Mixture of Experts

Sparse Models · Gating · Load Balancing · Scaling

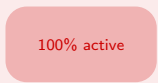
What is Mixture of Experts?

Core idea: divide a model into separate sub-networks (**experts**), each specializing in different aspects of the input.

A **router** decides which experts to activate for each input.

Dense model

Every input activates ALL parameters.



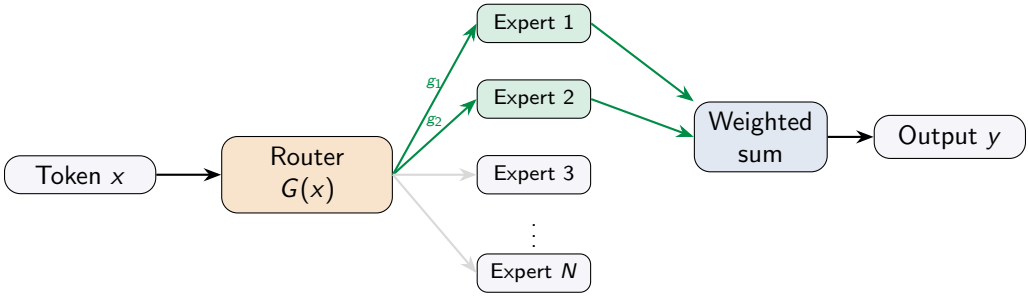
Sparse MoE model

Each input activates only a SUBSET of parameters.

Conditional computation: scale model capacity (total params) without proportionally increasing compute (FLOPs per token).

Jacobs, Jordan, Nowlan & Hinton, "Adaptive Mixtures of Local Experts," *Neural Computation*, 1991

MoE architecture



$$y = \sum_{i=1}^{N} G(x)_i \cdot E_i(x)$$

Most $G(x)_i = 0$ (sparse). Only top-k are nonzero.

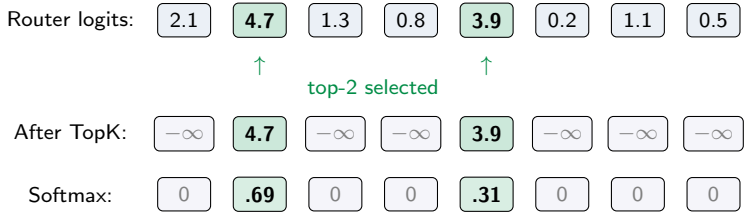
In a Transformer:
MoE replaces the FFN layer. Attention is shared.

The router (gating network)

$$G(x) = \text{Softmax}\big(\text{TopK}(x \cdot W_g, \quad k)\big)$$

$$W_g \in \mathbb{R}^{d_{\text{model}} \times N} \quad \text{(learned linear projection)}$$

How TopK works:



$$y = 0.69 \cdot E_2(x) + 0.31 \cdot E_5(x)$$

Top-1 (Switch Transformer): simplest, cheapest ·
Top-2 (Mixtral, GShard): more robust, most common

Noisy top-k gating

Problem: without noise, the router quickly converges to always selecting the same few experts $\Rightarrow$ most experts are wasted (**routing collapse**).

$$H(x)_i = (x \cdot W_g)_i + \mathcal{N}(0,1) \cdot \text{Softplus}\big((x \cdot W_{\text{noise}})_i\big)$$

$$G(x) = \text{Softmax}\big(\text{TopK}(H(x), \quad k)\big)$$

Add tunable Gaussian noise to logits **during training only** (disabled at inference)

Why it works:

Noise encourages exploration of different experts early on. Prevents premature lock-in. W_{noise} is learned — noise magnitude is input-dependent.

At inference:

Noise disabled for determinism. Router uses clean logits only. Selected experts are fixed given the input.

Shazeer et al., "Outrageously Large Neural Networks," ICLR 2017

Load balancing

Expert collapse: popular experts get more training → become better → get selected more → self-reinforcing

$$\mathcal{L}_{\text{balance}} = \alpha \cdot N \cdot \sum_{i=1}^{N} f_i \cdot P_i$$

f_i = fraction of tokens routed to expert i

P_i = mean router probability for expert i

Minimized when $f_i = P_i = 1/N$ for all experts (uniform distribution)

Capacity factor

Expert capacity = $\frac{\text{tokens}}{\text{experts}} \times C$

$C = 1.25$: each expert handles 25% more than fair share.

Overflow tokens are **dropped**.

Router z-loss

$$\mathcal{L}_z = \frac{1}{B} \sum_i \left[\log \sum_j e^{x_{ij}}\right]^2$$

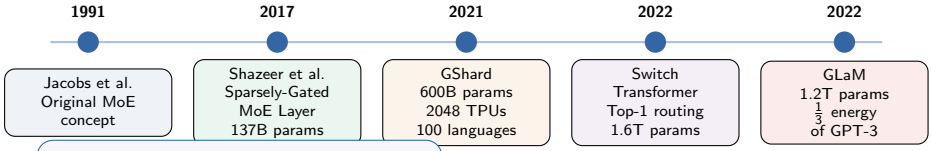
Penalizes large router logits.

