AIMS - 2025

Mixture-of-Experts (MoE)

Andrei A. Rusu

11/04/2025

Let's start at the beginning

We call an LLM **dense** if ...

essentially all of its parameters are "activated" and used during inference

E.g: GPT <=3, Llama <=3, Gemma, Qwen <=2.5, Mistral, Phi <=3

Goal

At the end of this lecture you will know …

how to scale up beyond dense models

TL;DR:

use Sparsely-Gated Mixtures-of-Experts (sparse MoEs)

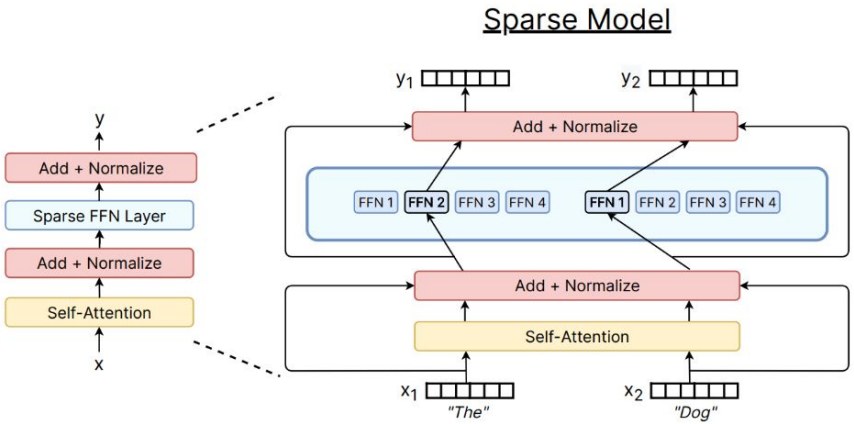


Figure: A Review of Sparse Expert Models in Deep Learning

Overview

Sparse MoEs:

- What (origins & modern)
- How (to train)
- Scaling laws
- Best practices

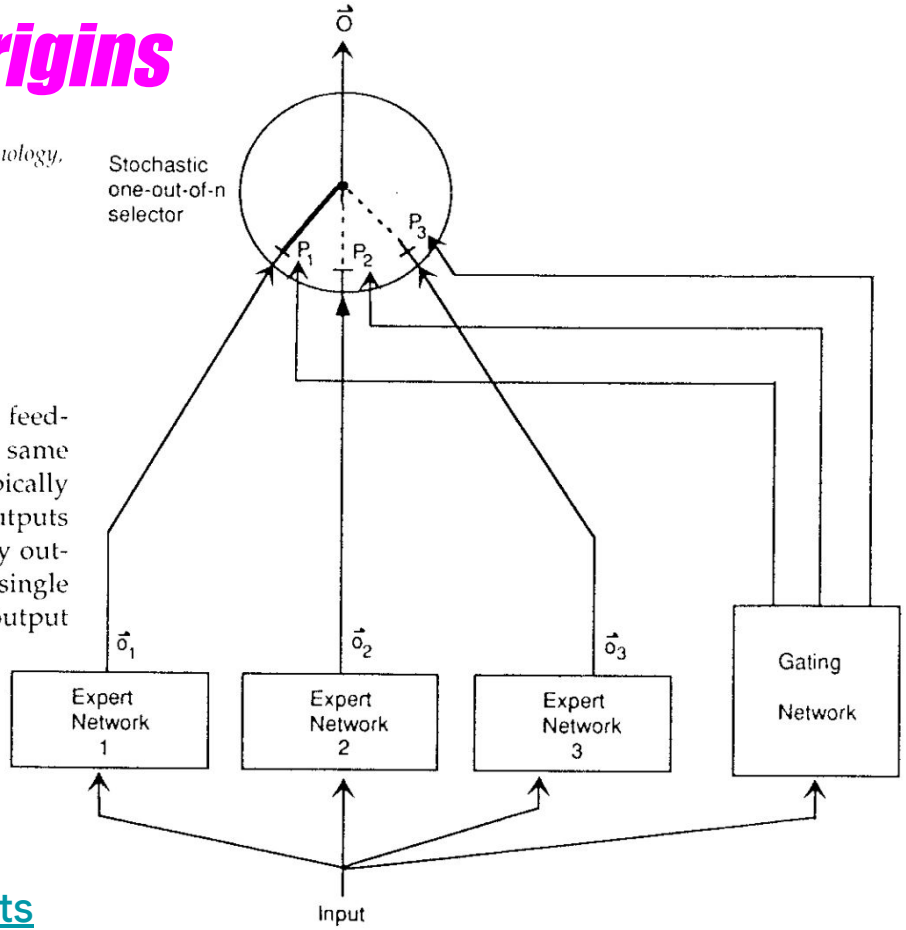
Origins

Adaptive Mixtures of Local Experts

Robert A. Jacobs
Michael I. Jordan
Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Steven J. Nowlan
Geoffrey E. Hinton
Department of Computer Science, University of Toronto, Toronto, Canada M5S 1A4

Figure 1: A system of expert and gating networks. Each expert is a feedforward network and all experts receive the same input and have the same number of outputs. The gating network is also feedforward, and typically receives the same input as the expert networks. It has normalized outputs $p_j = \exp(x_j)/\sum_i \exp(x_i)$, where x_j is the total weighted input received by output unit j of the gating network. The selector acts like a multiple input, single output stochastic switch; the probability that the switch will select the output from expert j is p_j.



From: [1991 - Adaptive Mixtures of Local Experts]

Origins

Hierarchical mixtures of experts and the EM algorithm

Michael I. Jordan
Department of Brain and Cognitive Sciences
MIT
Cambridge, MA 02139

Robert A. Jacobs
Department of Psychology
University of Rochester
Rochester, NY 14627

(1993)

We have presented a novel tree-structured architecture for supervised learning.

[...]

Learning is treated as a maximum likelihood problem; in particular, we present an Expectation-Maximization (EM) algorithm for adjusting the parameters of the architecture.

We also develop an on-line learning algorithm in which the parameters are updated incrementally. Comparative simulation results are presented in the **robot** dynamics domain.

From: Hierarchical Mixtures Of Experts And The EM Algorithm (1993)

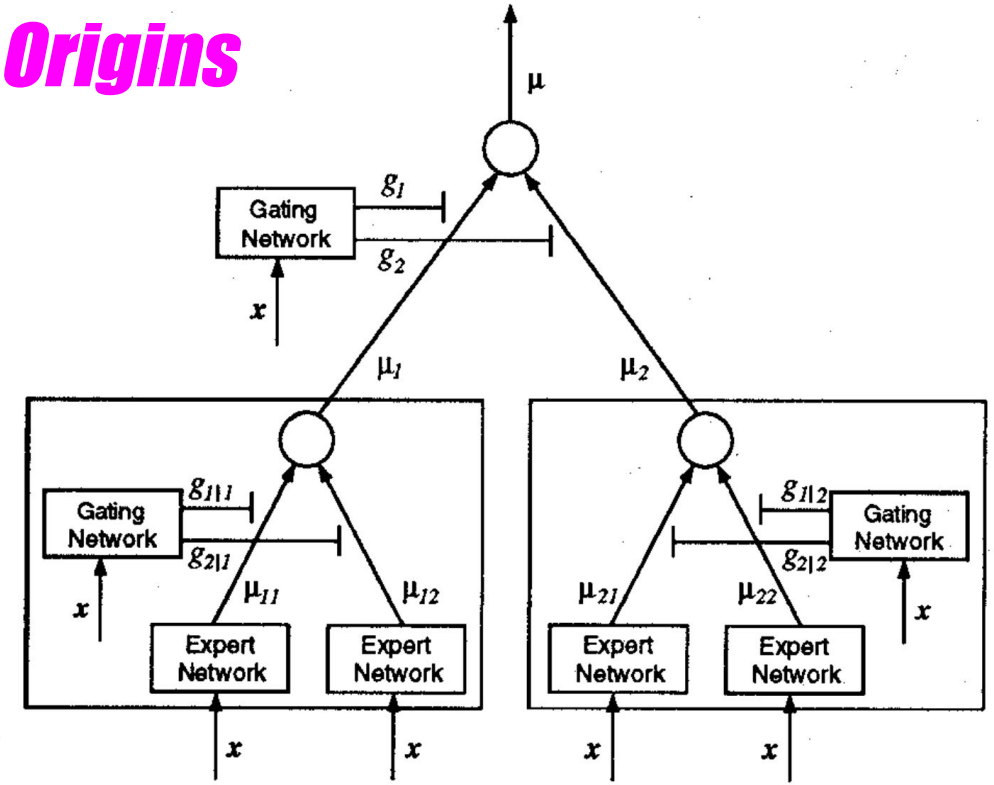


Figure 1: A two-level hierarchical mixture of experts.

Origins

Learning Factored Representations in a Deep Mixture of Experts

David Eigen [1,2] **Marc'Aurelio Ranzato** [1] * **Ilya Sutskever** [1]
[1] Google, Inc.
[2] Dept. of Computer Science, Courant Institute, NYU
deigen@cs.nyu.edu ranzato@fb.com ilyasu@google.com

(2014)

Such models show promise for building larger networks that are **still cheap to compute** at test time, and **more parallelizable** at training time.

In this this work, we extend the Mixture of Experts to a stacked model, the **Deep** Mixture of Experts, with multiple sets of gating and experts.

This **exponentially increases** the number of effective experts by associating each input with a combination of experts at each layer, yet *maintains a modest model size*.

From: Learning Factored Representations in a Deep Mixture of Experts

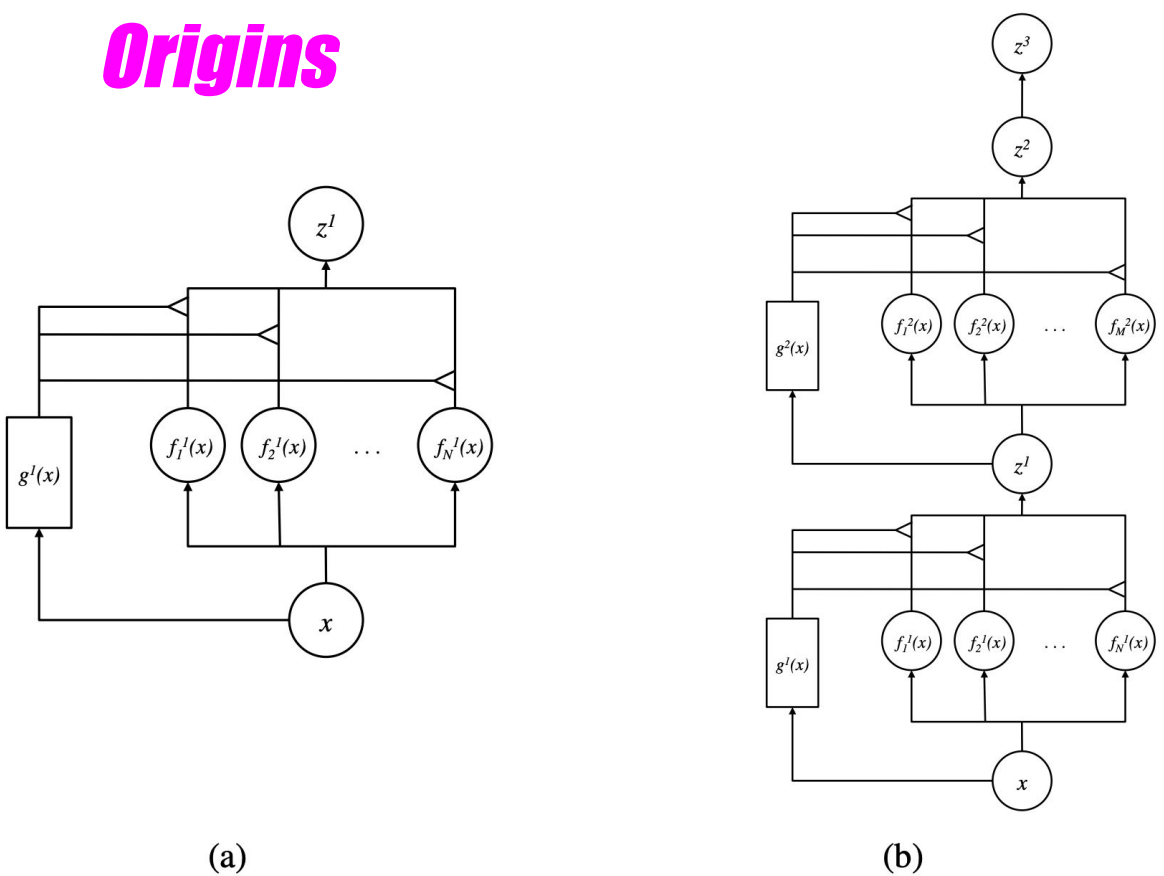


Figure 1: (a) Mixture of Experts; (b) Deep Mixture of Experts with two layers.

