Advanced LLM Architectures

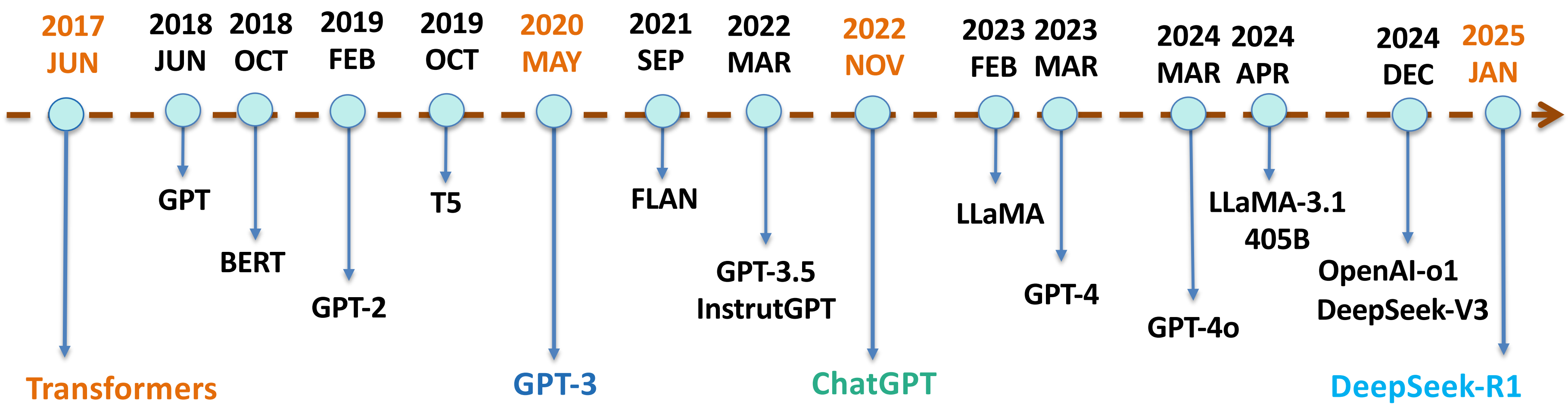
(Optional)

AI with Deep Learning
EE4016

Prof. Lai-Man Po
Department of Electrical Engineering
City University of Hong Kong

The Evolution of LLM Architectures

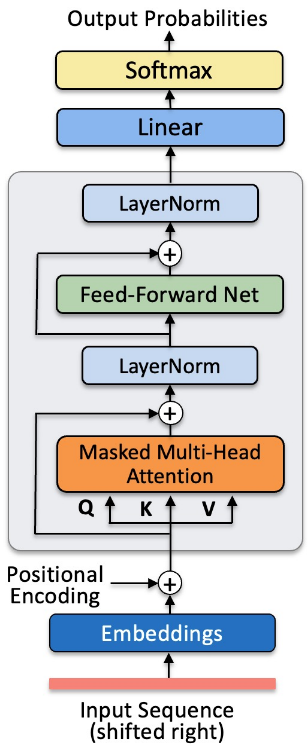
- Structural similarities from GPT-2 (2019) to 2025 models like DeepSeek-V3 and LLaMA 4.
- **Evolutions:** Absolute to RoPE positional embeddings; MHA to GQA; GELU to SwiGLU activations, etc.



GPT-1 vs GPT-2 Architectures

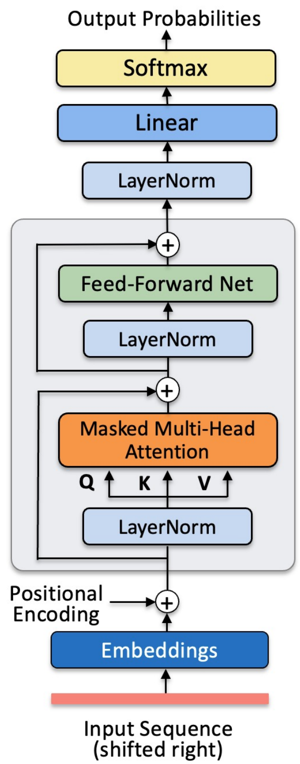
GPT-1 (2018)

- **117M Parameters**
- N = 12 Layers
- *h* = 12 heads
- Context window: 256 tokens
- GELU
- **Post-LayerNorm**
- Fine-tune for down-stream NLP tasks



GPT-2 (2019)

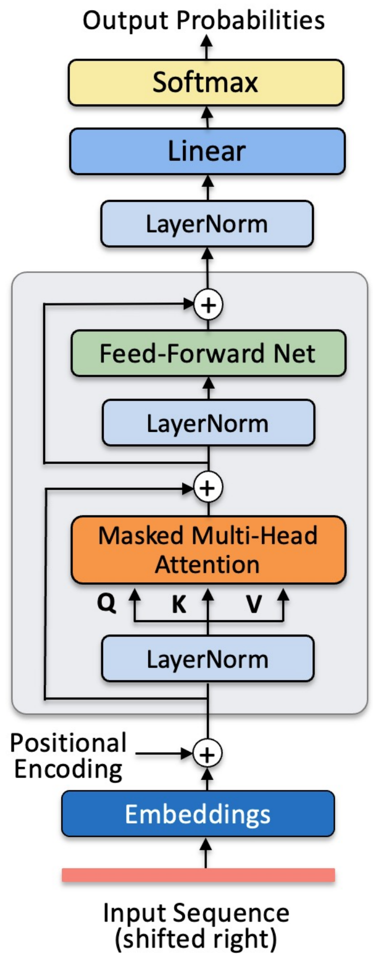
- **1.5B Parameters**
- N = 48 Layers
- *h* = 25 heads
- Context window: 1024 tokens
- GELU
- **Pre-LayerNorm**
- **Strong zero-shot performance**



GPT-2 proved the scaling hypothesis with a 10x parameter jump and strong zero-shot performance, but relied on the naive, baseline Transformer architecture.

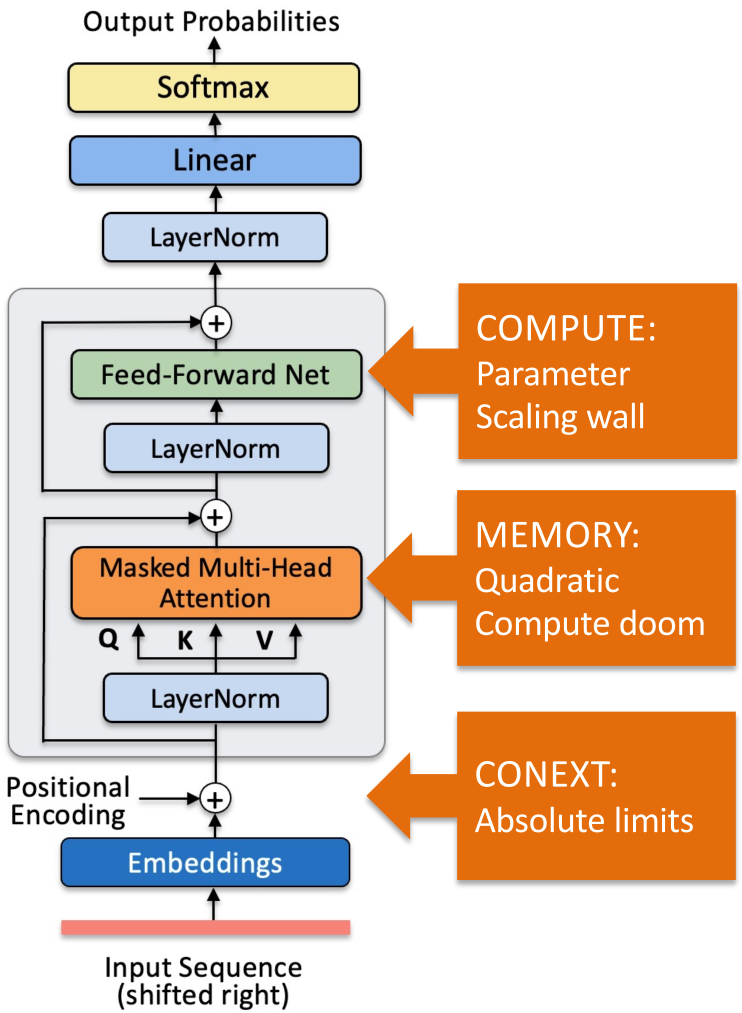
GPT-2 Architecture (2019)

- GPT-2 was an early decoder-only Transformer LLM with **1.5B** parameters, using stacked layers with key features include:
    - **Masked Multi-Head Attention (MHA)**
    - **Feed-Forward Net (FFN)**
    - **GELU** as a smooth activation function
    - **Dropout** after sub-layers to curb overfitting
    - **Sinusoidal positional embeddings** for token order
    - **Pre-LayerNorm** for stable training
- Despite generating coherent text, it suffered from quadratic attention complexity and overfitting on small datasets.



The 2019 baseline introduced fatal scaling bottlenecks

GPT-2 scaled to 1.5B parameters and demonstrated zero-shot capabilities, but its architecture contained three structural flaws that would choke future growth:



LLM Architecture Evolution from GPT-2

- **GPT-2 (2019)**:
  - 1.5B parameters, foundational transformer architecture.
  - Used dropout, absolute (sinusoidal) positional embeddings, LayerNorm and GELU.
- **Architecture Advancements of Modern Open-Weight LLMs**:
  - **Dropout Removed**: Not needed for single-epoch training on large datasets.
  - **RoPE**: Replaces absolute positional embeddings with rotary position encoding.
  - **RMSNorm**: Replaces LayerNorm for simpler, faster normalization.
  - **Swish/SwiGLU**: Replaces GELU for computational efficiency.
  - **Grouped Query Attention (GQA)**: More efficient than Multi-Head Attention.
  - **Sliding Window Attention**: Limits context to 128 tokens in alternating layers.
  - **Multi-Latent Attention (MLA):** Used in DeepSeek-V3/R1 to achieve further improvement.
  - **Mixture-of-Experts (MoE)**: Sparse, multi-feed-forward modules for efficiency.

