, Mambura is ahead on three of the four benchmarks the two models share and behind on GQA, at 256×256 and 384×384 against Show-o's 512-resolution variant. Against the 1.5B dense Janus-Pro-1B, which uses a separate understanding encoder, it is within three MMMU points but trails on the perception-heavy benchmarks (MME-Perception, MMBench, SEED-Bench), where a perception gap to the understanding-only models remains overall, though not on every one. These results are obtained with 0.39B activated parameters per token against 0.8B to 8.2B for the baselines, and with a generation pathway that the understanding-only models do not carry; Section 5 takes up the remaining gap.

Table 3: Text-to-image generation on GenEval, ordered by activated parameters per token within each group. Parameters are total / activated; dense models activate all of theirs. Baseline scores are as reported in the cited papers, except Chameleon's, taken from Chen et al. (2025).

Type	Method	Params (total / act.)	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall↑
Gen. Only	DALL-E 3 (Betker et al., 2023)	–	0.96	0.87	0.47	0.83	0.43	0.45	0.67
	Emu3-Gen (Wang et al., 2024)	8B / 8B	0.98	0.71	0.34	0.81	0.17	0.21	0.54
	SDXL (Podell et al., 2024)	2.6B / 2.6B	0.98	0.74	0.39	0.85	0.15	0.23	0.55
	SD3-Medium (Esser et al., 2024)	2B / 2B	0.99	0.94	0.72	0.89	0.33	0.60	0.74
	SDv1.5 (Rombach et al., 2022)	0.9B / 0.9B	0.97	0.38	0.35	0.76	0.04	0.06	0.43
	PixArt-α (Chen et al., 2024)	0.6B / 0.6B	0.98	0.50	0.44	0.80	0.08	0.07	0.48
Und. and Gen.	SenseNova-U1 (Diao et al., 2026)	16.4B / 8.2B	1.00	0.96	0.92	0.92	0.91	0.76	0.91
	BAGEL (Deng et al., 2025)	14B / 7B	0.99	0.94	0.81	0.88	0.64	0.63	0.82
	UniToken (Jiao et al., 2025)	7B / 7B	0.99	0.80	0.35	0.84	0.38	0.39	0.63
	Chameleon (Chameleon Team, 2024)	7B / 7B	–	–	–	–	–	–	0.39
	UniDDT (Wang et al., 2026)	5B / 5B	0.99	0.93	0.71	0.92	0.85	0.80	0.87
	InternVL-U (Tian et al., 2026)	4B / 3.7B	0.99	0.94	0.74	0.91	0.77	0.74	0.85
	D-DiT (Li et al., 2025)	2.0B / 2.0B	0.97	0.80	0.54	0.76	0.32	0.50	0.65
	Janus (Wu et al., 2024)	1.5B / 1.5B	0.97	0.68	0.30	0.84	0.46	0.42	0.61
	Janus-Pro-1B (Chen et al., 2025)	1.5B / 1.5B	0.98	0.82	0.51	0.89	0.65	0.56	0.73
	Show-o2 (Xie et al., 2025b)	1.5B / 1.5B	0.99	0.86	0.55	0.86	0.46	0.63	0.73
	Show-o (Xie et al., 2025a)	1.3B / 1.3B	0.95	0.52	0.49	0.82	0.11	0.28	0.53
	Mambura	**1.7B / 0.39B**	**0.98**	**0.82**	**0.45**	**0.83**	**0.69**	**0.41**	**0.70**

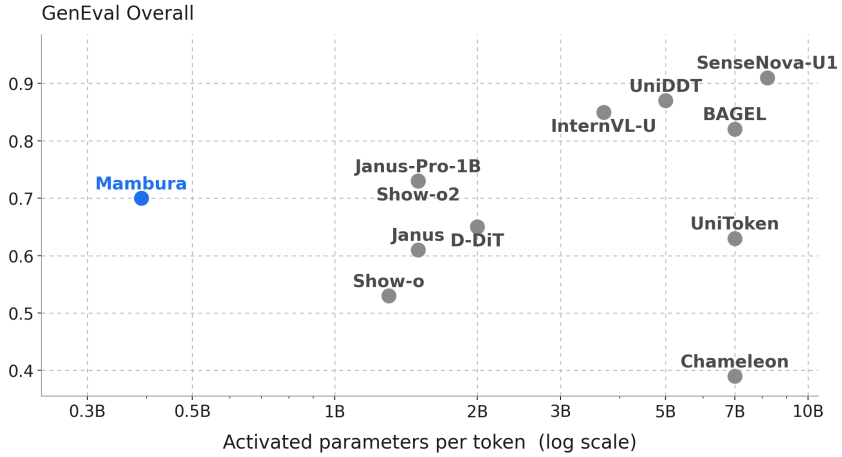

Figure 4: GenEval Overall against activated parameters per token for the unified models of Table 3.

4.4 Generation benchmarks

Table 3 reports GenEval per category, and Figure 4 plots Overall against activated parameters per token. Mambura reaches an Overall of 0.70, 0.03 below Janus-Pro-1B and Show-o2, which tie at 0.73 as the strongest same-scale unified models, while using only 26% of their activated parameters. The models above it all activate at least three times as many parameters per token, and no listed model approaches its score at a comparable activated budget. Per category, it matches Janus-Pro-1B on single-object (0.98) and two-object (0.82) accuracy and exceeds it on position accuracy (0.69 against 0.65); its largest deficit is in color attribution (0.41 against 0.56), followed by counting (0.45 against 0.51) and colors (0.83 against 0.89).

4.5 Ablation studies

We study the two levels of M^3's modality-aware routing (Section 3.2.2): *modality-aware*, the parameter-free Level-1 mask that confines each modality to its own expert pool, and *content routing*, the Level-2 selection of one expert within that pool. Each can be removed on its own, giving four arms: M^3; *modality-aware only*, where each modality is pinned to a fixed expert; *content routing only*, where selection ranges over all eight; and *neither*, a single shared projection group. All four keep eight projection experts and activate exactly one per token, so they are matched in activated capacity, and where a router is used it receives the modal embedding, so modality is observable to it. The arms start from the same Stage I checkpoint and train on the same data with matched batch sizes and schedules. Table 4 reports their benchmark scores; Appendix A.7 gives the budgets and protocol, plots the training curves, and reports the SigLIP and load-balancing comparisons.

