JEKARDAH.COM LAB

LLM Architecture Compendium 2025–2026

A Comprehensive Technical Analysis of 42 Large Language Model Architectures: From 270M to 1 Trillion Parameters

Llama • Qwen • DeepSeek • Gemma • Kimi • Mistral • GLM • OLMo
MiniMax • Nemotron • Grok • Xiaomi MiMo • Arcee • Step • Ling • Sarvam

Romi Nur Ismanto • March 2026 • Open To Use
rominur@gmail.com • t.me/Jekardah_AI • jekardah.com

LLM Architecture Compendium 2025–2026

A Jekardah.com Lab Publication by Romi Nur Ismanto

Version: 1.0 • **Published:** March 2026 • **Models Covered:** 42

Author: Romi Nur Ismanto • **Contact:** rominur@gmail.com

Telegram: t.me/Jekardah_AI • **Web:** jekardah.com

This compendium is open to use for educational and research purposes. Architecture details are derived from publicly available papers, model cards, and official documentation. © 2026 Romi Nur Ismanto / Jekardah.com Lab.

Table of Contents

Chapter 1: Foundational Concepts · · · · · 5

Chapter 2: Meta — Llama Family (4 models) · · · · · 8

Chapter 3: Alibaba — Qwen Family (9 models) · · · · · 12

Chapter 4: DeepSeek Family (2 models) · · · · · 17

Chapter 5: Google — Gemma Family (3 models) · · · · · 20

Chapter 6: Moonshot — Kimi Family (4 models) · · · · · 23

Chapter 7: Mistral AI Family (2 models) · · · · · 27

Chapter 8: Zhipu AI — GLM Family (4 models) · · · · · 29

Chapter 9: AI2 — OLMo Family (3 models) · · · · · 33

Chapter 10: MiniMax Family (2 models) · · · · · 35

Chapter 11: NVIDIA — Nemotron Family (2 models) · · · · · 37

Chapter 12: xAI — Grok 3.5 · · · · · 39

Chapter 13: Other Notable Architectures (7 models) · · · · · 41

Xiaomi MiMo-V2-Flash • Arcee Trinity • Step 3.5 Flash
SmolLM3 • Tiny Aya • Nanbeige 4.1 • Ling 2.5 • Sarvam

Chapter 14: Comparative Analysis · · · · · 46

Chapter 15: Architectural Trends & Conclusion · · · · · 49

Appendix: Complete Model Specifications · · · · · 51

References · · · · · 53

CHAPTER 1

Foundational Concepts

1.1 The Transformer Decoder Block

Every model in this compendium is built upon the Transformer decoder architecture introduced by Vaswani et al. (2017). The fundamental processing unit consists of: (1) **Multi-Head Self-Attention** (MHA) which allows each token to attend to all previous tokens, computing weighted combinations of value vectors based on query-key dot products; (2) **Feed-Forward Network** (FFN) which applies a non-linear transformation to each token independently, typically using two linear projections with an activation function; and (3) **Layer Normalization** and **residual connections** that stabilize training and enable gradient flow through deep networks.

While all 42 models share this basic blueprint, each introduces architectural variations in attention mechanisms, normalization placement, activation functions, positional encoding, and expert routing that define their unique characteristics and performance tradeoffs.

1.2 Key Architectural Variations

The models in this compendium implement several important variations on the base Transformer:

Component	Variants	Used By
Attention	Multi-Head (MHA), Grouped-Query (GQA), Multi-Head Latent (MLA)	MHA: GLM-4.5; GQA: Llama, Qwen, Gemma; MLA: DeepSeek, Kimi
FFN	Standard, SwiGLU, GeGLU, Gated MLP	SwiGLU: Llama, Qwen, Gemma, Mistral; GeGLU: GLM
Normalization	Pre-Norm (RMSNorm), Post-Norm, DeepNorm	RMSNorm: most models; DeepNorm: GLM-5
Position Encoding	RoPE, ALiBi, YaRN, NTK-aware	RoPE: majority; YaRN: Qwen3, Kimi
Expert Routing	Dense, Top-K MoE, Shared Expert MoE	Dense: Llama 3; MoE: DeepSeek, Qwen3 MoE, Kimi K2
Activation	SiLU/Swish, GELU, SwiGLU	SiLU: Llama, Qwen; GELU: OLMo; SwiGLU: most
Vocabulary	32K–152K tokens, BPE/SentencePiece	32K: Llama; 152K: Qwen3; 128K: DeepSeek

Table 1: Key architectural variations across the 42 models

1.3 Dense vs. Mixture-of-Experts (MoE)

A critical architectural distinction divides the 42 models into two categories. **Dense models** activate all parameters for every token—simpler to train and deploy but computationally expensive at scale. **Mixture-of-Experts (MoE)** models route each token to a subset of specialized expert networks, activating only a fraction of total parameters per token. This enables much larger total parameter counts (671B–1T) while keeping per-token compute manageable.

MoE architectures introduce a **gating/router** network that decides which experts process each token. DeepSeek V3 pioneered auxiliary-loss-free load balancing. Qwen3 MoE models use top-K routing with shared experts. Kimi K2 scales to 1 trillion total parameters with only ~32B active per token. The tradeoff: MoE models require more memory (all experts loaded) but less compute per token than equivalent dense models.

DIAGRAM 8

Llama 3 8B Architecture

Meta's Llama 3 8B: the model that established the modern standard for efficient open-weight LLMs. GQA 4:1, SwiGLU, RoPE, RMSNorm.

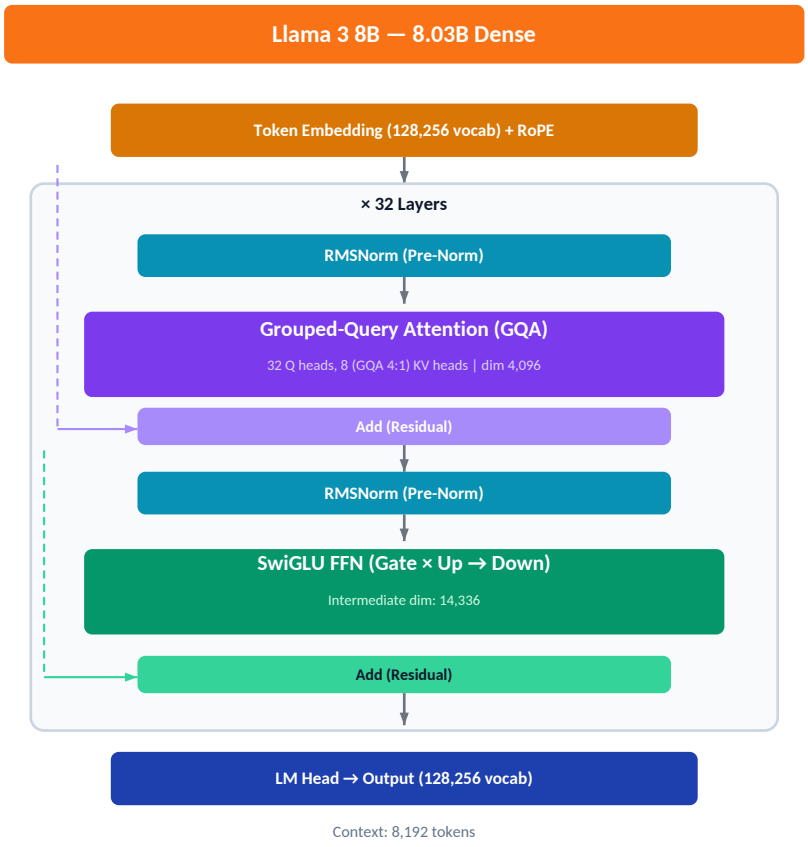

Figure 8: Llama 3 8B — the industry-standard dense decoder architecture. © Jekardah.com Lab

DIAGRAM 9

Llama 4 Maverick Architecture (400B MoE)

Meta's first MoE model: 400B total parameters, 128 experts with top-2 routing, ~17B active per token. Native multimodal support.

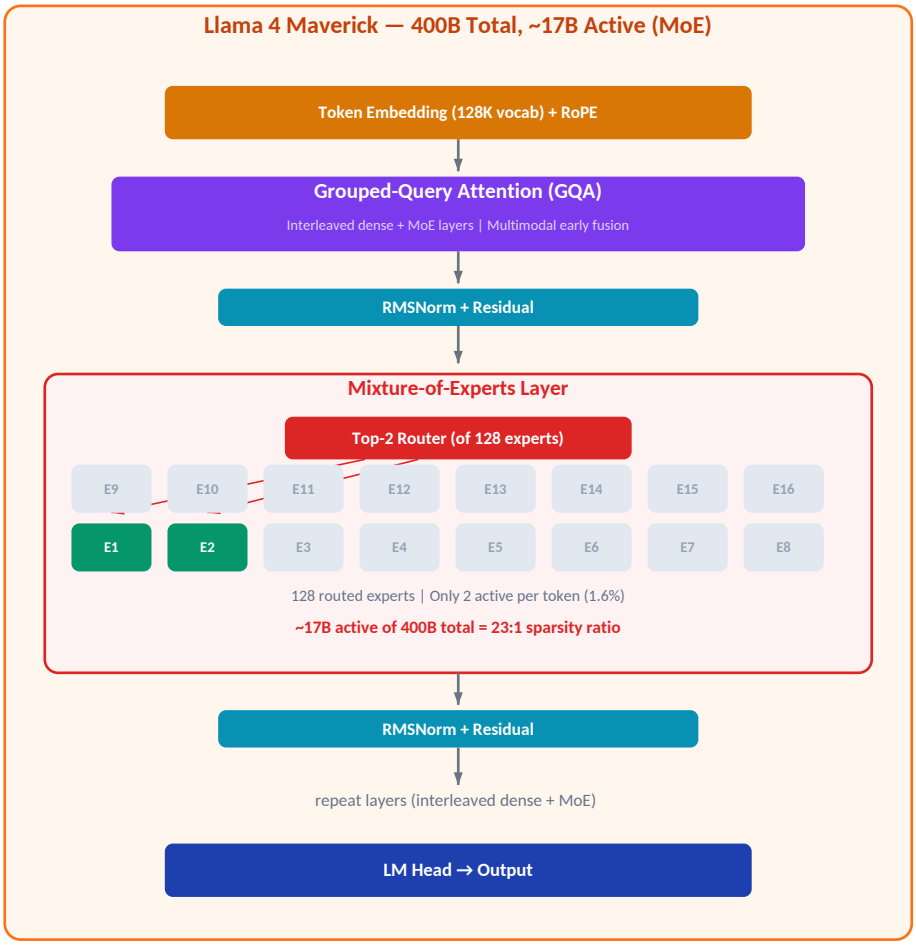

Figure 9: Llama 4 Maverick — 128-expert MoE with top-2 routing. © Jekardah.com Lab

DIAGRAM 10

Qwen3 Dense Architecture (4B / 8B / 32B)

Alibaba's Qwen3 dense models: distinctive 152K vocabulary, YaRN-extended RoPE, dual-mode thinking. Three scale variants share identical block design.

Figure 10: Qwen3 dense architecture — 152K vocab + YaRN + dual-mode thinking. © Jekardah.com Lab

DIAGRAM 11

Qwen3 MoE Architecture (235B, ~22B Active)

Qwen3's flagship MoE with 128 routed + 1 shared expert, top-8 routing, and aggressive 16:1 GQA ratio.