The Blueprint for the Modern LLM

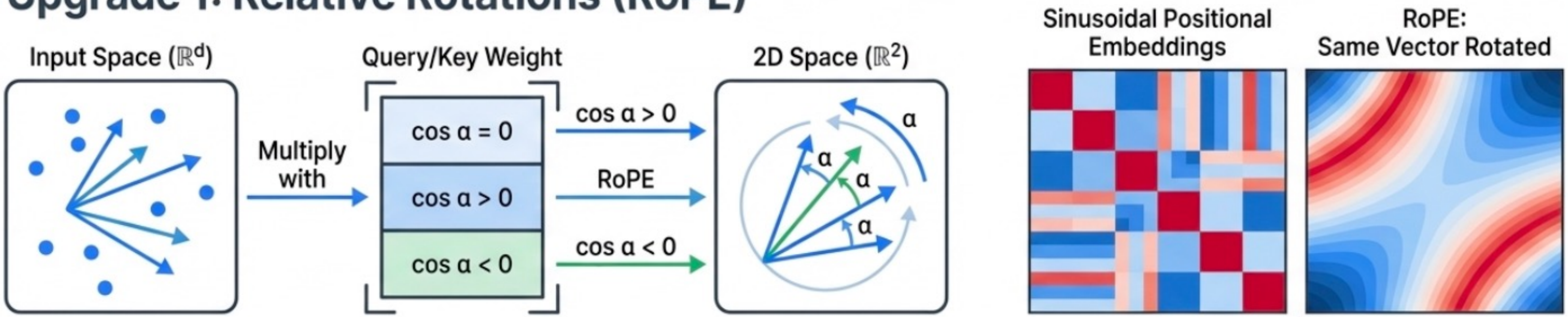
The Era Shift Matrix		
Component	Classical Era (GPT-2)	Modern Era (LLaMA, DeepSeek)
Positional Encoding	Sinusoidal Absolute	Rotary Positional Embeddings (RoPE)
Normalization	LayerNorm	RMSNorm
Activation Function	GELU	SwiGLU
Attention Mechanism	Multi-Head Attention (MHA)	GQA, SWA, MLA
Density	100% Dense Activation	Sparse Mixture-of-Experts (MoE)

Absolute Encodings

The Flaw:

Fails when processing texts longer than the specific absolute lengths seen during training data.

Upgrade 1: Relative Rotations (RoPE)



Introduced in 2021, Rotary Positional Embeddings (RoPE) apply rotations to query and key vectors. Attention scores now rely strictly on the relative geometric distance ($i - j$) between words.

Impact: Naturally enables massive context extrapolation (used natively in LLaMA, Mistral, Qwen, and DeepSeek).

https://arxiv.org/abs/2104.09864

Finding Our Place with RoPE

The Old Way

- Adding static sine/cosine signals.

The RoPE Way

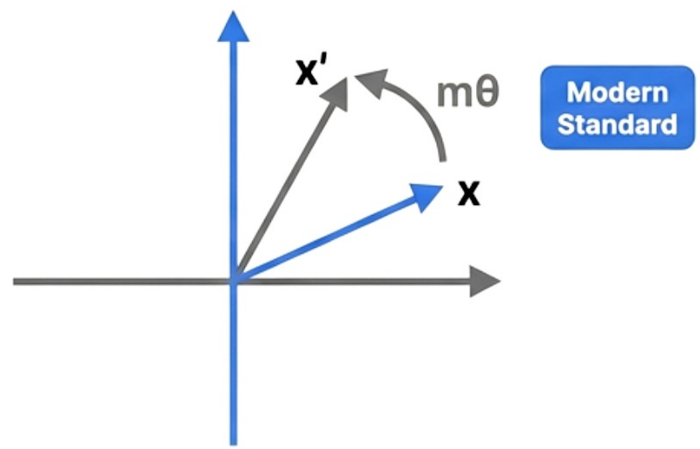
- Vectors are split into 2D pairs and rotated.
- The angle of rotation depends on token position.

The Result

- The dot product of rotated queries and keys naturally captures relative distance.
- It enables models to extrapolate to massive sequence lengths without changing the attention mechanism.

https://arxiv.org/abs/2104.09864

$$\theta_{p,i} = \frac{p}{10000^{2i/d}}, \quad R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$



Impact: Enables vast context windows (like LLaMA-3's 128K) without modifying the attention mechanism itself.

How RoPE Works

1. **Vector Splitting**: Split each embedding into pairs of dimensions (e.g., a 128D vector → 64 pairs).
2. **Rotation Application**: Rotate each pair in 2D by an angle that depends on:
    1. The token's position in the sequence
    2. The pair's index (each pair uses a different rotation frequency)

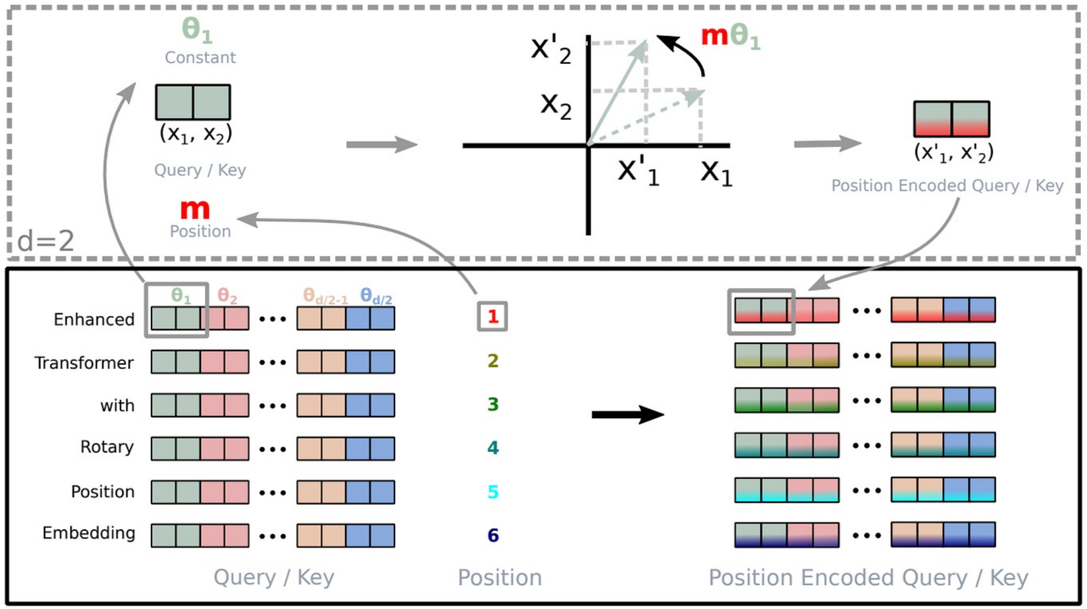


Figure 1: Implementation of Rotary Position Embedding(RoPE).

- **Why Pairs?:** Rotation is inherently 2D. Using multiple pairs with varying frequencies lets RoPE capture both fine-grained local order (fast rotations) and long-range structure (slow **rotations)—like clock hands moving at different speeds.**
- **Key Benefit**: The dot product of rotated query and key vectors automatically encodes their relative position—no changes to attention needed.

https://medium.com/@akdemir_bahadir/rotary-positional-encoding-rope-15bb30c4fa55

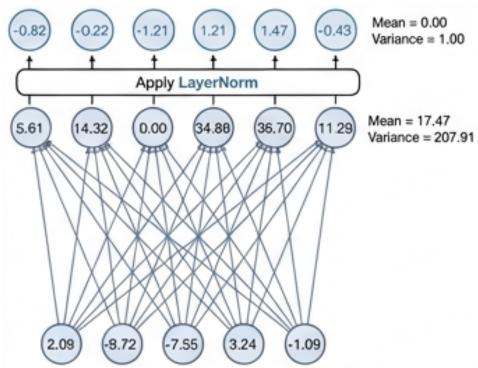
Upgrade 2: The Need for Speed (RMSNorm)

LayerNorm

$$\bar{a}_i = \frac{a_i - \mu}{\sigma} g_i$$

$$\mu = \frac{1}{n}\sum a)i \qquad \sigma = \sqrt{\frac{1}{n}\sum (a_i - \mu)^2}$$

Calculates full mean and variance.
Expensive and slow.



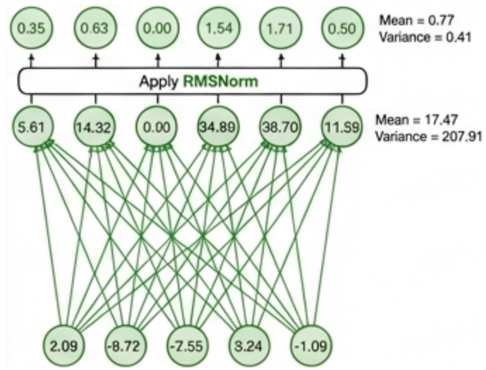
MRSNorm

$$\bar{a}_i = \frac{a_i}{\mathrm{RMS}(a)} g_i \qquad \mathrm{RMS}(a) = \sqrt{\frac{1}{n}\sum a_i^2}$$

Drops the mean calculation entirely.

Normalizes by RMS only.