Improves numerical stability.

Zoph et al., ST-MoE, 2022

Switch Transformer: $\alpha = 10^{-2}$ · Total loss: $\mathcal{L} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{balance}} + \mathcal{L}_z$

Landmark MoE papers



Switch Transformer

Simplified to **top-1** routing
(each token $\rightarrow$ 1 expert).
Halved communication cost.
Up to **7x pre-training speedup**
over T5 with same compute.

Fedus, Zoph & Shazeer, JMLR 2022

GLaM

1.2T total, 97B active (8%).
64 experts, top-2 routing.
Better than GPT-3 quality
at **1/3 the training energy**.

Du et al., ICML 2022

Each generation scaled MoE by $\sim$10x while simplifying the routing mechanism.

Modern MoE models

Model	Total	Active	Experts	Top-k	Notes
Mixtral 8x7B	46.7B	12.9B	8	2	Beats LLaMA2-70B
Mixtral 8x22B	176B	44B	8	2	Larger variant
DeepSeek-V2	236B	21B	160+2	6+2	Fine-grained MoE
DeepSeek-V3	671B	37B	256+1	8+1	\$5.5M training cost
DBRX	132B	36B	16	4	Fine-grained, 16-choose-4
Grok-1	314B	$\sim$79B	8	2	Open-source (xAI)
Arctic	480B	17B	128	2	Dense+MoE hybrid

As of 2025, nearly all frontier models use MoE: GPT-4 (rumored 16 experts), Gemini 1.5, DeepSeek-R1, Qwen3-235B. Dense-only models are the exception.

MoE vs. dense models

MoE wins when:

Pre-training speed matters
(4–7x speedup over dense)

Inference throughput matters
(2–3x higher than same-size dense)

Knowledge-intensive tasks
(broad factual knowledge)

Budget-constrained training
(GLaM: $\frac{1}{3}$ energy of GPT-3)

Dense wins when:

Small-data fine-tuning
(MoE overfits more easily)

Memory-constrained deployment
(all experts must be loaded)

Latency-critical serving
(memory bandwidth bottleneck)

Simple, well-defined tasks
(reasoning benchmarks)

The trade-off: Mixtral 8x7B has **46.7B total params** but only **12.9B active**.

Compute per token $\approx$ a 14B dense model.
Quality $\approx$ a 70B dense model.
Memory $\approx$ a 47B dense model (all experts loaded).

MoE decouples model capacity from compute cost — more knowledge per FLOP.

Gating mechanisms compared

Method	Who chooses?	Balance	Used by
Top-1	Token $\rightarrow$ 1 expert	Aux loss	Switch Transformer
Top-2	Token $\rightarrow$ 2 experts	Aux loss	Mixtral, GShard
Expert Choice	Expert $\rightarrow$ top-k tokens	Perfect	Google (2022)
Soft MoE	Weighted avg (all)	N/A	Puigcerver (2024)
Hash routing	Hash function	Random	Roller et al. (2021)

Expert Choice

Inverts routing: *experts* pick their top-k tokens.
Perfect balance by construction.

2x faster convergence.

Zhou et al., NeurIPS 2022

Soft MoE

No discrete routing at all.
Each expert gets a different soft combination of all tokens.

No collapse, no dropping.

Puigcerver et al., ICLR 2024

What do experts learn?

Do experts specialize in domains ("biology expert", "math expert")?
Mostly no. Experts tend to specialize at the **token/syntax level**.

What experts do learn:

Parts-of-speech categories
Syntactic patterns
Punctuation and formatting

What they don't learn:

"Psychology expert"
"Biology expert"
"Code expert"

Nuance (2025): some experts show lower activation for specific domains, and in-context demonstrations can activate a stable subset of "domain experts."

Functional roles: **domain experts** (universal knowledge) vs. **driver experts** (task-specific mediation)

ST-MoE (Zoph et al., 2022) conducted extensive analysis of expert specialization patterns.

Fine-grained MoE (DeepSeek)

Standard MoE

8 large experts, pick 2

Fine-grained MoE (DeepSeek)

shared many small routed experts, pick 6–8

Strategy 1: Segmentation

Split each large expert into m smaller experts. More combinations $\Rightarrow$ more flexible specialization.

Strategy 2: Shared experts

Designate K_s experts as always-active (shared). Capture common knowledge. Routed experts can specialize.

DeepSeek-V3: 671B total, 37B active, 256 routed + 1 shared expert. Training cost: **$5.5M** (fraction of comparable models). Rivals GPT-4o and Claude 3.5.

Training challenges

Communication

Two extra all-to-all ops per MoE layer (dispatch + aggregate). Primary bottleneck at scale.

Instability

Router logits grow unboundedly large. Softmax numerical issues. Fix: router z-loss, bfloat16 training.

Routing collapse

Without aux losses, router converges to same few experts. Self-reinforcing loop. Most experts die.

Expert parallelism: experts distributed across GPUs. Tokens must be shuffled to the correct GPU (**all-to-all**).

Combined with data parallelism (non-expert params) and tensor parallelism (within experts).

