Modern Language Model Architectures

From Papers to Practice: What Actually Works

Amit Kumar

Lead AI/NLP Engineer
E42.ai

March 22, 2026

Table of Contents

19+ New Dense Models in 2024-2025

- Qwen 2.5, Gemma 3, Command R
- OLMo 2, SmolLM2, InternLM2
- Each with architectural variations

But they all converge to similar patterns...

Most Models are LLaMA-like

- Main differences: normalization, position embeddings, activation functions
- These explain only **10-30%** of performance variance
- The rest? Training data, scale, and compute

What We'll Cover: Part 1

Architecture Variations

- Normalization strategies
- Position embeddings evolution
- Activation functions and gating
- Attention mechanisms

What We'll Cover: Part 2

Hyperparameter Consensus

- FFN dimension ratios
- Model depth vs width
- Attention head configurations

What We'll Cover: Part 3

Training Stability

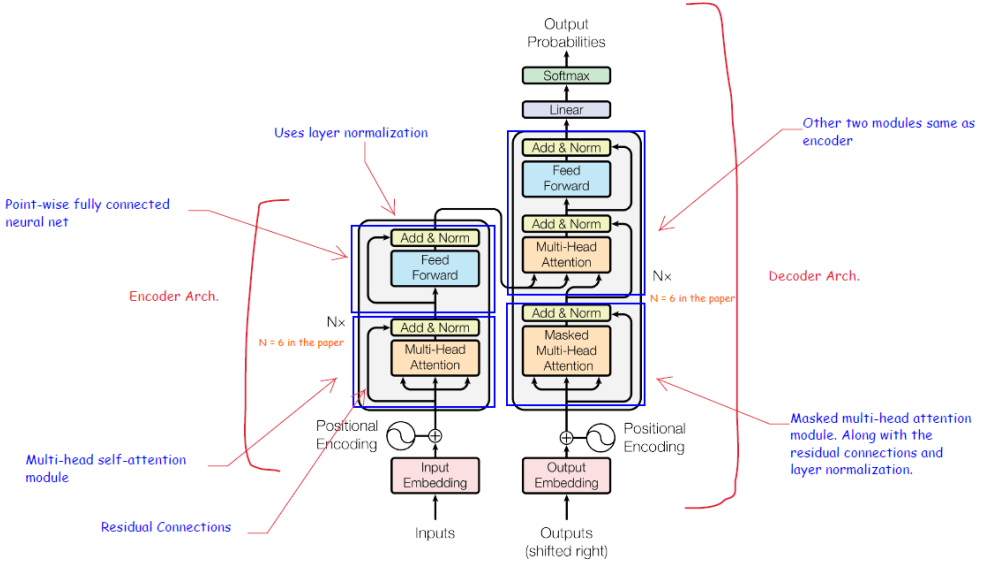
- Softmax overflow prevention
- QK-Normalization
- Z-Loss auxiliary term

What We'll Cover: Part 4

Efficiency Techniques

- Multi-Query Attention (MQA)
- Grouped-Query Attention (GQA)
- Sliding Window Attention

Attention Recap



Where should LayerNorm go?

Original (2017)

Post-norm architecture

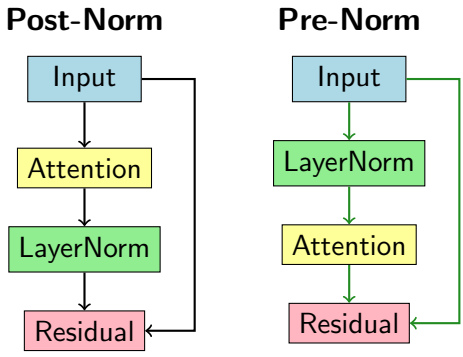
Issue: Gradient attenuation

Modern (2021+)

Pre-norm architecture

Better gradient flow

Pre-Norm vs Post-Norm



Why Pre-Norm Wins

Three Key Advantages:

1. Prevents gradient attenuation
2. No learning rate warmup needed
3. Stable training at 100B+ parameters

Pillar 2: RMSNorm

LayerNorm $\rightarrow$ RMSNorm

LayerNorm:

$$y = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \cdot \gamma + \beta$$

RMSNorm:

$$y = \frac{x}{\sqrt{\frac{1}{d}\sum_i x_i^2 + \epsilon}} \cdot \gamma$$

RMSNorm Advantages

	LayerNorm	**RMSNorm**
Operations	More	Fewer
Parameters	$2d$	d
Speed	Baseline	$+5\%$ faster

Used in: LLaMA, Qwen, Mistral, DeepSeek

Pillar 3: Gated Activations

Standard FFN $\rightarrow$ Gated FFN

Standard:

$$\text{FFN}(x) = \text{ReLU}(xW_1) \cdot W_2$$

SwiGLU:

$$\text{SwiGLU}(x) = (\text{Swish}(xW) \odot xV)\, W_2$$

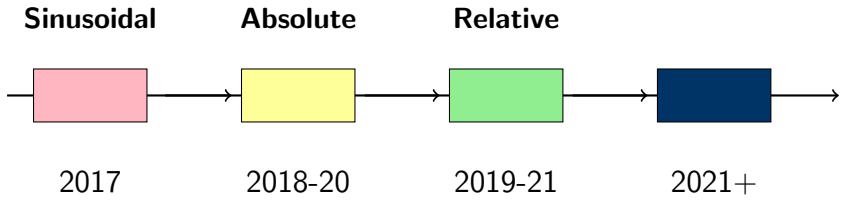
Why Gating Works

Key Benefits:

- Element-wise multiplication acts as a gate
- Per-token feature selection
- **2-5%** performance improvement

Used in: LLaMA, Mistral, PaLM, Gemma

Evolution Timeline



Rotary Position Embeddings

- Used in 90%+ of 2024+ models
- Truly relative positioning
- Scales to arbitrary context lengths

Core Principle

Inner products are invariant to rotation

- Rotate query/key by angle based on position
- Relative angle between vectors preserved
- Position becomes implicit in rotation

RoPE Mathematics

For each position m and dimension pair:

$$\begin{pmatrix} q'_{2i} \\ q'_{2i+1} \end{pmatrix} = \begin{pmatrix} \cos(m\theta_i) & -\sin(m\theta_i) \\ \sin(m\theta_i) & \cos(m\theta_i) \end{pmatrix} \begin{pmatrix} q_{2i} \\ q_{2i+1} \end{pmatrix}$$

where $\theta_i = 10000^{-2i/d}$

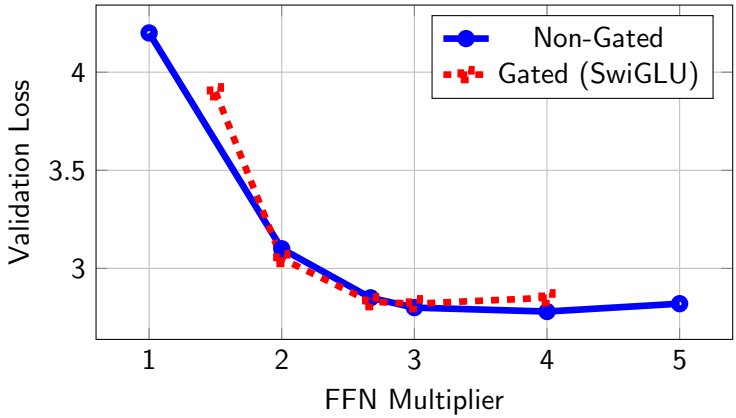
Attention between pos m and n depends only on $m - n$

How big should FFN be?

$$\boxed{d_{ffn} = 4 \times d_{model}} \quad \text{(non-gated)}$$

$$\boxed{d_{ffn} = 2.67 \times d_{model}} \quad \text{(gated)}$$

FFN Performance



Should models be deep or wide?

