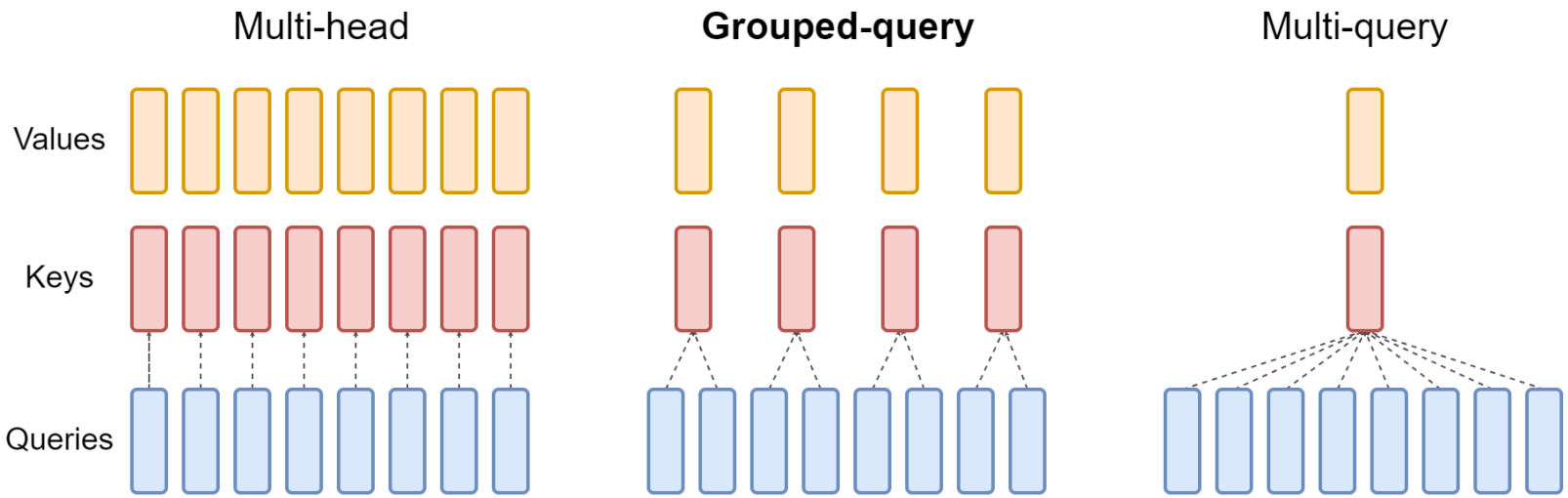
Arquiteturas de LLMs

Tópicos em Ciência de Dados

Prof. Dr. Denis Mayr Lima Martins

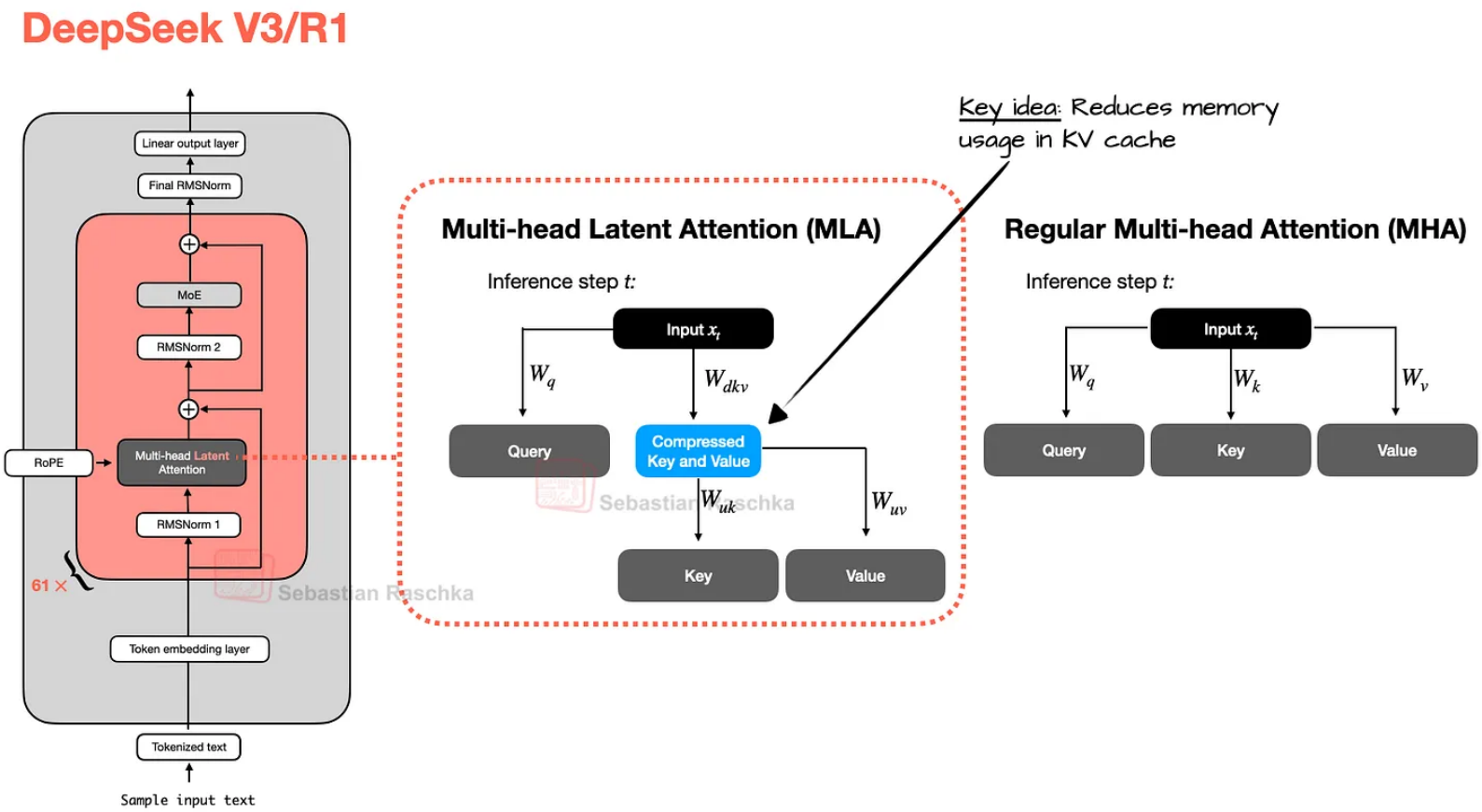
Pontifícia Universidade Católica de Campinas