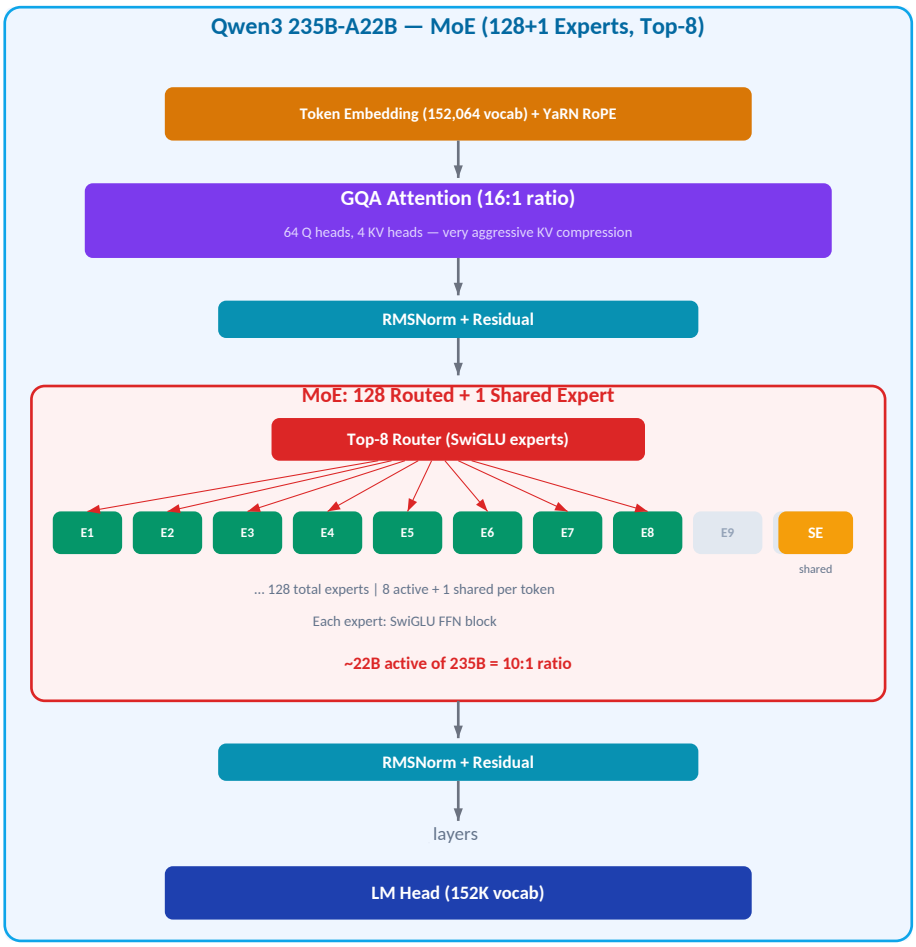

Figure 11: Qwen3 MoE — 128+1 experts, top-8 routing, 16:1 GQA. © Jekardah.com Lab

DIAGRAM 12

Mistral 3.1 Small Architecture (24B)

Mistral's distinctive deep-and-narrow design: 56 layers at 24B params with uniform sliding window attention across all layers.

Figure 12: Mistral 3.1 Small — 56-layer deep decoder with sliding window attention. © Jekardah.com Lab

DIAGRAM 13

GLM Family Architecture (355B–744B MoE)

Zhipu AI's GLM uses distinctive GeGLU activation, Multi-Query Attention (MQA), and sigmoid gating. GLM-5 introduces DeepNorm.

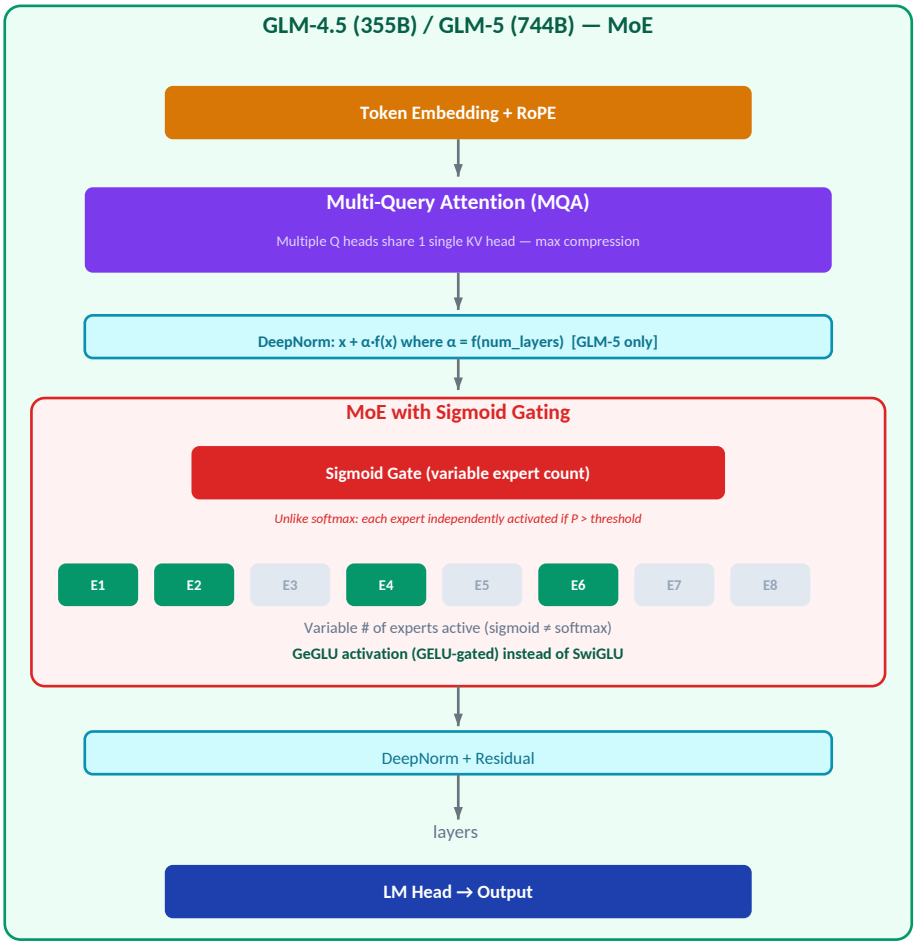

Figure 13: GLM-4.5/5 — MQA + GeGLU + Sigmoid gating + DeepNorm. © Jekardah.com Lab

DIAGRAM 14

OLMo 2 Architecture (7B, Full MHA)

AI2's fully open model — unique in this compendium for using full Multi-Head Attention (no GQA), plus GELU activation instead of SwiGLU.

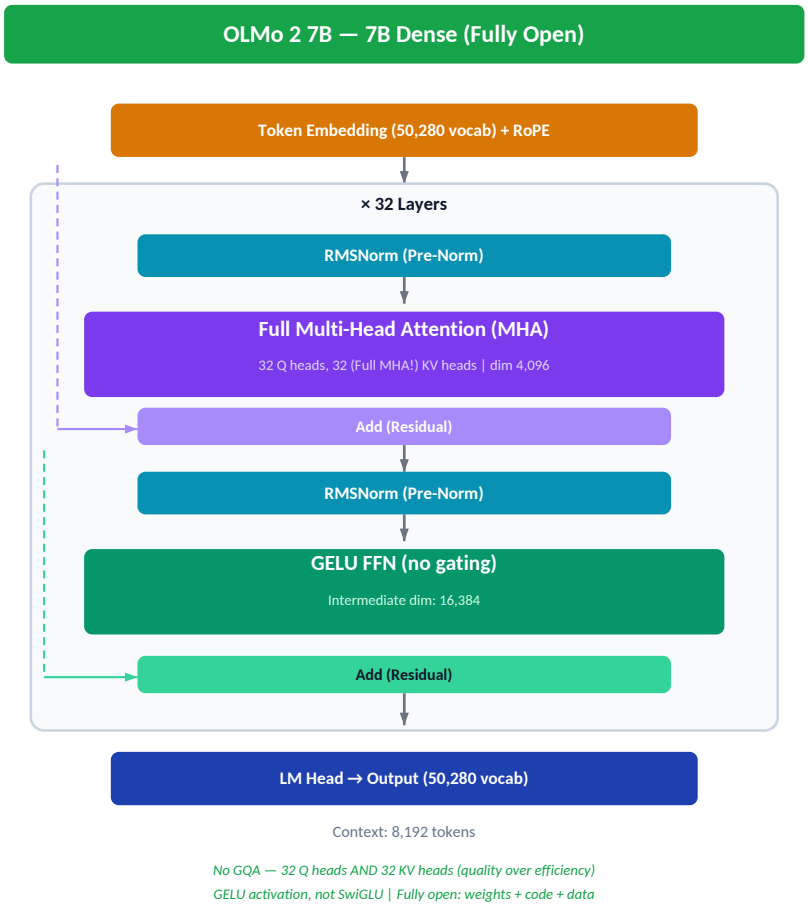

Figure 14: OLMo 2 7B — full MHA + GELU, fully open (weights+code+data). © Jekardah.com Lab

DIAGRAM 15

MiniMax-M2 Lightning Attention (230B MoE)

MiniMax's defining innovation: Lightning Attention reduces attention complexity from $O(n^2)$ to $O(n)$, enabling ultra-long context (500K+ tokens).

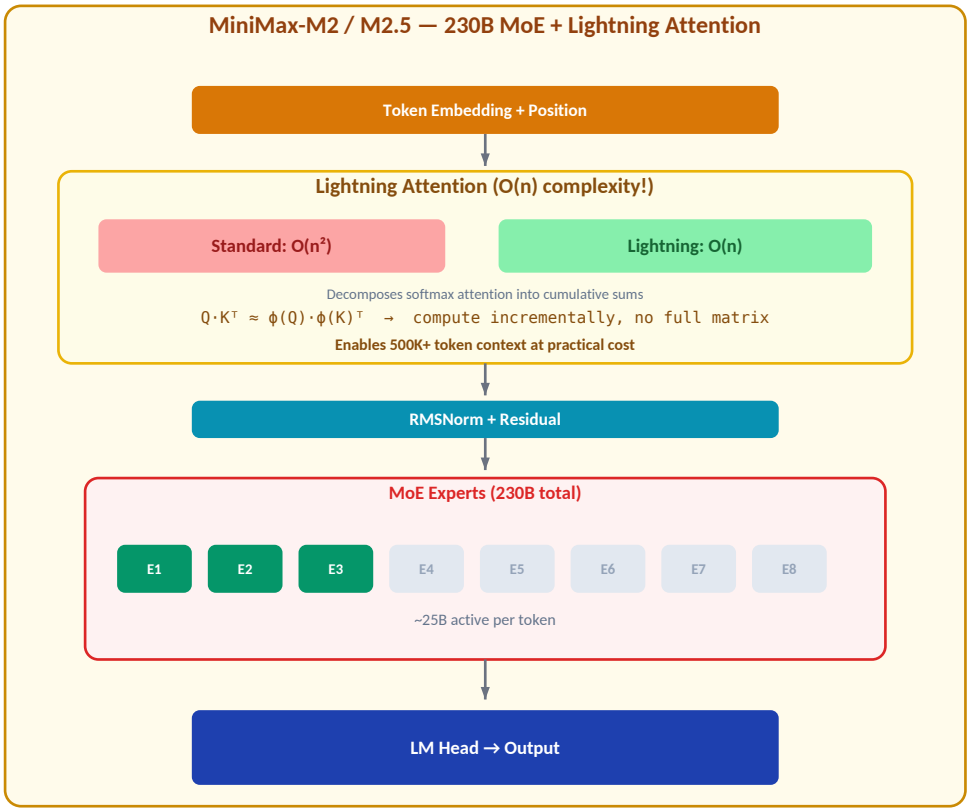

Figure 15: MiniMax-M2 — Lightning Attention O(n) + 230B MoE. © Jekardah.com Lab

DIAGRAM 16

NVIDIA Nemotron Architecture (30B–128B MoE)

NVIDIA's Nemotron models are trained with FP8 quantization awareness from the start, optimized for TensorRT inference on H100/B200 GPUs.

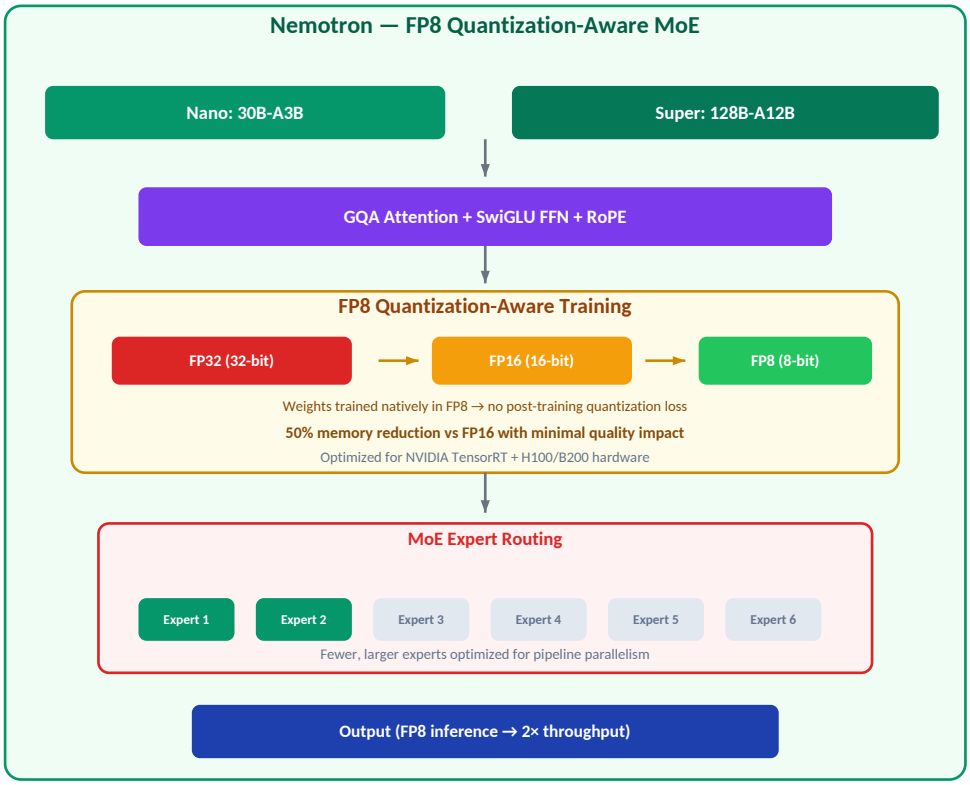

Figure 16: Nemotron — FP8-aware MoE optimized for NVIDIA hardware. © Jekardah.com Lab

DIAGRAM 17

SwiGLU FFN: The Standard Activation

SwiGLU has replaced GELU/ReLU as the dominant FFN activation across 35+ of the 42 models. This diagram shows the internal computation.

