Jamba

Jamba is a state-of-the-art, hybrid SSM-Transformer LLM. Jamba is the first production-scale Mamba implementation, which opens up interesting research and application opportunities.

This model card is for the base version of Jamba. It’s a pretrained, mixture-of-experts (MoE) generative text model, with 12B active parameters and a total of 52B parameters across all experts. It supports a 256K context length, and can fit up to 140K tokens on a single 80GB GPU.

Paper

Jamba:A Hybrid Transformer-Mamba Language Model

Model: https://huggingface.co/ai21labs/Jamba-v0.1

- we end up with a powerful model that fits in a single 80GB GPU
- the model presents strong results for up to 256K tokens context length
- Taking advantage of both model families, Jamba combines Transformer and Mamba layers, at a certain ratio. Varying the ratio of Transformer/Mamba layers allows balancing memory usage, efficient training, and long context capabilities.

Model Architecture

Jamba is a hybrid decoder architecture

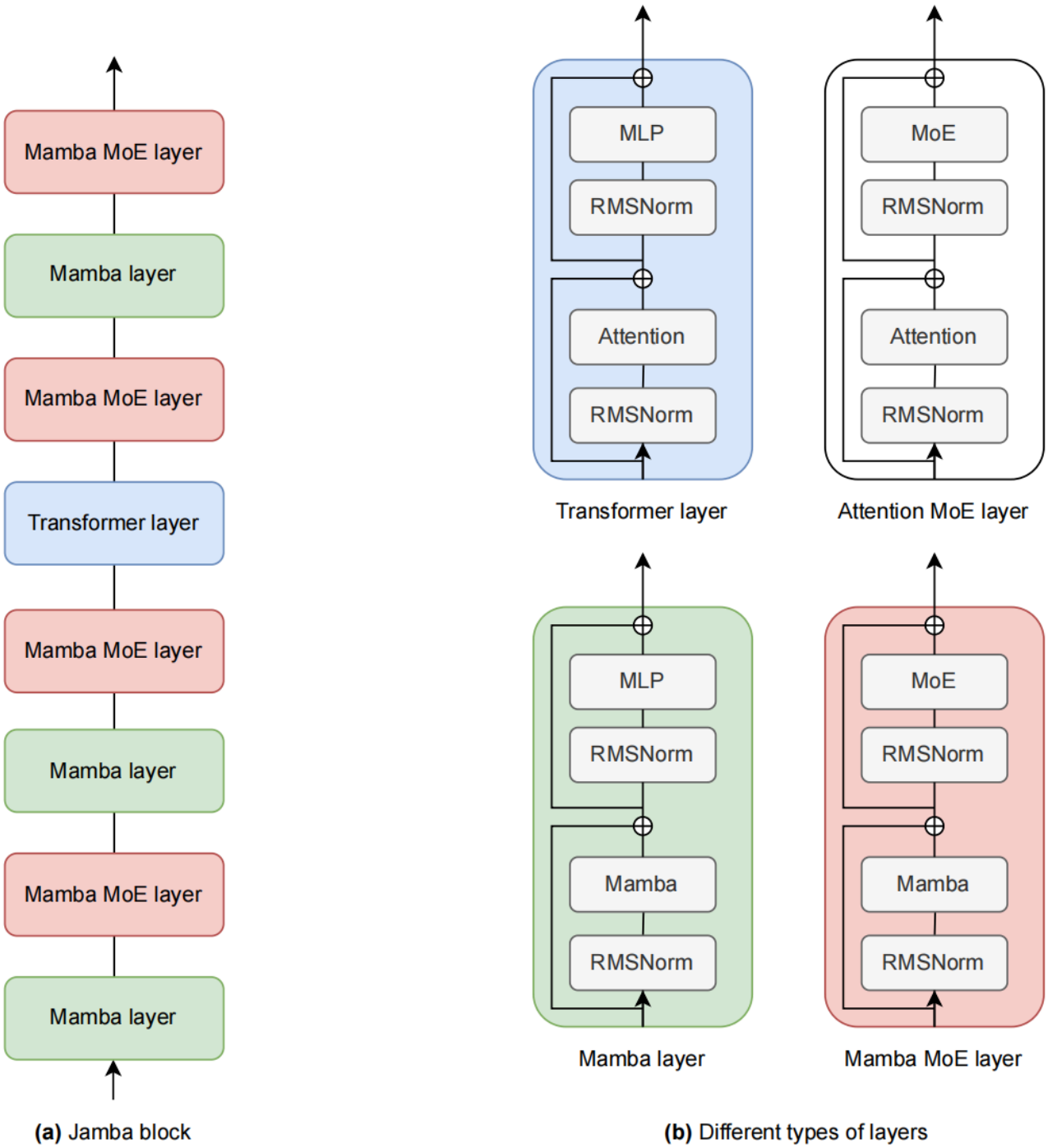