Grouped-Query Attention (GQA)



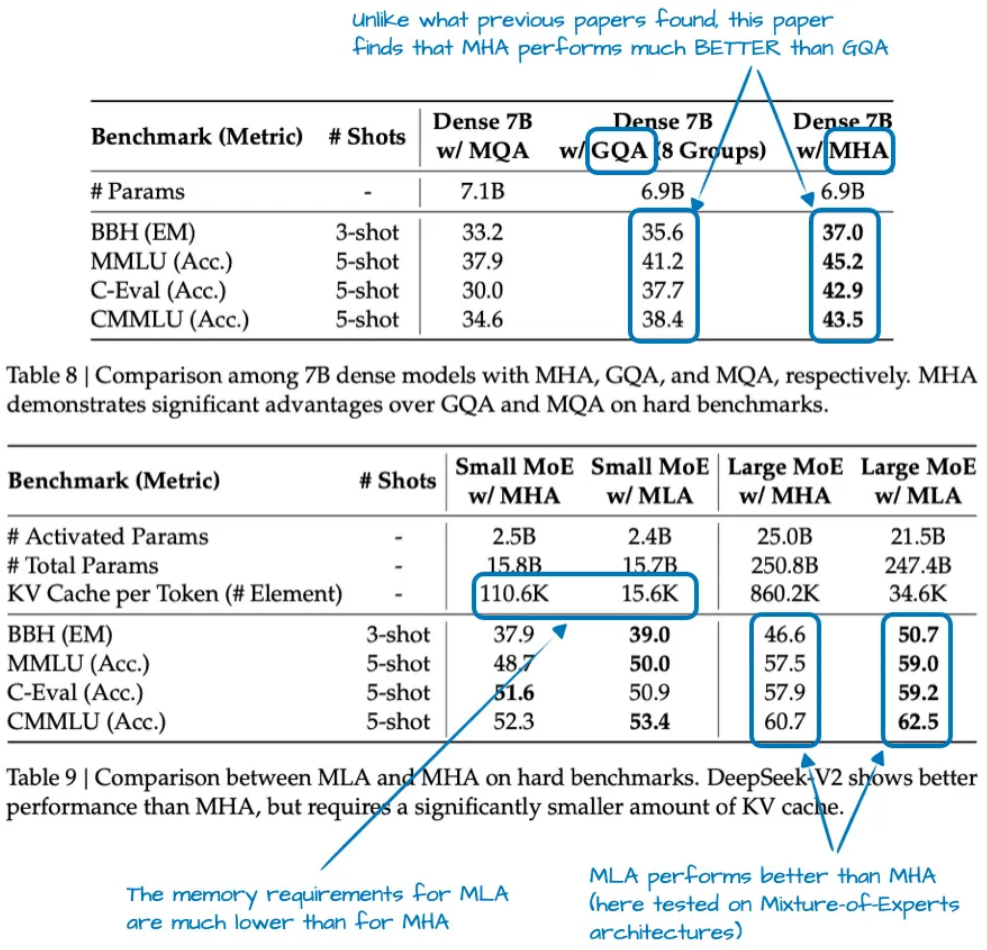
Comparação MHA (esquerda), GQA (centro) e MQA (direita). Fonte: GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints.

DeepSeek V3/R1



Multihead Latent Attention (MLA). Fonte: Raschka.

DeepSeek V3/R1: Resultados de MLA



Benchmark (Metric)	# Shots	Dense 7B w/ MQA	Dense 7B w/ GQA (8 Groups)	Dense 7B w/ MHA
# Params	-	7.1B	6.9B	6.9B
BBH (EM)	3-shot	33.2	35.6	**37.0**
MMLU (Acc.)	5-shot	37.9	41.2	**45.2**
C-Eval (Acc.)	5-shot	30.0	37.7	**42.9**
CMMLU (Acc.)	5-shot	34.6	38.4	**43.5**

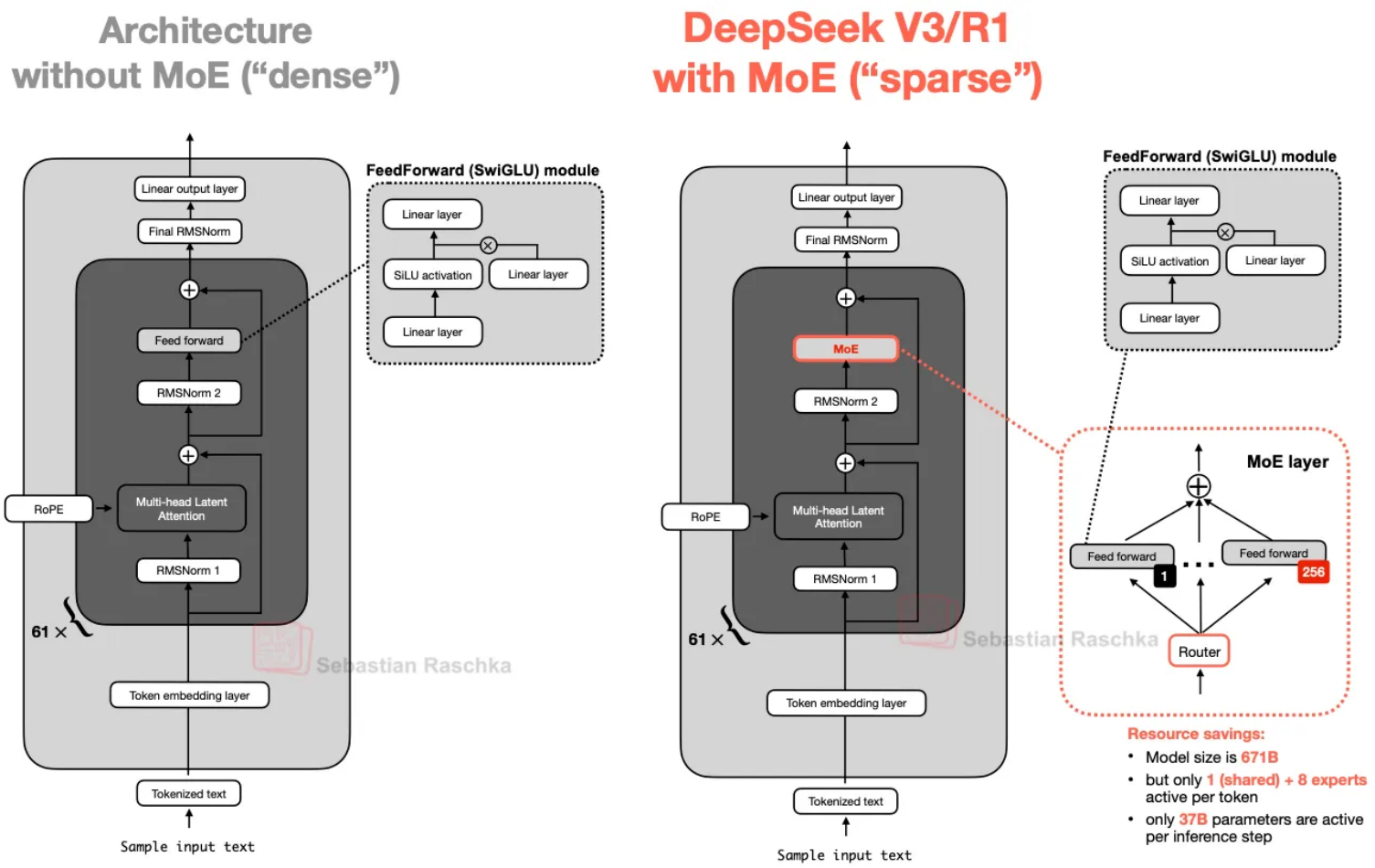
Table 8 | Comparison among 7B dense models with MHA, GQA, and MQA, respectively. MHA demonstrates significant advantages over GQA and MQA on hard benchmarks.