Reusing excellent slides from
Mixture of Experts CMU Course

A Modern View

A dense mixture of experts gives every expert a (non-zero) voice in the output

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

Dense MoE

- Dense Softmax

  $$G(\mathbf{x}) = \text{softmax}(\mathbf{x} \cdot \mathbf{W}_g)$$

- Each expert is just a feed-forward network

  $$E_i(\mathbf{x}) = \text{FFN}_{\text{ReLU}}(\mathbf{x}) = \text{ReLU}(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$

From: Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer (2017)

Origins

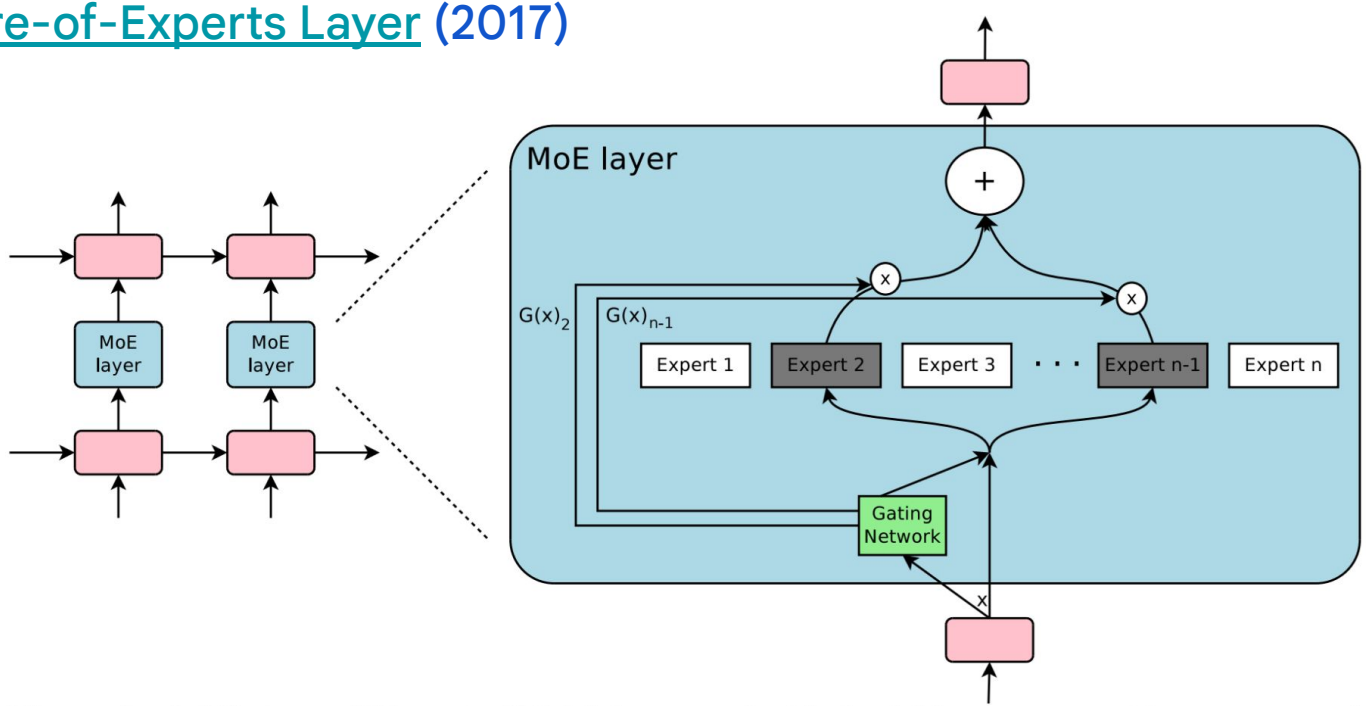


Figure 1: A Mixture of Experts (MoE) layer embedded within a recurrent language model. In this case, the sparse gating function selects two experts to perform computations. Their outputs are modulated by the outputs of the gating network.

Sparsely-Gated Mixtures-of-Experts

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

Sparse MoE

- Softmax over Top-K Gating

$$G(\mathbf{x}) = \text{softmax}(\text{topk}(\mathbf{x} \cdot \mathbf{W}_g + \mathbf{b}_g, k))$$

- Each expert is just a feed-forward network

$$E_i(\mathbf{x}) = \text{FFN}_{\text{ReLU}}(\mathbf{x}) = \text{ReLU}(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$

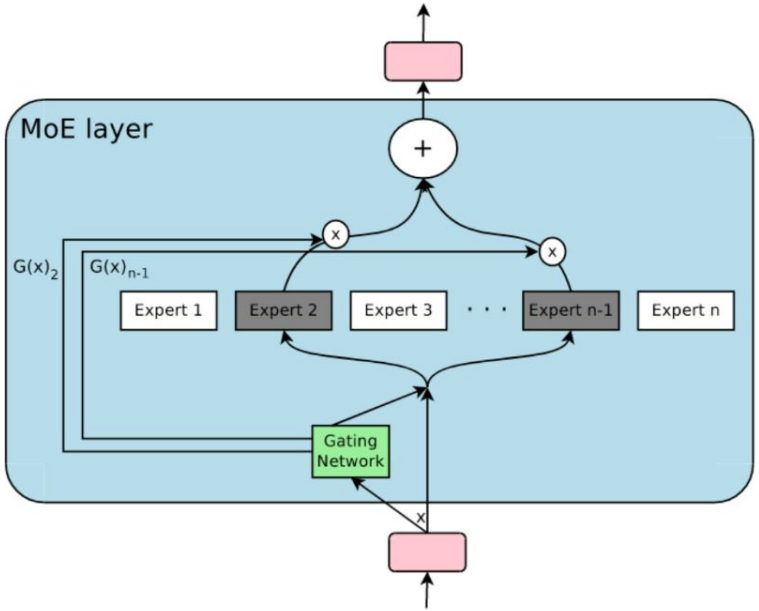


Figure: Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer (2017)

Sparsely-Gated Mixtures-of-Experts

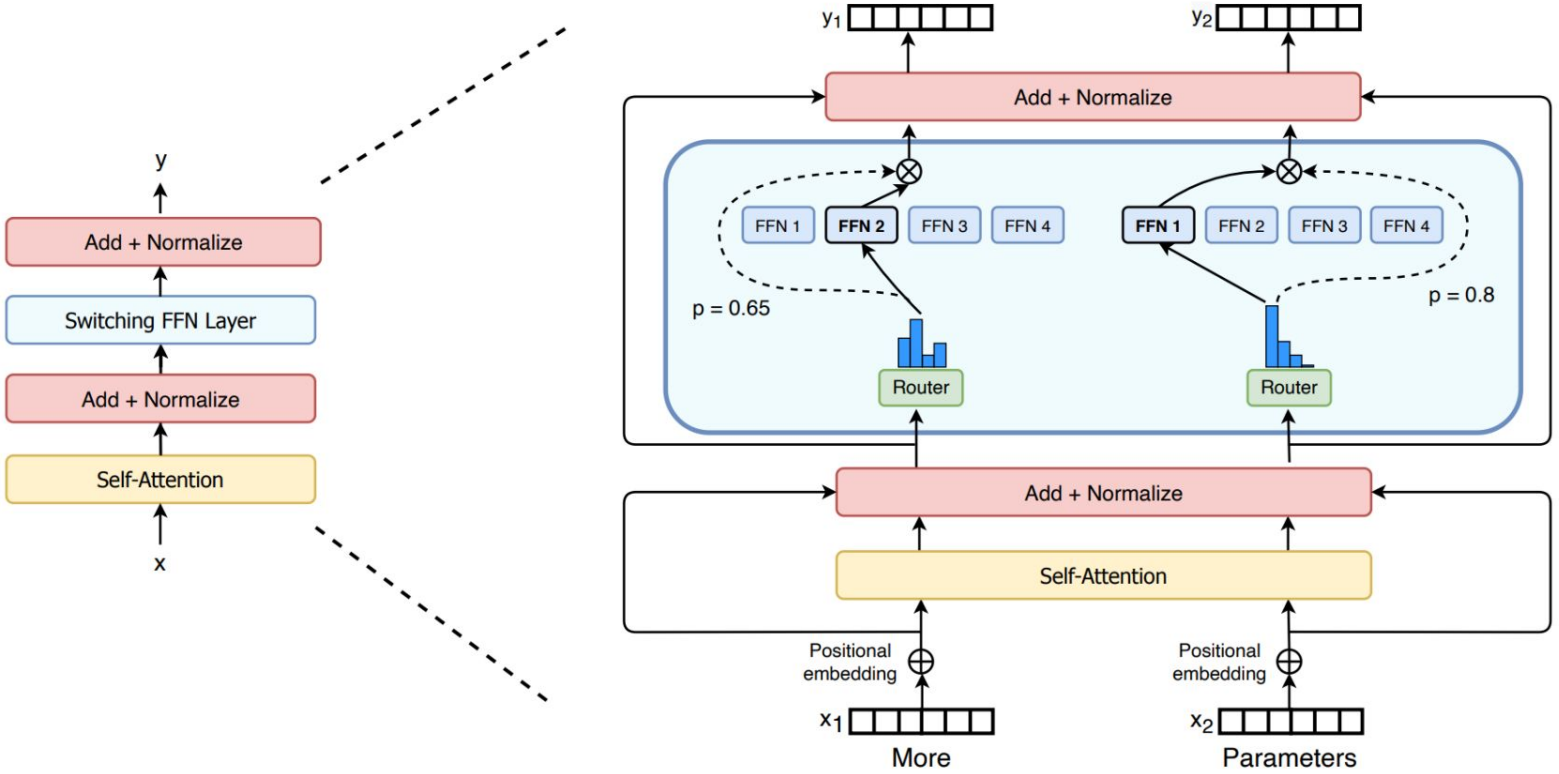


Figure: [Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity](#) (2022)