Figure 1: **(a)** A single Jamba block. **(b)** Different types of layers. The implementation shown here is with $l = 8$, $a : m = 1 : 7$ ratio of attention-to-Mamba layers, and MoE applied every $e = 2$ layers.

1. the KV cache – the memory required to store the attention keys and values in the context. When scaling Transformer models to long contexts, the KV cache becomes a limiting factor. Trading off attention layers for Mamba layers reduces the total size of the KV cache.

Our architecture aims to provide not only a small number of active parameters but also an 8x smaller KV cache compared to a vanilla Transformer.

	Available params	Active params	KV cache (256K context, 16bit)
LLAMA-2	6.7B	6.7B	128GB
Mistral	7.2B	7.2B	32GB
Mixtral	46.7B	12.9B	32GB
Jamba	52B	12B	4GB

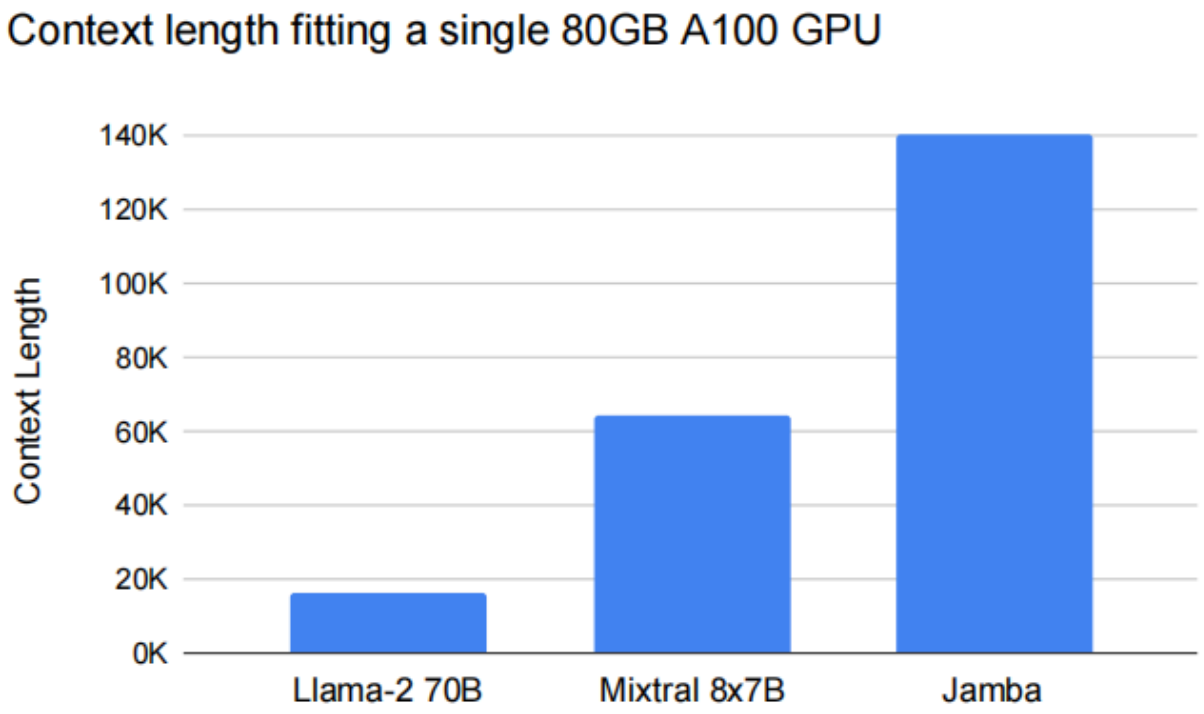
Table 1: Comparison of Jamba and recent open models in terms of total available parameters, active parameters, and KV cache memory on long contexts. Jamba provides a substantial reduction in the KV cache memory requirements.