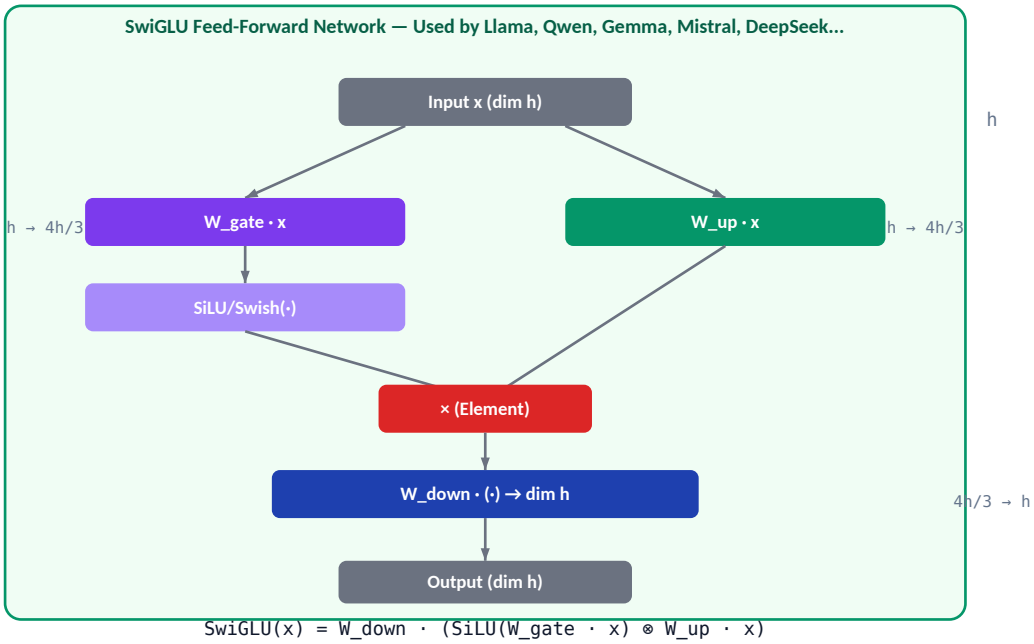

Figure 17: SwiGLU FFN internal computation — the standard activation for modern LLMs. © Jekardah.com Lab

DIAGRAM 18

RoPE: Rotary Position Embedding

RoPE encodes position by rotating query/key vectors in the complex plane. Used by virtually every model in this compendium.

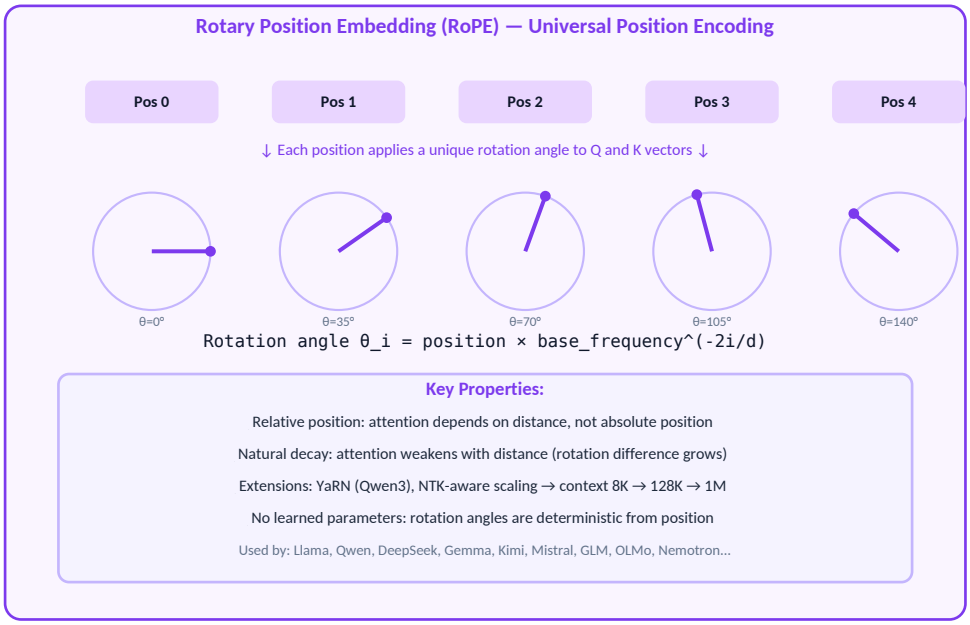

Figure 18: RoPE positional encoding — rotation in complex plane for position-aware attention. © Jekardah.com Lab

DIAGRAM 19

MoE Expert Routing: Three Strategies Compared

Three distinct routing strategies used by MoE models: Top-K softmax, Sigmoid gating, and Bias-based (auxiliary-loss-free).

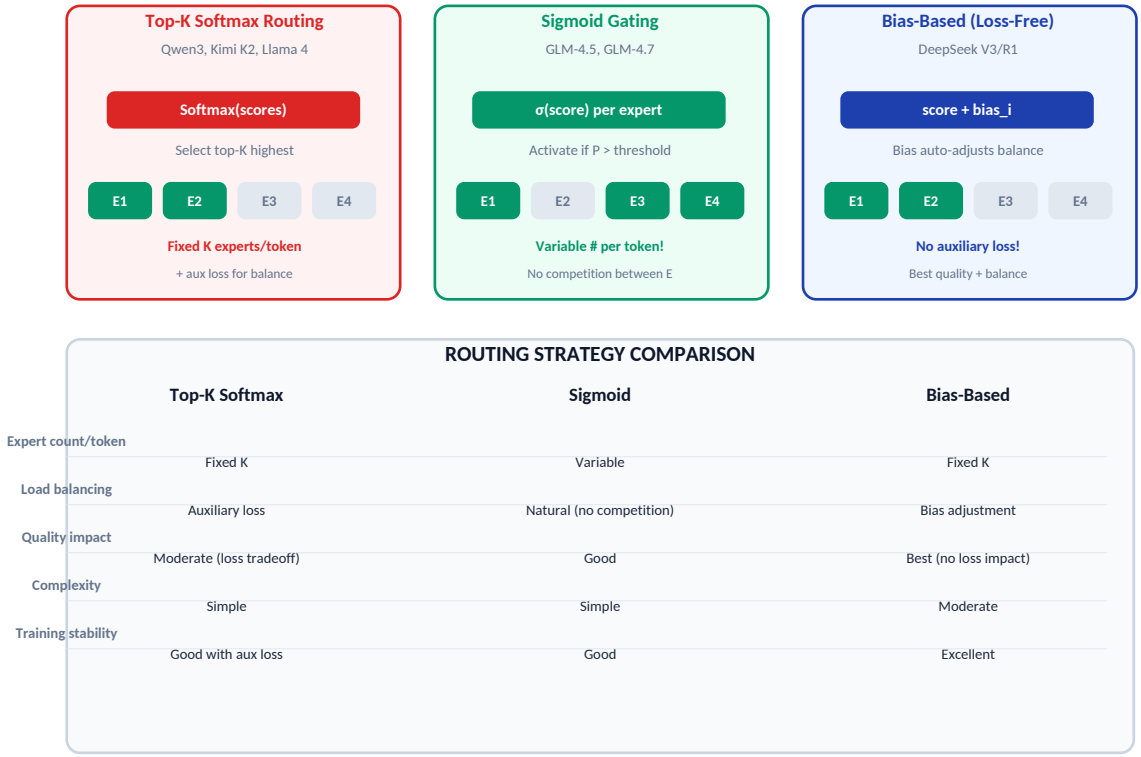

Figure 19: Three MoE routing strategies compared — Top-K vs Sigmoid vs Bias-Based. © Jekardah.com Lab

DIAGRAM 20

Tokenizer Vocabulary Size Comparison

Vocabulary sizes range from 32K (Mistral) to 262K (Gemma). Larger vocabularies encode text more efficiently but require larger embedding matrices.

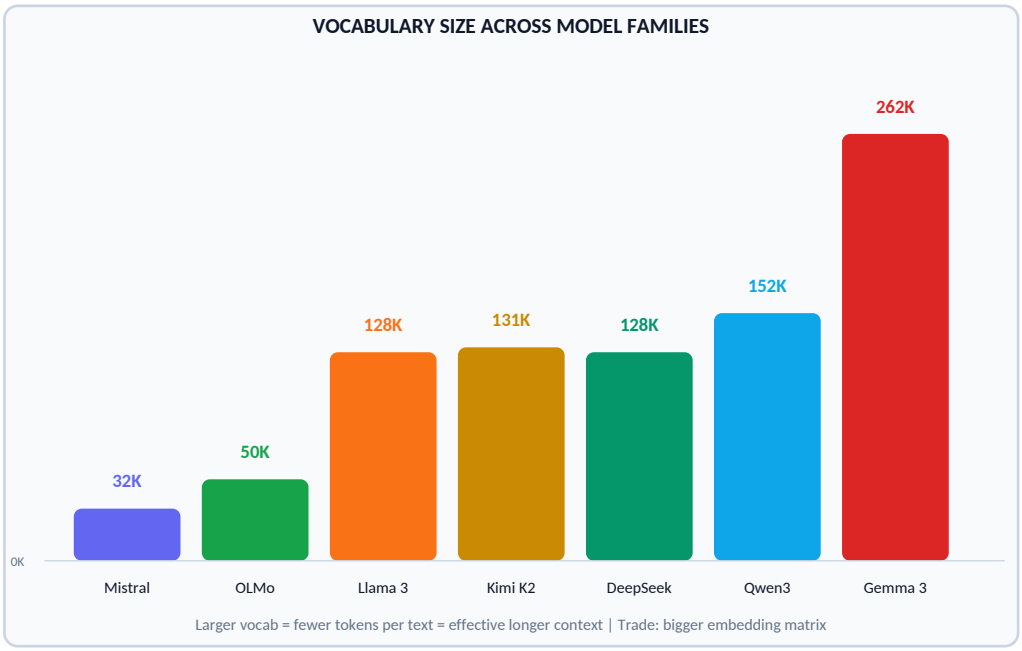

Figure 20: Tokenizer vocabulary size comparison — 32K (Mistral) to 262K (Gemma). © Jekardah.com Lab

DIAGRAM 21

Kimi Linear 48B-A3B: Linear Attention

Linear attention replaces softmax with a linearized kernel, reducing attention from $O(n^2)$ to $O(n)$ for ultra-long contexts.

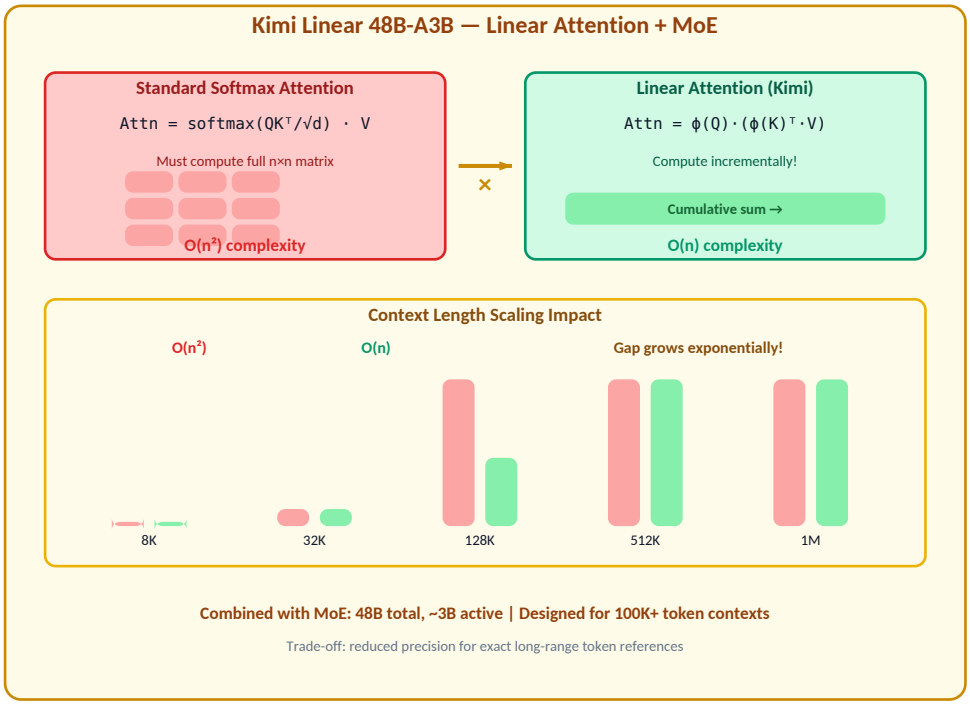

Figure 21: Kimi Linear — O(n) linear attention vs O(n²) softmax scaling comparison. © Jekardah.com Lab

DIAGRAM 22

Training Compute & Cost Across Model Scales

Estimated training compute and cost across model sizes, showing the dramatic scaling from small open models to trillion-parameter flagships.