Routing Strategies

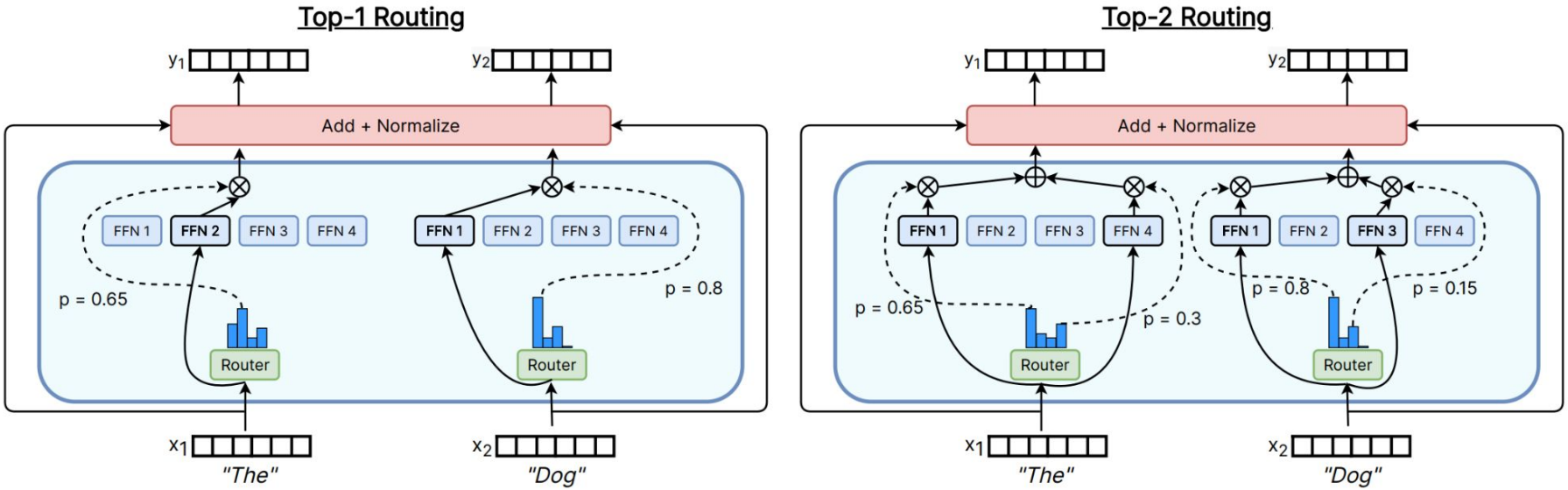


Figure: A Review of Sparse Expert Models in Deep Learning

Routing Strategies

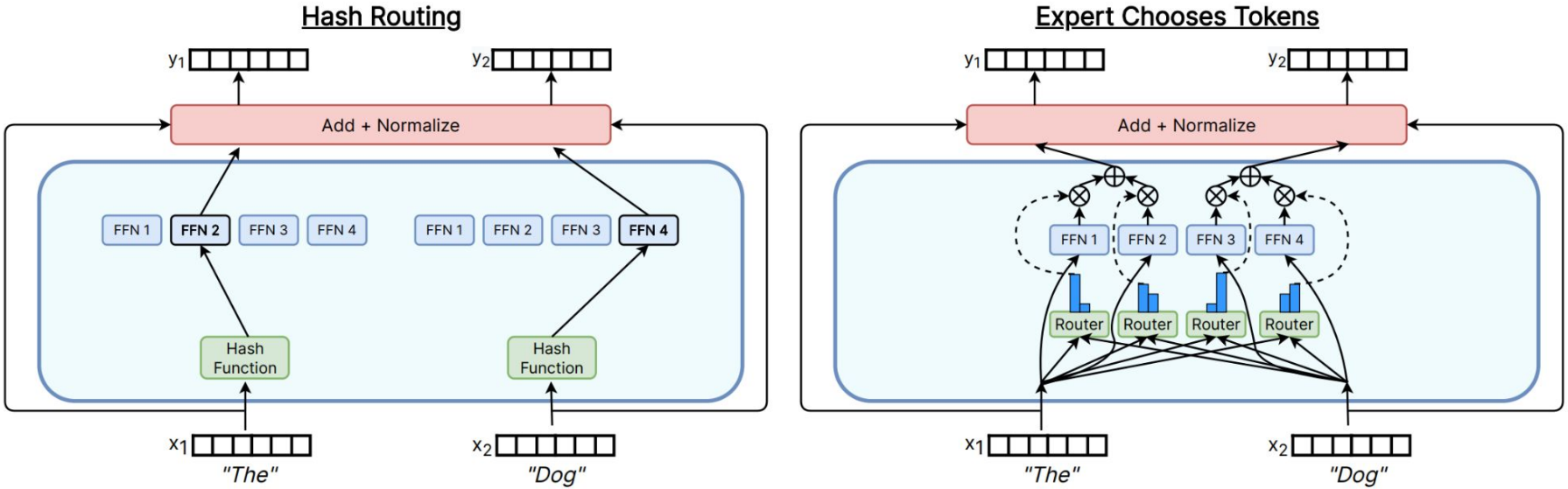


Figure: A Review of Sparse Expert Models in Deep Learning

Routing Strategies

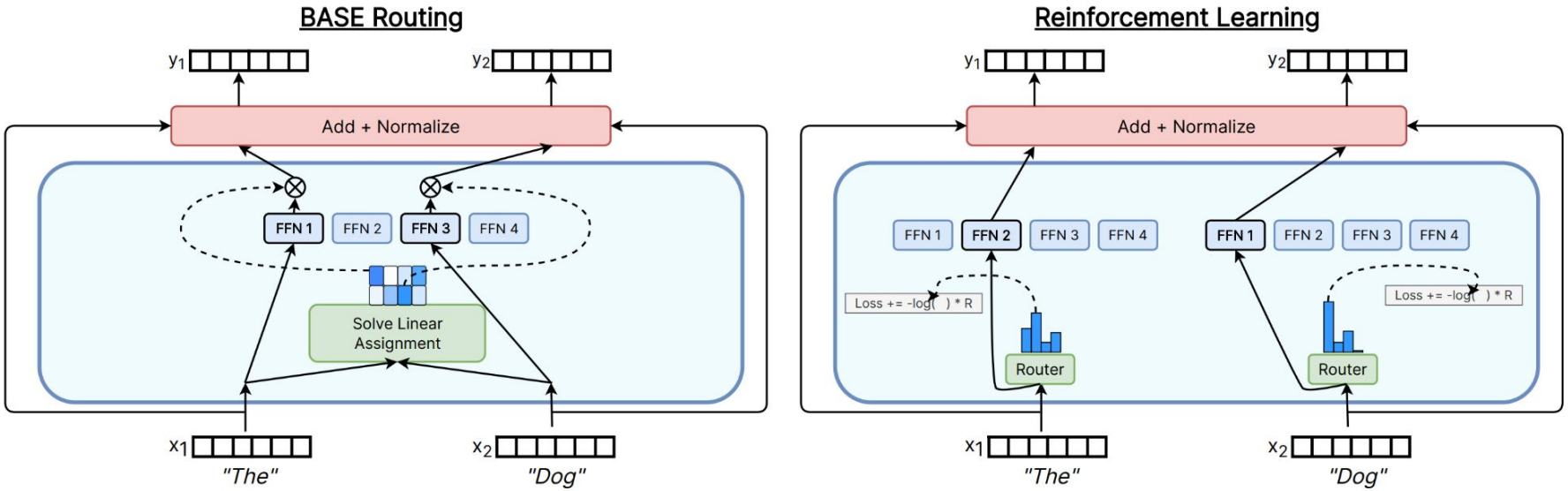


Figure: A Review of Sparse Expert Models in Deep Learning

Routing Strategies

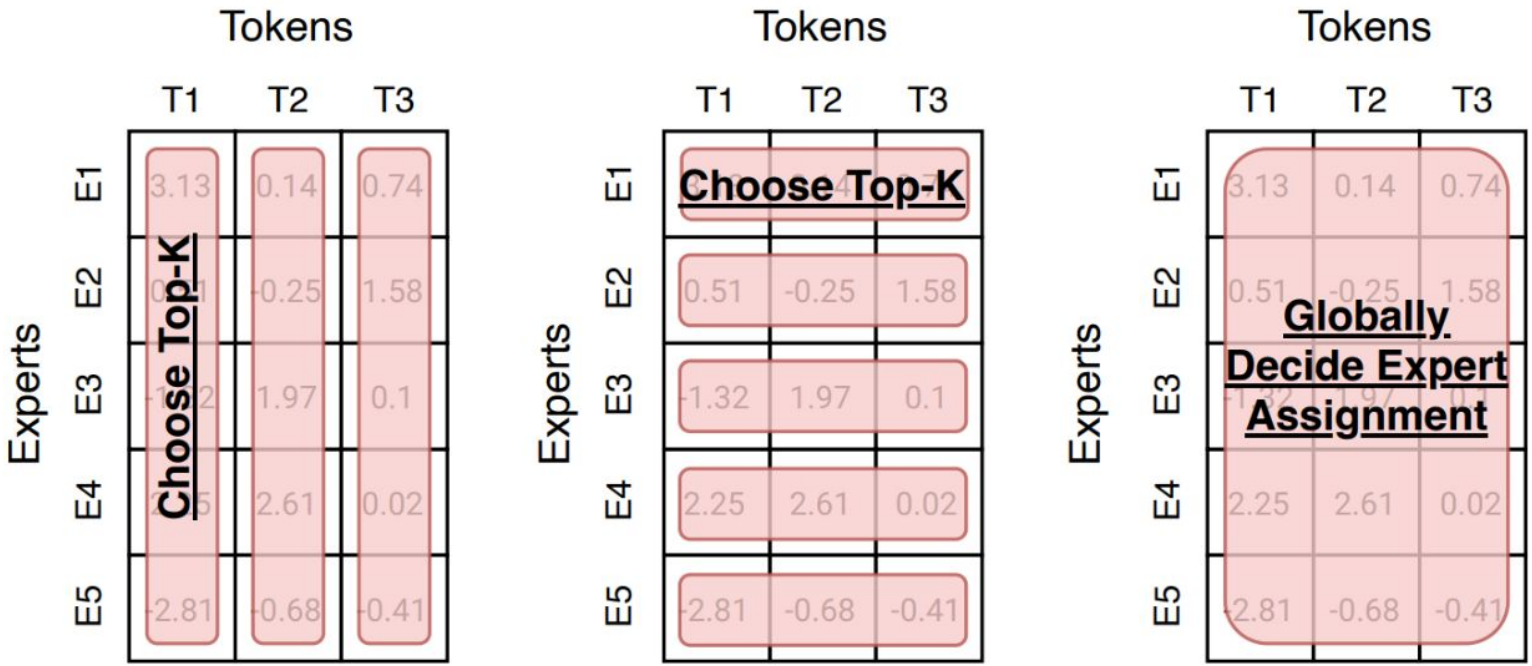


Figure: A Review of Sparse Expert Models in Deep Learning

What Recap

The "typical" Sparse MoE Transformer:

Decoder-only

FFN is MoE every block (or every other block)