Benchmark (Metric)	# Shots	Small MoE w/ MHA	Small MoE w/ MLA	Large MoE w/ MHA	Large MoE w/ MLA
# Activated Params	-	2.5B	2.4B	25.0B	21.5B
# Total Params	-	15.8B	15.7B	250.8B	247.4B
KV Cache per Token (# Element)	-	110.6K	15.6K	860.2K	34.6K
BBH (EM)	3-shot	37.9	**39.0**	46.6	**50.7**
MMLU (Acc.)	5-shot	48.7	**50.0**	57.5	**59.0**
C-Eval (Acc.)	5-shot	**51.6**	50.9	57.9	**59.2**
CMMLU (Acc.)	5-shot	52.3	**53.4**	60.7	**62.5**

Table 9 | Comparison between MLA and MHA on hard benchmarks. DeepSeek-V2 shows better performance than MHA, but requires a significantly smaller amount of KV cache.

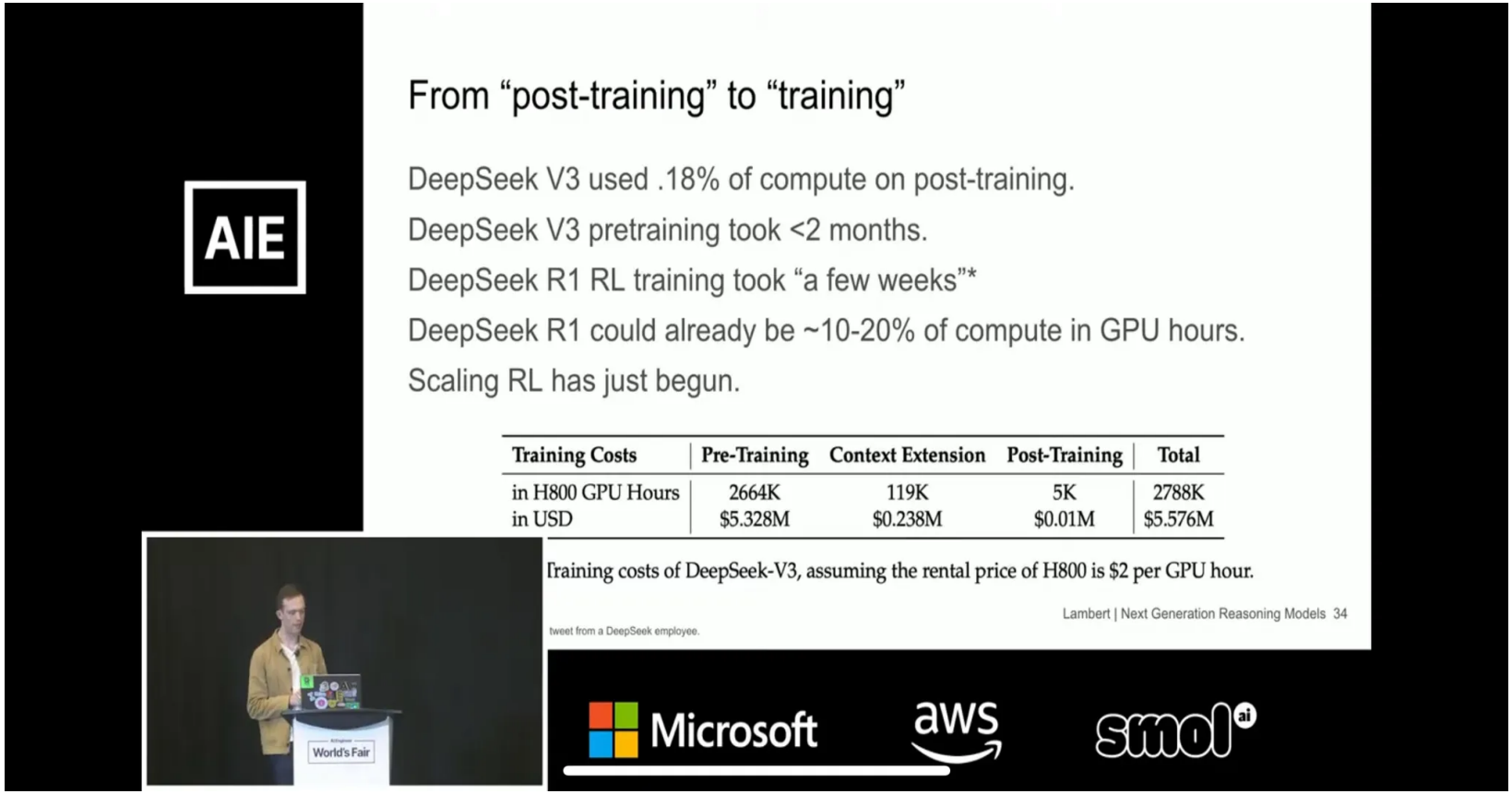
Multihead Latent Attention (MLA). Fonte: Raschka.