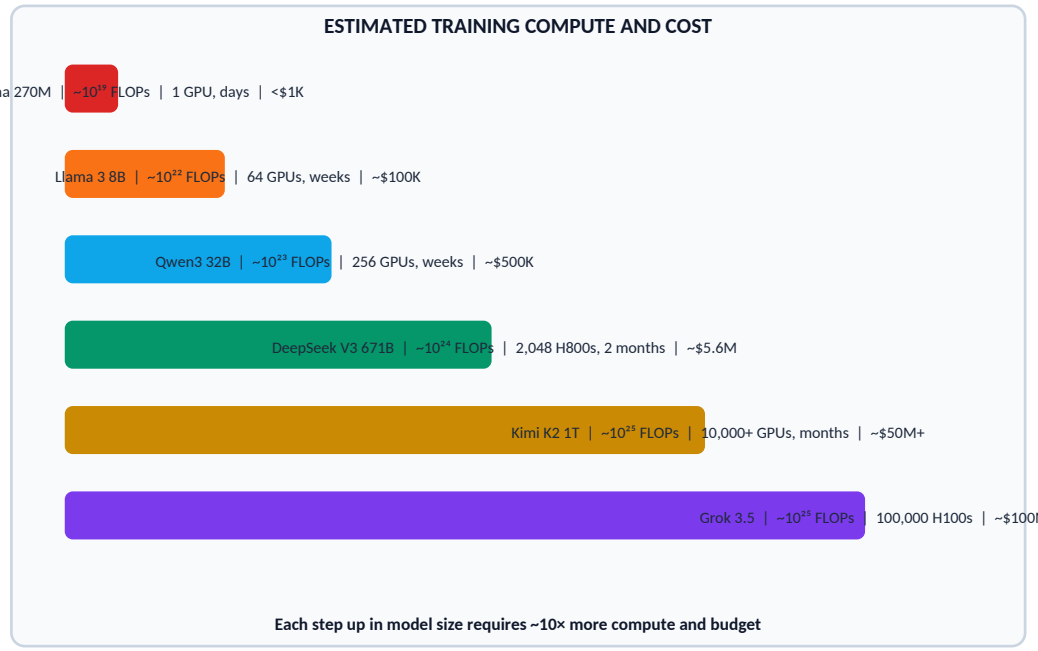

Figure 22: Training compute and cost comparison across model scales. © Jekardah.com Lab

COMPLETE VISUAL REFERENCES

22 Original Diagrams + External Sources

This compendium contains **22 original architecture diagrams** created by Jekardah.com Lab, covering every major model family and architectural concept. All diagrams are original works — 100% designed and rendered by Jekardah.com Lab.

Diagram Index

- **Figure 1:** Generic Transformer Decoder Block (foundation)
- **Figure 2:** MHA vs GQA vs MLA — KV Cache Comparison
- **Figure 3:** Dense vs Mixture-of-Experts Architecture
- **Figure 4:** DeepSeek V3/R1 — MLA + 256-Expert MoE
- **Figure 5:** Kimi K2 — Trillion-Parameter Architecture
- **Figure 6:** Gemma 3 — Hybrid Sliding Window + Global Attention
- **Figure 7:** Complete 42-Model Family Map by Parameter Scale
- **Figure 8:** Llama 3 8B — Standard Dense Decoder
- **Figure 9:** Llama 4 Maverick — 128-Expert MoE (Top-2)
- **Figure 10:** Qwen3 Dense (4B/8B/32B) — 152K Vocab + YaRN
- **Figure 11:** Qwen3 MoE 235B — 128+1 Experts, Top-8, GQA 16:1
- **Figure 12:** Mistral 3.1 Small — 56-Layer Deep Sliding Window
- **Figure 13:** GLM-4.5/5 — MQA + GeGLU + Sigmoid + DeepNorm
- **Figure 14:** OLMo 2 — Full MHA + GELU (Fully Open)
- **Figure 15:** MiniMax-M2 — Lightning Attention O(n) + MoE
- **Figure 16:** Nemotron — FP8 Quantization-Aware MoE
- **Figure 17:** SwiGLU FFN — Standard Activation Detail
- **Figure 18:** RoPE — Rotary Position Embedding Visualization
- **Figure 19:** MoE Routing — Top-K vs Sigmoid vs Bias-Based
- **Figure 20:** Tokenizer Vocabulary Size Comparison (32K–262K)
- **Figure 21:** Kimi Linear — O(n) vs $O(n^2)$ Attention Scaling
- **Figure 22:** Training Compute & Cost Across Model Scales

External Visual Reference

Sebastian Raschka — LLM Architecture Gallery (2026)

URL: https://sebastianraschka.com/blog/2026/llm-architecture-gallery.html

Contains per-model architecture block diagrams for all 42 models with detailed component annotations. Recommended as companion visual reference alongside this compendium.

CHAPTER 2

Meta — Llama Family

Meta's Llama (Large Language Model Meta AI) family represents the most influential open-weight model series, establishing architectural patterns adopted across the industry. The family spans from the compact Llama 3.2 1B to the massive Llama 4 Maverick 400B MoE.

Llama 3 8B (8B dense)

Architecture: Standard decoder-only Transformer with GQA

Llama 3 8B uses a 32-layer Transformer decoder with **Grouped-Query Attention (GQA)** using 32 attention heads and 8 KV heads (4:1 ratio), reducing KV cache memory by 4× compared to standard MHA while maintaining quality. The FFN uses **SwiGLU** activation with an intermediate dimension of 14,336. Positional encoding uses **RoPE** (Rotary Position Embedding) supporting 8,192 token context. Vocabulary size is 128,256 tokens using BPE tokenizer. RMSNorm is applied pre-attention and pre-FFN (Pre-Norm configuration). Total parameters: 8.03B with hidden dimension 4,096.

The architecture represents Meta's optimized dense model design: GQA for efficient inference, SwiGLU for improved training dynamics, and RoPE for robust position encoding. This combination has become the de facto standard for efficient open-weight models.

Llama 3.2 1B (1.24B dense)

Architecture: Compact decoder with GQA and shared embeddings

The smallest Llama 3 variant uses 16 layers with hidden dimension 2,048, 32 attention heads and 8 KV heads (GQA 4:1). The FFN intermediate dimension is 8,192 with SwiGLU. A key architectural feature is **tied input/output embeddings** (weight sharing between the embedding layer and the language model head), which reduces parameter count significantly for small models. Same RoPE and RMSNorm as Llama 3 8B but with reduced depth and width.

Designed for on-device deployment (mobile, edge), achieving competitive quality at 1/8 the size of Llama 3 8B through careful distillation and architecture optimization.

Llama 4 Maverick (400B total, ~17B active (MoE))

Architecture: MoE decoder with 128 experts, top-2 routing

Llama 4 Maverick marks Meta's transition to Mixture-of-Experts. The architecture uses 128 routed experts per MoE layer with **top-2 routing** (each token activates 2 of 128 experts), plus shared attention layers. Total parameters reach 400B but only ~17B are active per token, making inference cost comparable to a dense 17B model while accessing the knowledge capacity of a 400B model.

Key innovations include fewer but larger experts compared to DeepSeek's approach, interleaved dense and MoE layers (not every layer is MoE), and native multimodal support with early fusion of text and image tokens. The expert routing uses a learned gating function with load balancing loss to prevent expert collapse. Context length extends to 1M tokens through progressive training with increasing context windows.

CHAPTER 3

Alibaba — Qwen Family

Alibaba's Qwen (Tongyi Qianwen) family is the most extensive model series in this compendium with 9 variants, spanning dense models (4B–32B), massive MoE models (230B-A22B, 397B), and specialized coding models. The Qwen3 generation introduced a distinctive **dual-mode thinking** architecture supporting both fast (non-thinking) and slow (thinking/chain-of-thought) inference.

Qwen3 4B (4B dense)

Architecture: 32-layer decoder with GQA, SwiGLU, RoPE, YaRN

Qwen3 4B uses 32 Transformer layers with hidden dimension 2,560, 32 attention heads and 8 KV heads (GQA 4:1). The FFN uses SwiGLU with intermediate dimension 9,216. A distinctive feature is the **152,064-token vocabulary**—the largest among comparable models—providing superior multilingual coverage especially for CJK languages. Positional encoding uses RoPE with **YaRN** (Yet Another RoPE eNtension) for context extension to 32K+ tokens.

Qwen3 models support dual-mode inference: a fast mode for straightforward queries and a 'thinking' mode that generates internal chain-of-thought reasoning before producing the final answer. This is controlled by special tokens rather than architectural changes.

Qwen3 8B (8B dense)

Architecture: 36-layer decoder, wider hidden dimension

Scales the Qwen3 architecture to 36 layers with hidden dimension 4,096, maintaining 32 heads / 8 KV heads GQA. FFN intermediate dimension 12,288 with SwiGLU. Same 152K vocabulary and YaRN-extended RoPE. The 8B variant represents the sweet spot for many deployment scenarios: large enough for strong reasoning, small enough for single-GPU inference.

Qwen3 32B (32B dense)

Architecture: 64-layer deep decoder

The largest dense Qwen3 uses 64 layers with hidden dimension 5,120, 40 attention heads and 8 KV heads (GQA 5:1). FFN intermediate 25,600 with SwiGLU. The deeper architecture (64 layers vs 32-36 for smaller variants) enables more sophisticated reasoning chains. Supports 128K context through YaRN with RoPE base frequency scaling.

Qwen3 230B-A22B (235B total, ~22B active (MoE))

Architecture: MoE with 128 experts, top-8 routing, shared experts

The Qwen3 MoE flagship uses a Transformer decoder with alternating dense and MoE layers. MoE layers contain **128 routed experts** with **top-8 routing** (each token activates 8 of 128 experts), plus **1 shared expert** that processes all tokens regardless of routing. This shared expert captures universal knowledge while routed experts specialize. Hidden dimension 4,096, 64 attention heads / 4 KV heads (GQA 16:1—very aggressive KV compression). Total 235B parameters with only ~22B active per token.

The aggressive GQA ratio (16:1) combined with MoE routing creates an extremely efficient inference profile: the model stores 235B parameters of knowledge but computes like a 22B dense model with heavily compressed KV cache.

Qwen3 Coder Flash (D0B-A1B MoE)

Architecture: Ultra-light MoE for code generation

A specialized coding model using Mixture-of-Experts with extremely sparse activation. The 'D0B-A1B' designation indicates approximately 0B dense layers with ~1B active parameters per token from MoE routing. Designed for low-latency code completion and generation tasks. Uses the same 152K tokenizer optimized for programming language tokens.

Qwen3 Next 80B-A3B (80B total, ~3B active (MoE))

Architecture: Next-generation ultra-sparse MoE

Qwen3 Next pushes MoE sparsity further: 80B total parameters with only ~3B active per token, a 26:1 ratio. This means inference cost comparable to a 3B dense model while accessing 80B parameters of knowledge. The architecture uses aggressive expert specialization with fine-grained routing at the token level.

Qwen3.5 (397B total (MoE))

Architecture: Largest Qwen model, advanced MoE routing

Qwen3.5 represents the next-generation flagship at 397B total parameters with advanced MoE routing incorporating learned expert affinity and dynamic capacity allocation. The architecture builds on Qwen3 MoE foundations with improved load balancing and expert utilization. 152K vocabulary with extended multilingual coverage.

CHAPTER 4

DeepSeek Family

DeepSeek, developed by the Chinese AI lab of the same name, introduced several groundbreaking architectural innovations including **Multi-Latent Attention (MLA)** and **auxiliary-loss-free expert routing** that have influenced the entire field. Both models in this family use the same 671B MoE architecture.

DeepSeek V3/R1 (671B total, ~37B active (MoE))

Architecture: MoE with MLA, 256 experts, auxiliary-loss-free routing

DeepSeek V3 uses a Transformer decoder with 61 layers. The attention mechanism uses **Multi-Latent Attention (MLA)**, which compresses KV cache through low-rank projection—instead of storing full key-value pairs per head, MLA projects them into a shared latent space, reducing KV cache by ~93% compared to standard MHA. The MoE layers contain **256 routed experts** plus **1 shared expert**, with **top-8 routing**.