Table 4: The four arms at 30K steps. Understanding is likelihood-scored; scores are percentages except GenEval Overall. MMB denotes MMBench-dev, bold the best arm per column.

	Understanding benchmarks				Generation benchmark (GenEval)						
Arm	POPE	MMB	AI2D	MMMU	Single	Two	Count	Color	Pos.	Attr.	Overall
M^3	67.8	**26.7**	**28.7**	29.7	81.6	23.7	10.6	53.5	21.2	9.2	0.333
modality-aware only	65.2	**26.7**	28.4	29.7	**89.7**	**37.1**	**14.4**	**63.6**	**30.0**	**12.2**	**0.412**
content routing only	67.1	26.4	28.0	30.2	77.2	17.7	1.9	47.6	10.5	6.8	0.269
neither	**70.8**	26.0	27.9	**30.6**	4.1	0.0	0.0	1.6	0.0	0.0	0.009

Modality-aware only. Within the masked pair, the fixed-expert arm leads M^3 on GenEval at 30K steps. Both arms retain the same modality partition and activate one expert per token; they differ in whether each modality uses one fixed expert or selects among four by content. On identical probe batches, the fixed expert receives two to five times the gradient norm of an individual M^3 expert (Appendix A.7), and its generation training loss is lower at the same step budget. Together, the gradient measurements and loss trajectories support the interpretation that concentrating training signals in fewer experts favors early convergence, at the cost of a smaller reachable expert pool. Whether the larger pool yields a longer-term advantage remains open: when training is extended to 100K steps, the generation training-loss gap peaks between 30K and 40K and then narrows slowly without crossing, while the per-expert gradient disparity persists (Appendix A.7).

Content routing only. Under the initialization, balancing rules and 30K-step budget used here, removing the mask lowers generation quality whether or not content routing is present, and the arm with neither mechanism barely generates. Without the mask, the *content routing only* arm outperforms the *neither* arm on generation, but still trails M^3. A routing probe at 10K steps finds image tokens distributed over all eight experts with overlapping text and image expert utilization, despite the modal embedding in the router's input; the arm has not recovered M^3's disjoint modality-specific expert pools. Sharing projection experts across modalities may make image-specific specialization more difficult, providing a possible explanation for the lower generation score. The routing probe is consistent with this interpretation, but does not establish it as the cause. This comparison does not separate the mask from the initialization and balancing conditions, which Appendix A.7 states. The understanding benchmarks do not separate the arms at this budget and are reported for completeness.

4.6 Qualitative results

After Stage II we instruction-tune Mambura with the same two-pass training structure. Figure 7 shows visual question answering on four photographs and Figure 8 text-to-image samples for eight prompts; both, with observations, are in Appendix A.8.

5 Conclusion and limitations

We presented Mambura, to our knowledge the first Mamba–Transformer hybrid to unify image understanding and generation within one backbone. Our work makes three contributions: a hybrid backbone that combines recurrent efficiency with direct global interaction; the M^3 layer, which shares a state-space core while using modality-specific projection experts; and a native training recipe that jointly optimizes both tasks from the first multimodal step without a separate frozen-backbone alignment stage. Its principal advantage is inference efficiency at long context, where the full-attention unified models we compare against exhaust the device entirely, and it reaches that while activating a fraction of the parameters those models use on every token. These results provide a promising direction for overcoming the long-context inference bottleneck of unified multimodal models.

This study has two main limitations. First, constraints on the multimodal training data, particularly on the understanding side, limit Mambura's performance on several understanding benchmarks. Future work will expand both the scale and diversity of the understanding data to strengthen its perception and reasoning capabilities. Second, due to our available compute budget, we evaluate only the current 1.7B-parameter model and cannot yet establish how the architecture scales. We plan to train a 7B-scale Mambura and evaluate it across understanding, generation, and long-context efficiency benchmarks to determine whether its quality and efficiency advantages persist at a larger scale.

AI USE STATEMENT

In this work, we used generative AI tools for two tasks that require disclosure. First, part of the Stage II generation corpus (Table 5) was produced with AI models: the synthetic compositional pool consists of images generated by FLUX.1-schnell from enumerated prompts, and the dense-recaptioned pool uses captions written by a captioning model. Second, an LLM-based assistant was used in research execution and analysis: it ran evaluation and probing scripts, aggregated logged training and evaluation records into the figures and tables, and assisted in interpreting the ablation results; in addition, MM-Vet is scored by an open-weight LLM judge as stated in the evaluation setup. We have not used generative AI tools to develop the theoretical formulation or to formulate or prove mathematical claims, and translation and qualitative or thematic analysis are not applicable to this work. Additionally, we used generative AI tools to polish the writing for clarity and consistency of terminology and to format references. The research questions, the ablation design and the training recipe were formulated by the authors. We have reviewed all AI-assisted work: every reported number was checked by the authors against the source training logs, evaluation outputs or the cited papers, and all AI-assisted text was read and approved by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The training and evaluation setup is given in Section 4.1. The appendix provides the architecture configuration (Appendix A.2), the load-balancing term and the complete training objective (Appendices A.3 and A.4), the Stage II data sources and training recipe (Appendix A.5), the inference measurement protocol for every model in Figures 3 and 5 (Appendix A.6), and the training and sampling settings of the ablations (Appendix A.7).

ETHICS STATEMENT

Mambura generates images and is trained in part on synthetic images produced by an open text-to-image model (Table 5). Like other generative models it may reproduce biases present in its training data or be used to produce misleading images. The model is a research prototype for architecture study, and we have not evaluated it for safety in deployment.

REFERENCES

Sangmin Bae, Bilge Acun, Chien-Yu Lin, Haroun Habeeb, Seungyeon Kim, Liang Luo, Junjie Wang, and Carole-Jean Wu. Hybrid architectures for language models: Systematic analysis and design insights. *arXiv:2510.04800*, 2025.

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. *arXiv:2308.12966*, 2023.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. Technical report, OpenAI, 2023.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv:2405.09818*, 2024.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-Pro: Unified multimodal understanding and generation with data and model scaling. *arXiv:2501.17811*, 2025.

Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, and Chunhua Shen. MobileVLM: A fast, strong and open vision language assistant for mobile devices. *arXiv:2312.16886*, 2023.

Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, and Chunhua Shen. MobileVLM V2: Faster and stronger baseline for vision language model. *arXiv:2402.03766*, 2024.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *ICML*, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv:2505.14683*, 2025.

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, Yan Li, Yubo Wang, Zhijie Cao, Zhiqian Lin, Zhitao Yang, Zhongang Cai, et al. SenseNova-U1: Unifying multimodal understanding and generation with NEO-unify architecture. *arXiv:2605.12500*, 2026.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming Transformers for high-resolution image synthesis. In *CVPR*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. Zamba: A compact 7b SSM hybrid model. *arXiv:2405.16712*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv:2207.12598*, 2022.

Yang Jiao, Haibo Qiu, Zequn Jie, Shaoxiang Chen, Jingjing Chen, Lin Ma, and Yu-Gang Jiang. UniToken: Harmonizing multimodal understanding and generation through unified visual encoding. In *CVPR Workshop*, 2025.

Kimi Team. Kimi Linear: An expressive, efficient attention architecture. *arXiv:2510.26692*, 2025.

Haopeng Li, Jinyue Yang, Kexin Wang, Xuerui Qiu, Yuhong Chou, Xin Li, and Guoqi Li. Scalable autoregressive image generation with Mamba. *arXiv:2408.12245*, 2024.

Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, and Peng Wang. Dual diffusion for unified image generation and understanding. *arXiv:2501.00289*, 2025.

Weixin Liang, Junhong Shen, Genghan Zhang, Ning Dong, Luke Zettlemoyer, and Lili Yu. Mixture-of-Mamba: Enhancing multi-modal state-space models with modality-aware sparsity. *arXiv:2501.16295*, 2025.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid Transformer-Mamba language model. *arXiv:2403.19887*, 2024.

NVIDIA. Nemotron-H: A family of accurate and efficient hybrid Mamba-Transformer models. *arXiv:2504.03624*, 2025.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

Yanyuan Qiao, Zheng Yu, Longteng Guo, Sihan Chen, Zijia Zhao, Mingzhen Sun, Qi Wu, and Jing Liu. VL-Mamba: Exploring state space models for multimodal learning. *arXiv:2403.13600*, 2024.

Qwen Team. Qwen2.5 technical report. *arXiv:2412.15115*, 2024.

Qwen Team. Qwen3-Next: Towards ultimate training & inference efficiency. https://qwen.ai/blog?id=e34c4305036ce60d55a0791b170337c2b70ae51d, 2025. Blog post, September 10, 2025.

Qwen Team. Qwen3.5. https://huggingface.co/Qwen/Qwen3.5-2B, 2026. Model cards, February 2026.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. In *ICLR*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Noam Shazeer. GLU variants improve Transformer. *arXiv:2002.05202*, 2020.

SlimPajama. Slimpajama-627b. https://huggingface.co/datasets/cerebras/SlimPajama-627B, 2023.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with rotary position embedding. *arXiv:2104.09864*, 2021.

Changyao Tian, Danni Yang, Guanzhou Chen, Erfei Cui, Zhaokai Wang, Yuchen Duan, Penghao Yin, Sitao Chen, Ganlin Yang, Mingxin Liu, Zirun Zhu, Ziqian Fan, Leyao Gu, Haomin Wang, Qi Wei, Jinhui Yin, Xue Yang, Zhihang Zhong, Qi Qin, Yi Xin, Bin Fu, Yihao Liu, Jiaye Ge, Qipeng Guo, Gen Luo, Hongsheng Li, Yu Qiao, Kai Chen, and Hongjie Zhang. InternVL-U: Democratizing unified multimodal models for understanding, reasoning, generation and editing. *arXiv:2603.09877*, 2026.

Shuai Wang, Liang Li, Yang Chen, Ruopeng Gao, Yao Teng, and Limin Wang. UniDDT: Unifying multimodal understanding and generation with decoupled diffusion transformer. *arXiv:2606.16255*, 2026.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need. *arXiv:2409.18869*, 2024.

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

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single Transformer to unify multimodal understanding and generation. In *ICLR*, 2025a.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. In *Advances in Neural Information Processing Systems*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv:2505.09388*, 2025a.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated Delta Networks: Improving Mamba2 with delta rule. In *ICLR*, 2025b.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

Zheng Zhan, Liliang Ren, Shuohang Wang, Liyuan Liu, Yang Liu, Yeyun Gong, Yanzhi Wang, and Yelong Shen. Routing Mamba: Scaling state space models with mixture-of-experts projection. *arXiv:2506.18145*, 2025.

Kaichen Zhang, Bo Li, Sicheng Pu, Songyang Liu, Yuanhan Cui, Linrui Yu, Yifei Pan, Hao Zhang, Yikai Yang, Wenhao Mai, Wenchao Liu, Tianle Pang, Yang Cao, Linyu Liu, Tianyi Liu, Pan Lu, Edward J. Hu, Yifan Yang, and Ziwei Liu. LMMs-Eval: Reality check on the evaluation of large multimodal models. *arXiv:2407.12772*, 2024.

Han Zhao, Min Zhang, Wei Zhao, Pengxiang Ding, Siteng Huang, and Donglin Wang. Cobra: Extending Mamba to multi-modal large language model for efficient inference. In *AAAI*, 2025.

Yichen Zhu, Minjie Zhu, Ning Liu, Zhicai Ou, Xiaofeng Mou, and Jian Tang. LLaVA-Phi: Efficient multi-modal assistant with small language model. *arXiv:2401.02330*, 2024.

Jialv Zou, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. OmniMamba: Efficient and unified multimodal understanding and generation via state space models. *arXiv:2503.08686*, 2025.