Mixture of Experts (MoE)



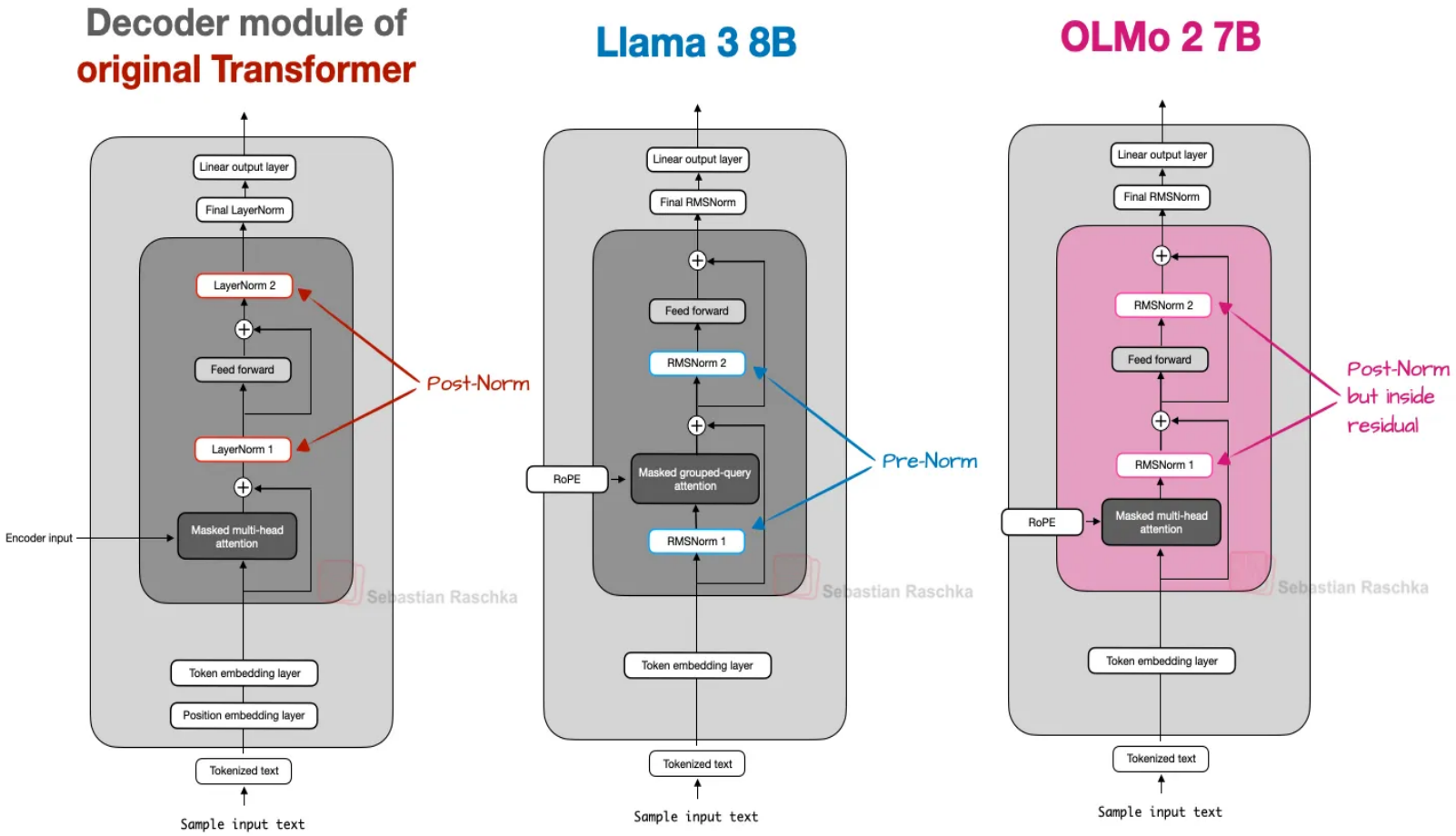
MoE no DeepSeek V3. Fonte: Raschka.

DeepSeek V3/R1



DeepSeek Metrics. Fonte: Lance Martin.

Normalization Layer



Post-Norm x Pre-Norm. Fonte: Raschka.

Root Mean Square Layer Normalization

$\text{RMSNorm}(x) = \gamma \odot \frac{x}{\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2 + \epsilon}}$, onde $\odot$ é a multiplicação elemento-a-elemento.

```python
class RMSNorm(nn.Module):
    def __init__(self, normalized_shape: list | tuple,
        eps: float = 1e-5, element_affine: bool = True,
    ):
        super().__init__()
        self.eps = eps
        self.element_affine = element_affine
        if self.element_affine:
            self.gamma = nn.Parameter(torch.ones(normalized_shape))
        else:
            self.register_parameter("gamma", None)

    def forward(self, x: torch.Tensor):
        x = x * torch.rsqrt(self.eps + x.pow(2).mean(dim=-1,
keepdim=True))
        return x if self.gamma is None else x * self.gamma
```

Query-Key Normalization

Ideia: Aplicar L2 norm às matrizes Q e K antes do cálculo de atenção.