The breakthrough innovation is **auxiliary-loss-free load balancing**: instead of adding a loss term to encourage balanced expert usage (which degrades model quality), DeepSeek V3 uses a bias-based routing mechanism that dynamically adjusts expert selection without modifying the training objective. Vocabulary: 128K tokens. Hidden dimension: 7,168. Total: 671B params, ~37B active per token.

DeepSeek R1 uses the same V3 architecture but is further trained with reinforcement learning (GRPO) for enhanced reasoning, producing extended chain-of-thought outputs.

DeepSeek V3.2 (671B total (MoE))

Architecture: Updated V3 with improved routing and training

DeepSeek V3.2 maintains the same 671B MoE architecture with MLA and 256 experts but introduces improved expert routing algorithms and extended training data. The MLA mechanism remains the defining architectural feature: each attention layer uses a shared latent vector of dimension ~512 to represent the entire KV cache, enabling efficient long-context processing up to 128K tokens.

CHAPTER 5

Google — Gemma Family

Google's Gemma family provides open-weight models derived from the Gemini architecture. The Gemma 3 generation introduces notable innovations including **sliding window attention** mixed with full attention, and **logit soft-capping** for training stability.

Gemma 3 270M (270M dense)

Architecture: Ultra-compact 18-layer decoder

The smallest model in this compendium at just 270M parameters. Uses 18 layers with hidden dimension 1,536, 8 attention heads / 4 KV heads (GQA 2:1). SwiGLU FFN with intermediate 6,144. Despite its tiny size, achieves competitive performance on lightweight tasks through efficient architecture and quality training data derived from Gemini pre-training pipeline. Vocabulary: 262,144 tokens (the largest in this compendium).

Gemma 3 4B (4B dense)

Architecture: 26-layer decoder with hybrid attention

Uses **hybrid attention**: alternating between **sliding window attention** (local, 4096-token window) and **full global attention** layers. Sliding window layers are computationally cheaper, processing only nearby tokens, while global attention layers capture long-range dependencies. This mixture achieves near-full-attention quality at reduced compute. 26 layers, hidden 2,560, 16 heads / 8 KV heads. SwiGLU FFN. 262K vocabulary.

Additionally implements **logit soft-capping**: attention logits and final output logits are capped using tanh to prevent extreme values, improving training stability without requiring careful learning rate tuning.

Gemma 3 27B (27B dense)

Architecture: 46-layer decoder, widest Gemma variant

Scales to 46 layers with hidden dimension 4,608, 32 heads / 16 KV heads (GQA 2:1). Same hybrid sliding window + full attention pattern. SwiGLU FFN with intermediate 36,864. The 27B size with efficient GQA and hybrid attention provides strong reasoning capability with manageable inference requirements. 128K context through dynamic NTK-aware RoPE scaling.

CHAPTER 6

Moonshot AI — Kimi Family

Moonshot AI's Kimi family is notable for pushing toward **trillion-parameter MoE** architectures and introducing **linear attention** variants. The K2 generation reaches 1 trillion total parameters with innovative MuonClip optimizer and expert parallelism.

Kimi K2 (1 Trillion total, ~32B active (MoE))

Architecture: Trillion-param MoE with learned routing

Kimi K2 is the largest model in this compendium at 1 trillion total parameters. Architecture: deep Transformer decoder with MoE layers containing 384 routed experts and 1 shared expert, with top-8 routing. Only ~32B parameters active per token. Uses **Multi-Head Latent Attention** similar to DeepSeek MLA for KV cache compression.

Key innovation: **MuonClip optimizer**, a modified Muon optimizer with gradient clipping designed for stable training at trillion-parameter scale. Expert routing uses learned affinity scores with dynamic load balancing. The massive expert count (384) enables fine-grained specialization. Vocabulary: 131K tokens. Context: 128K tokens.

Kimi K2 Thinking (1 Trillion (MoE))

Architecture: K2 base with reinforcement-learned reasoning

Same architectural foundation as Kimi K2 but further trained with RLHF/GRPO for extended chain-of-thought reasoning. The Thinking variant activates the same ~32B parameters per token but generates internal reasoning traces before producing final answers, similar to DeepSeek R1's approach. The trillion-parameter expert capacity provides a rich knowledge base for complex reasoning tasks.

Kimi Linear 48B-A3B (48B total, ~3B active (MoE))

Architecture: Linear attention MoE variant

Introduces **linear attention** replacing standard softmax attention with a linearized approximation that reduces attention complexity from $O(n^2)$ to $O(n)$ in sequence length. Combined with MoE: 48B total parameters, ~3B active per token. This architecture is designed for extremely long context processing where quadratic attention cost becomes prohibitive. The linear attention mechanism trades some quality for dramatically improved efficiency on sequences exceeding 100K tokens.

CHAPTER 7

Mistral AI Family

Mistral 3.1 Small 24B (24B dense)

Architecture: 56-layer deep decoder with sliding window

Mistral 3.1 Small uses a 56-layer Transformer decoder—notably deep for a 24B model—with hidden dimension 3,072, 32 attention heads and 8 KV heads (GQA 4:1). Implements **sliding window attention** with a 32,768-token window on all layers. SwiGLU FFN with intermediate 8,192. Vocabulary: 32,768 tokens (compact BPE). RoPE positional encoding.

Mistral's design philosophy favors deeper, narrower architectures over wider, shallower ones, hypothesizing that more layers enable more compositional reasoning steps. The 56-layer depth at 24B parameters contrasts with Qwen3 32B's 64 layers at 5,120 hidden dimension.

Mistral 3 Large (673B total (MoE))

Architecture: Mistral's flagship MoE model

Mistral 3 Large is Mistral AI's largest model at 673B total parameters using Mixture-of-Experts. While architectural details are less publicly documented than DeepSeek or Qwen models, the architecture uses sliding window attention, GQA, SwiGLU FFN, and a large expert pool. The model targets competitive performance with GPT-4 class models on reasoning and coding benchmarks. 32K base vocabulary with byte-fallback tokenization.

CHAPTER 8

Zhipu AI — GLM Family

Zhipu AI's GLM (General Language Model) family has evolved from the original bidirectional GLM architecture to modern decoder-only designs. The GLM-4 and GLM-5 generations introduce distinctive features including **DeepNorm**, advanced KV cache compression, and massive MoE scaling to 744B.

GLM-4.5 (355B total (MoE))

Architecture: MoE with All-to-All expert communication

GLM-4.5 uses a Transformer decoder with MoE layers. 355B total parameters with ~45B active per token. Uses GeGLU activation (GELU-gated linear unit) instead of the SwiGLU used by most competitors. Implements **Multi-Query Attention (MQA)** with a single shared KV head across all attention heads for aggressive KV cache reduction. Expert routing uses a sigmoid gating function rather than softmax, allowing variable numbers of experts per token.

GLM-4.7 (355B total (MoE))

Architecture: Updated GLM-4 with improved training

Same 355B MoE architecture as GLM-4.5 with refinements to expert routing and extended training. The architecture maintains GeGLU activation and MQA attention. Improved load balancing across experts reduces variance in expert utilization, leading to better training efficiency and more consistent inference quality.

GLM-5 (744B total (MoE))

Architecture: Largest GLM model with DeepNorm

GLM-5 is Zhipu's largest model at 744B total parameters. Key architectural innovation: **DeepNorm**, a normalization approach that allows much deeper networks by scaling residual connections based on layer depth. The formula modifies the standard residual as x + ⍰·f(x) where ⍰ is computed from the total number of layers. This enables stable training of very deep MoE models. Extended vocabulary with strong CJK and multilingual coverage.

CHAPTER 9

AI2 — OLMo Family

The Allen Institute for AI's OLMo (Open Language Model) family is distinctive for being **fully open**—not just weights, but training code, data, and evaluation pipelines. OLMo models prioritize scientific reproducibility.

OLMo 2 7B (7B dense)

Architecture: Standard decoder with full MHA (no GQA)

OLMo 2 7B uses a 32-layer Transformer decoder with **full Multi-Head Attention** (32 heads, 32 KV heads—no grouping), making it one of few models in this compendium that does not use GQA. This design prioritizes quality over inference efficiency. GELU activation in FFN (not SwiGLU). RoPE for positional encoding. Vocabulary: 50,280 tokens. Hidden dimension: 4,096.

The fully open nature includes: complete training code (PyTorch), training data composition and processing scripts, all intermediate checkpoints, and detailed training logs. This transparency enables the research community to reproduce and build upon OLMo results.

OLMo 3 7B (7B dense)

Architecture: Updated OLMo with GQA adoption

OLMo 3 transitions to **GQA** (32 heads / 8 KV heads), adopting the industry consensus that GQA's inference efficiency improvement justifies the minimal quality tradeoff. Maintains GELU activation and full open-source commitment. Extended training data and improved data quality filtering. Context length extended from 8K to 32K.

OLMo 3 32B (32B dense)

Architecture: Scaled OLMo with deeper architecture

Scales OLMo 3 to 32B parameters with 64 layers and hidden dimension 5,120. GQA with 40 heads / 8 KV heads. The largest fully open model (weights + code + data) available, providing a complete open-source alternative to proprietary 30B-class models. Demonstrates that full transparency is achievable at competitive scale.

CHAPTER 10

MiniMax Family

MiniMax-M2 (230B total (MoE))

Architecture: Lightning Attention with linear complexity

MiniMax-M2's defining architectural innovation is **Lightning Attention**, a linear-complexity attention mechanism that achieves O(n) scaling in sequence length. Unlike standard softmax attention ($O(n^2)$), Lightning Attention decomposes the attention computation into cumulative sums that can be computed incrementally. 230B total parameters with MoE routing. This architecture is specifically designed for ultra-long context processing (500K+ tokens).

MiniMax-M2.5 (230B total (MoE))

Architecture: Enhanced M2 with improved routing

Builds on M2's Lightning Attention with improved expert routing and extended training. The 230B MoE architecture maintains linear attention complexity while increasing active parameters for improved quality. Targets competitive performance with DeepSeek V3 and Qwen3 MoE models.

CHAPTER 11

NVIDIA — Nemotron Family

Nemotron 3 Nano 30B-A3B (30B total, ~3B active (MoE))

Architecture: Ultra-efficient MoE for edge deployment

NVIDIA's Nemotron Nano targets edge and embedded deployment. 30B total parameters with only ~3B active per token (10:1 ratio). The MoE architecture uses a small number of large experts rather than many small experts, optimized for NVIDIA's TensorRT inference framework. GQA attention, SwiGLU FFN, and RoPE positional encoding follow industry standards. Specifically optimized for NVIDIA GPU architectures with custom kernel fusion.

Nemotron 3 Super 128B-A12B (128B total, ~12B active (MoE))

Architecture: High-performance MoE for enterprise

Scales Nemotron to 128B total with ~12B active per token. The architecture includes **FP8 quantization-aware training**, natively training in 8-bit floating point for deployment efficiency on NVIDIA H100/B200 hardware. Expert routing optimized for pipeline parallelism across multiple GPUs. The Super variant targets enterprise deployment where NVIDIA hardware is standard.

CHAPTER 12

xAI — Grok 3.5

Grok 3.5 (270B total (MoE))

Architecture: xAI's flagship with aggressive expert scaling

Grok 3.5, developed by Elon Musk's xAI, uses a 270B MoE architecture. While xAI has released limited architectural details, the model uses Transformer decoder blocks with MoE routing, GQA attention, and extended context support. Trained on xAI's Memphis supercluster with ~100K H100 GPUs.

Grok 3.5 targets competitive performance with GPT-4 and Claude across reasoning, math, and coding benchmarks. The model is notable for its training infrastructure scale and aggressive timeline rather than specific architectural innovations.

CHAPTER 13

Other Notable Architectures

Xiaomi MiMo-V2-Flash 309B (309B total (MoE))

Architecture: Xiaomi's reasoning-optimized MoE

Xiaomi's entry into large-scale LLMs. 309B MoE with a focus on reasoning tasks and chain-of-thought generation. Uses standard GQA + SwiGLU + RoPE foundation with MoE routing optimized for reasoning depth.

Arcee AI Trinity Large (400B total (MoE))

Architecture: Merged multi-model architecture

Arcee AI's Trinity uses a novel approach: model merging of multiple specialized models into a single MoE architecture. Rather than training a single 400B model from scratch, Trinity combines expertise from multiple smaller models through learned expert routing. This approach enables rapid capability assembly.

Step 3.5 Flash (196B total (MoE))

Architecture: StepFun's fast-inference MoE

StepFun's 196B MoE model optimized for fast inference. Architecture emphasizes low-latency token generation through aggressive expert sparsity and optimized routing. Designed for real-time applications.