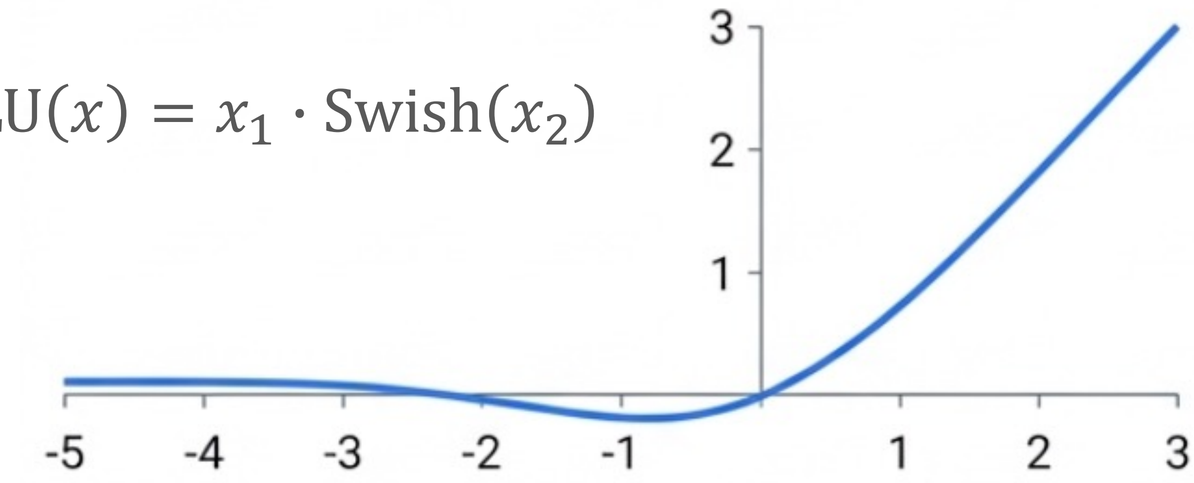
Impart: Faster processing with zero loss in model stability. Now the standard in modern efficient LLMs.



Upgrade 3: Upgrading Activation with SwiGLU

SwiGLU replaces the older GELU function. By combining Swish activation with Gated Linear Units, it provides superior expressiveness and large-scale training efficiency.

$$\mathrm{SwiGLU}(x) = x_1 \cdot \mathrm{Swish}(x_2)$$

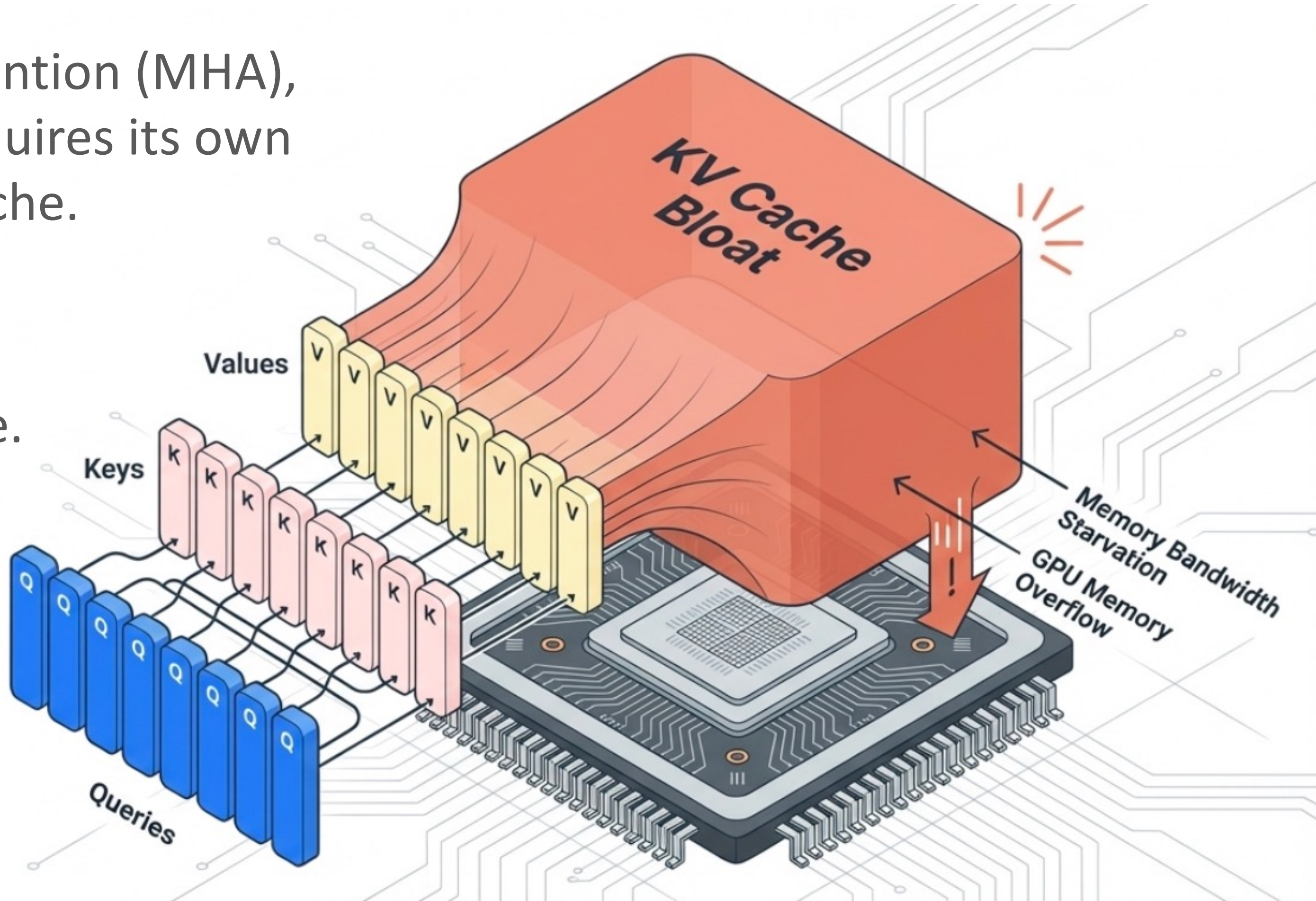


Impact: These two micro-optimizations of RMSNorm and SwiGLU became the baseline standard for LLaMA, Gemma, and nearly all post-2023 open models.

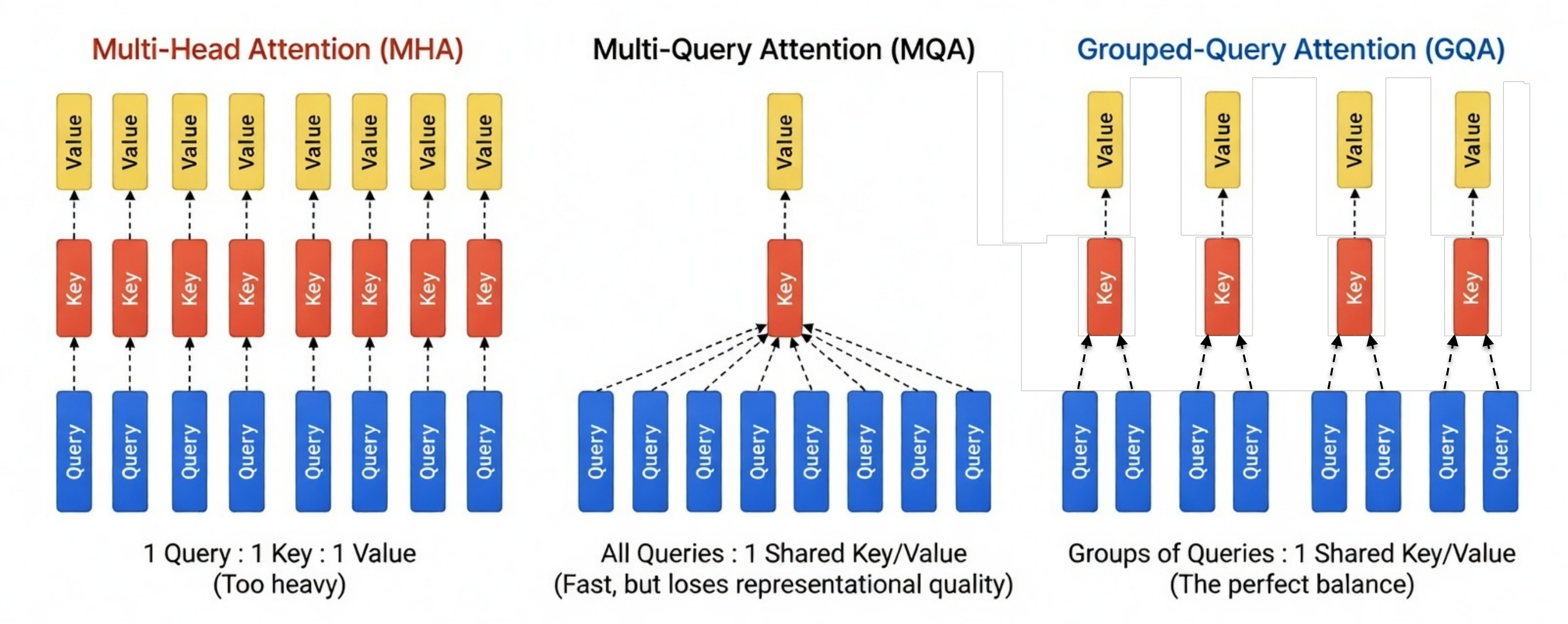
The Attention Memory Crisis

In standard Multi-Head Attention (MHA), every single Query head requires its own dedicated Key and Value cache.

During token generation, this KV cache grows massive. It starves the GPU of memory bandwidth, turning inference into a slow, memory-bound crawl.



Upgrade 4a: The Grouped-Query Compromise



GQA radically reduces the size of the K/V cache-lowering memory bandwidth requirements during inference-while maintaining the high reasoning quality of traditional MHA.

Upgrade 4b: Sliding Window Attention

Breaking the Length Barrier (100k+ Tokens)

Restricts attention to a fixed local window, changing computational complexity from $O(n^2)$ to $O(n \cdot W)$ (linear time).