Reaping the Benefits

Jamba Implementation for a Single 80GB GPU

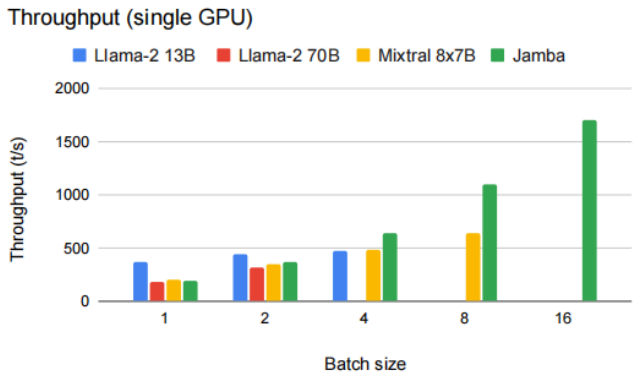
Jamba Implementation for a Single 80GB GPU. In our implementation we have a sequence of 4 Jamba blocks. Each Jamba block has the following configuration:

- l = 8: The number of layers.
- a : m = 1 : 7: ratio attention-to-Mamba layers.
- e = 2: how often to use MoE instead of a single MLP.
- n = 16: total number of experts.
- K = 2: number of top experts used at each token

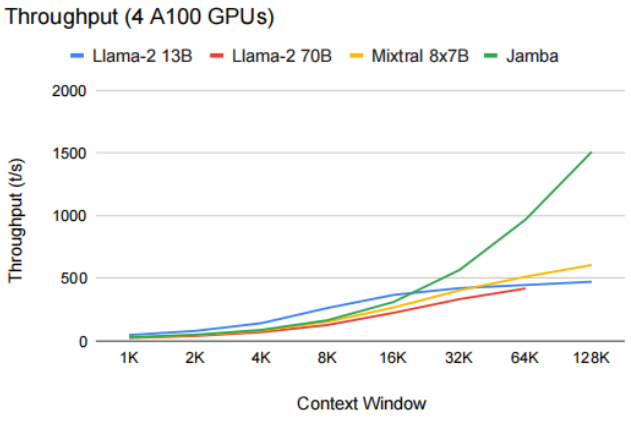


Overall, our Jamba implementation was successfully trained on context lengths of up to 1M tokens. The released model supports lengths of up to 256K tokens.

Throughput Analysis



(a) Throughput at different batch sizes (single A100 GPU, 8K context length). Jamba allows processing large batches, with a throughput 3x greater than Mixtral.



(b) Throughput at different context lengths (single batch, 4 A100 GPUs). With a context of 128K tokens, Jamba obtains 3x the throughput of Mixtral, while Llama-2-70B does not fit with this long context.

Training Infrastructure and Dataset

The model was trained on NVIDIA H100 GPUs.

Evaluation

Academic Benchmarks

	Reasoning						
	HellaSwag	**WinoGrande**	**ARC-E**	**ARC-C**	**PIQA**	**NQ**	**TruthfulQA**
Llama-2 13B	80.7	72.8	77.3	59.4	80.5	37.7	37.4
Llama-2 70B	85.3	80.2	80.2	**67.3**	82.8	**46.9**	44.9
Gemma	81.2	72.3	**81.5**	53.2	81.2	32.6	44.8
Mixtral	86.7	81.2	77.6	66	83	44.8	**46.8**
Jamba	**87.1**	**82.5**	73.5	64.4	**83.2**	45.9	46.4

	Comprehension				Aggregate	
	BoolQ	**QuAC**	**GSM8K**	**HumanEval**	**MMLU**	**BBH**
Llama-2 13B	81.7	**42.7**	34.7	18.3	54.8	39.4
Llama-2 70B	85	42.4	55.3	29.9	69.8	51.2
Gemma	87.2	39.2	54.5	32.3	64.3	**55.1**
Mixtral	**88.4**	40.9	**60.4**	**34.8**	**70.6**	50.3
Jamba	88.2	40.9	59.9	29.3	67.4	45.4

Table 2: Comparison of Jamba with other publicly available models. Jamba obtains similar or better performance with much better throughput.

In summary, Jamba demostrates the ability of hybrid architectures to reach the performance of state-of-the-art Transformer based models of the same size class, while having the benefits of anSSM.

Long-Context Evaluations

We have successfully trained Jamba models with context lengths of up to 1M tokens.