Frameworks: DeepSpeed-MoE, Tutel, FasterMoE (17x speedup with topology-aware gating)

Inference challenges

The memory problem: all experts must be loaded even though only a fraction is activated per token. Memory $\propto$ total params, compute $\propto$ active

Mixtral 8x7B

Compute: $\sim$14B dense
Memory: $\sim$47B dense
3.6x memory overhead
vs. equivalent compute

DeepSeek-V3

Compute: $\sim$37B dense
Memory: $\sim$671B dense
18x memory overhead
vs. equivalent compute

Solutions:

Expert parallelism
(distribute across GPUs)
Expert offloading
(CPU $\leftrightarrow$ GPU swap)
Quantization
(QMoE: 20x compression)

When FLOPs are the bottleneck:

MoE wins — fewer FLOPs per token
$\Rightarrow$ faster inference at scale
(high-throughput serving)

When bandwidth is the bottleneck:

Dense may win — MoE must load all expert weights
(single-user, low-batch scenarios)

Pre-gated MoE (ISCA 2024): predict next layer's experts to overlap data transfer with compute

MoE and scaling laws

Dense scaling (Chinchilla)

For optimal training:
model size and data
should scale equally.

$L(N, D) \propto N^{-\alpha} + D^{-\beta}$

Doubling model $\Rightarrow$
double compute AND data

MoE scaling

MoE decouples **capacity**
(total params) from **compute**
(active params per token).

Can scale capacity $\gg$ compute.

Same FLOPs, much more
knowledge stored.

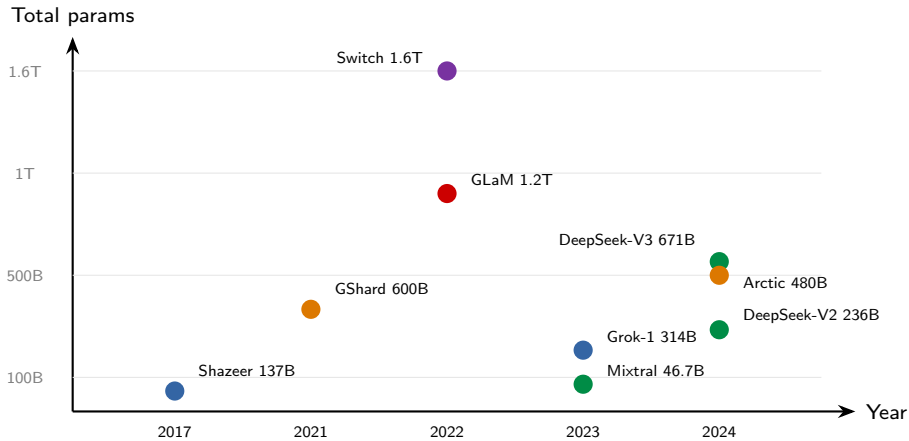
Key insight: given a fixed compute budget, MoE achieves **lower loss** than a dense model because it can store more knowledge in its extra parameters.

GLaM (1.2T MoE) $>$ GPT-3 quality at $\frac{1}{3}$ the energy.
Mixtral (46.7B MoE) $\approx$ LLaMA2-70B quality at $\frac{1}{5}$ active params.

2025 trend: from "few large experts" (8) to "many small experts" (64–256).

DeepSeek-V3: 256 experts · Qwen3: 128 experts · Arctic: 128 experts

The MoE landscape



Case study: DeepSeek-V3

DeepSeek-V3 (Dec 2024): 671B total, 37B active. 256 routed experts + 1 shared expert. Rivals GPT-4o and Claude 3.5 Sonnet.

Aux-loss-free balancing

Expert-wise bias on routing scores, updated dynamically. Avoids quality-vs-balance trade-off of aux losses.

Multi-head Latent Attention (MLA)

Compresses KV cache to low-dimensional latent space. Reduces inference memory dramatically.

Multi-Token Prediction (MTP)

Training objective: predict multiple future tokens simultaneously. Densifies training signal. Can speed up inference.

Training cost: 2.788M H800 GPU hours $\approx$ **$5.5M**

Compare: LLaMA3-405B estimated at $60–100M. GPT-4 estimated at $100M+.

MoE + engineering innovations $\Rightarrow$ 10–20x cost reduction.

DeepSeek-R1 (reasoning model) is built on DeepSeek-V3's MoE backbone + GRPO alignment.

Practical guide

When should I use MoE?

Large-scale pre-training?

⇒ **Yes, MoE**

4–7x training speedup.
Dominant architecture for frontier models.

High-throughput serving?

⇒ **Yes, MoE**

Lower cost per token.
Better quality per FLOP.

Memory-constrained?

⇒ **Dense model**

All experts must fit in memory.
Or use expert offloading.

Fine-tuning on small data?

⇒ **Dense or LoRA on MoE**

MoE overfits more.
ST-MoE: fine-tune only non-expert layers.

Many small experts or few large?

⇒ **Fine-grained (DeepSeek)**

More combinations = better.
Shared + routed pattern.

Want to explore MoE?

⇒ **Mixtral is the best start**

Open-source, well-documented, strong community support.

Questions?

All DL4NLP topics complete!