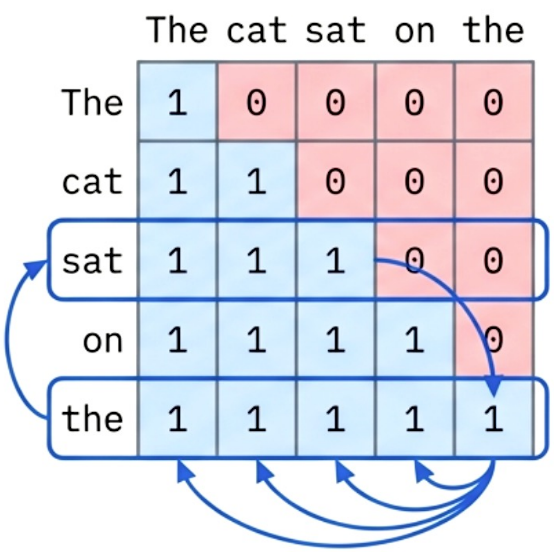
Stacking layers expands the receptive field.

The Application

Used alongside GQA (SW-GQA) in architectures like Mistral 7B to process massive 10k+ sequences efficiently without running out of memory.

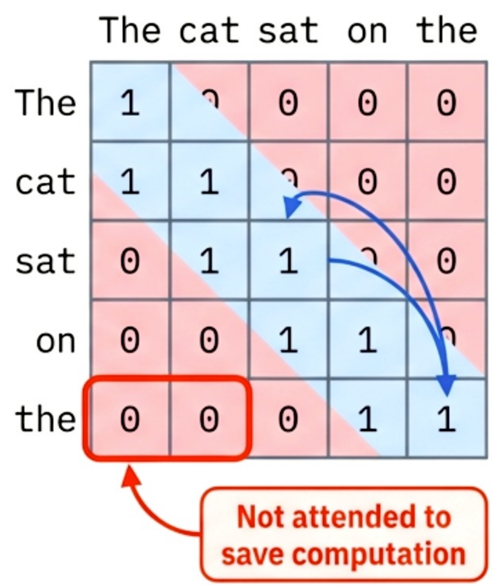
MISTRAL AI_

Regular causal self-attention mask



Using a causal attention mask, the current token can only attend previous tokens (+ itself)

New sliding window attention



Using a causal attention mask, the current token can only attend previous tokens within a certain limit

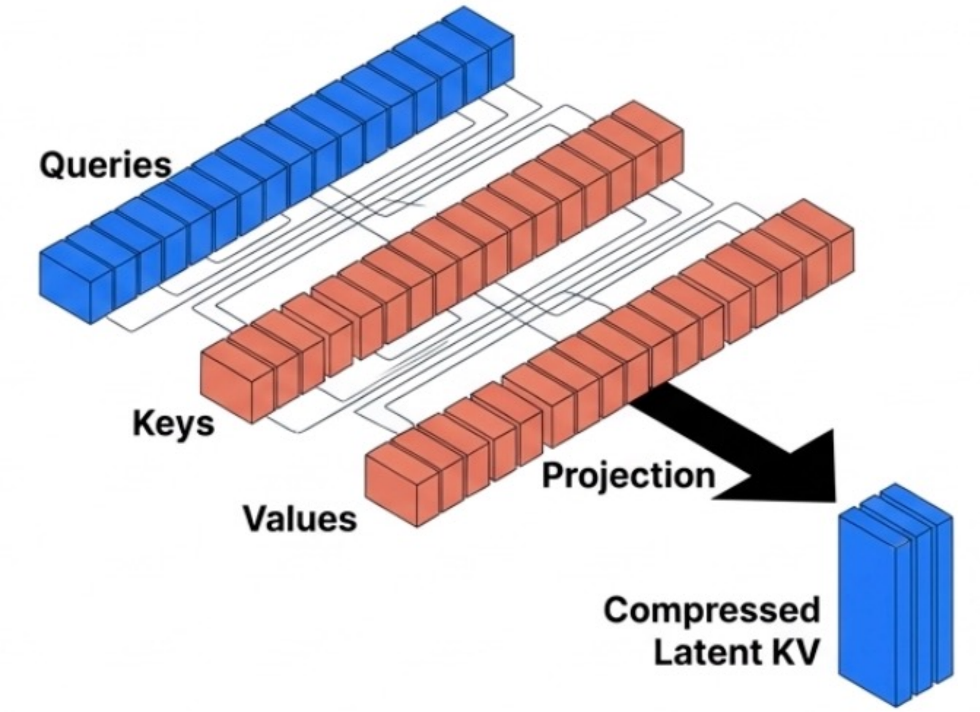
Upgrade 4c: Multi-Head Latent Attention (MLA)

Compresses key-value pairs into a tiny, low-dimensional latent space before caching with *m* « *n*.

Two-Stage Process

1. Compress Keys and Values into latent memory.
2. Retrieve by querying latent for output.

Impact: Achieves radical K/V cache reduction for 100,000+ token processing.

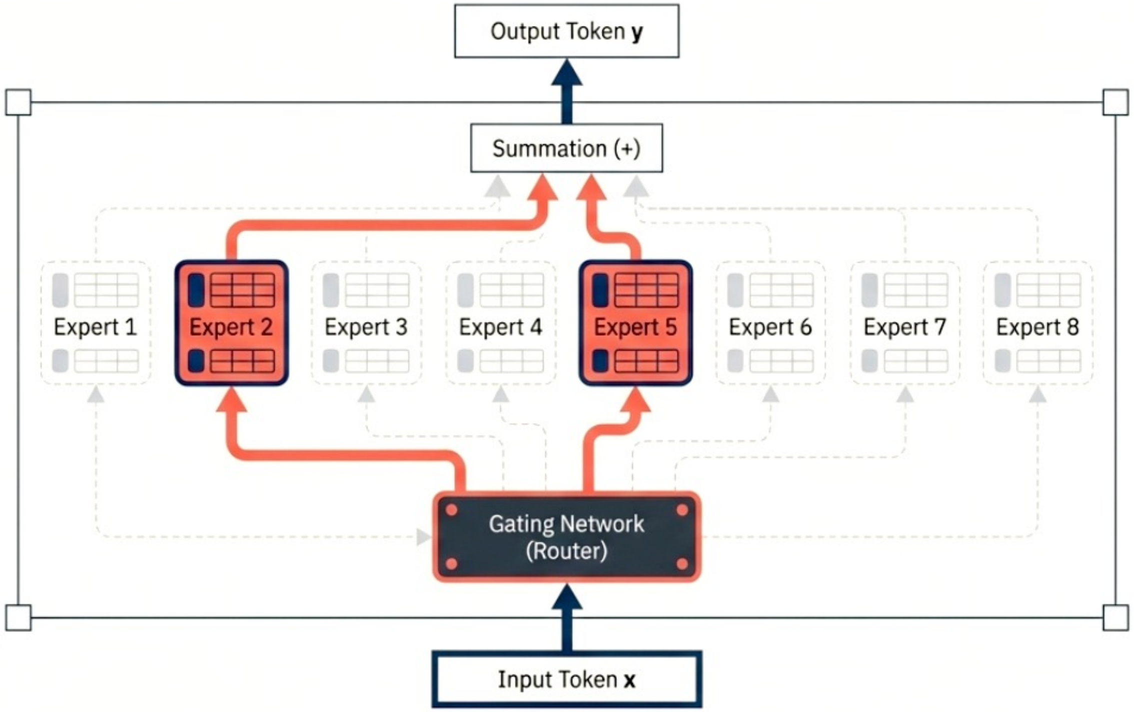


Upgrade 5: The Sparsity Paradigm (Mixture-of-Experts)

1. **The Input:** A token arrives at the layer.
2. **The Router:** A Gating Network acts as a switchboard.
3. **Sparsity in Action:** The router fires the token to only 2 Expert networks out of a possible 8.