A Appendix

A.1 State-space model preliminaries

State-space models map an input $x(t)$ through a latent state $h(t)$ to an output $y(t)$ via the linear ODE with parameters A, B, C:

$$\frac{dh(t)}{dt} = A\,h(t) + B\,x(t), \qquad y(t) = C\,h(t). \tag{4}$$

Applying zero-order-hold discretization with a time step Δ_t (Gu & Dao, 2023) gives:

$$\overline{A}_t = \exp(\Delta_t A), \qquad \overline{B}_t = (\Delta_t A)^{-1}(\exp(\Delta_t A) - I) \cdot \Delta_t B_t, \tag{5}$$

$$h_t = \overline{A}_t\,h_{t-1} + \overline{B}_t\,x_t, \qquad y_t = C_t\,h_t. \tag{6}$$

Because B_t, C_t and Δ_t depend on the input, the resulting transition and readout vary across tokens, making the model selective. Mamba-2 (Dao & Gu, 2024) restricts A to a scalar per head and uses the first-order approximation $\overline{B}_t \approx \Delta_t B_t$, yielding the recurrence $h_t = e^{\Delta_t A} h_{t-1} + \Delta_t B_t x_t$ used in the main text. This recurrence is computed in the state-space dual (SSD) form.

A.2 Architecture details

Visual encoders. The VQGAN (Chameleon checkpoint (Chameleon Team, 2024)) uses a codebook of 16,384 entries and a downsampling factor of 16, producing $16\times16 = 256$ tokens per 256×256 image. SigLIP-SO400M-Patch14-384 processes the same source image resized to 384×384, yielding $27\times27 = 729$ patches of dimension 1152, projected to the model dimension 1024 by a 2-layer MLP with a GELU non-linearity. Both encoders are frozen during all training stages. VQ tokens are computed offline and stored with the training data, whereas SigLIP features are extracted online from the frozen encoder during training.

Ordering of the two representations. In our preliminary experiments, with the projector weights drawn from $\mathcal{N}(0, 0.02^2)$ (the initialization also used in the final model), placing SigLIP features before any VQ token caused the recurrent state to diverge during training, whereas placing VQ tokens first trained stably. Our working explanation is as follows. SigLIP features come from a frozen external encoder through a newly initialized projector, so at the start of training their statistics need not match what the pretrained SSM expects, and in a recurrent model a mismatch at the head of the sequence enters the state before any other evidence is present. VQ tokens are discrete codebook IDs embedded through the model's own image embedding table, whose scale is matched to the text embeddings, and they encode the same image that the subsequent SigLIP features represent. We hypothesize that a VQ prefix lets the recurrent state settle on representations closer to the pretrained regime before the projected features arrive. Matching the embedding scale does not by itself align the two representation spaces, and we have not isolated this mechanism beyond the ordering experiment; we report it as an explanation consistent with the observation rather than as an established cause.

Scaling axes. Beyond vertical scaling through the number of blocks N, the architecture scales horizontally along two axes, the number of modalities K and the number of experts per modality n_k, and can in principle extend to $K > 2$ through per-modality tokenizers, expert partitions, and output heads.

A.3 Load-balancing objective

Because $\arg\max$ selection is not differentiable and no straight-through estimator is used, the task loss does not reach W_{router} through expert selection; the selected expert projections receive the task gradient directly. An auxiliary per-modality load-balancing objective can be defined on the soft routing probabilities,

$$\mathcal{L}_{\text{LB}} = \frac{1}{K}\sum_{k=0}^{K-1} n_k \sum_{j=s_k}^{s_k+n_k-1} \left(\bar{p}_j^k\right)^2, \qquad \bar{p}_j^k = \frac{1}{|T_k|}\sum_{t\in T_k} \mathrm{softmax}(\mathbf{g}_t)_j, \tag{7}$$

where $T_k = \{t : m_t = k\}$ and $\bar{p}_j^k$ is the mean routing probability of expert j over tokens of modality k. Each modality-specific term, and therefore their average, has a minimum value of 1 when its mean

routing probabilities are uniform, and grows quadratically as they concentrate. We compute this objective independently at every M^3 layer and average it across layers for each task pass. It constrains the average soft probability rather than the realized top-1 assignment, so it encourages balanced mean routing probabilities without guaranteeing balanced expert utilization. Our main model is trained with this term switched off; its effect when enabled is examined in Appendix A.7.

A.4 Training objective

During Stage II, the model is trained with autoregressive next-token prediction. The understanding loss is the cross-entropy over valid text token positions:

$$\mathcal{L}_u = \frac{1}{N_u} \sum_{t \in \mathcal{T}_u} \mathrm{CE}(\mathbf{z}_t \in \mathbb{R}^{V_{\text{total}}},\ x_{t+1}) \tag{8}$$

where $\mathcal{T}_u = \{t : x_{t+1} \text{ is a non-padding text token}\}$ is the set of prediction positions whose target supervises the text stream, and $N_u = |\mathcal{T}_u|$. The logits $\mathbf{z}_t$ are the concatenation of the text-head and image-head outputs, so the cross-entropy is computed over the full vocabulary $V_{\text{total}} = V + V_{\text{img}} = 151{,}936 + 16{,}384 = 168{,}320$ (Qwen3 vocabulary plus VQ codebook), while the supervision targets are text tokens only.

The generation loss is the cross-entropy over the 256 VQ token positions:

$$\mathcal{L}_g = \frac{1}{N_g} \sum_{i=0}^{N_g - 1} \mathrm{CE}(\mathbf{z}_i \in \mathbb{R}^{V_{\text{img}}},\ v_i) \tag{9}$$

where $N_g = 256$ (all VQ positions are valid; no padding), $\mathbf{z}_i \in \mathbb{R}^{V_{\text{img}}}$ is the image logits for predicting the i-th VQ token v_i given the caption, the BOI token and $v_{<i}$ ($V_{\text{img}} = 16{,}384$ for the VQ codebook).

The joint loss combines both task losses, optionally with the load-balancing term of Eq. 7, which each task pass produces once, written $\mathcal{L}^u_{\text{LB}}$ and $\mathcal{L}^g_{\text{LB}}$:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_u + \lambda_g \mathcal{L}_g + \lambda \left(\mathcal{L}^u_{\text{LB}} + \mathcal{L}^g_{\text{LB}}\right). \tag{10}$$

Our main model uses $\lambda_g = 1$ and $\lambda = 0$, so the balancing term is inactive and the router receives no gradient; the effect of $\lambda = 0.01$ is examined in the ablations.