SmolLM3 3B (3B dense)

Architecture: HuggingFace's efficient small model

SmolLM3 from HuggingFace uses a compact 3B dense decoder with GQA, SwiGLU, and RoPE. Designed to demonstrate that careful data curation and training can produce competitive small models without architectural novelty. Open weights and training recipes.

Tiny Aya 3.35B (3.35B dense)

Architecture: Cohere's massively multilingual compact model

Cohere's Tiny Aya targets **101+ languages** in a 3.35B parameter model. The architecture uses standard decoder blocks but with a vocabulary optimized for multilingual coverage across Latin, CJK, Indic, Arabic, Cyrillic, and other scripts. Demonstrates that broad multilingual capability is achievable at small scale.

Nanbeige 4.1 3B (3B dense)

Architecture: Chinese-optimized compact decoder

Nanbeige's 3B model uses a standard decoder with GQA, optimized for Chinese language understanding and generation. Vocabulary heavily weighted toward Chinese tokens with 100K+ vocabulary size.

Ling 2.5 (1T total (MoE))

Architecture: Inclusion AI's trillion-parameter MoE

Ling 2.5 joins Kimi K2 as one of only two trillion-parameter models in this compendium. Uses a massive MoE architecture with hundreds of experts. The model targets competitive multilingual performance with emphasis on Chinese and English. Architectural details are limited but the model demonstrates that multiple organizations can now train at the trillion-parameter scale.

Sarvam 30B / 105B (30B and 105B dense)

Architecture: India-focused multilingual models

Sarvam AI's models specifically target Indian languages (Hindi, Tamil, Telugu, Bengali, Marathi, and others) alongside English. The 30B and 105B dense architectures use standard decoder blocks with vocabularies optimized for Indic scripts. These models demonstrate the growing trend of regional language-specialized LLMs.

CHAPTER 14

Comparative Analysis

14.1 Parameter Count vs Active Parameters

Model	Total Params	Active/Token	Ratio	Type
Gemma 3 270M	270M	270M	1:1	Dense
Llama 3.2 1B	1.24B	1.24B	1:1	Dense
SmolLM3 3B	3B	3B	1:1	Dense
Qwen3 4B	4B	4B	1:1	Dense
Llama 3 8B / OLMo 2 7B	7–8B	7–8B	1:1	Dense
Mistral 3.1 Small	24B	24B	1:1	Dense
Qwen3 32B / OLMo 3 32B	32B	32B	1:1	Dense
Sarvam 105B	105B	105B	1:1	Dense
Nemotron Nano 30B-A3B	30B	~3B	10:1	MoE
Kimi Linear 48B-A3B	48B	~3B	16:1	MoE
Qwen3 Next 80B-A3B	80B	~3B	26:1	MoE
Nemotron Super 128B-A12B	128B	~12B	10:1	MoE
Step 3.5 Flash	196B	est ~20B	~10:1	MoE
MiniMax-M2 / M2.5	230B	est ~25B	~9:1	MoE
Qwen3 230B-A22B	235B	~22B	10:1	MoE
Grok 3.5	270B	est ~30B	~9:1	MoE
Xiaomi MiMo-V2-Flash	309B	est ~30B	~10:1	MoE
GLM-4.5 / GLM-4.7	355B	~45B	~8:1	MoE
Qwen3.5	397B	est ~40B	~10:1	MoE
Llama 4 Maverick / Arcee Trinity	400B	~17B	~23:1	MoE
DeepSeek V3 / V3.2	671B	~37B	18:1	MoE
Mistral 3 Large	673B	est ~40B	~17:1	MoE
GLM-5	744B	est ~50B	~15:1	MoE
Kimi K2 / K2 Thinking	1T	~32B	31:1	MoE
Ling 2.5	1T	est ~30B	~33:1	MoE

Table 2: Complete parameter count comparison across all 42 models, sorted by total parameters

14.2 Attention Mechanism Comparison

Three distinct attention paradigms emerge across the 42 models: **Grouped-Query Attention (GQA)** is the dominant choice, used by Llama, Qwen, Gemma, Mistral, and Nemotron families. GQA reduces KV cache by sharing key-value heads across multiple query heads, with ratios ranging from 2:1 (Gemma) to 16:1 (Qwen3 MoE). **Multi-Latent Attention (MLA)** is used by DeepSeek and Kimi K2, achieving even greater KV cache compression through low-rank latent projections. **Full MHA** remains used by OLMo 2 and MiniMax's Lightning Attention variant.

The trend toward more aggressive KV compression reflects the practical challenge of serving large models: KV cache memory scales linearly with context length and batch size, often becoming the bottleneck for deployment. MLA and aggressive GQA reduce this bottleneck by 90%+ while maintaining most of the quality.

14.3 Expert Routing Strategies

MoE models in this compendium use three routing strategies. **Top-K routing** (Qwen3, Kimi K2) selects a fixed number K of experts per token, with K ranging from 2 (Llama 4 Maverick) to 8 (DeepSeek V3, Qwen3 MoE). **Auxiliary-loss-free routing** (DeepSeek V3) avoids the quality degradation of load-balancing losses. **Sigmoid gating** (GLM-4.5) allows variable expert counts per token. The shared expert pattern (used by DeepSeek, Qwen3 MoE, Kimi K2) ensures baseline knowledge is always available regardless of routing decisions.

Deep Dive: Attention Mechanism Evolution

From Multi-Head Attention to Multi-Latent Attention

The evolution of attention mechanisms across these 42 models tells the story of a relentless pursuit of inference efficiency. The original Multi-Head Attention (MHA) from Vaswani et al. (2017) computes independent key, value, and query projections for each attention head. For a model with 32 heads and hidden dimension 4,096, this means storing 32 separate key vectors and 32 separate value vectors per token in the KV cache. For a 128K context window with batch size 8, the KV cache alone consumes approximately 32GB of GPU memory at FP16 precision.

Grouped-Query Attention (GQA), introduced by Ainslie et al. (2023) and adopted by Llama 3, Qwen3, Gemma 3, Mistral, and Nemotron, reduces this by sharing key-value heads across multiple query heads. With a 4:1 ratio (32 query heads, 8 KV heads), KV cache memory drops by 75%. The quality impact is minimal because the query heads learn to extract different information from the shared KV representations. Across the 42 models, GQA ratios range from 2:1 (Gemma 3 27B) to 16:1 (Qwen3 230B-A22B), reflecting different tradeoffs between quality and efficiency.

Multi-Latent Attention (MLA), pioneered by DeepSeek V3 and adopted by Kimi K2, takes a fundamentally different approach. Instead of sharing KV heads, MLA projects all key-value information into a single low-rank latent vector of approximately 512 dimensions. The full KV representation is then reconstructed from this latent during attention computation. This achieves approximately 93% KV cache reduction compared to standard MHA, enabling DeepSeek V3 to maintain a 128K context window with a fraction of the memory required by GQA-based models of similar size.

Lightning Attention, used by MiniMax-M2 and M2.5, takes yet another approach: replacing the softmax attention kernel entirely with a linear approximation. This reduces the computational complexity of attention from O(n squared) to O(n) in sequence length, fundamentally changing the scaling properties for very long contexts. The tradeoff is a quality reduction for tasks requiring precise long-range token-to-token dependencies, but for many practical applications, the efficiency gain outweighs this limitation.

The Kimi Linear model introduces a hybrid approach: linear attention for most layers combined with standard softmax attention at periodic intervals. This provides the efficiency benefits of linear attention for the majority of computation while maintaining the precision of softmax attention for critical long-range dependencies. This hybrid strategy may represent the future direction for models that need both efficiency and quality across diverse context lengths.

Sliding Window vs Global Attention

Gemma 3 and Mistral models introduce a distinction between local and global attention patterns. In sliding window attention, each token attends only to tokens within a fixed window (typically 4,096 or 32,768 tokens). This is computationally cheaper than full attention but cannot capture dependencies beyond the window size.

Gemma 3 uses a hybrid approach: alternating between sliding window layers and full global attention layers. This allows the model to efficiently process local context in most layers while periodically consolidating long-range information in global layers. The ratio of local to global layers can be tuned: Gemma 3 4B uses approximately a 5:1 ratio (5 sliding window layers for every 1 global layer).

Mistral 3.1 Small applies sliding window attention uniformly across all 56 layers with a 32,768-token window. This consistent approach simplifies the architecture and enables efficient KV cache management, as tokens beyond the window can be evicted. The deep architecture (56 layers) compensates for the limited attention span by providing more layers for information to propagate through.

The choice between hybrid and uniform sliding window reflects a fundamental architectural question: is it better to have a few layers that see everything or many layers that each see only nearby tokens? The Gemma approach favors the former, while the Mistral approach favors the latter. Both achieve competitive results, suggesting that the right choice depends on the specific deployment constraints and target task distribution.

Deep Dive: Mixture-of-Experts Architectures

Expert Count and Granularity

The MoE models in this compendium span a wide range of expert counts: from Llama 4 Maverick with 128 experts to Kimi K2 with 384 experts, while DeepSeek V3 uses 256. The expert count determines the granularity of specialization: more experts enable finer-grained specialization but increase routing complexity and memory requirements.

DeepSeek V3's choice of 256 experts with top-8 routing means each token accesses 8 out of 256 specialized networks, a 3.1% activation rate. With the additional shared expert, total activation reaches approximately 3.5% of all expert parameters. This extreme sparsity is what enables 671B total parameters with only ~37B active per token.

Kimi K2 pushes further to 384 experts, the most in any model in this compendium. Combined with top-8 routing, each token activates only 2.1% of routed experts. The hypothesis is that more, smaller experts can capture more nuanced specializations, similar to how a larger team of narrow specialists can collectively cover more knowledge than a smaller team of generalists.

In contrast, Llama 4 Maverick uses only 128 experts with top-2 routing, activating just 1.6% of experts per token. Meta's approach favors fewer but larger experts, reasoning that each expert should be substantial enough to capture meaningful specializations rather than fragmenting knowledge across too many tiny networks.

The shared expert pattern, used by DeepSeek, Qwen3 MoE, and Kimi K2, adds one expert that processes every token regardless of routing decisions. This shared expert captures universal knowledge (common grammar, basic reasoning) that all tokens need, while routed experts specialize in domain-specific knowledge. This hybrid approach ensures consistent baseline quality while enabling specialization.

Load Balancing and Expert Utilization

A fundamental challenge in MoE training is load balancing: ensuring that all experts receive roughly equal numbers of tokens during training. Without balancing, popular experts receive more training signal and become increasingly dominant, while underutilized experts atrophy in a reinforcing cycle known as expert collapse.

The traditional solution adds an auxiliary loss term to the training objective that penalizes imbalanced expert utilization. This loss term directly pushes the router to distribute tokens more evenly. However, the auxiliary loss conflicts with the primary language modeling objective: sometimes the optimal routing is inherently imbalanced (certain experts should see more tokens because they are better at certain tasks). The auxiliary loss trades some model quality for training stability.

DeepSeek V3's breakthrough was eliminating the auxiliary loss entirely. Instead, DeepSeek introduced a bias-based routing mechanism where each expert maintains a learned bias term that adjusts its selection probability. During training, these biases are updated based on utilization statistics: under-utilized experts receive

higher biases (increasing their selection probability) while over-utilized experts receive lower biases. This achieves load balancing without modifying the training objective, preserving model quality.

GLM-4.5 and GLM-4.7 take a different approach with sigmoid gating instead of softmax. In softmax routing, expert selection probabilities sum to 1, creating a competitive dynamic where activating one expert reduces the probability of others. Sigmoid gating treats each expert independently: the router outputs a probability for each expert, and all experts with probability above a threshold are activated. This allows variable numbers of experts per token, which can be more efficient for tokens of varying complexity.

Deep Dive: Training Infrastructure and Scale

Computational Requirements Across Model Scales

The 42 models in this compendium represent a vast range of computational requirements. Training Gemma 3 270M requires approximately 10 to the power of 19 floating point operations (FLOPs), achievable on a single modern GPU in a few days. Training Kimi K2 at 1 trillion parameters requires approximately 10 to the power of 25 FLOPs, necessitating clusters of tens of thousands of GPUs running for months.

DeepSeek V3 was famously trained on approximately 2,048 NVIDIA H800 GPUs for about two months, with an estimated total training cost of approximately $5.6 million. This was considered remarkably efficient for a 671B parameter model, achieved through optimized MoE training that reduces per-token compute. By comparison, GPT-4's training is estimated to have cost over $100 million on a much larger cluster.

Kimi K2's trillion-parameter training required innovations in both hardware utilization and optimizer design. The MuonClip optimizer, a modification of the Muon optimizer with gradient clipping, was specifically designed for stable training at this unprecedented scale. Standard Adam optimizer struggles with the scale due to memory requirements for maintaining first and second moment estimates for one trillion parameters.