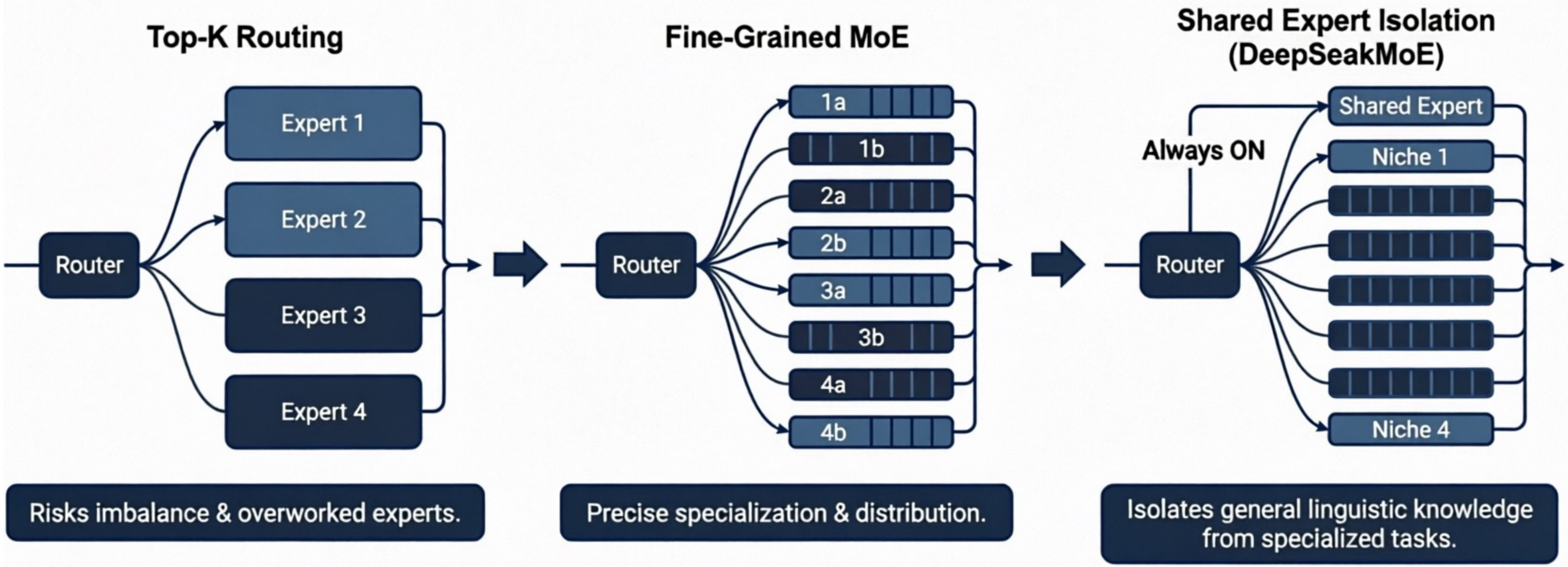
Core Equation

$$y = \sum g_i \cdot E_i(x)$$



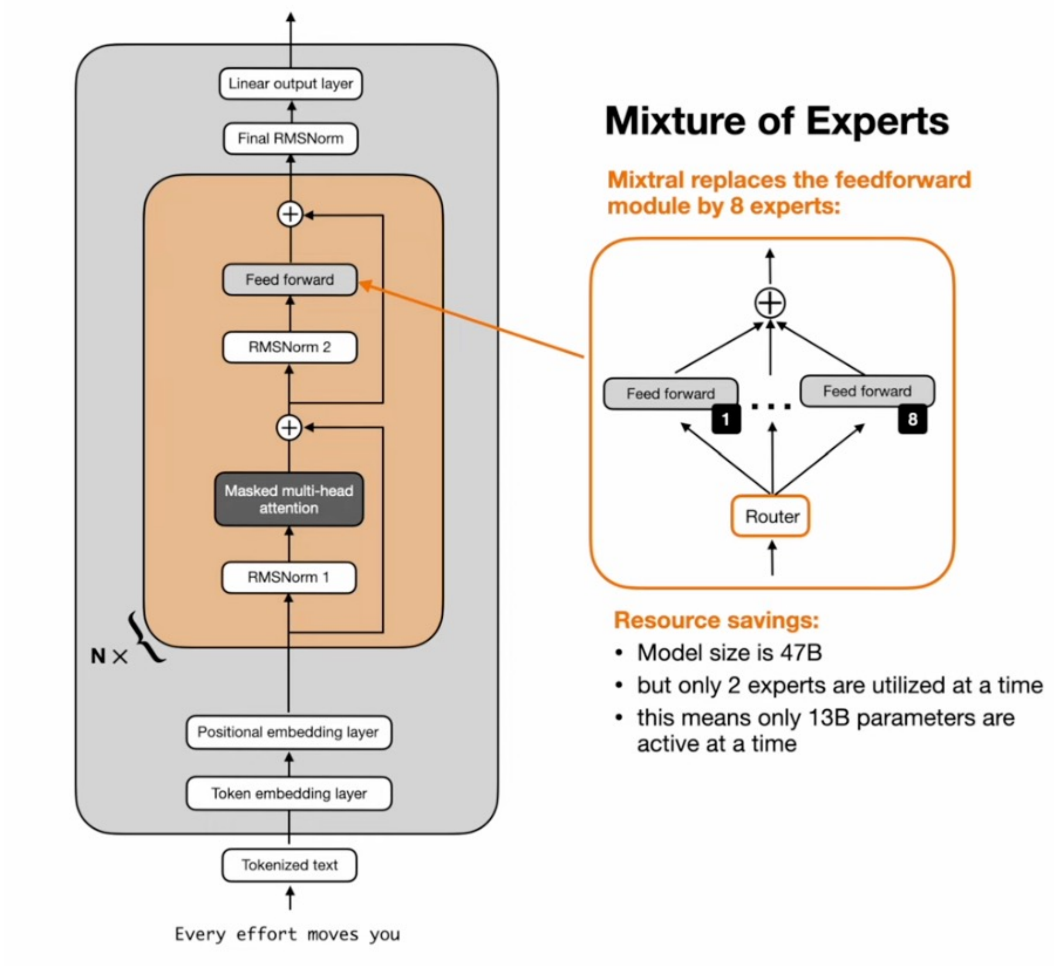
MoE replaces the massive dense layer with multiple smaller networks. A router analyzes each token and activates only the 2 or 3 experts best suited for it. This decouples total parameter size from inference compute.

Routing evolution prevents expert collapse



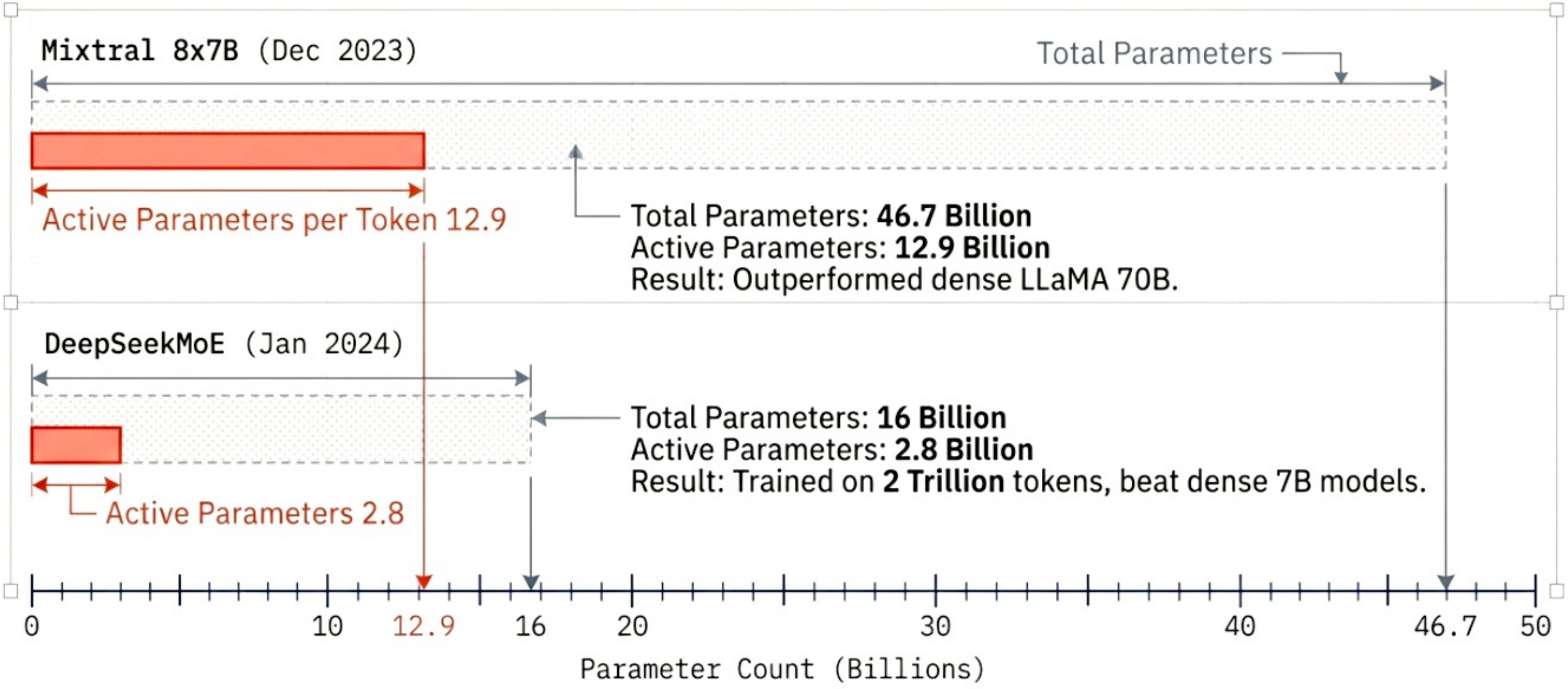
Evolution of MoE routing logic, from simple top-k selection to fine-grained segmentation and finally shared expert isolation, enhances model efficiency, specialization, and prevents expert collapse by separating general knowledge from niche tasks.

Well-known MoE Model: Mixture-8x7B

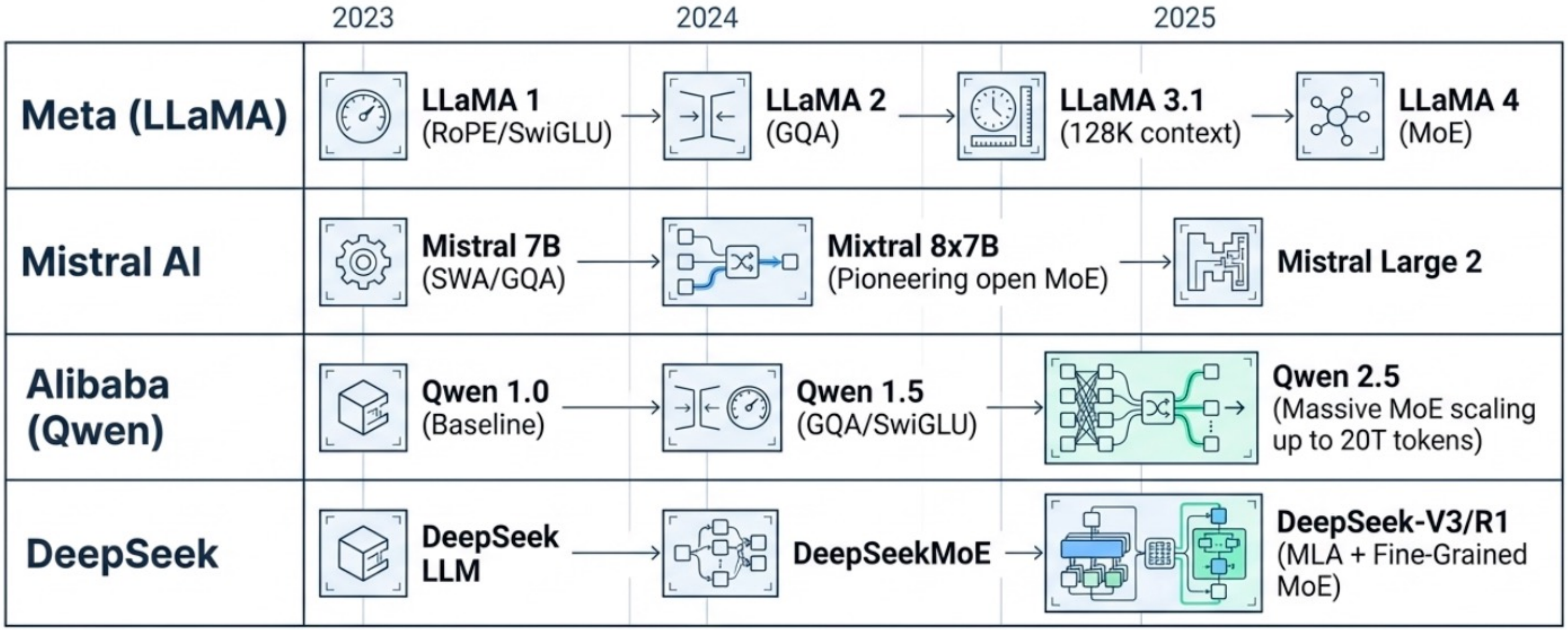


Visualizing Sparsity: MoE by the Numbers

Total size dictates knowledge capacity. Active size dictates inference cost.