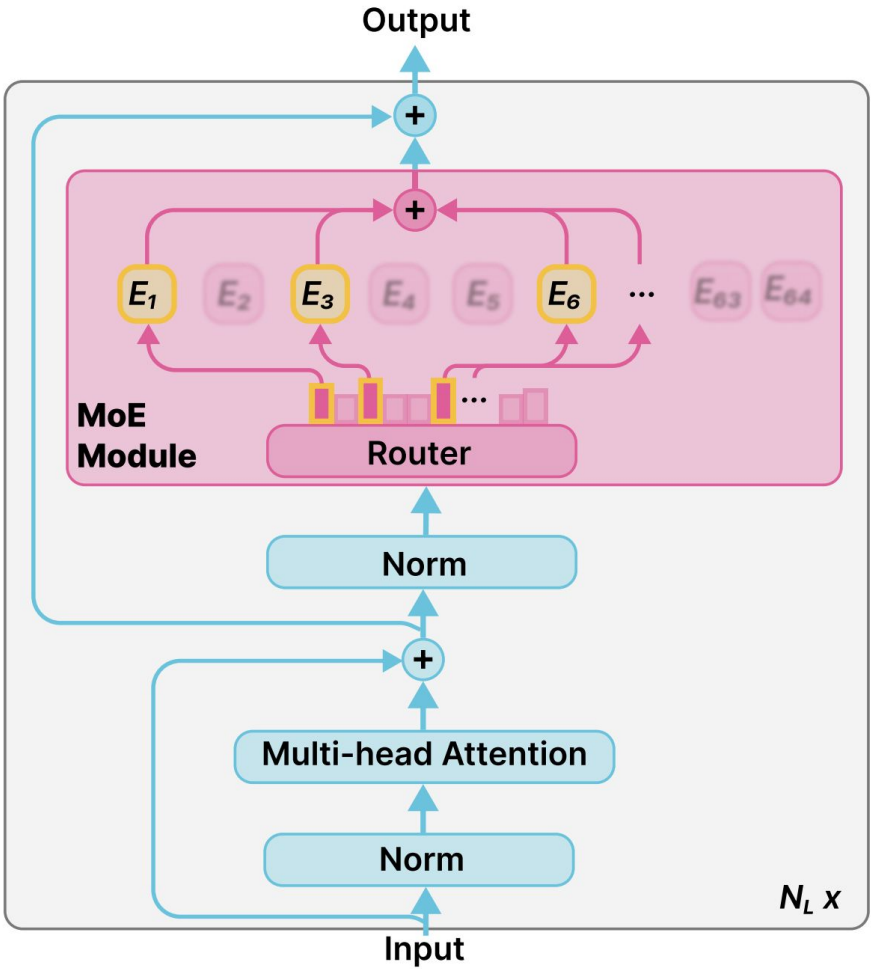
Experts are single hidden layer MLPs

Softmax routers

Top-k choices per token: k = 1 or 2

Everything else similar to Dense models (normalization, attention, residual connections)

Figure: OLMoE: Open Mixture-of-Experts Language Models



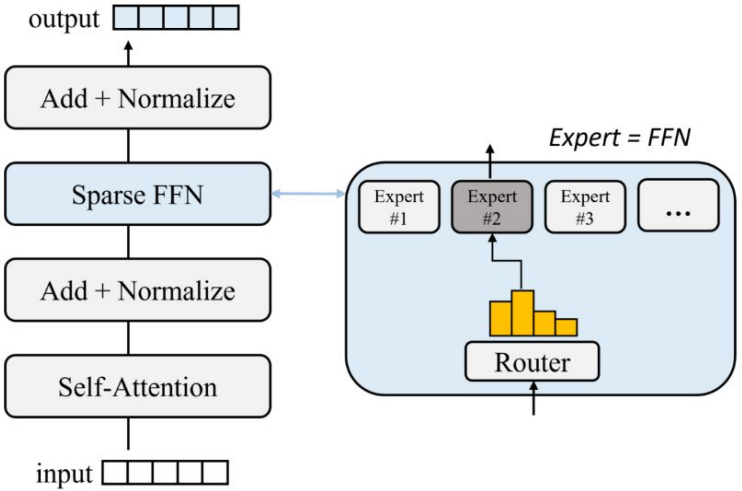
Overview

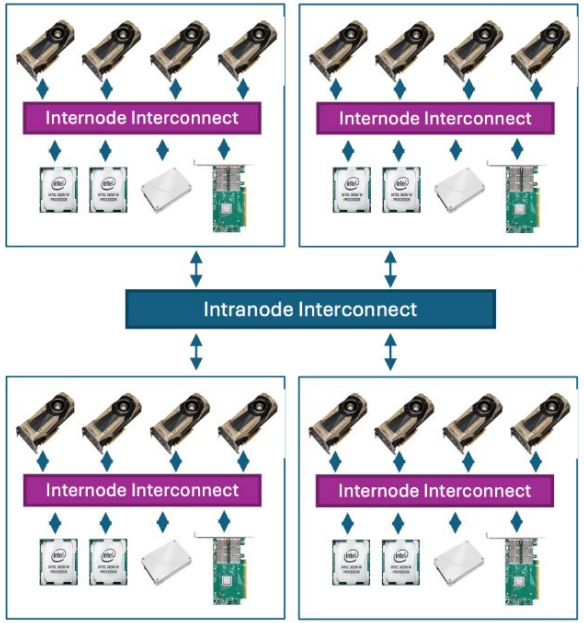
Sparse MoEs:

- What (origins & modern)
- How (to train)
- Scaling laws
- Best practices

MoE Training on Modern Hardware

- How to break the memory wall to enable massive MoEs?
- How to efficiently route tokens to different experts across GPUs?
- How to minimize communication overhead while achieving high per-GPU compute throughput?

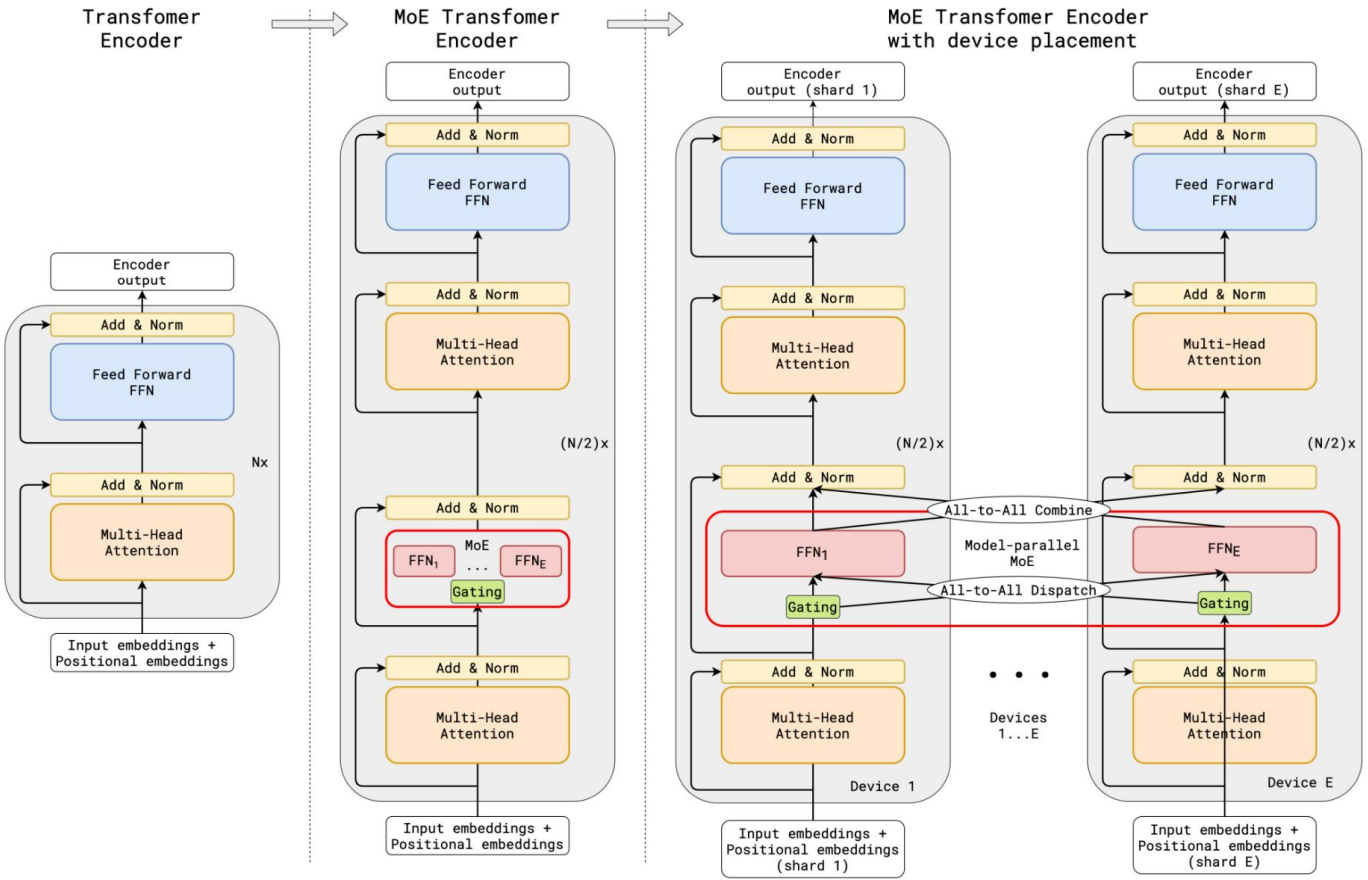




Source: [Mixture-of-Experts in the Era of LLMs A New Odyssey]

Figure: GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding

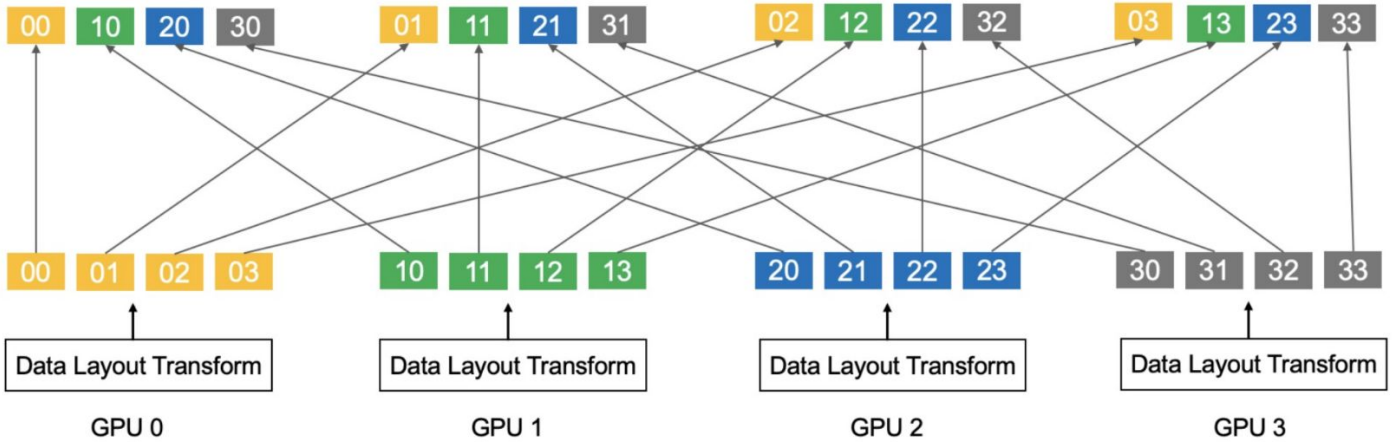
Expert Parallelism (GShard)