Ablations and Insights

we found useful: explicit positional information is not needed in Jamba, and
Mamba layers necessitate special normalization to stabilize training at large scale.

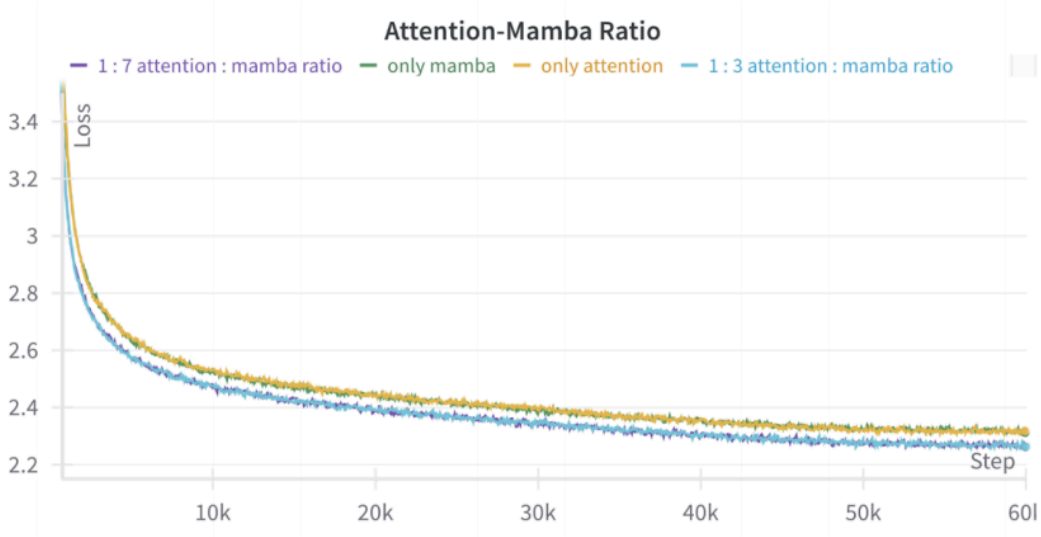


Figure 5: Training loss curves for pure Attention, pure Mamba, and Attention-Mamba hybrids (no MoE), with ratios $a : m$ of 1:3 and 1:4. All models are 1.3B parameters. The two hybrids achieve better loss throughout this training run, without any noticeable difference between the different Attention/Mamba ratios.

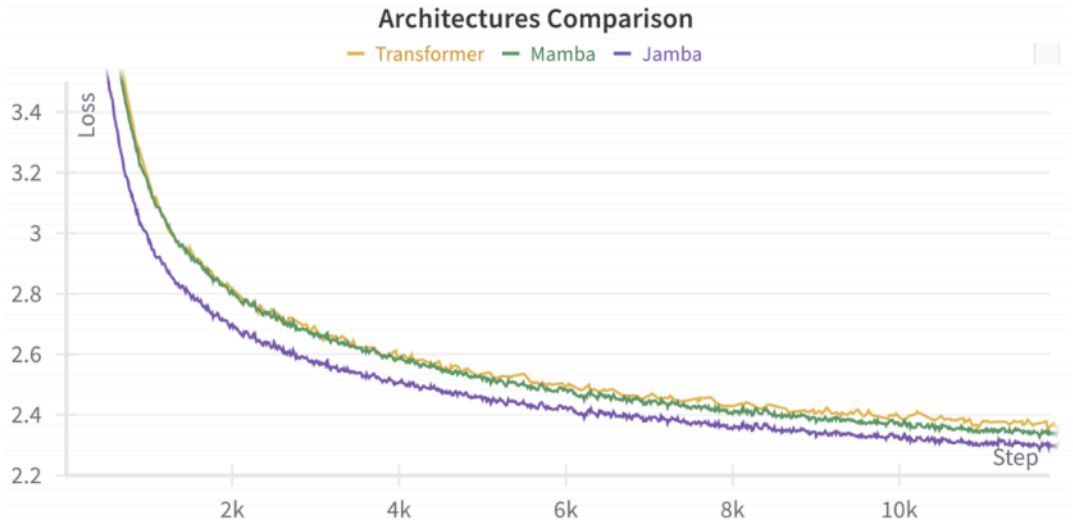


Figure 6: Training loss curves for pure Attention, pure Mamba, and an Attention-Mamba hybrid (no MoE). All models are 7B parameters. the hybrid achives better loss throughout this training run.