The Architectural Convergence (2023-2025)



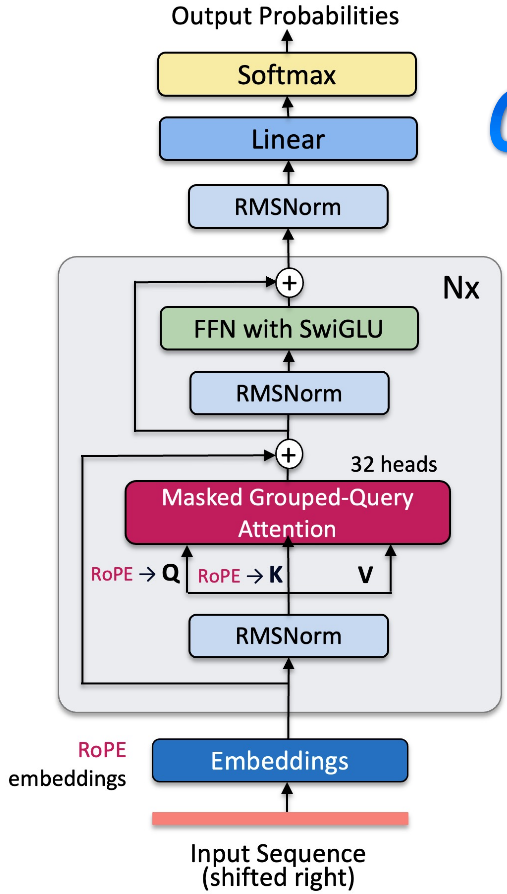
Key Takeaway: Over just two years, RoPE, GQA, and MoE transformed from experimental academic papers into the mandatory, de facto standard engineering blueprint for all frontier open-weight models.

The Open-Weight Standard: Meta's LLaMA Blueprint

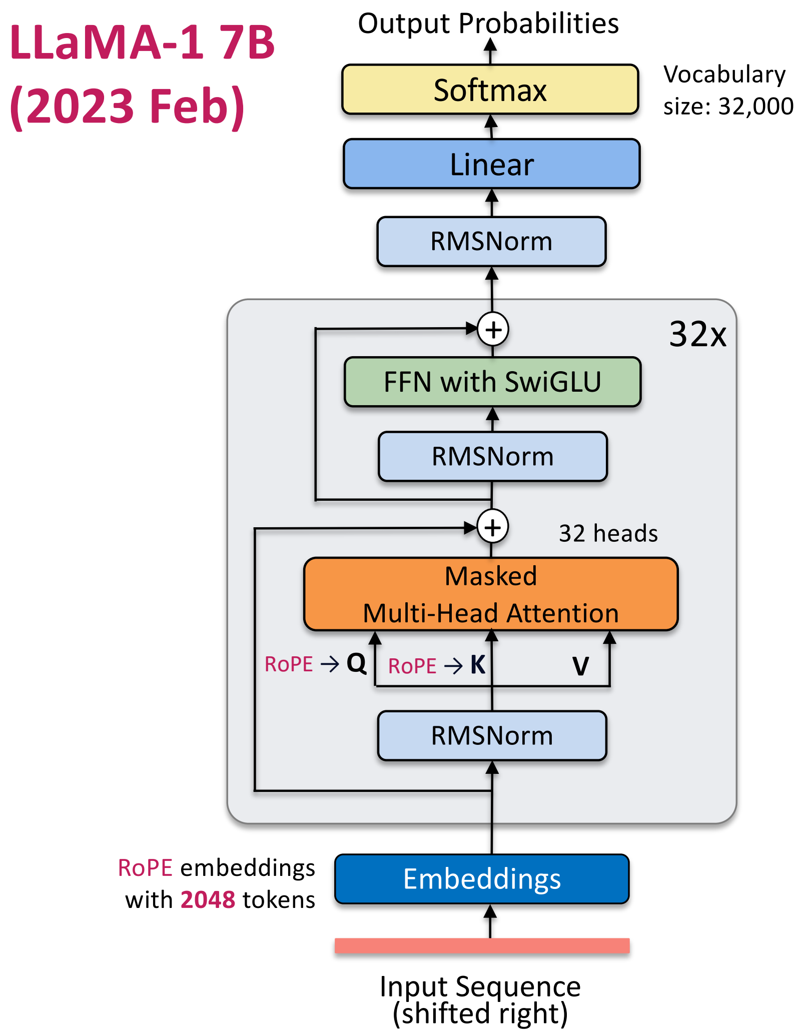
The Winning Recipe

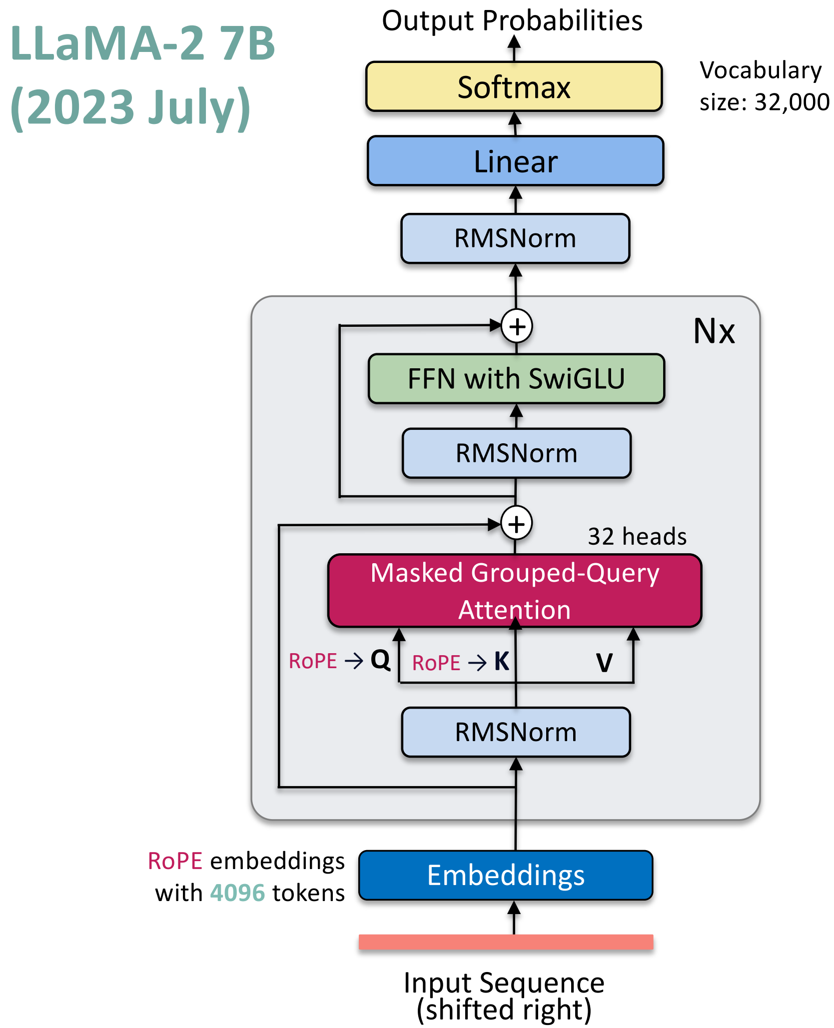
- RoPE replacing absolute positional embeddings.
- RMSNorm replacing LayerNorm
- SwiGLU replacing GELU.
- GQA replacing standard MHA (in later versions).
- Dropped dropout entirely.