Expert Parallelism

Source: Mixture-of-Experts in the Era of LLMs A New Odyssey

1. Gating function: decide target experts for each token
2. **Dispatch phase**:
   a. 1st layout transformation: tokens to the same target experts are grouped in a continuous memory buffer
   b. 1st All2All: dispatch tokens to their corresponding experts
3. Expert compute: each expert process its tokens
4. **Combine phase**:
   a. 2nd All2All: combine processed tokens back to their GPUs
   b. 2nd layout transform: restore tokens to their original positions



Multidimensional Parallelism

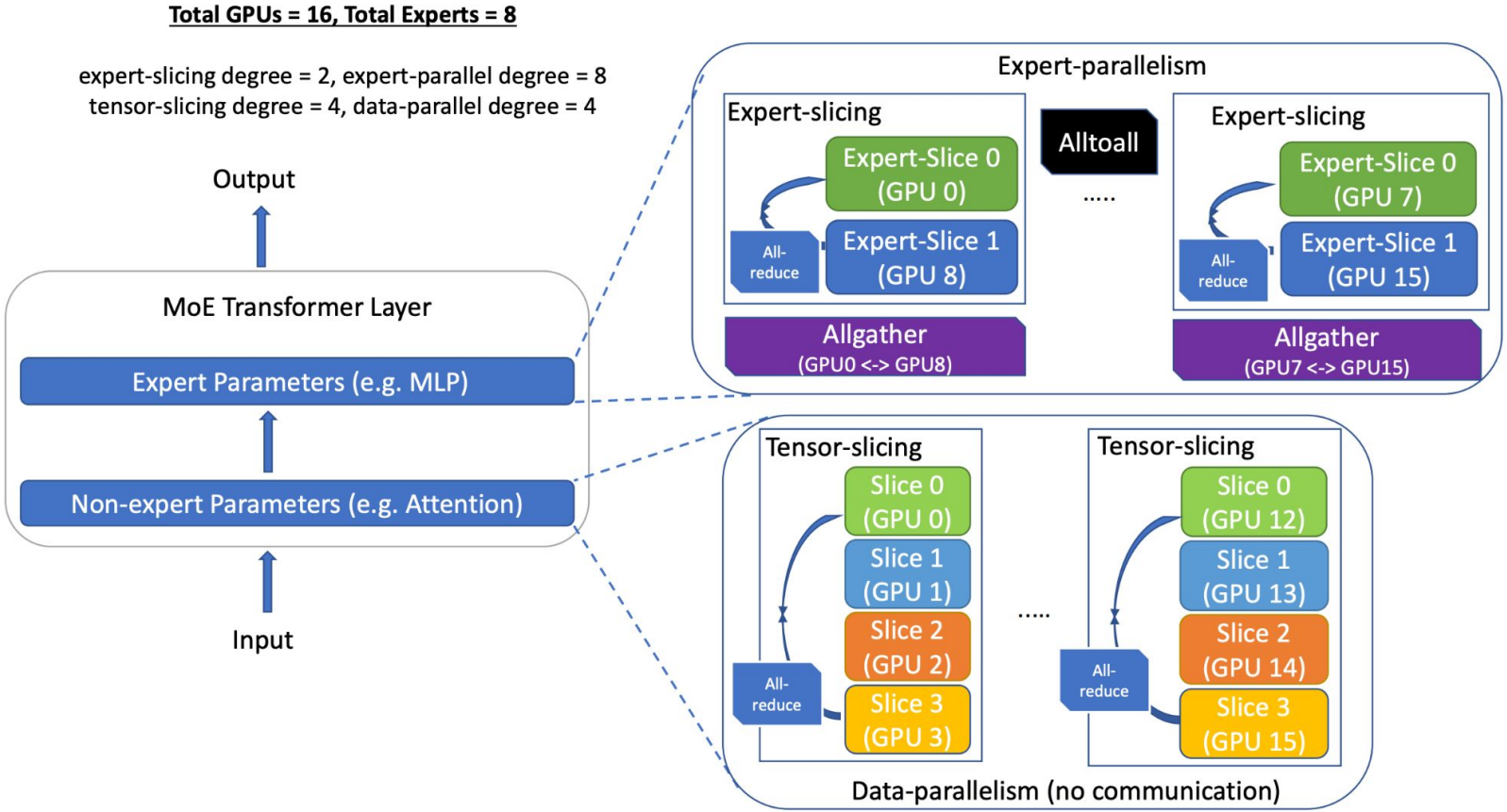
Source: Mixture-of-Experts in the Era of LLMs A New Odyssey

Short Name	Flexible Parallelism Combinations	Benefit
E	Expert	Scales the model size by increasing the number of experts
E+D	Expert + Data	Accelerates training throughput by scaling to multiple data parallel groups
E+Z	Expert + ZeRO	Partitions the nonexpert parameters to support larger base models
E+D+M	Expert + Data + Model	Supports massive hidden sizes and even larger base models than E+Z
E+D+Z	Expert + Data + ZeRO	
E+Z-Off+M	Expert + ZeRO-Offload + Model	Leverages both GPU and CPU memory for large MoE models on limited GPU resources

Optimal parallelism strategy depends on model and hardware specifics

Figure:
DeepSpeed-MoE

Expert Parallelism in Practice



Routing "collapse"

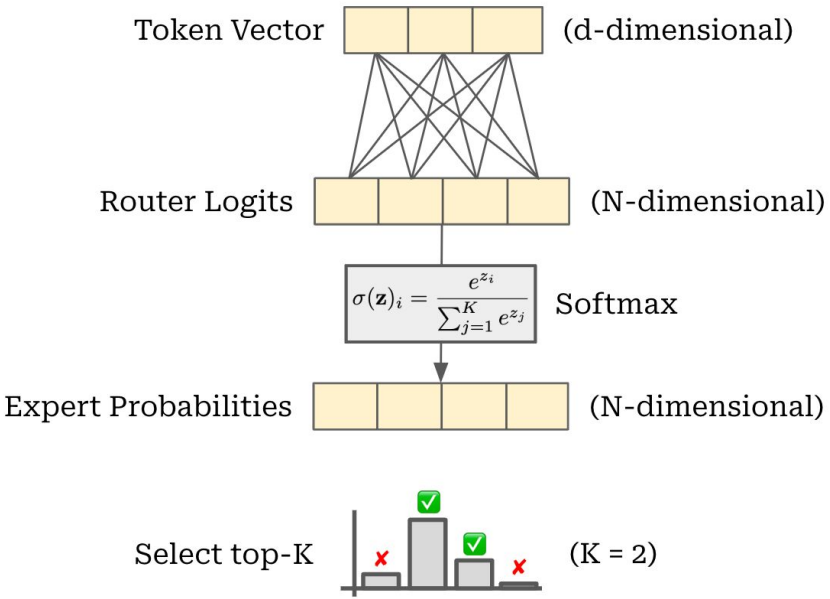


Figure: Mixture-of-Experts (MoE) LLMs

"We have observed that the gating network tends to converge to a state where it always produces large weights for the same few experts."

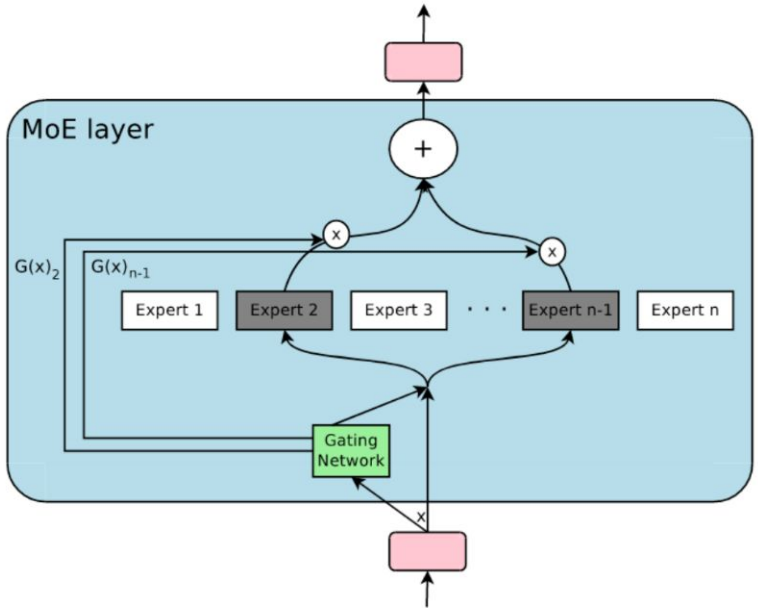
"This imbalance is self-reinforcing, as the favored experts are trained more rapidly and thus are selected even more by the gating network."

From: Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer

Noisy Top-K Routing

Sparsely-Gated MoEs were originally introduced for RNNs, but are generally applicable and now popular for Transformers

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



Sparse MoE

- Softmax over Top-K Gating

$$G(\mathbf{x}) = \text{softmax}(\text{topk}(\mathbf{x} \cdot \mathbf{W}_g + \mathbf{b}_g + \mathbf{r}_{\text{noise}}(\mathbf{x}), k))$$

$$\mathbf{r}_{\text{noise}}(\mathbf{x}) = N(\mathbf{0}, \mathbf{I}) \cdot \text{sigma}(\mathbf{x}\mathbf{W}_{\text{noise}} + \mathbf{b}_{\text{noise}})$$

- Each expert is just a feed-forward network

$$E_i(\mathbf{x}) = \text{FFN}_{\text{ReLU}}(\mathbf{x}) = \text{ReLU}(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$

Figure: Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer

Router Initialization

Sparsely-Gated MoEs were originally introd
are generally applicable and now popular f

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

Mixtral

- Softmax over Top-K Gating

$$G(\mathbf{x}) = \text{softmax}(\text{topk}(\mathbf{x} \cdot \mathbf{W}_g + \mathbf{b}_g + \mathbf{r}_{\text{noise}}(\mathbf{x}), k))$$

$$\mathbf{r}_{\text{noise}}(\mathbf{x}) = N(\mathbf{0}, \mathbf{I}) \cdot \text{sigma}(\mathbf{x}\mathbf{W}_{\text{noise}} + \mathbf{b}_{\text{noise}})$$

- Each expert is just a feed-forward network

$$E_i(\mathbf{x}) = \text{SwiGLU}(\mathbf{x}) = \text{Swish}(\mathbf{x}\mathbf{W} + \mathbf{b}) \otimes (\mathbf{x}\mathbf{V} + \mathbf{c})$$

Initialization:

- We initialize W_g and W_{noise} to all zeros
- Effectively provides no signal and a small amount of noise