De: $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$ Para:

$\text{QK-Norm}(Q, K, V) = \text{softmax}(\frac{Q}{\|Q\|_2} \cdot (\frac{K}{\|K\|_2})^T)V$

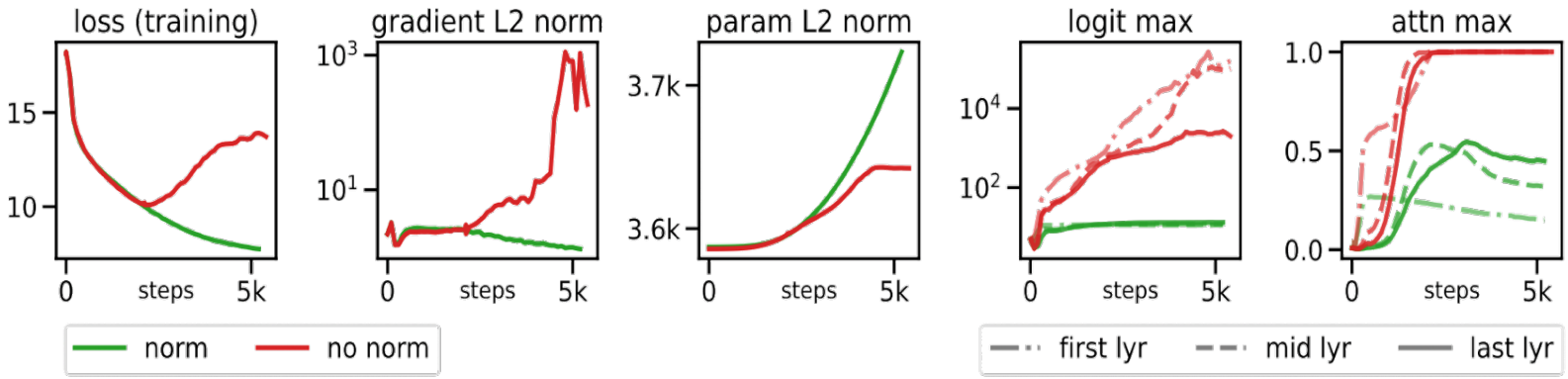
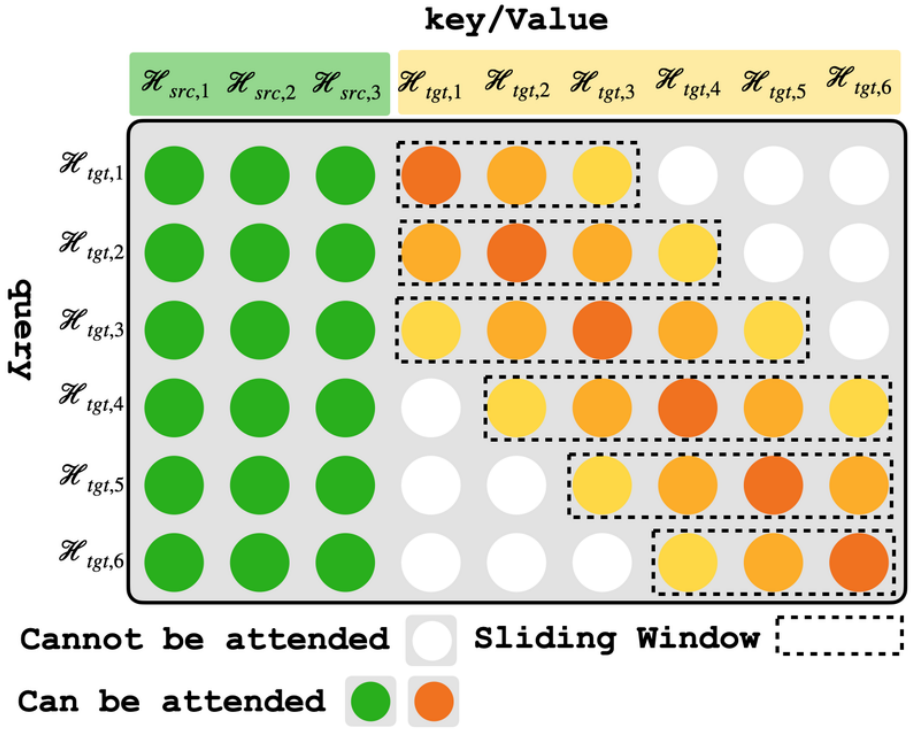


Figure 1: Effect of query/key normalization on an 8B parameter model.

Resultados de QK-Norm. Fonte: https://arxiv.org/abs/2302.05442.

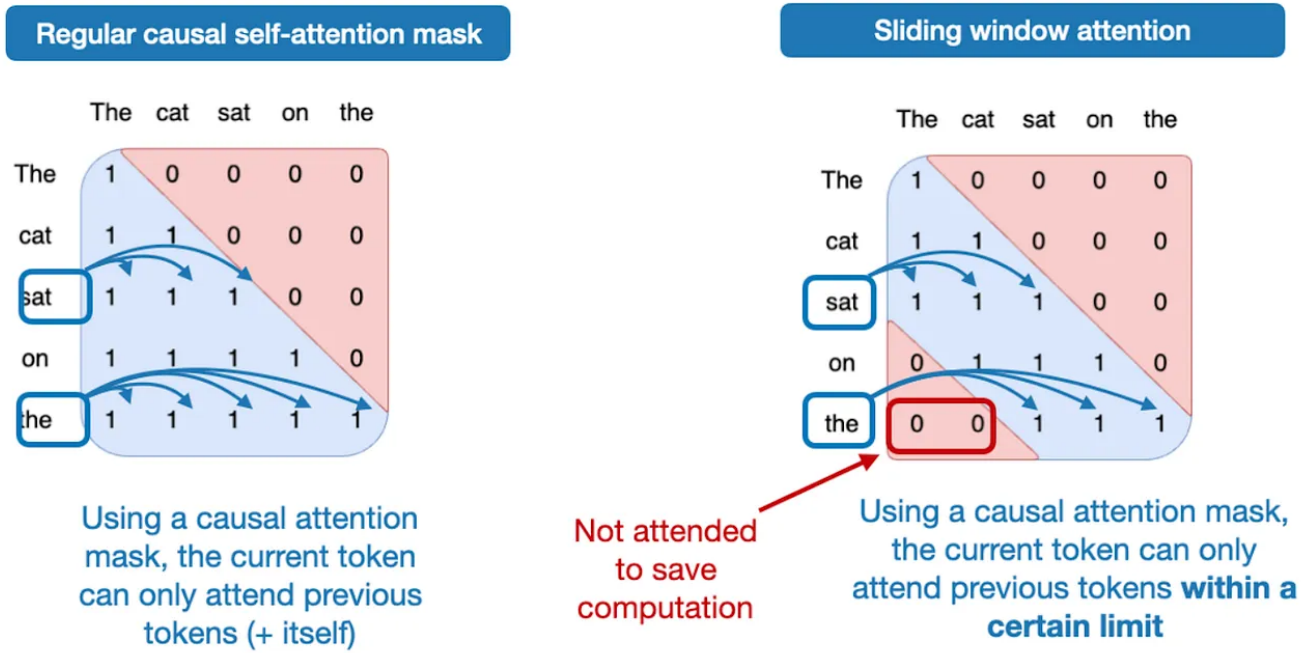
Sliding Window Attention (SWA)



Exemplo de SWA. Fonte: Research Gate.