Sweet spot:

$$\frac{d_{model}}{n_{layers}} \approx 100 - 130$$

Model Aspect Ratios

Model	Ratio	Note
GPT-2	33	Too narrow
GPT-3	128	Optimal
LLaMA 2	102	Optimal
PaLM	156	Wide

Attention Heads

$$\boxed{d_{head} \times n_{heads} = d_{model}}$$

- Head dimension: typically 64-128
- Most models follow this strictly

Head Configuration Examples

Model	Heads	Head Dim
GPT-3	96	128
LLaMA 2 7B	32	128
LLaMA 2 70B	64	128
Mistral 7B	32	128

Training Collapse at 100B+

- Losses spike unexpectedly
- Gradient norms explode
- NaN values appear

Softmax Operations

Attention:

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)$$

Issue: Exponentials overflow with large logits

Solution 1: QK-Normalization

Normalize before softmax

$$Q' = \text{LayerNorm}(Q), \quad K' = \text{LayerNorm}(K)$$

$$\text{Attn} = \text{softmax}\left(\frac{Q'K'^T}{\sqrt{d}}\right)$$

QK-Norm Benefits

Key Advantages:

- Prevents logit overflow
- Stable attention weights
- No performance loss
- Scales to longer sequences

Solution 2: Z-Loss

Control partition function

$$Z = \sum_j e^{z_j}$$

Auxiliary loss:

$$L_{\text{total}} = L_{\text{CE}} + \alpha \cdot (\log Z)^2$$

Z-Loss Benefits

How it works:

- Penalizes large partition functions
- Keeps logits bounded
- Typically $\alpha = 0.001$ to 0.01

Used in: PaLM, OLMo 2, DCLM

The KV Cache Problem

Memory bottleneck during inference

$$\text{Memory} = \text{batch} \times \text{heads} \times \text{seq} \times d_{head}$$

Three Approaches

1. **Multi-Head Attention (MHA)**
   Each query head has dedicated KV

2. **Multi-Query Attention (MQA)**
   All queries share one KV head

3. **Grouped-Query Attention (GQA)**
   Groups of queries share KV heads

GQA: The Sweet Spot

Best of both worlds

- Example: 32 query : 4 KV ratio
- Minimal accuracy loss
- Major speedup

Used in: Llama 3, Mistral, Command R

Local attention for long sequences

Query at position i attends to $[i - W, i + W]$

$O(nW)$ instead of $O(n^2)$

When to Use Sliding Window

Context Length	**Strategy**
$< 4K$	Full attention
4K-100K	Hybrid approach
$> 100K$	Sliding window

What Works in 2024-2025

- Pre-Norm + RMSNorm
- SwiGLU/GeGLU activation
- RoPE position embeddings
- GQA or Full attention

Modern Architecture Components

Component	Choice
Normalization	Pre + RMSNorm
FFN	SwiGLU/GeGLU
FFN Size	2.67-4x
Position	RoPE
Stability	QK + Z-Loss
Attention	GQA/Full

Following this recipe:

- LLaMA 3.1 series
- Qwen 2.5 series
- DeepSeek models
- Phi-4
- OLMo 2, SmolLM2

Reading Papers: Checklist 1

Questions to ask:

1. Pre-norm or post-norm?
2. LayerNorm or RMSNorm?
3. QK-normalization included?

Questions to ask:

4. Position embedding type?
5. FFN size ratio?
6. Gated or standard activation?

Reading Papers: Checklist 3

Questions to ask:

7. Z-loss included?
8. Full, MQA, or GQA attention?
9. Sliding window for long context?

Architecture Convergence

- Most models are LLaMA-like
- The "big three": Pre-norm, RMSNorm, Gated activations
- Differences explain 10-30% variance

Hyperparameters are Consistent

- FFN: 2.67-4x model dimension
- Aspect ratio: 100
- Head config follows formula

Stability Over Cleverness

- QK-norm prevents overflow
- Z-loss controls growth
- Critical for 100B+ models

RoPE is Standard

- Truly relative positioning
- Scales to long context
- 90%+ of models use it

Efficiency Matters

- GQA reduces KV cache
- Sliding window for long sequences
- Minimal accuracy tradeoffs

Building an LLM in 2026

- Pre-Norm + RMSNorm
- RoPE embeddings
- SwiGLU FFN (2.67x)
- GQA attention
- QK-Norm + Z-Loss

= Stable, efficient, state-of-the-art LLM

Thank You!

Questions?