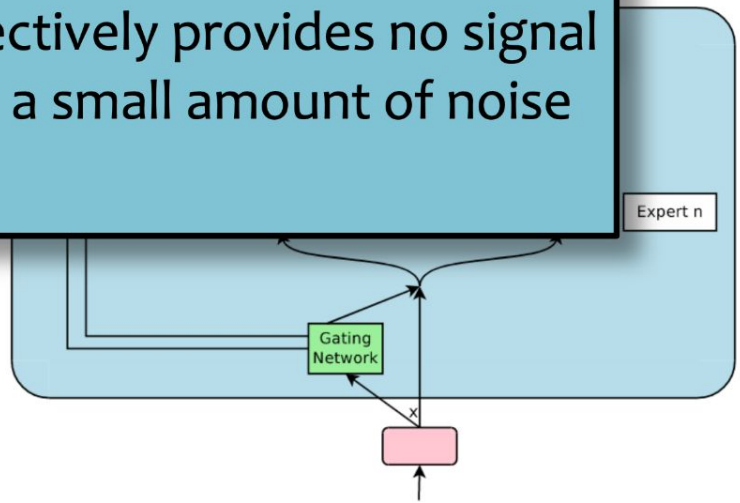


Figure: Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer

Importance Loss

N-dimensional vector of expert probabilities for token x

Linear transformation of token vector

$$G(x) = \texttt{Softmax}(\texttt{top-}k(x \cdot W_g))$$

$$G_i(x) := \text{probability of } i\text{-th expert}$$

N-dimensional vector of expert importance values across all tokens

$$\texttt{Importance}(\mathcal{X}) = \sum_{x \in \mathcal{X}} G(x)$$

Set of all tokens in batch

Weighted scalar importance loss derived from all expert importances

Minimized when experts receive equal importance

$$L_{\texttt{Importance}}(\mathcal{X}) = w_{\texttt{Importance}} \cdot \texttt{CV}(\texttt{Importance}(\mathcal{X}))^2$$

Constant scaling factor for the importance loss

Figure: Mixture-of-Experts (MoE) LLMs

Load Balancing Loss

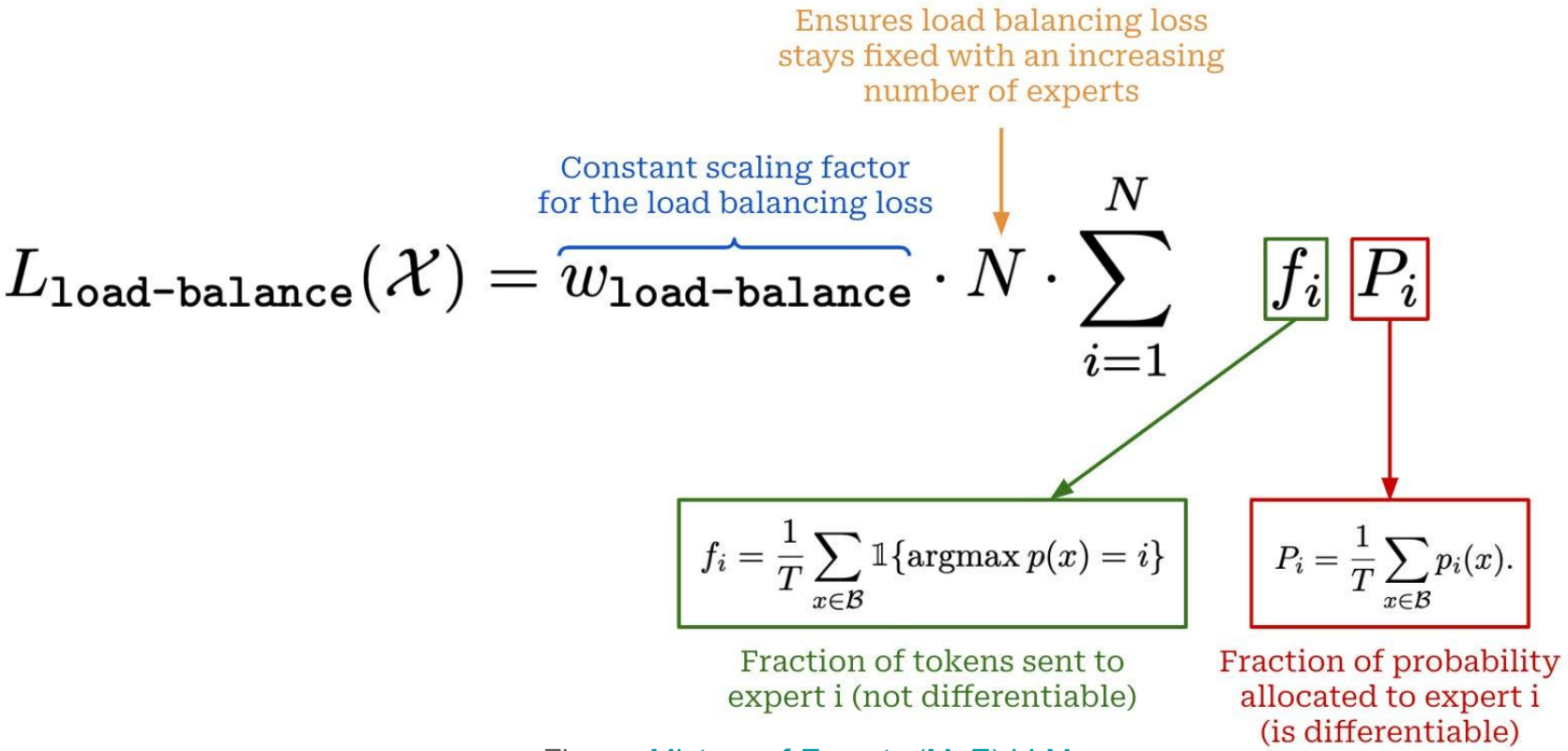


Figure: Mixture-of-Experts (MoE) LLMs

Router z-loss

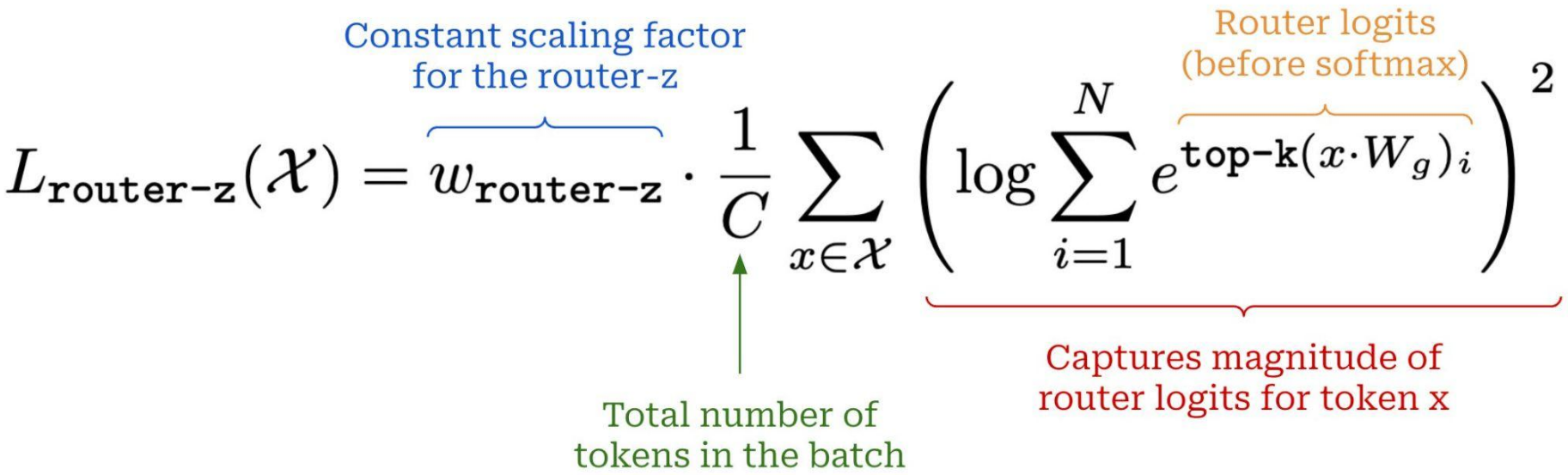


Figure: Mixture-of-Experts (MoE) LLMs

Full Training Loss

Standard Language Modeling Loss

Weighted load balancing loss

Weighted router-z loss

$$L_{\texttt{full}}(\mathcal{X}) = L_{\texttt{LM}}(\mathcal{X}) + w_{\texttt{load-balance}} \cdot L_{\texttt{load-balance}}(\mathcal{X}) + w_{\texttt{router-z}} \cdot L_{\texttt{router-z}}(\mathcal{X})$$

Figure: Mixture-of-Experts (MoE) LLMs

Expert Capacity (& E. C. Factor)

- **Expert Capacity:** Batch size of each expert. Calculated as
- (tokens_per_batch / num_experts) * capacity_factor

- **Capacity Factor:** Used when calculating expert capacity. Expert capacity allows more buffer to help mitigate token overflow during routing.