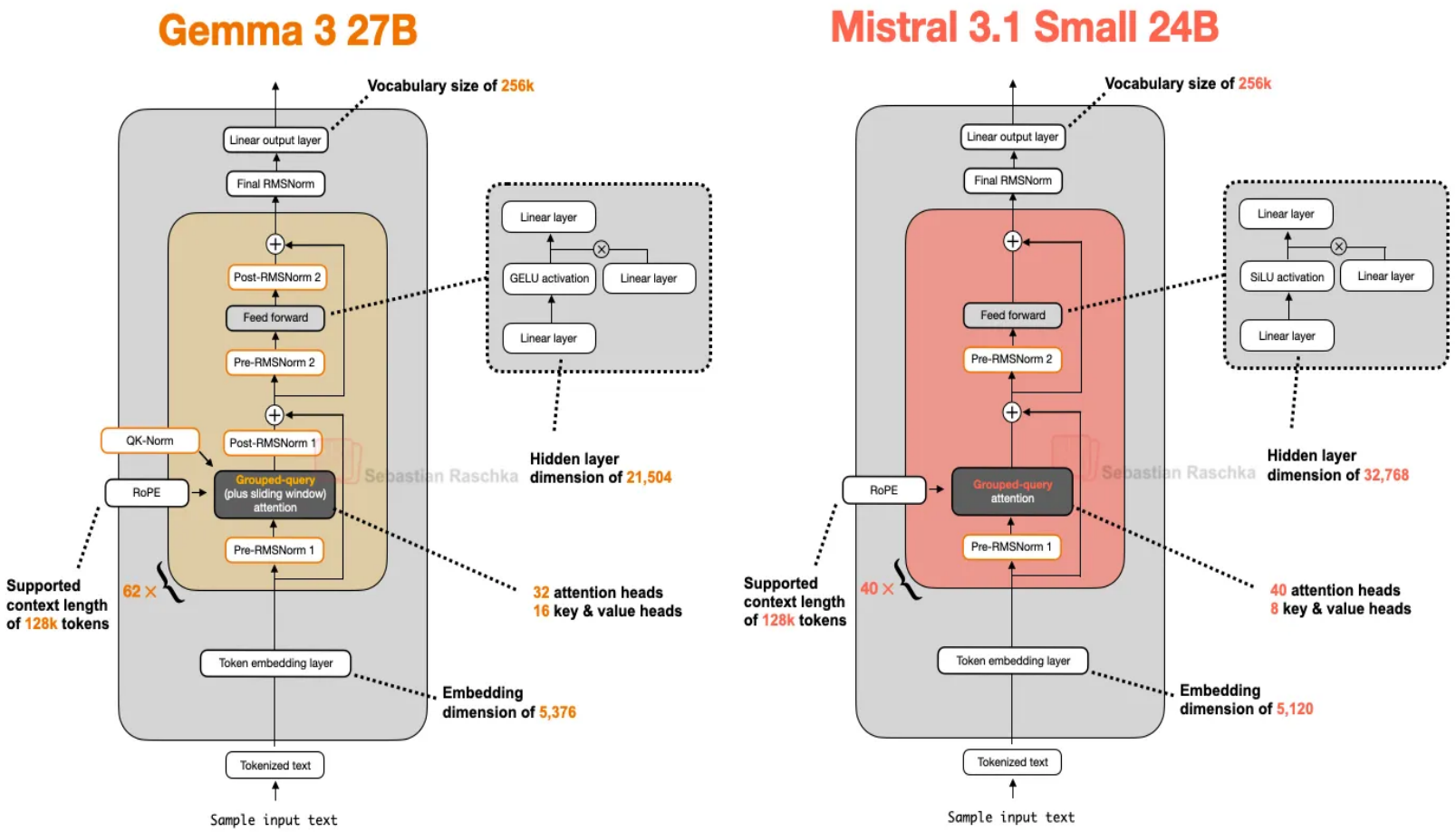
> *Detalhes em: https://arxiv.org/html/2502.18845v2*

Sliding Window Attention (SWA)



Comparação Causal Attention x SWA. Fonte: Raschka.

Gemma 3 e Mistral 3.1



Comparação entre Gemma 3 27B e Mistral 3.1 Small 24B. Fonte: Raschka.