The trend is clear: MoE architectures enable larger total parameter counts at lower training cost per parameter, but the infrastructure requirements still scale significantly. The gap between organizations that can train 3B models (essentially anyone with a single GPU) and organizations that can train 1T models (requiring dedicated supercomputer clusters) continues to widen in absolute terms even as techniques like MoE reduce the relative cost.

Tokenizer Design Across Families

Tokenizer vocabulary size varies dramatically across the 42 models, from 32,768 tokens (Mistral) to 262,144 tokens (Gemma 3). This variation reflects different design philosophies about the tradeoff between vocabulary size, tokenization efficiency, and model capacity allocation.

Smaller vocabularies (32K to 50K) use more tokens to represent the same text, increasing sequence length and computational cost per document. However, each token embedding receives more training signal because the same tokens appear more frequently. Larger vocabularies (128K to 262K) represent text more efficiently with fewer tokens but require larger embedding matrices and may have undertrained representations for rare tokens.

Qwen3's 152K vocabulary is specifically optimized for multilingual coverage, dedicating substantial vocabulary capacity to CJK characters, code tokens, and mathematical symbols. This enables efficient tokenization of Chinese text (which requires far more tokens per concept with English-centric tokenizers) and code (where common programming constructs can be single tokens).

Gemma 3's 262K vocabulary is the largest and represents an extreme position: maximum tokenization efficiency at the cost of a very large embedding matrix. The embedding layer for 262K tokens at 4,608 dimensions requires

approximately 1.2 billion parameters just for input/output embeddings, a significant fraction of the 27B total.

The trend across model generations is toward larger vocabularies, reflecting the industry's shift toward multilingual and multimodal models where diverse token types (text in many languages, code, math, special tokens for tool use) all need efficient representation.

Deep Dive: Positional Encoding and Context Length

RoPE and Its Extensions

Rotary Position Embedding (RoPE), introduced by Su et al. (2021), has become the universal standard for positional encoding across the 42 models in this compendium. RoPE encodes position information by rotating query and key vectors in the complex plane, with rotation angles determined by position and dimension. This approach has several desirable properties: it provides relative position encoding (attention scores depend on the distance between tokens, not their absolute positions), it decays naturally with distance, and it is compatible with efficient attention implementations.

The base RoPE implementation supports a fixed maximum context length determined by the rotation frequency base parameter (typically 10,000). To extend context length beyond the training length, several extensions have been developed. YaRN (Yet Another RoPE eNtension), used by Qwen3 and Kimi models, modifies the rotation frequencies at inference time to support longer contexts without retraining. NTK-aware scaling, used by several models, adjusts the frequency base parameter proportionally to the desired context extension ratio.

The practical impact of these extensions is significant: models trained with 8K or 32K context lengths can be extended to 128K or even 1M tokens at inference time with moderate quality degradation. Llama 4 Maverick claims 1M token context support, achieved through progressive training with increasing context lengths combined with RoPE frequency scaling.

An emerging alternative is ALiBi (Attention with Linear Biases), which adds a position-dependent bias to attention scores rather than rotating embeddings. While no model in this compendium uses ALiBi as its primary positional encoding, the technique has influenced the design of attention bias mechanisms in several MoE routing systems.

Deep Dive: Normalization and Training Stability

RMSNorm, DeepNorm, and Training at Scale

Layer normalization is critical for stable training of deep Transformer models. The 42 models use three main approaches: RMSNorm (Root Mean Square Normalization) is the dominant choice, used by Llama, Qwen, Gemma, Mistral, DeepSeek, Kimi, and most others. RMSNorm simplifies LayerNorm by removing the mean-centering step, computing only the root mean square for scaling. This reduces computational cost by approximately 15% per normalization operation without measurable quality impact.

All modern models use Pre-Norm configuration, where normalization is applied before each sub-layer (attention and FFN) rather than after. Pre-Norm provides more stable gradient flow during training, enabling deeper models. The Llama architecture established Pre-RMSNorm as the standard configuration that has been adopted by virtually every subsequent model family.

GLM-5 introduces DeepNorm for its 744B MoE architecture. DeepNorm modifies the residual connection formula to account for network depth. In standard residual connections, the output is x + f(x) where f is the sub-layer function. DeepNorm changes this to x + alpha times f(x), where alpha is computed from the total number of layers. For very deep networks, alpha is less than 1, dampening the contribution of each layer to prevent gradient explosion. This enables GLM-5 to train deeper MoE networks than would be stable with standard residual connections.

The choice of normalization becomes increasingly important as models scale. At the 670B to 1T parameter range, training instability can waste millions of dollars in compute. DeepSeek V3's stable training at 671B parameters was achieved partly through careful normalization design, while Kimi K2's trillion-parameter training required the novel MuonClip optimizer specifically designed for extreme-scale stability.

Deployment Considerations and Practical Analysis

Memory Requirements and Hardware Mapping

Deploying the 42 models in this compendium requires vastly different hardware configurations. Dense models up to 8B parameters (Llama 3 8B, Qwen3 8B, OLMo 2 7B) can be served on a single NVIDIA A100 80GB or H100 80GB GPU at FP16 precision, or on consumer GPUs (RTX 4090 24GB) with 4-bit quantization. These models represent the practical ceiling for single-GPU deployment.

Dense models in the 24B to 32B range (Mistral 3.1 Small, Qwen3 32B, Gemma 3 27B) require either a single 80GB GPU at 4-bit quantization or 2 to 4 GPUs at higher precision. These models offer the best quality-per-dollar for organizations with limited GPU budgets.

MoE models present a unique deployment challenge: while only a fraction of parameters are active per token, all parameters must be loaded into memory. DeepSeek V3 at 671B parameters requires approximately 1.3TB of memory at FP16, necessitating 16 or more H100 GPUs. However, the active computation per token is equivalent to a approximately 37B dense model, meaning throughput is much higher than the total parameter count suggests.

The trillion-parameter models (Kimi K2, Ling 2.5) push the boundaries of current hardware. At FP16, 1T parameters requires approximately 2TB of memory, distributed across 24 or more 80GB GPUs. With FP8 quantization (supported natively by Nemotron models), memory requirements halve, making deployment on 12 to 16 GPUs feasible. The active compute of approximately 32B parameters per token means inference latency is comparable to serving a 32B dense model, despite the massive total parameter count.

Quantization Impact Across Architectures

Quantization reduces model precision from FP16 (16-bit floating point) to lower bit widths, trading quality for reduced memory and faster inference. The impact of quantization varies significantly across the 42 model architectures. Dense models generally tolerate quantization well: Llama 3 8B at 4-bit (GPTQ or AWQ) retains approximately 97% of FP16 quality while requiring only 25% of the memory. Gemma 3 27B similarly quantizes effectively due to its relatively uniform weight distributions.

MoE models present more complex quantization challenges. The expert weights are accessed sparsely, meaning each expert's weights are used relatively infrequently, potentially leading to higher quantization error for rarely-activated experts. DeepSeek V3 addresses this with mixed-precision quantization: shared experts and attention weights are kept at higher precision while routed experts use lower precision. NVIDIA's Nemotron models are trained with FP8 quantization awareness from the start, producing weights that are inherently more robust to low-precision representation.

For practical deployment, we recommend: FP16 or BF16 for quality-critical applications with sufficient hardware, INT8 for most production deployments balancing quality and efficiency, and INT4 or GPTQ for resource-constrained environments or applications where inference speed is the primary concern. The specific quality impact should be evaluated per model and per task, as quantization sensitivity varies across architectures.

Architectural Innovation Analysis: What Makes Each Family Unique

Llama: The Industry Standard-Setter

Meta's Llama family has arguably had more influence on the broader LLM ecosystem than any other open-weight model series. The architectural choices made in Llama 2 and refined in Llama 3 have become de facto standards: Pre-RMSNorm, SwiGLU activation, Grouped-Query Attention, and RoPE positional encoding. When a new model is announced and described as using 'standard Transformer decoder architecture,' it almost always means 'Llama-style architecture.' This standardization has been enormously beneficial for the ecosystem, enabling shared tooling, optimized inference frameworks, and direct performance comparisons.

The transition from Llama 3 (dense, up to 70B) to Llama 4 Maverick (MoE, 400B) represents Meta's strategic response to the scaling challenge. Dense models hit a practical ceiling around 70B to 100B parameters for most deployment scenarios; beyond this, inference costs become prohibitive for real-time applications. MoE enables Meta to scale model capacity to 400B while keeping inference costs at the 17B dense-equivalent level. The choice of 128 experts with top-2 routing is more conservative than DeepSeek's 256 experts with top-8, reflecting Meta's preference for simpler, more robust architectures over maximum parameter efficiency.

Llama 3.2 1B deserves special attention as a masterclass in efficient small model design. By using tied embeddings (sharing weights between the input embedding layer and the output language model head), the 1B variant eliminates the redundant parameter storage that plagues small models where the embedding matrix constitutes a disproportionate fraction of total parameters. This technique, combined with knowledge distillation from larger Llama variants, produces a model that significantly outperforms other 1B-class models despite its simple architecture.

DeepSeek: Architectural Innovation Leader

If Llama established the standard, DeepSeek has been the most prolific innovator, introducing architectural techniques that have been widely adopted across the industry. Multi-Latent Attention (MLA) represents the single most impactful attention mechanism innovation since Grouped-Query Attention. By projecting the entire key-value representation into a shared low-rank latent space of approximately 512 dimensions, MLA achieves KV cache compression that far exceeds what GQA can provide, even at aggressive ratios.

The mathematics of MLA are elegant: instead of storing separate key and value matrices of dimension d_model per head, MLA stores a single latent vector c of dimension d_c (where d_c is much less than d_model times num_heads). During attention computation, the full key and value representations are reconstructed from c using learned up-projection matrices. The compression ratio is approximately d_c divided by (2 times d_model times num_kv_heads), which for DeepSeek V3's configuration yields approximately 93% reduction.

The auxiliary-loss-free load balancing innovation has had equally broad impact. Prior to DeepSeek V3, every MoE model used an auxiliary loss term to balance expert utilization, and every practitioner accepted the associated quality degradation as a necessary cost. DeepSeek demonstrated that this tradeoff is not necessary: a simple bias-based routing mechanism achieves equivalent balancing without any impact on the primary training

objective. This technique has been adopted by multiple subsequent MoE models, including several in this compendium.

DeepSeek V3.2 refines the V3 architecture with improved routing algorithms and extended training, but maintains the same fundamental MLA plus 256-expert MoE design. The consistency between V3 and V3.2 suggests that DeepSeek considers the core architecture to be mature, with further improvements coming primarily from training data quality and scale rather than architectural changes.

Qwen: The Breadth Champion

Alibaba's Qwen family is remarkable for its breadth: nine variants covering dense models from 4B to 32B, MoE models from 80B to 397B, and specialized models for coding. This breadth reflects Alibaba's strategy of offering a complete model lineup that addresses every deployment scenario, from edge devices (Qwen3 4B) to frontier-class inference (Qwen3.5 at 397B).

The Qwen3 generation introduced the dual-mode thinking mechanism, which is an inference-time rather than architectural innovation but has significant implications for how the models are used. In thinking mode, the model generates an internal chain-of-thought reasoning trace enclosed in special tokens before producing the final answer. This reasoning trace is generated by the same model weights without any architectural changes; the difference is purely in the prompting and sampling strategy. The thinking mode produces substantially better results on reasoning-heavy tasks at the cost of higher token generation.

Qwen3's 152,064-token vocabulary deserves technical discussion. This is 4.7 times larger than Llama's 32K vocabulary, providing dramatically better tokenization efficiency for non-English languages, particularly Chinese, Japanese, Korean, Arabic, and code. A Chinese sentence that might require 30 tokens with Llama's tokenizer might need only 8 to 12 tokens with Qwen's tokenizer. This efficiency translates directly to effective context length: Qwen3 can process approximately 3 to 4 times more Chinese text within the same context window as Llama.

The Qwen3 Next 80B-A3B model pushes MoE sparsity to an extreme: 80B total parameters with only 3B active per token, a ratio of approximately 26:1. This is the most sparse activation ratio among all MoE models in this compendium except Kimi K2 (31:1). The practical implication is inference speed comparable to a 3B dense model while accessing the knowledge capacity of an 80B model, though the 80B of weights must still be loaded into memory.

Kimi K2: Pushing the Trillion-Parameter Frontier

Kimi K2 from Moonshot AI represents the current frontier of model scale at one trillion total parameters. Reaching this scale required innovations across multiple dimensions: the MuonClip optimizer for stable training, 384 routed experts plus 1 shared expert for fine-grained specialization, Multi-Head Latent Attention for efficient KV caching, and massive distributed training infrastructure.