LLaMA 1	65B	2k context
LLaMA 2	70B	4k context
LLaMA 3.1	405B	128k context

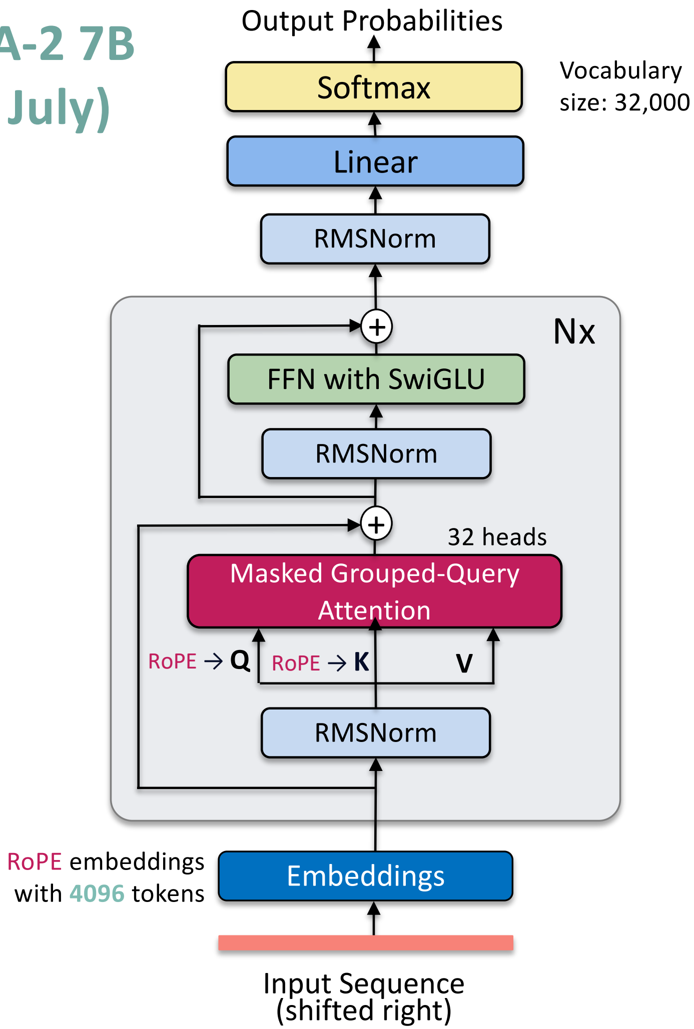


Meta

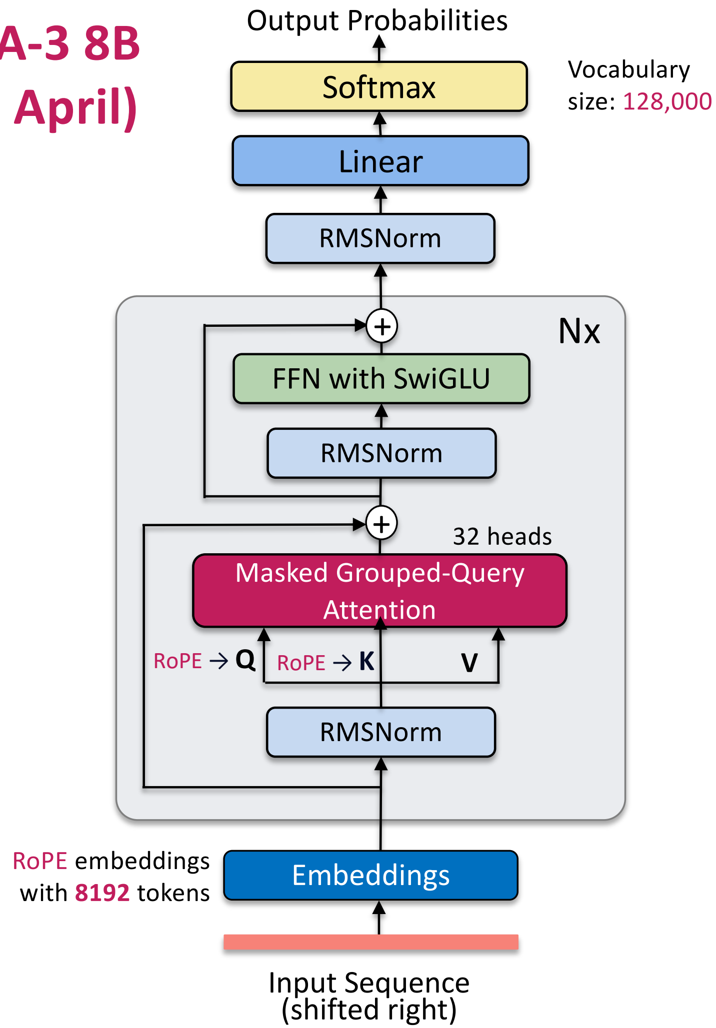

LLaMA-1 7B
(2023 Feb)
Output Probabilities
Softmax
Vocabulary
size: 32,000
Linear
RMSNorm
32x
FFN with SwiGLU
RMSNorm
32 heads
Masked
Multi-Head Attention
RoPE → Q
RoPE → K
V
RMSNorm
RoPE embeddings
with 2048 tokens
Embeddings
Input Sequence
(shifted right)


LLaMA-2 7B
(2023 July)
Output Probabilities
Softmax
Vocabulary
size: 32,000
Linear
RMSNorm
Nx
FFN with SwiGLU
RMSNorm
32 heads
Masked Grouped-Query
Attention
RoPE → Q
RoPE → K
V
RMSNorm
RoPE embeddings
with 4096 tokens
Embeddings
Input Sequence
(shifted right)

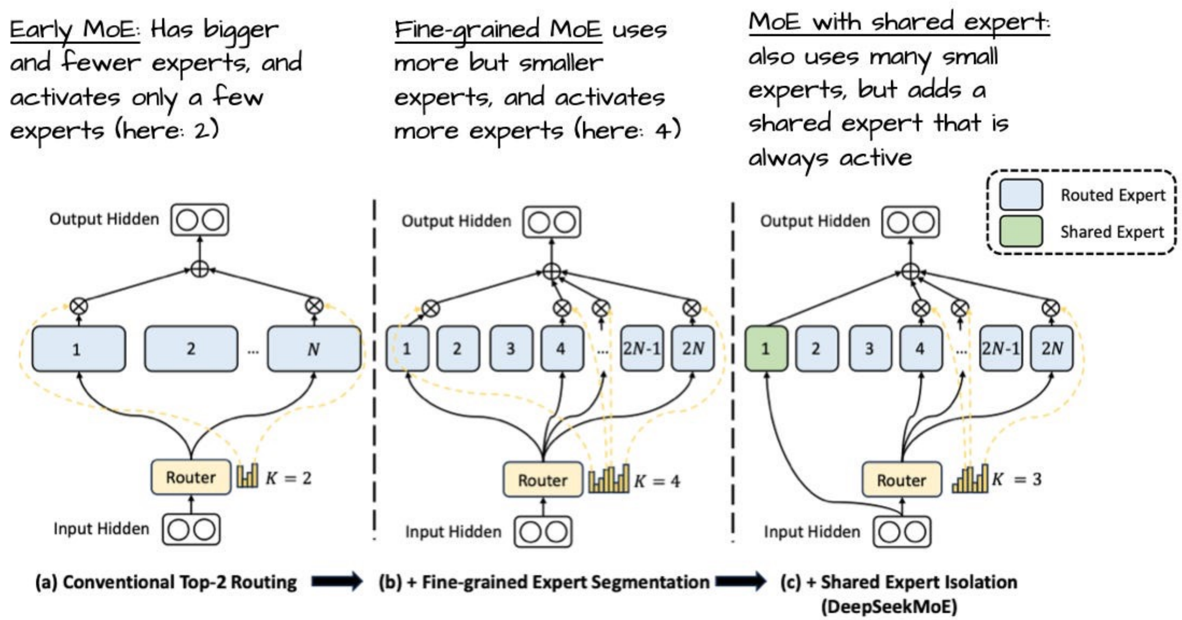
LLaMA-2 7B
(2023 July)



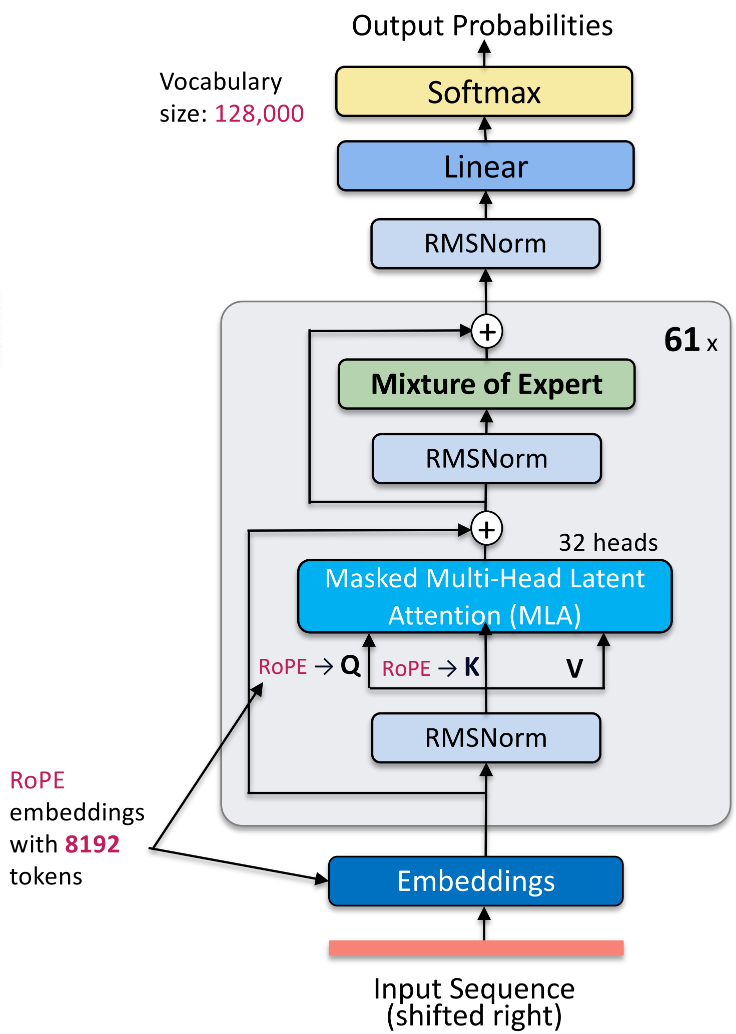
LLaMA-3 8B
(2024 April)



DeepSeek V3/R1 (2025)