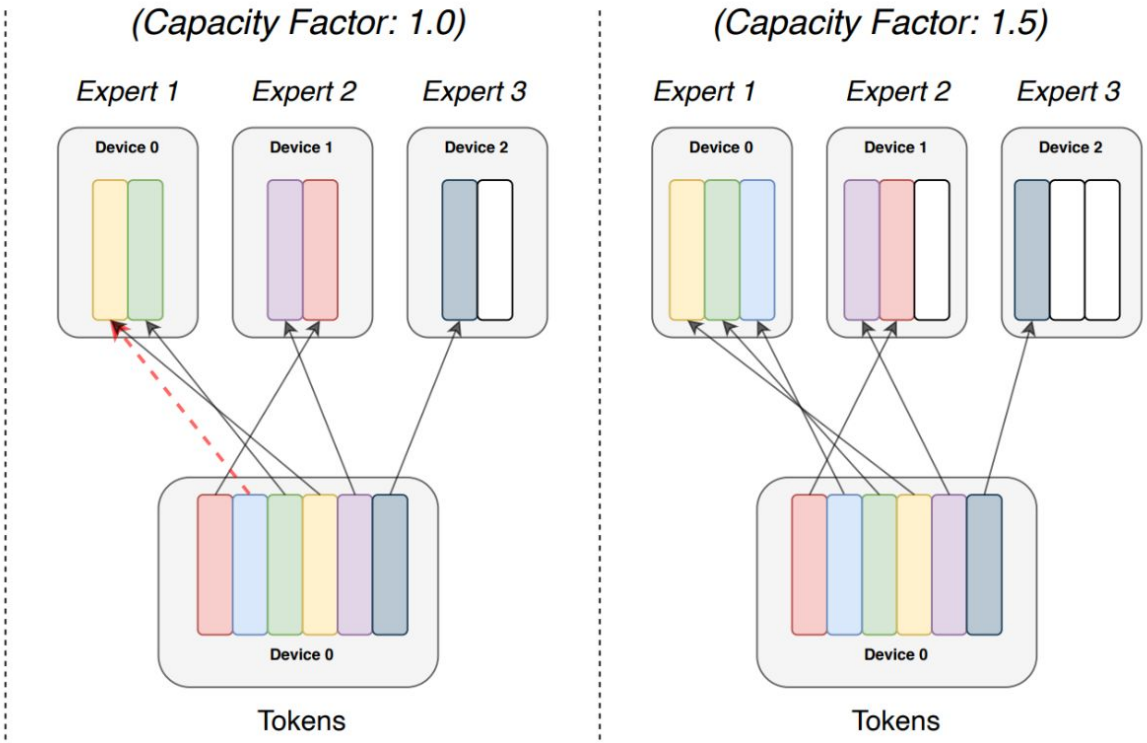


Figure: Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity

Outcome

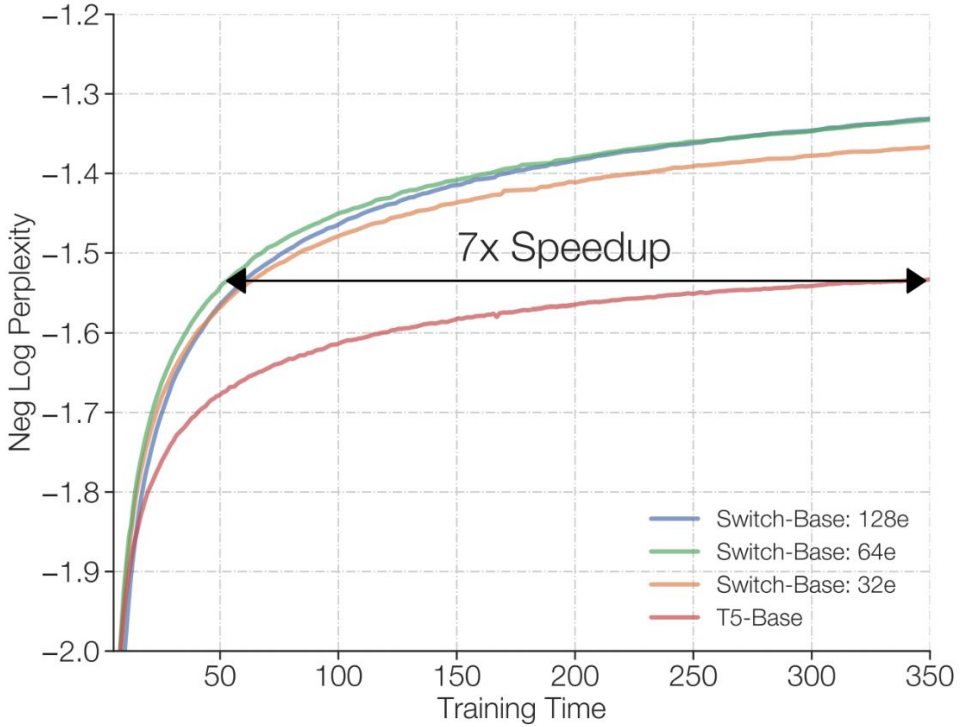


Figure: Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity

Overview

Sparse MoEs:

- What (origins & modern)
- How (to train)
- Scaling laws
- Best practices

Unified Scaling Laws

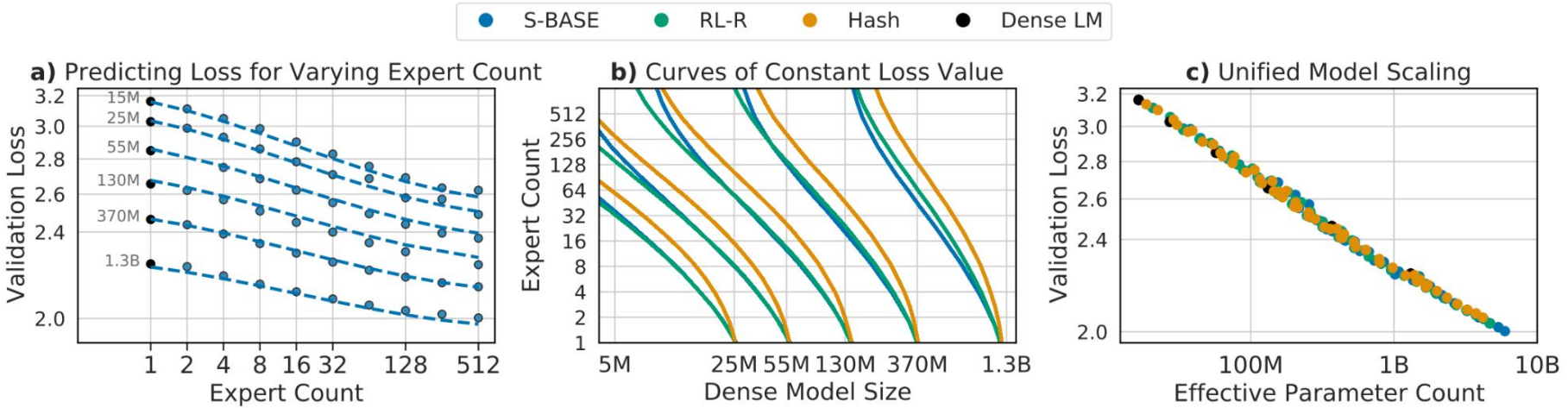


Figure 1: **(a)** The performance achieved by Routing Networks when varying the number of experts for a fixed dense model size is described by a bilinear function (Eq. 1), **(b)** whose level curves indicate how to trade model size with expert count to maintain a fixed performance, **(c)** and which can be manipulated to align dense and routed model performance under a shared power law.

Figure: Unified Scaling Laws for Routed Language Models

Unified Scaling Laws

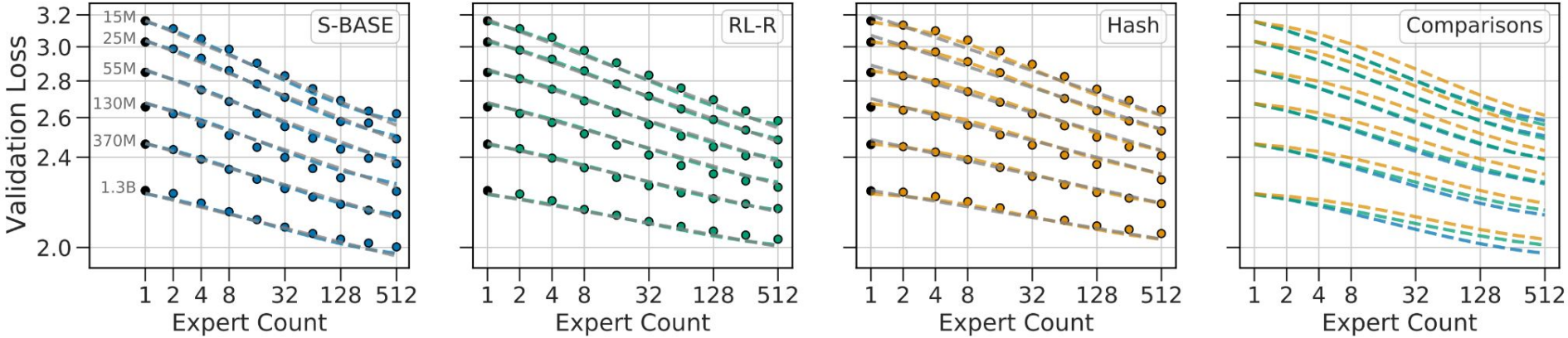


Figure: [Unified Scaling Laws for Routed Language Models]