The Effect of Mixture-of-Experts (MoE)

	OLLM	Hella Swag	Wino Grande	NQ	log-prob C4	log-prob Books	log-prob Code
Jamba (no MoE)	36.6	62.5	58.8	15.4	-0.547	-0.658	-0.340
Jamba+MoE	38.1	66.0	61.2	18.9	-0.534	-0.645	-0.326

Table 7: Mixture-of-experts improves the Attention-Mamba hybrid.

Stabilizing Mamba at large scale

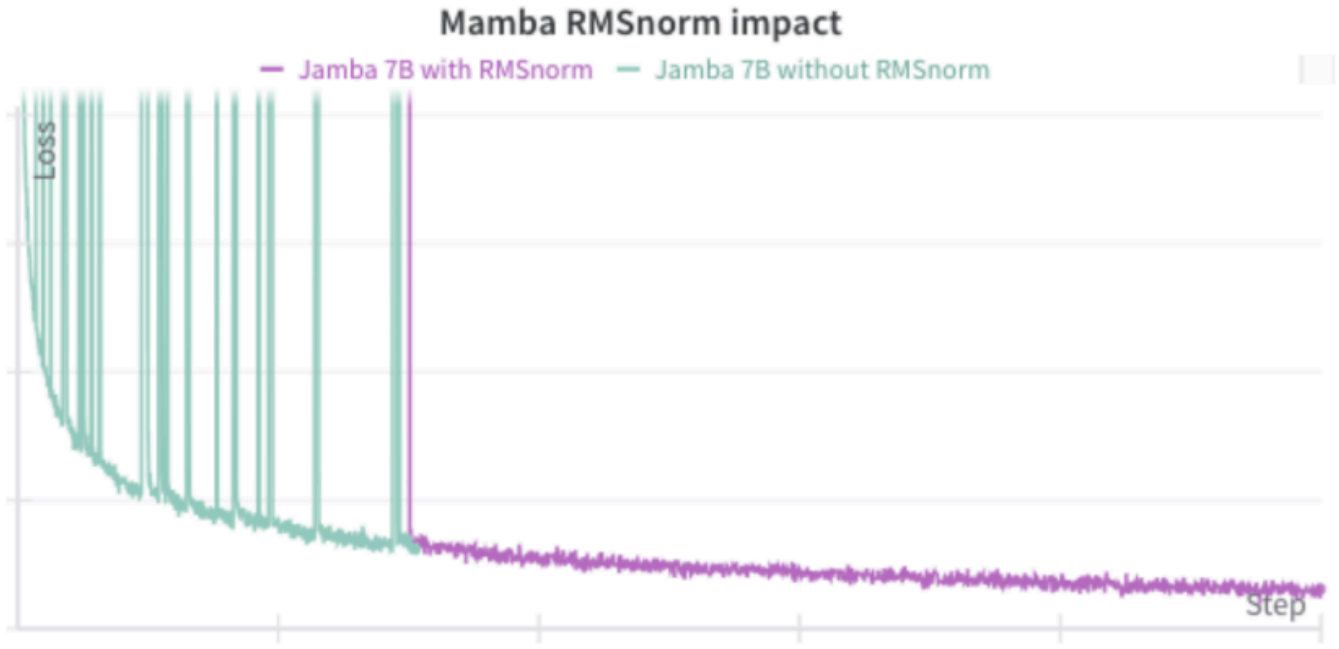


Figure 8: Adding RMSNorm to Mamba layers prevents loss spikes.

Jamba does not Require Explicit Positional Information

								log-prob		
	OLLM	Hella Swag	Wino Grande	ARC-C	Narrative QA	NQ	BoolQ	C4	Books	Code
Jamba	39.6	71.5	64.2	40.7	50.5	22.2	68.9	-0.516	-0.623	-0.299
Jamba+RoPE	40.1	71.8	65.5	40.4	46.2	22.2	67.9	-0.516	-0.623	-0.299

Table 8: Comparison of Jamba with and without explicit positional information.

Intallation

```
pip install git+https://github.com/huggingface/transformers
```

Install from local source

```
git clone https://github.com/huggingface/transformers.git
cd transformers
pip install -e .
```

Jamba requires you use transformers version 4.39.0 or higher:

```
pip install mamba-ssm causal-conv1d>=1.2.0
pip install transformers>=4.39.0
```

https://huggingface.co/ai21labs/Jamba-v0.1