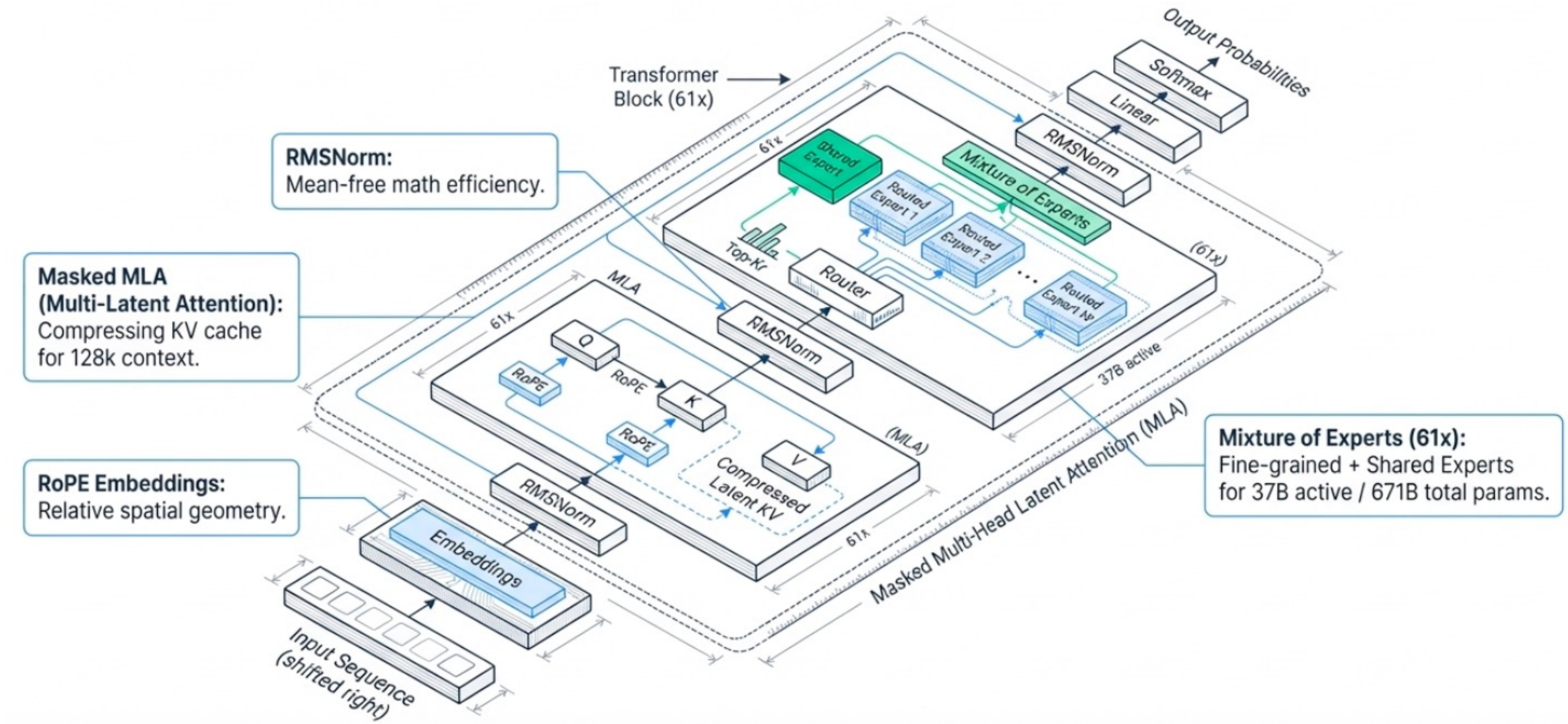


An annotated figure from "DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models", https://arxiv.org/abs/2401.06066

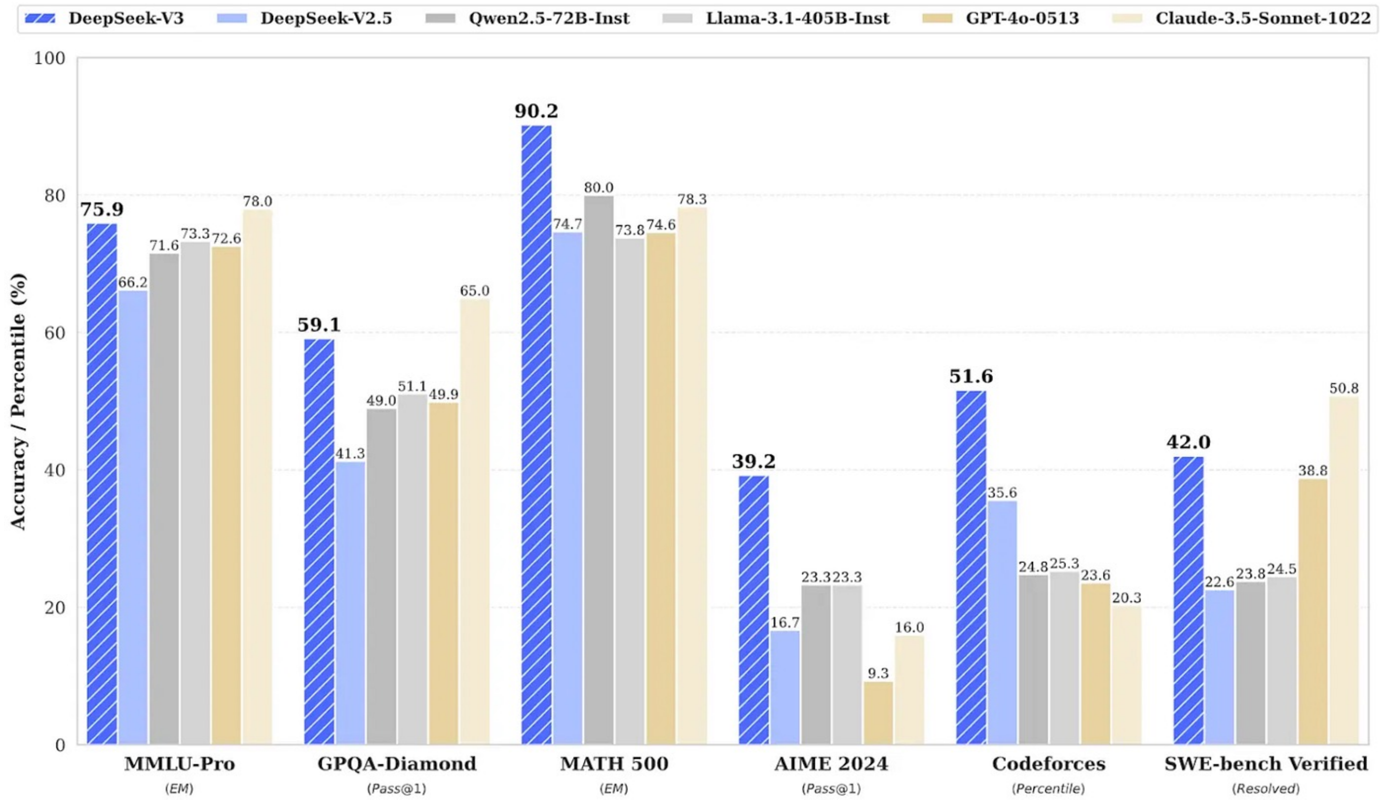


The State of the Art: DeepSeek V3 & R1

The modern frontier model is a masterpiece o f assembled efficiency. It integrates every major breakthrough to solve the physical constraints of scaling.



Architectural Efficiency vs. Compute Brute Force



Training Cost Comparison

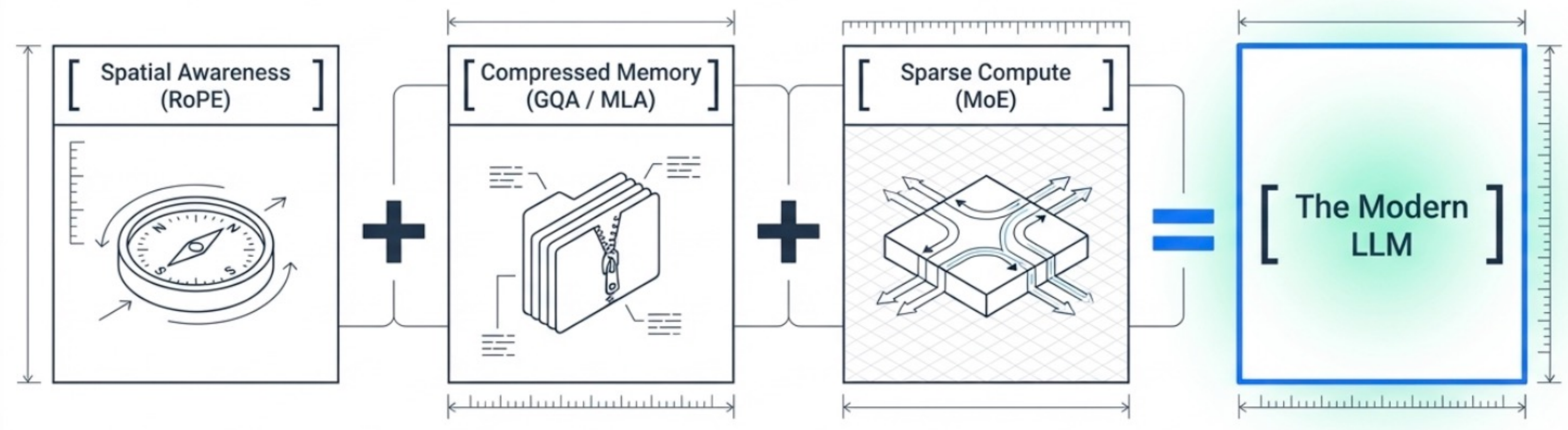
LLaMA 3 (Dense): Required 30 Million GPU hours to train.

DeepSeek-V3 (MoE + MLA): Required just 2.8 Million GPU hours to train.

Conclusion: Smart architecture (MoE, MLA, FP8) decisively beats sheer compute scale, yielding state-of-the-art reasoning (MATH 500, Codeforces) at a fraction of the cost.

The Era of Intelligent Scaling

The journey from GPT-1 to today was not just about plugging in more GPUs. It was a triumph of intelligent design-solving the physical bounds of memory and compute through mathematical elegance.



The future of AI is not just larger models, but smarter architectures.

Efficiency, sparsity, and intelligent scaling have defined the last 7 years of LLM design. As we move toward native multimodal and reasoning models, the blueprint for the next generation is already being drawn.