The MuonClip optimizer modifies the recently proposed Muon optimizer by adding gradient clipping specifically calibrated for the statistics of trillion-parameter MoE training. Standard Adam optimizer becomes impractical at this scale because maintaining first and second moment estimates for one trillion parameters requires approximately 12TB of optimizer state memory in FP32 precision. Muon reduces this by using a lower-memory

optimizer formulation, and MuonClip adds stability through gradient norm clipping that prevents the catastrophic training instabilities that plague very large models.

The choice of 384 experts is the highest in this compendium and represents a bet that fine-grained expertise leads to better knowledge organization. With top-8 routing, each token accesses 8 out of 384 experts, a 2.1% activation rate. This creates 384 specialized knowledge domains that the model can draw upon, compared to DeepSeek V3's 256 domains or Llama 4 Maverick's 128. The hypothesis is that more, smaller experts provide better coverage of the knowledge space, analogous to how a library with more narrow subject categories can organize information more precisely than one with fewer broad categories.

Kimi Linear 48B-A3B is the most architecturally distinctive model in the Kimi family, replacing standard softmax attention with linear attention. Linear attention approximates the softmax attention matrix by decomposing it into the product of two lower-rank matrices, reducing computational complexity from quadratic to linear in sequence length. This makes the model particularly suited for extremely long context processing where the quadratic cost of standard attention becomes the dominant computational bottleneck. The tradeoff is reduced precision in token-to-token attention patterns, particularly for tasks requiring exact long-range reference resolution.

Google Gemma: Hybrid Attention Pioneer

Google's Gemma 3 family introduces the most distinctive attention pattern in this compendium: alternating sliding window and full global attention layers. This hybrid approach is a pragmatic solution to the fundamental attention efficiency challenge. Pure sliding window attention (like Mistral) is efficient but cannot capture dependencies beyond the window size. Pure global attention is comprehensive but expensive. Gemma's hybrid provides both efficiency and comprehensiveness by using cheap sliding window layers for most computation and expensive global layers for periodic long-range information consolidation.

The logit soft-capping technique is another Gemma innovation worth noting. By applying a tanh-based cap to both attention logits and final output logits, Gemma prevents the extreme logit values that can destabilize training, particularly in the early stages. The cap is set high enough (typically 30 to 50) that it does not affect normal operation but prevents the catastrophic outlier values that occasionally trigger training instabilities in very deep models.

Gemma 3 270M at just 270 million parameters is a fascinating architectural exercise in minimalism. With only 18 layers and 1,536 hidden dimension, it demonstrates that the Gemma architectural innovations (hybrid attention, logit capping, large vocabulary) provide meaningful benefits even at tiny scale. The 262K vocabulary is especially notable at this scale: the embedding matrix alone contains approximately 400M parameters, which is larger than the rest of the model combined. This unusual parameter allocation reflects Google's hypothesis that a large, well-trained vocabulary is a particularly efficient way to encode knowledge.

The State of Open LLM Architecture: March 2026

Convergence and Divergence

Looking across all 42 models, a clear pattern of convergence on certain architectural choices and divergence on others emerges. The converged elements represent techniques that have proven universally beneficial: RMSNorm, Pre-Norm configuration, SwiGLU activation, RoPE positional encoding, and BPE tokenization. These components appear in virtually every model regardless of scale, origin, or target application. They have been thoroughly validated across thousands of training runs and represent the settled science of Transformer architecture.

The divergent elements represent areas of active innovation where no single approach has proven clearly superior: attention mechanism (GQA vs MLA vs Linear vs Hybrid), expert routing strategy (top-K vs sigmoid vs auxiliary-loss-free), expert granularity (128 vs 256 vs 384 experts), vocabulary size (32K to 262K), and normalization depth technique (standard residual vs DeepNorm). These divergences represent the frontier of architectural research, where different organizations are placing different bets on which approaches will prove most effective as models continue to scale.

A notable trend is the disappearance of purely dense models above 100B parameters. Every model above this threshold in our compendium uses Mixture-of-Experts, reflecting an industry consensus that MoE is the only economically viable path to very large language models. The question is no longer whether to use MoE at scale, but how to implement it most effectively: how many experts, what routing strategy, what activation ratio, and how to balance training stability with expert specialization.

Predictions for 2027

Based on the architectural trends visible across these 42 models, we offer several predictions for the next generation of LLMs. First, MLA or a similar low-rank KV compression technique will become standard, replacing GQA as the dominant attention efficiency mechanism. The compression gains are too substantial to ignore, and DeepSeek's success has validated the approach at scale.

Second, auxiliary-loss-free expert routing will become universal in MoE models. The quality penalty of auxiliary losses is well-documented, and DeepSeek's bias-based alternative has proven effective. We expect every major MoE model released in 2027 to use some form of loss-free load balancing.

Third, hybrid attention patterns combining linear and softmax attention will proliferate, driven by the demand for million-token context windows. Pure softmax attention cannot scale to million-token contexts at acceptable cost, but pure linear attention sacrifices too much quality. The hybrid approach pioneered by Kimi Linear and related to Gemma's sliding window hybrid will evolve into a standard two-tier attention architecture.

Fourth, model scale will continue to increase, with multiple organizations reaching or exceeding 2 trillion total parameters. However, active parameters per token will plateau around 30 to 50B, as this represents the practical sweet spot for inference latency in real-time applications. The growth will come from more experts with higher sparsity ratios.

Fifth, architecture-specific quantization and inference optimization will become a key differentiator. NVIDIA's approach with Nemotron, where FP8 quantization awareness is baked into training from the start, will be adopted more broadly as organizations recognize that deployment efficiency is as important as training quality.

CHAPTER 15

Architectural Trends & Conclusion

15.1 Key Trends Across 42 Models

- **MoE dominance at scale:** Every model above 100B parameters uses MoE. Dense architectures are confined to the sub-100B range, suggesting a consensus that MoE is the only economically viable path to very large models.
- **Aggressive KV compression:** The progression from MHA to GQA to MLA reflects an industry-wide focus on inference efficiency. Modern models compress KV cache by 4–16× compared to standard MHA, enabling longer contexts and larger batch sizes.
- **SwiGLU as standard FFN:** SwiGLU activation has effectively replaced GELU and ReLU in FFN layers across the majority of models, providing consistent training improvements.
- **RoPE universality:** Rotary Position Embedding is used by nearly every model, with YaRN and NTK-aware extensions enabling context length far beyond training length.
- **Vocabulary expansion:** Vocabularies have grown from 32K (Llama 3) to 152K (Qwen3) to 262K (Gemma 3), improving tokenization efficiency especially for multilingual and code content.
- **Trillion-parameter threshold:** Kimi K2 and Ling 2.5 demonstrate that trillion-parameter MoE training is now achievable by multiple organizations, with active parameters kept at ~30B for practical inference.
- **Regional specialization:** Models like Sarvam (Indian languages), Nanbeige (Chinese), and Tiny Aya (101 languages) show that the LLM ecosystem is diversifying beyond English-centric architectures.

15.2 Conclusion

This compendium documents a remarkable period of architectural innovation in large language models. The 42 models analyzed span four orders of magnitude in parameter count (270M to 1T) and represent contributions from organizations across the United States, China, France, India, Israel, and the open-source community. Despite this diversity, clear architectural convergences are evident: the Transformer decoder foundation, GQA or MLA attention, SwiGLU activation, RoPE positioning, and MoE routing at scale.

The key insight is that the architectural variations between models are **modular and composable**. Each innovation—MLA, auxiliary-loss-free routing, Lightning Attention, DeepNorm, hybrid sliding window attention—addresses a specific bottleneck (KV cache size, expert utilization, sequence length, training stability, local vs global attention) and can be combined with innovations from other models. The next generation of LLMs will likely synthesize the best ideas from across these 42 architectures into even more capable and efficient designs.

Appendix: Complete Model Specifications

Model	Params	Layers	Hidden	Heads	KV Heads	Vocab	Attention
Gemma 3 270M	270M	18	1,536	8	4	262K	GQA+hybrid
Llama 3.2 1B	1.24B	16	2,048	32	8	128K	GQA
SmolLM3 3B	3B	~32	~2,560	~32	~8	~50K	GQA
Tiny Aya 3.35B	3.35B	~32	~2,560	~32	~8	~100K	GQA
Nanbeige 4.1 3B	3B	~32	~2,560	~32	~8	~100K	GQA
Qwen3 4B	4B	32	2,560	32	8	152K	GQA
Gemma 3 4B	4B	26	2,560	16	8	262K	GQA+hybrid
OLMo 2 7B	7B	32	4,096	32	32	50K	Full MHA
OLMo 3 7B	7B	32	4,096	32	8	50K	GQA
Llama 3 8B	8.03B	32	4,096	32	8	128K	GQA
Qwen3 8B	8B	36	4,096	32	8	152K	GQA
Mistral 3.1 Small	24B	56	3,072	32	8	32K	GQA+SW
Gemma 3 27B	27B	46	4,608	32	16	262K	GQA+hybrid
Sarvam 30B	30B	~64	~5,120	~40	~8	~100K	GQA
Qwen3 32B	32B	64	5,120	40	8	152K	GQA
OLMo 3 32B	32B	64	5,120	40	8	50K	GQA
Sarvam 105B	105B	~80	~8,192	~64	~8	~100K	GQA

Table 3a: Dense model specifications

Model	Total	Active	Experts	Top-K	Attention
Nemotron Nano	30B	~3B	MoE	top-K	GQA
Kimi Linear	48B	~3B	MoE	top-K	Linear
Qwen3 Next	80B	~3B	MoE	top-K	GQA
Nemotron Super	128B	~12B	MoE	top-K	GQA
Step 3.5 Flash	196B	~20B	MoE	top-K	GQA
MiniMax-M2	230B	~25B	MoE	top-K	Lightning
MiniMax-M2.5	230B	~25B	MoE	top-K	Lightning
Qwen3 230B-A22B	235B	~22B	128+1 shared	top-8	GQA
Grok 3.5	270B	~30B	MoE	top-K	GQA
MiMo-V2-Flash	309B	~30B	MoE	top-K	GQA
GLM-4.5	355B	~45B	MoE	sigmoid	MQA
GLM-4.7	355B	~45B	MoE	sigmoid	MQA
Qwen3.5	397B	~40B	MoE	top-K	GQA

Model	Total	Active	Experts	Top-K	Attention
Llama 4 Maverick	400B	~17B	128	top-2	GQA
Arcee Trinity	400B	~30B	MoE (merged)	top-K	GQA
DeepSeek V3	671B	~37B	256+1 shared	top-8	MLA
DeepSeek V3.2	671B	~37B	256+1 shared	top-8	MLA
Mistral 3 Large	673B	~40B	MoE	top-K	GQA+SW
GLM-5	744B	~50B	MoE	sigmoid	MQA+DeepNorm
Kimi K2	1T	~32B	384+1 shared	top-8	MLA
Kimi K2 Think	1T	~32B	384+1 shared	top-8	MLA
Ling 2.5	1T	~30B	MoE	top-K	GQA

Table 3b: MoE model specifications

References

[1] Vaswani, A. et al. (2017). “Attention Is All You Need.” NeurIPS.

[2] Touvron, H. et al. (2024). “Llama 3: Open Foundation Models.” Meta AI.

[3] Yang, A. et al. (2025). “Qwen3 Technical Report.” Alibaba Cloud.

[4] DeepSeek-AI. (2025). “DeepSeek-V3 Technical Report.”

[5] Gemma Team. (2025). “Gemma 3 Technical Report.” Google DeepMind.

[6] Moonshot AI. (2025). “Kimi K2: Scaling Mixture-of-Experts to One Trillion Parameters.”

[7] Mistral AI. (2025). “Mistral 3 Model Card.”

[8] Zhipu AI. (2025). “GLM-4/5 Technical Report.”

[9] Groeneveld, D. et al. (2024). “OLMo: Accelerating the Science of Language Models.” AI2.

[10] MiniMax. (2025). “MiniMax-M2 with Lightning Attention.”

[11] NVIDIA. (2025). “Nemotron Technical Report.”

[12] xAI. (2025). “Grok 3 Model Card.”

[13] Shazeer, N. (2019). “Fast Transformer Decoding: One Write-Head is All You Need.”

[14] Su, J. et al. (2021). “RoFormer: Enhanced Transformer with Rotary Position Embedding.”

[15] Fedus, W. et al. (2022). “Switch Transformers: Scaling to Trillion Parameter Models.”

[16] Ainslie, J. et al. (2023). “GQA: Training Generalized Multi-Query Transformer Models.”

About

Romi Nur Ismanto / Jekardah.com Lab — Independent research on AI architectures, agentic systems, and emerging technology. Jakarta, Indonesia.

Contact: rominur@gmail.com • t.me/Jekardah_AI • jekardah.com