RoPE

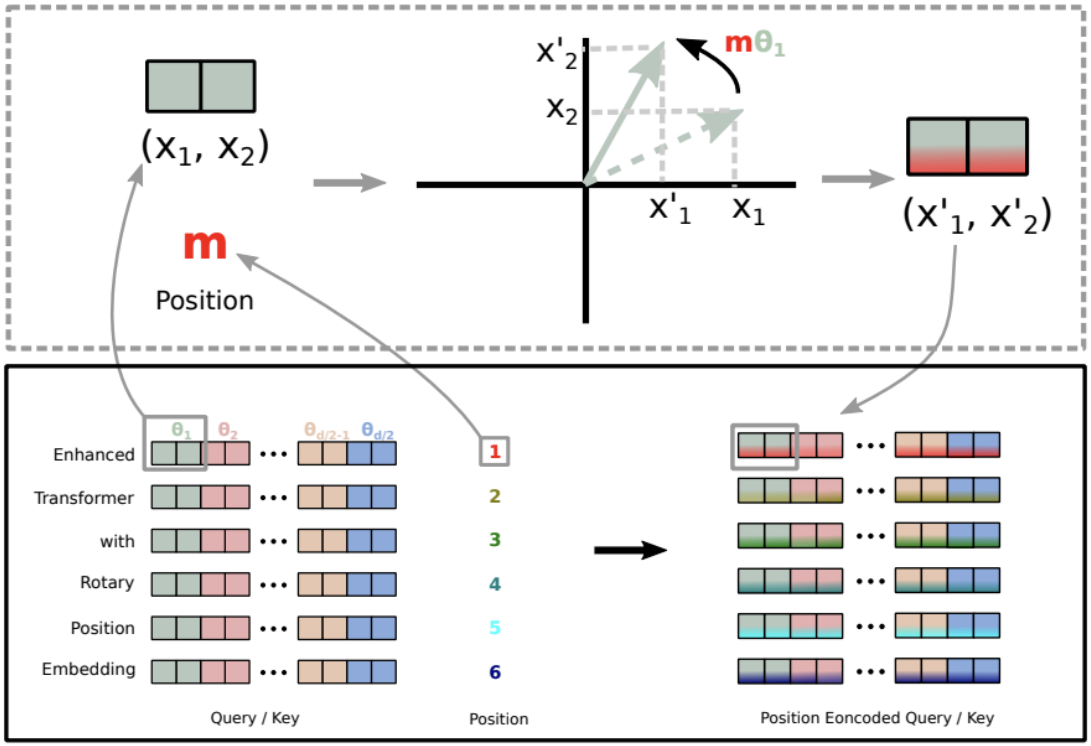
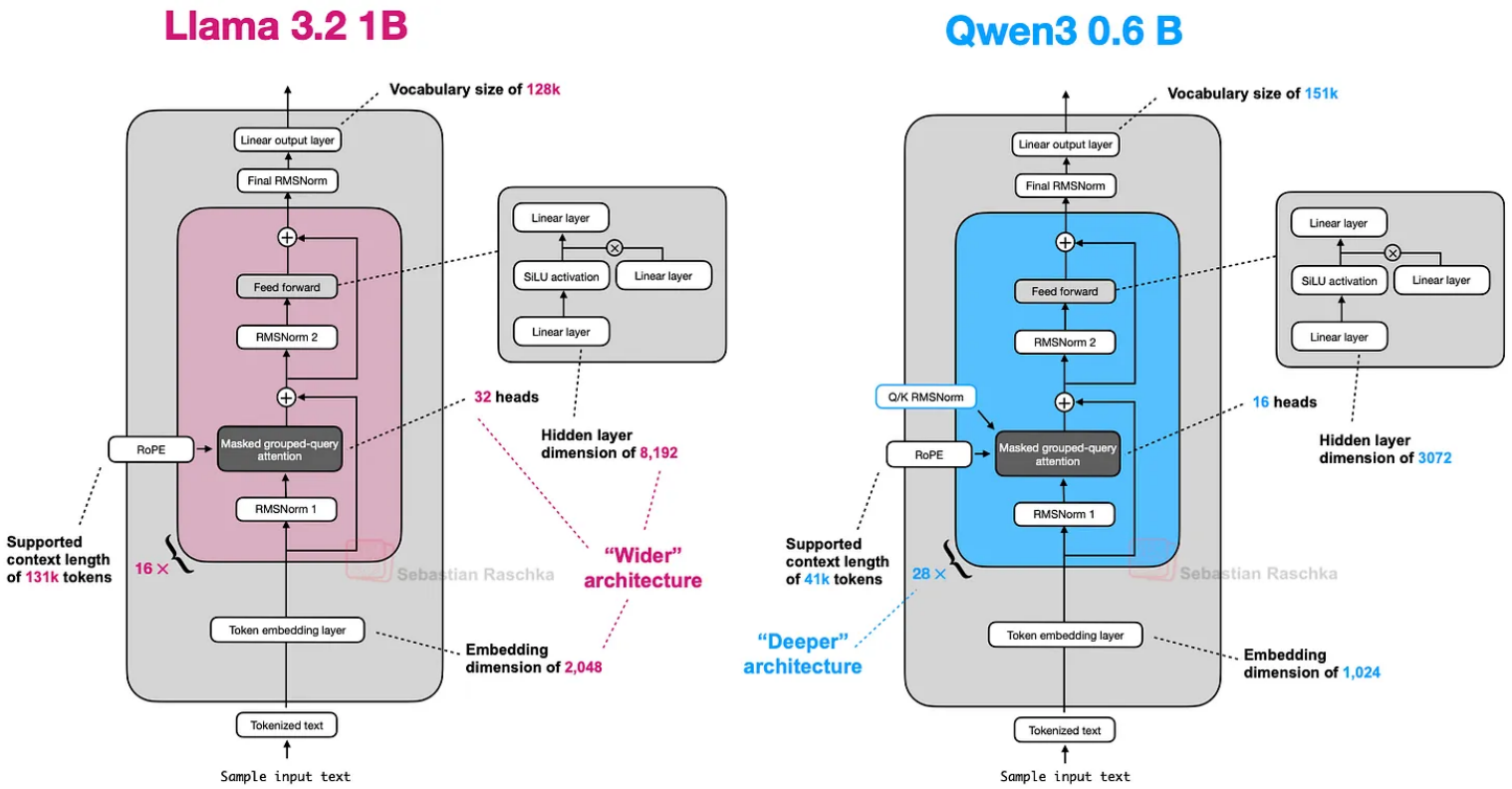


Figure 1: Implementation of Rotary Position Embedding(RoPE).

Rotary Positional Embedding. Fonte: https://arxiv.org/abs/2104.09864.