Per-Mamba-layer condition injection. The design motivation is that, in a recurrent layer, the influence of the caption on later image positions may weaken with distance. We therefore add a pooled caption embedding to the hidden states after each M^3 layer on the generation pathway:

$$\mathbf{c} = 0.5 \cdot \mathrm{MeanPool}\big(\mathrm{Emb}_{\text{text}}(\text{caption tokens})\big), \qquad \mathbf{H}_\ell \leftarrow \mathbf{H}_\ell + \mathbf{c}, \tag{11}$$

where the mean is taken over the non-padding caption tokens only.

Learnable VQ position embedding. Mamba processes tokens sequentially and has no explicit two-dimensional positional encoding, which matters for image tokens flattened from a 16×16 grid via raster scan. To address this, we introduce a learnable position embedding table for VQ tokens:

$$\mathbf{e}^{\text{vq}}_i = \mathrm{Emb}_{\text{image}}(v_i) + \mathrm{Emb}_{\text{pos}}(i), \quad i \in [0, 255] \tag{12}$$

where $\mathrm{Emb}_{\text{pos}} \in \mathbb{R}^{256 \times 1024}$ is initialized with $\mathcal{N}(0, 0.02^2)$. This adds only 0.26M parameters and negligible computational overhead.

A.5 Training recipe details

Comparison with prior training pipelines. Chameleon (Chameleon Team, 2024) pretrains a mixed-modal Transformer jointly from scratch and subsequently applies an SFT-based alignment stage, without a frozen-backbone modality-alignment phase. Janus (Wu et al., 2024) and Janus-Pro (Chen et al., 2025) use three stages: train the adapters and image head with the language backbone frozen, jointly train the backbone and image-side components, and then instruction-tune the model. Show-o (Xie et al., 2025a) also uses three stages, first learning image-token embeddings and pixel dependencies, then image-text alignment, and finally fine-tuning on high-quality data; both task streams are present beyond the first stage. BAGEL (Deng et al., 2025) uses four stages with increasing

data scale and resolution: its initial alignment stage trains only the vision connector with the language backbone frozen, while all parameters except the VAE are trainable during subsequent pretraining.

Joint training without a separate alignment stage. Mambura instead enters joint training directly from the Stage I checkpoint: the backbone, token embeddings, output heads and vision projector are trainable from the first step, with the VQGAN and SigLIP encoders frozen. The newly initialized image-side components enter at small scale (projector weights drawn from $\mathcal{N}(0, 0.02^2)$), and the learning-rate warmup described below limits the size of the early updates; in our runs this was sufficient for stable joint training from the start.

Why a long-warmup schedule. At the start of Stage II, the image-side components (VQ token embeddings, vision projector, per-modality image experts, and learnable VQ position embeddings) remain at their initialization after Stage I, while the text backbone is pretrained. The warmup raises the learning rate gradually, which limits early parameter updates while these components adapt.

Classifier-free guidance. On the generation pathway, the text condition is replaced with a null condition with probability 0.1 during training, enabling CFG sampling at inference time.

The recipe on other backbones. Using the same Stage II data mixture, two-pass objective and training schedule, dense Qwen2.5-1.5B and Qwen2.5-0.5B backbones (Qwen Team, 2024) reach GenEval scores of 0.713 at 120K steps and 0.677 at 100K steps respectively. These results show that the recipe transfers beyond the M^3 architecture.

Stage II data composition. Table 5 lists the Stage II sources by the role they play. The understanding-only sources provide image-conditioned text supervision for visual perception and reasoning; the four generation-only pools cover the VQ token distribution required for image generation, spanning artistic, photographic and synthetic compositional images; and the bridge class is included in both dataloaders, so the same image serves as understanding input and as generation target over the course of training, and its sample count contributes to both streams. The two dataloaders sample independently.

Table 5: Stage II training data, counted after per-source sampling as the records each dataloader draws from. The bridge class is read by both dataloaders and contributes to both streams; the four generation-only pools feed the generation pass only.

Role	Und. records	Gen. records	Sources
Understanding-only	2.37M	—	LLaVA, GQA, ALLaVA-VFlan, ChartQA, DocVQA, UReader, SciGraphQA, Geo170K, OCRVQA, TextVQA
Bridge (both streams)	2.77M	2.77M	COCO-Cap, SA-1B-Cap, Visual Genome, LLaVA-Pretrain, ALLaVA-LAION
Generation-only pools			
Artistic pairs	—	4.71M	JourneyDB, Midjourney V6
Dense-recaptioned photographs	—	10.52M	ImageNet-21K, COCO and Visual Genome images with model-written captions
Web photographs	—	6.73M	ReLAION
Synthetic compositional	—	1.57M	FLUX.1-schnell generations from enumerated compositional prompts
Total	5.14M	26.29M	

A.6 Inference measurement details

Each model in Figures 3 and 5 runs in its own process on one H200 in bf16, at batch size 1 and 8 respectively. Time to first token is the wall-clock time of the prefill; baseline decode throughput is measured over 49 autoregressive steps after prefill, while for Mambura it is measured over 40 decode-step evaluations per trial at a fixed context length; peak memory is the maximum allocated during prefill. Prefill is chunked at $4{,}096$ tokens for every implementation that supports multi-token continuation. For Show-o and Janus-Pro we benchmark the autoregressive language backbone used by the released unified model, with its maximum position range extended to 1M so that rotary buffers cover