Qwen3 0.6B

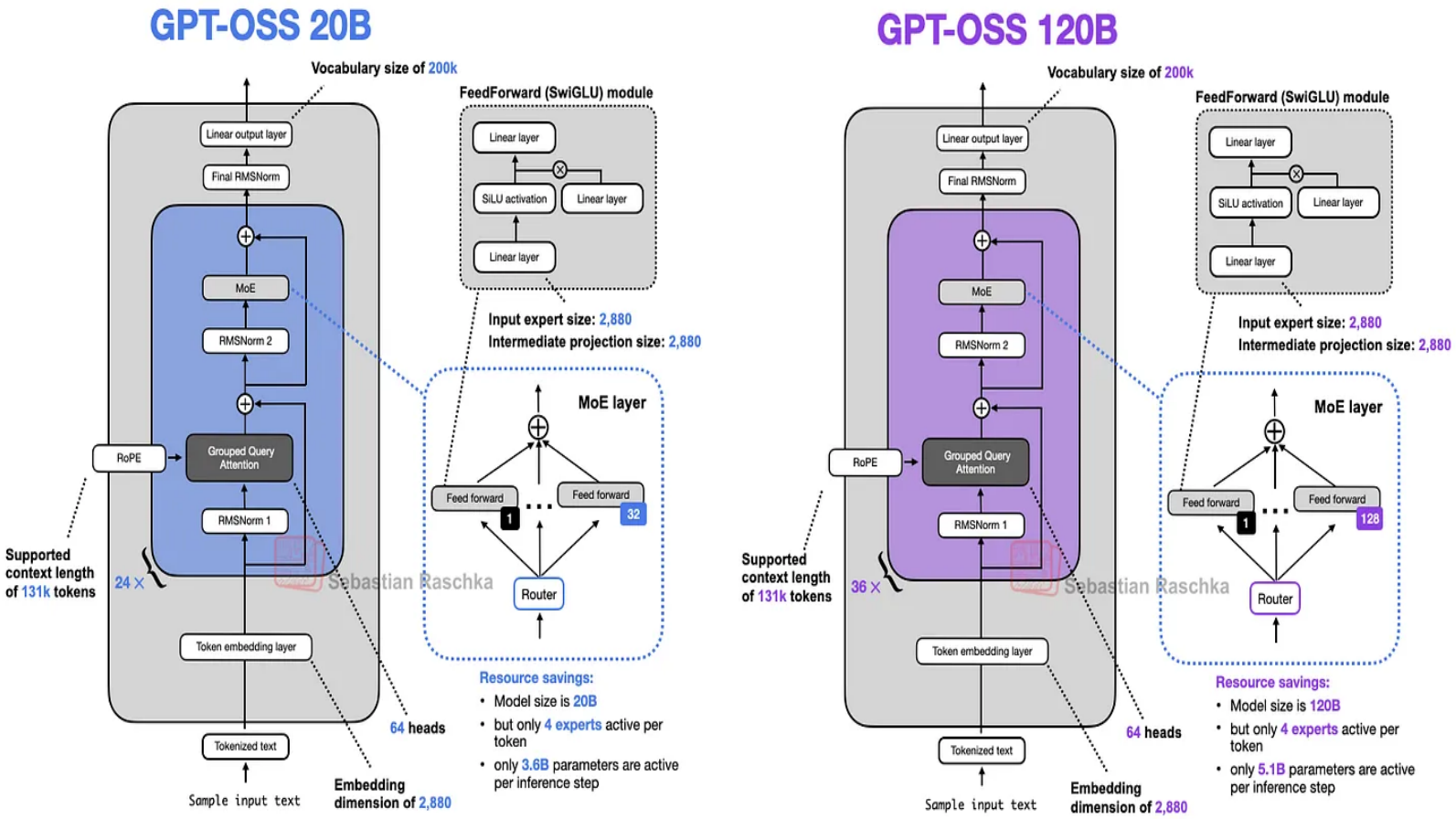


	Tokens/sec	Memory
Llama3Model 1B	42	2.91 GB
Llama3Model 1B compiled	170	3.12 GB

	Tokens/sec	Memory
Qwen3Model 0.6B	24	1.49 GB
Qwen3Model 0.6B compiled	101	1.99 GB

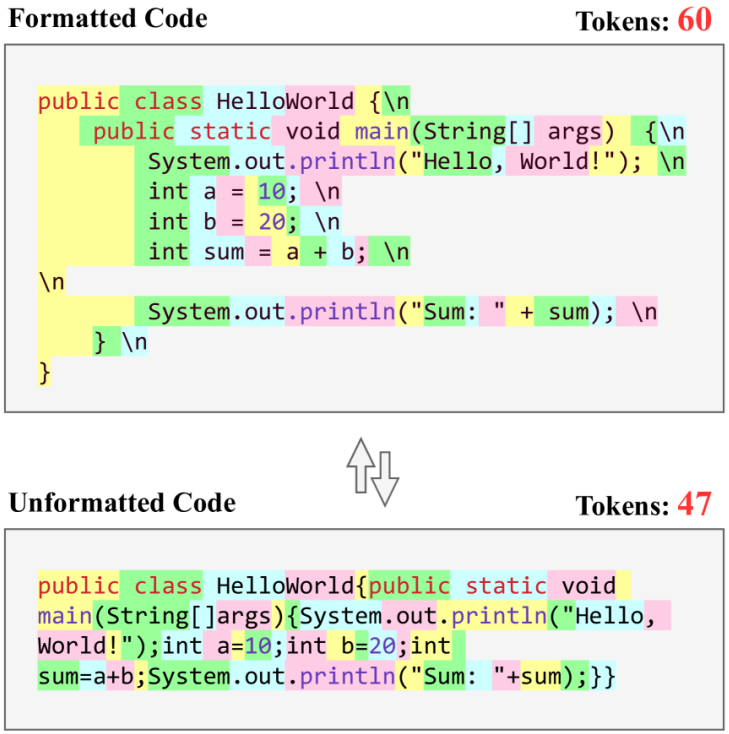
Comparação entre Llama 3.2 1B e Qwen3 0.6B. Fonte: Raschka.

GPT-OSS



Comparação modelos GPT-OSS. Fonte: Raschka.

Extra: O preço da formatação



Comparação de tokens no tokenizer GPT-4o. Formatação de código no prompt tem seu preço. Fonte: https://arxiv.org/html/2508.13666v1.

Leitura Recomendada

- Sebastian Raschka blog (Ahead of AI): https://magazine.sebastianraschka.com/p/the-big-llm-architecture-comparison
- The Transformer++: https://www.gleech.org/tplus
- Applied LLMs: https://applied-llms.org/
- Eugene Yan blog: https://eugeneyan.com/
- Hugging Face Papers (antigo Papers with Code): https://huggingface.co/papers/trending