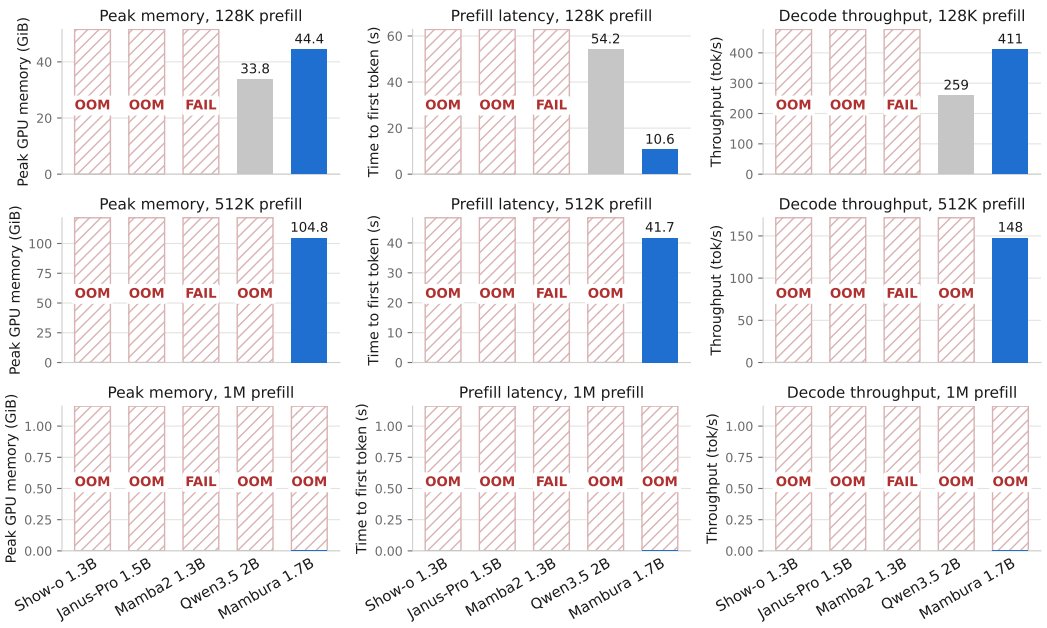

Figure 5: **Inference efficiency at batch size** 8, otherwise the protocol of Figure 3 (one H200, bf16; mean of five trials). Hatched bars mark runs that did not complete: OOM is an out-of-memory failure and FAIL is Mamba2's 32-bit indexing limit. No model completes at 1M.

the evaluated lengths; Qwen3.5-2B runs its released `Qwen3_5ForConditionalGeneration` class in bf16 with no kernel or implementation overrides.

Failure modes of the baselines. Show-o and Janus-Pro use full multi-head attention: their peak memory reaches about 54 GiB at 256K and exceeds the device at 512K, and their decode throughput decreases sharply between 128K and 256K as the cache grows. Mamba2's native selective-scan kernel hits a 32-bit indexing limit under one-shot prefill beyond 128K, and its reference inference API supports only single-token continuation after the initial prefill, which blocks a chunked workaround; Mambura's chunked prefill keeps every kernel call below that bound.

Decode behaviour at long context. Mambura's five attention layers use full multi-head attention with a hidden size of 1024. In bf16 their KV cache requires about 20 KiB per token across the five layers, totalling 20 GiB at 1,048,576 tokens, whereas the recurrent states of the twenty M^3 layers have a fixed size independent of sequence length. The growing attention cache and the corresponding full-attention computation reduce decode throughput from 141 to 56 tok/s between 128K and 1M. Qwen3.5-2B carries six full-attention layers with grouped-query attention, two key-value heads of dimension 256, which is 12 KiB per token. Mambura's cache is therefore about $1.7\times$ larger per token even though its hidden size is half as large, because multi-head attention stores all sixteen heads while grouped-query attention stores two. Qwen3.5-2B's decode throughput is not monotonic over the evaluated lengths, 31, 51 and 33 tok/s at 128K, 512K and 1M, whereas Mambura's decreases with context length. Mambura is faster at every evaluated length, by $1.7\times$ at 1M.

Batch behaviour. Figure 5 repeats Figure 3 at batch size 8. Batching multiplies every length-dependent term, so the context each architecture can reach falls: at 128K only the two hybrids complete, at 512K only Mambura, and at 1M none. At 128K Mambura prefills $5.1\times$ faster and decodes $1.6\times$ faster than Qwen3.5-2B but holds 31% more memory, up from 3% at batch 1, because batching scales the per-token cache difference above while the fixed terms do not grow.

A.7 Additional ablations

Training budgets and evaluation. The four-arm ablation in Section 4.5 uses a five-pool generation mixture (artistic pairs, joint image–text pairs, dense-recaptioned photographs, web photographs and synthetic compositional images) on eight GPUs, a 20K-step warmup to 2×10^{-4} followed by a constant learning rate, and a 30K-step budget with $\lambda = 0.01$. The earlier single-shared-

projection comparison uses the main-model recipe with $\lambda = 0$ for 20K steps. The reported training cross-entropies for these two comparisons are averages over their respective final 2K steps. Ablation understanding benchmarks use answer-option likelihood on POPE, MMBench-dev, AI2D and MMMU. All GenEval ablations use CFG 5.0, temperature 1.0 and no top-k filtering, with the same sampling seeds across the arms of a comparison.

Modality-awareness results and scope. At 30K steps, M^3 achieves mean training cross-entropies of 1.2480 for understanding and 7.7803 for generation, compared with 1.2908 and 7.9090 respectively for the *content routing only* arm. Figure 6 shows the trajectories, and Table 4 reports the downstream scores. The generation benchmark improves across categories, while understanding benchmarks show no consistent ordering at this budget.

Each arm is a single training run. Their common Stage I initialization already contains modality structure: experts 0–3 were trained on text, while experts 4–7 remain at initialization. Removing the mask therefore exposes text tokens to previously untrained experts. In addition, the masked arm uses the per-modality balancing objective of Eq. 7, whereas the *content routing only* arm balances mean routing probabilities globally over all eight experts. Equal λ does not make these objectives identical. This comparison evaluates masking under the inherited initialization and these balancing rules; it does not isolate the mask independently of either condition. A 10K-step routing probe finds that image tokens in the *content routing only* arm use all eight experts, with a utilization entropy of 1.84 out of a maximum of $\log 8 \approx 2.079$. Writing p^{img} and p^{txt} for the top-1 expert utilization histograms of the two modalities over the eight experts of a layer, their overlap is $\sum_j \min(p_j^{\text{img}}, p_j^{\text{txt}})$, which is 0 when the two modalities use disjoint experts and 1 when they use the same distribution. Averaged over the twenty M^3 layers, as the entropies above are, this overlap is 0.29; the arm has not recovered a disjoint modality partition. This observation does not establish why its generation score is lower.

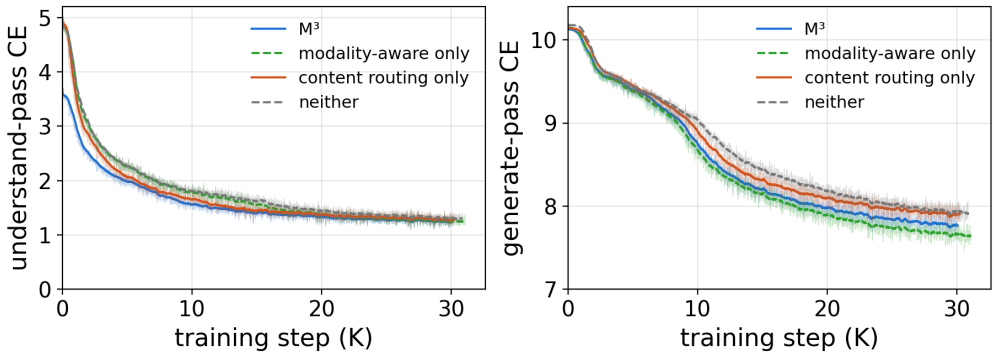

Figure 6: Training cross-entropy for the four arms; heavier lines are 300-step means.

A single shared projection group. An earlier comparison routes all tokens to one fixed projection group and bypasses the router. Each token still activates one group, but the number of groups used across tokens falls from eight to one. At 20K steps under the main-model recipe, M^3 leads on both objectives (understanding cross-entropy 1.217 against 1.317, accuracy 69.6% against 68.0%, generation cross-entropy 7.733 against 7.797), POPE is 78.5 against 61.8, and GenEval 0.030 against 0.007. This comparison jointly changes expert sharing and selection, so it cannot attribute the gain to either mechanism alone. Within the four-arm ablation in Section 4.5, M^3 and the *content routing only* arm each retain eight experts in total and activate one per token; their comparison is subject to the initialization and balancing conditions described above.

The SigLIP arm of the dual tokenizer. Two runs with the same data, schedule and step budget differ only in whether the SigLIP arm is present. Table 6 reports likelihood-scored understanding at 100K and 150K and GenEval at 80K and 110K using the ablation evaluation settings above. The arm carries perception: POPE is about 35 points higher with it at both checkpoints and AI2D 3 to 4 points higher, while MMBench and MMMU differ by less than one percentage point between the two arms, with the direction of the differences reversing between 100K and 150K steps. On the generation side, GenEval with the arm is lower by 0.010 at 80K and by 0.003 at 110K. These results support the dual-tokenizer design: adding the SigLIP branch substantially improves POPE and AI2D, with small observed reductions in GenEval at the evaluated checkpoints.

Table 6: The SigLIP arm on and off. Left: likelihood-scored understanding at 100K and 150K. Right: GenEval at 80K and 110K.

Understanding (likelihood-scored)				
Arm	POPE	MMB	AI2D	MMMU
without SigLIP arm, 100K	50.4	26.7	24.6	31.3
with SigLIP arm, 100K	**85.6**	27.1	29.0	31.9
without SigLIP arm, 150K	50.2	27.9	26.1	31.4
with SigLIP arm, 150K	**85.1**	27.7	29.0	30.6

GenEval		
Arm	80K	110K
without SigLIP arm	0.585	0.620
with SigLIP arm	0.575	0.617

The load-balancing term. Our main model is trained with the auxiliary term of Eq. 7 inactive. A rerun that wires the term in with $\lambda = 0.01$ and changes nothing else brings the soft-routing balance metric to about 1.002 within 2K steps, close to its minimum of 1, but at the last checkpoint the two runs share (30K) it trails the unbalanced run on every benchmark (Table 7), so the term stays off at this scale.

Table 7: The load-balancing term at 30K steps, same recipe and seeds.

Arm	GenEval	POPE	MMB	AI2D	MMMU
term inactive (shipped)	**0.386**	**76.9**	26.7	29.6	30.0
term active, $\lambda = 0.01$	0.353	72.1	26.1	28.5	29.8

Capacity against gradient density. The single-expert and four-expert masked arms of Section 4.5 trade image-side projection capacity against the gradient each expert receives. At 30K steps the single-expert arm leads on GenEval, and its advantage in generate-pass training cross-entropy is still widening over the final five 2K-step windows (0.091, 0.098, 0.104, 0.107 and 0.109 nats). Both arms were then carried to 100K steps. The gap peaks at about 0.112 nats between 30K and 40K and narrows slowly thereafter, to 0.105 nats over 90K to 100K, without crossing. A gradient probe on the 30K and 100K checkpoints measures the signal each expert actually receives (Table 8). On identical batches, the single expert of the *modality-aware only* arm receives a batch-mean gradient of norm 0.35 on the text side and 0.54 on the image side, whereas each of M^3's four experts receives 0.11 to 0.15. On the image side the four M^3 experts together receive about 0.48, close to the single expert's 0.54, so the same signal is being divided among them. At 100K the picture is unchanged: the single expert receives 0.34 and 0.59, each M^3 expert 0.11 to 0.15, and the four image experts together 0.58 against 0.59. Together, the gradient measurements and loss trajectories support the interpretation that concentrating training signals in fewer experts favors early convergence, at the cost of a smaller reachable expert pool. Whether the larger expert pool eventually yields a performance advantage remains open: the fixed-expert arm retains a lower generation training loss through 100K steps, although the gap narrows after peaking between 30K and 40K.

Table 8: Gradient norm received by each expert at 30K and 100K steps. Both arms see the same 1,024 understanding and 1,024 generation samples (four global batches of 256); the entry is the L2 norm of the batch-mean gradient over an expert's three projection matrices across all twenty M^3 layers, with the understanding and generation passes summed as in training. Image experts are global experts 4 to 7. The zeros are exact: those experts receive no gradient.

	Text experts 0–3				Image experts 4–7			
Arm	0	1	2	3	4	5	6	7
M^3, 30K	0.114	0.153	0.127	0.115	0.115	0.123	0.123	0.122
modality-aware only, 30K	0.351	0	0	0	0.537	0	0	0
M^3, 100K	0.112	0.122	0.126	0.110	0.143	0.145	0.141	0.153
modality-aware only, 100K	0.342	0	0	0	0.590	0	0	0

A.8 Qualitative results

Understanding. Figure 7 presents four photographs with targeted questions after the joint instruction tuning described in Section 4.6. The responses identify a Highland cow; list glasses, a laptop and a

mouse on a table, noting that the laptop is closed; read the visible word “LONDON” on the boat; and describe the foreground bicycle with a wicker basket on its rear rack. These observations illustrate recognition, enumeration, text reading and spatial description. The figure reproduces the model’s answers verbatim.

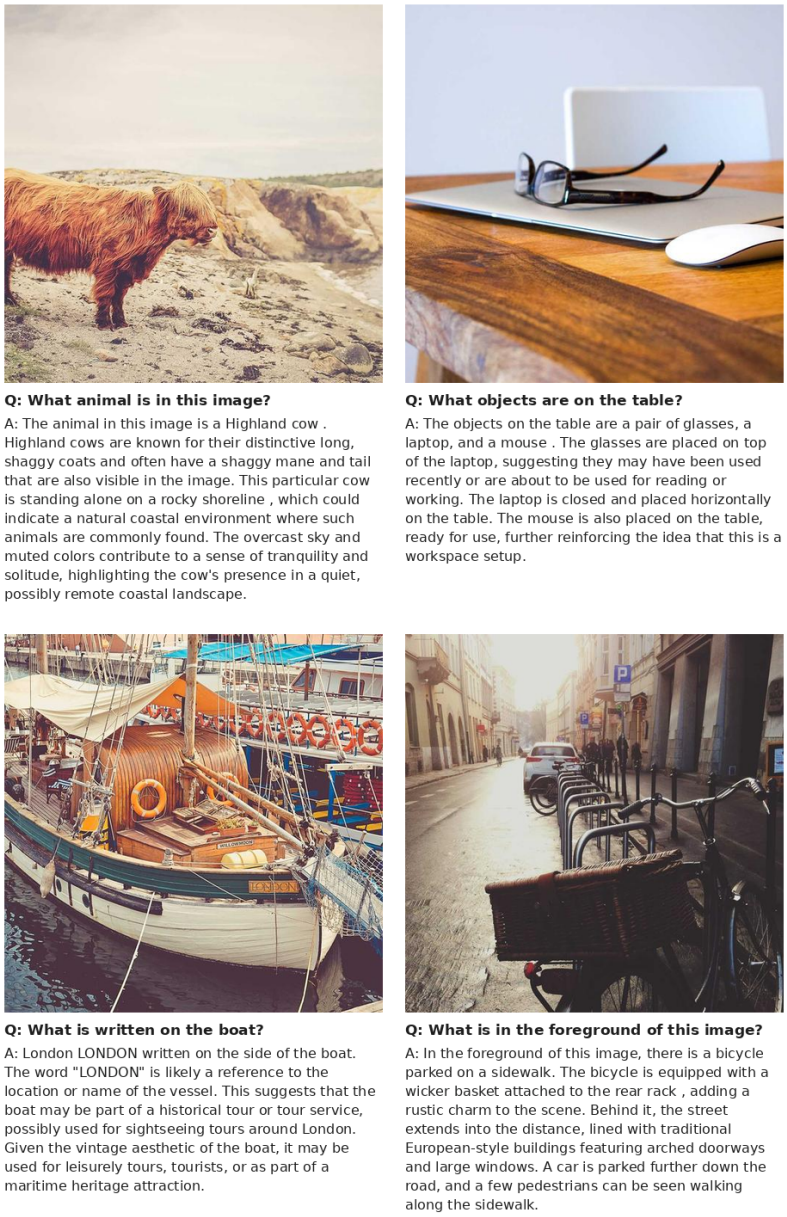

Figure 7: Visual question answering after instruction fine-tuning: four photographs, one targeted question each, and the model’s answer verbatim.

Generation. Figure 8 presents one 256×256 sample for each of eight prompts, ranging from short object descriptions to longer instructions about setting, lighting and style. The dog portrait and pizza have recognizable subjects; the living room and coastline show coherent scene layouts; and the cathedral includes visible columns and arched windows. These examples illustrate the range of generated content, while fine details remain imperfect. Comparisons of fine visual detail with higher-resolution models should account for differences in output resolution and visual token budgets.

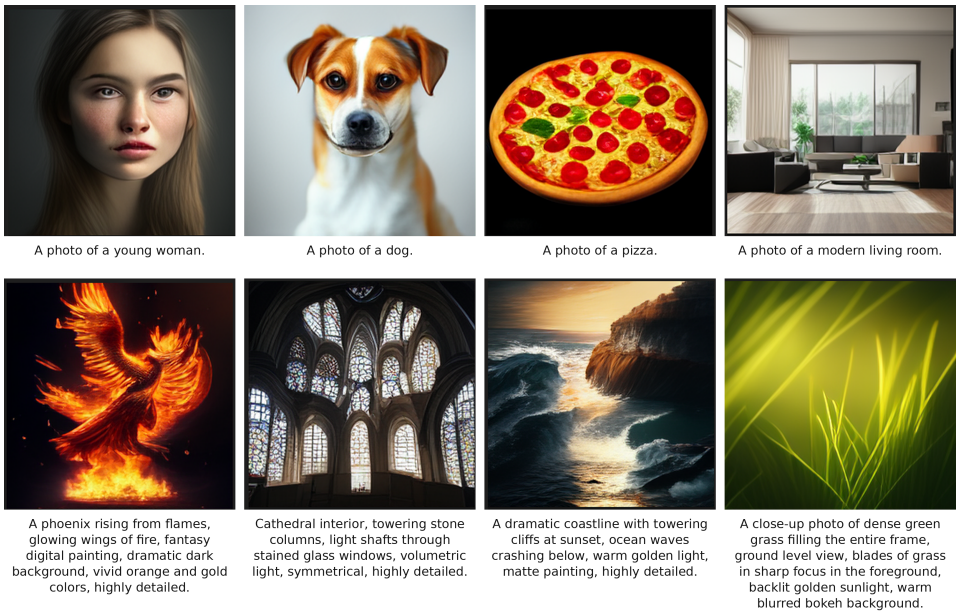

Figure 8: Text-to-image samples from Mambura across eight prompts. Each image is generated at 256×256 resolution using only 256 discrete visual tokens arranged on a 16×16 grid. The examples demonstrate recognizable subjects and coherent scene layouts within this compact visual token budget.