Unified Scaling Laws

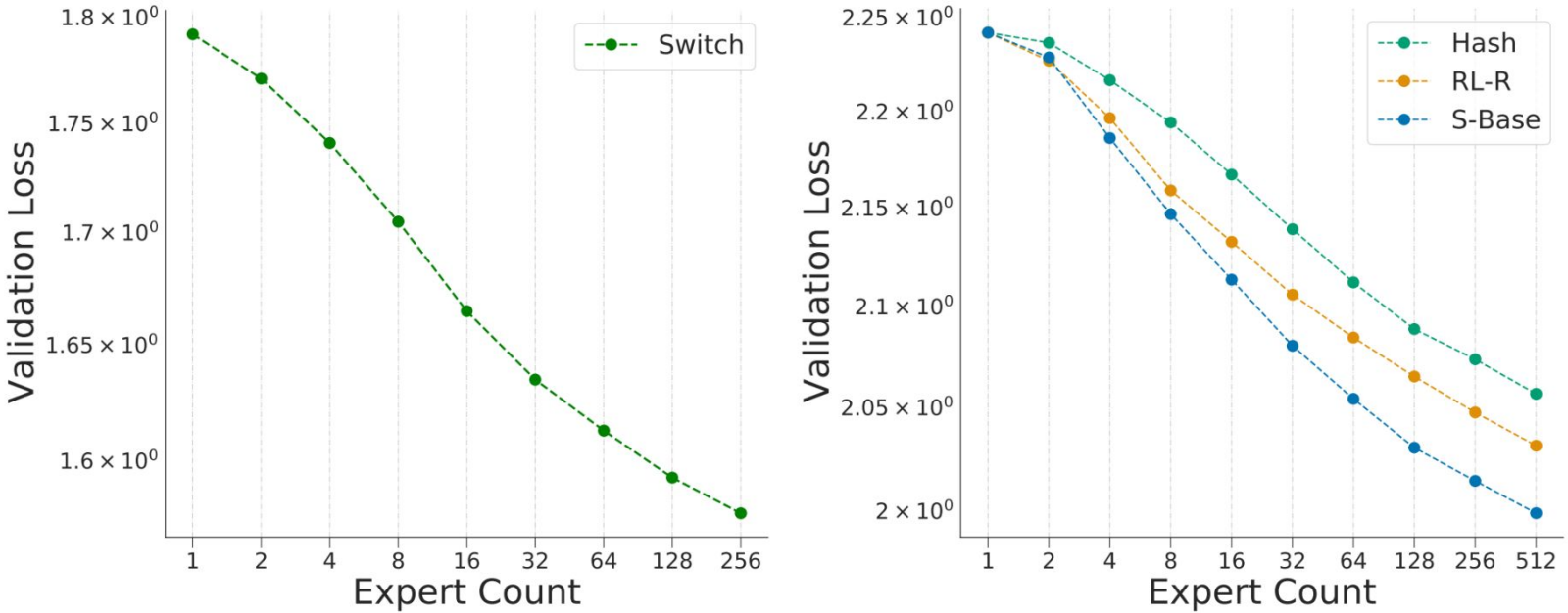


Figure: Unified Scaling Laws for Routed Language Models

FINE-GRAINED MoE Scaling Laws

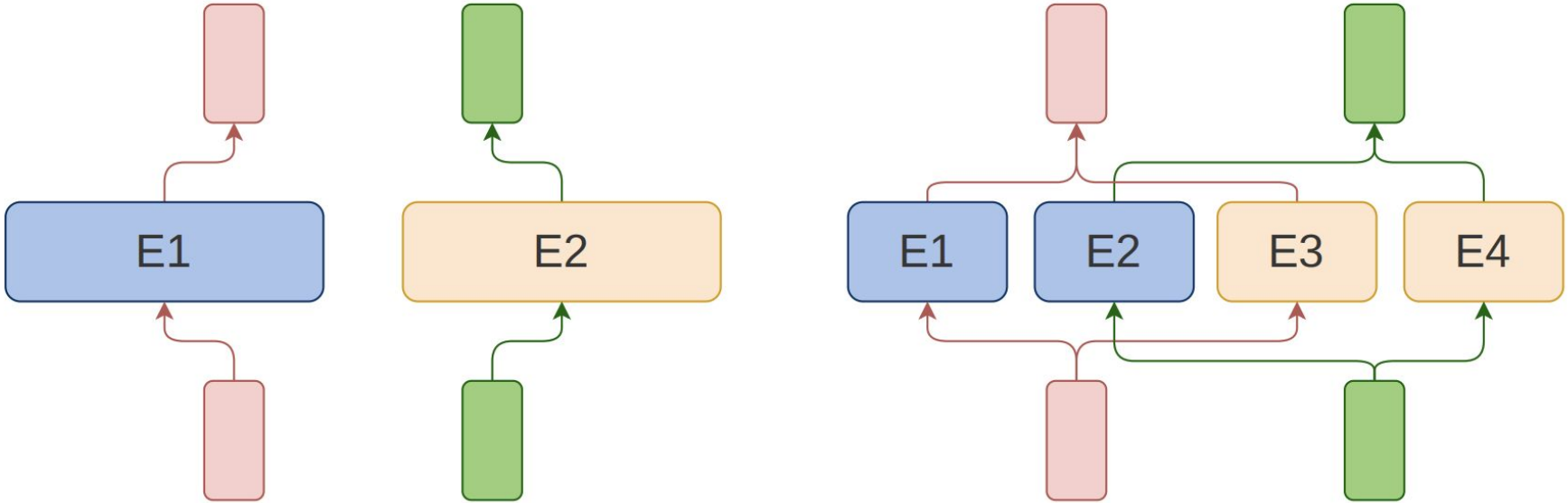


Figure: Scaling Laws for Fine-Grained Mixture of Experts

FINE-GRAINED MoE Scaling Laws

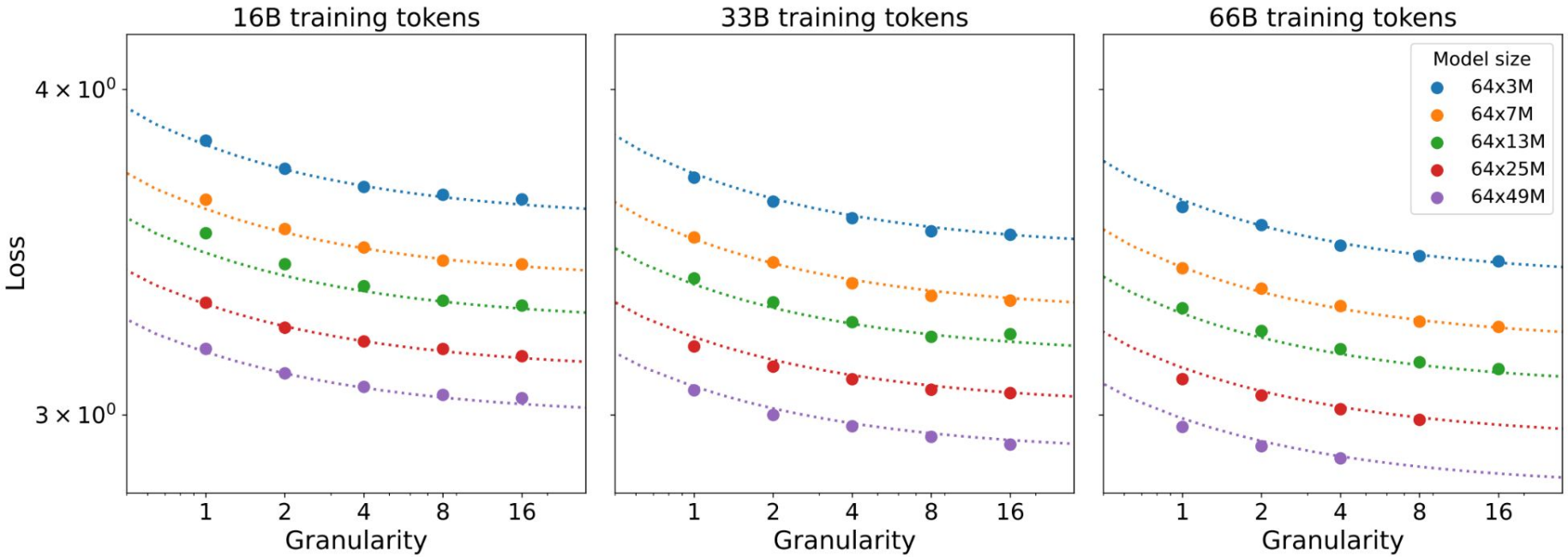


Figure: [Scaling Laws for Fine-Grained Mixture of Experts]

FINE-GRAINED MoE Scaling Laws

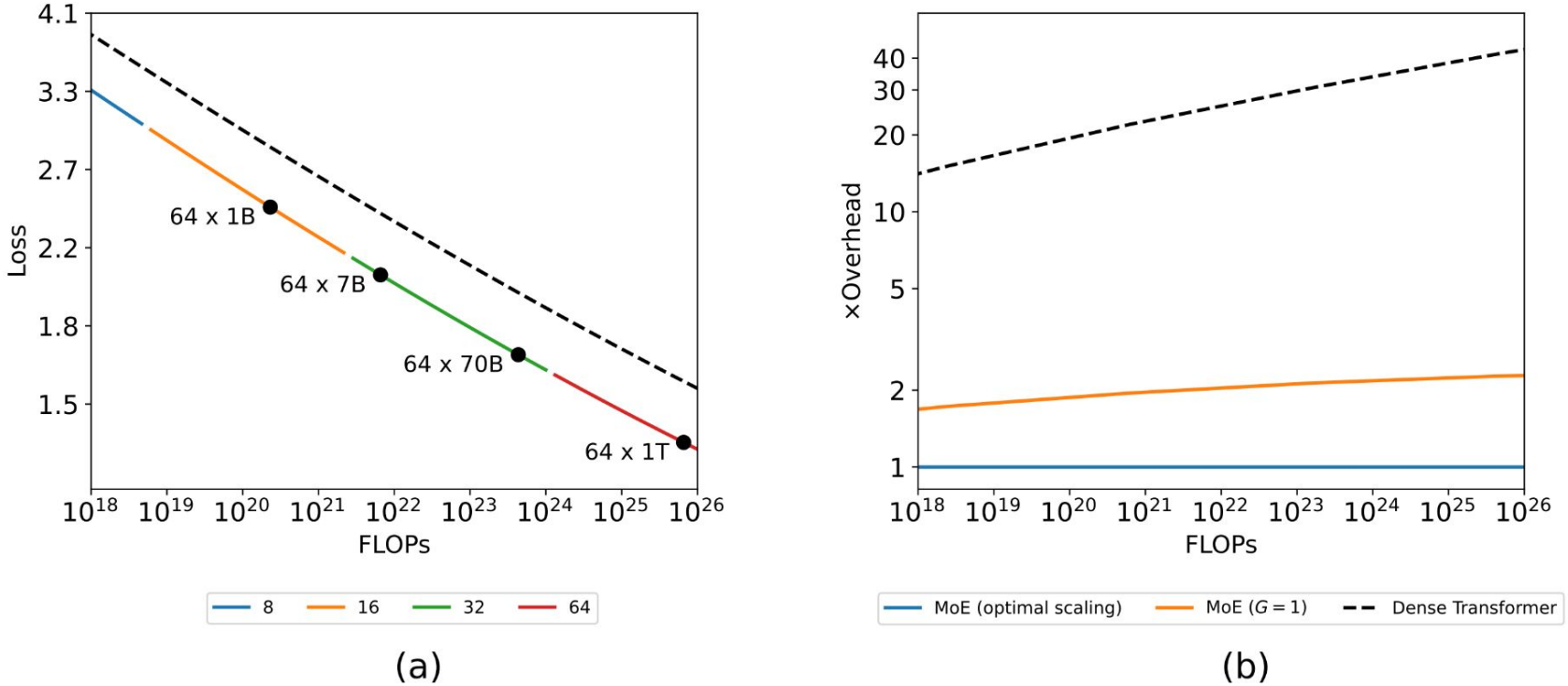


Figure: [Scaling Laws for Fine-Grained Mixture of Experts]

FINE-GRAINED MoE Scaling Laws

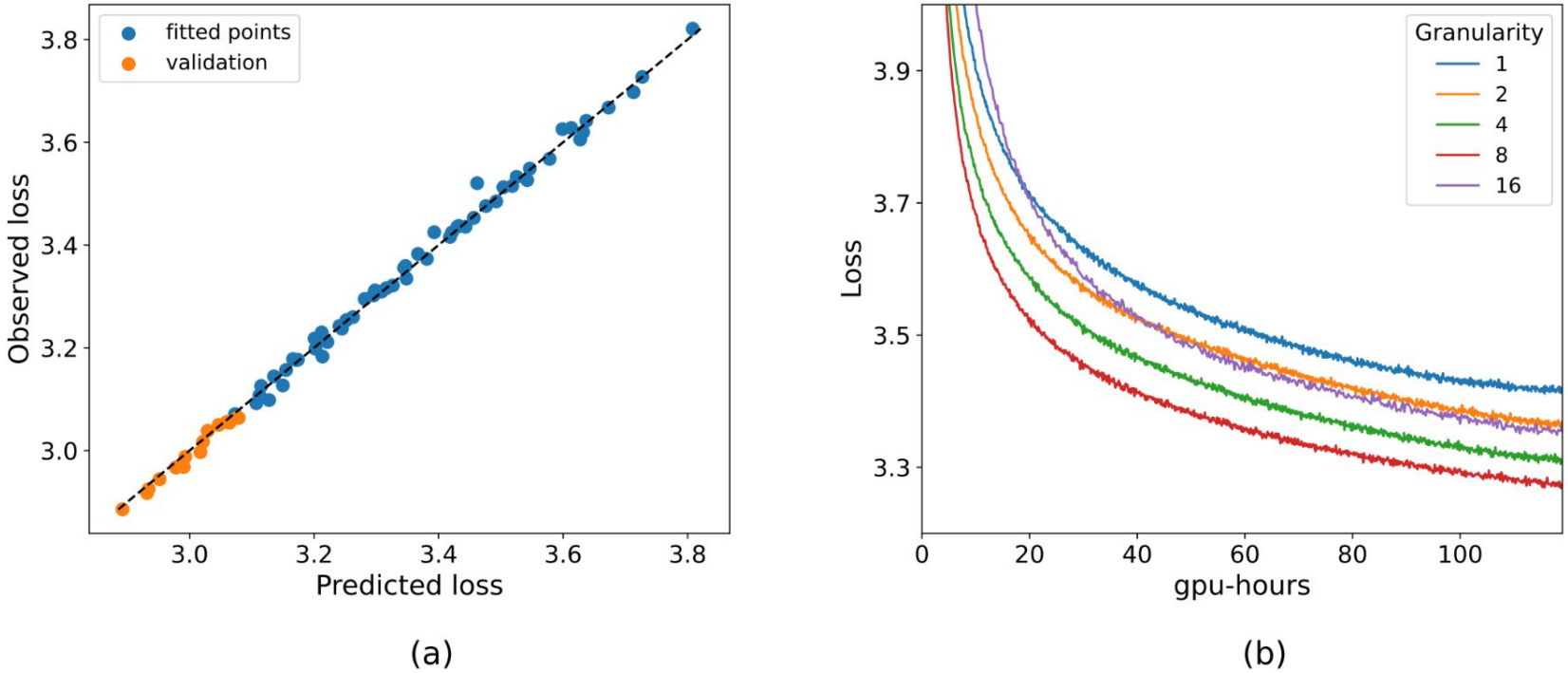


Figure: [Scaling Laws for Fine-Grained Mixture of Experts]

Scaling Laws Recap

Well configured sparsely-gated MoE Transformers are more compute efficient than FLOPs-matched Dense variants.

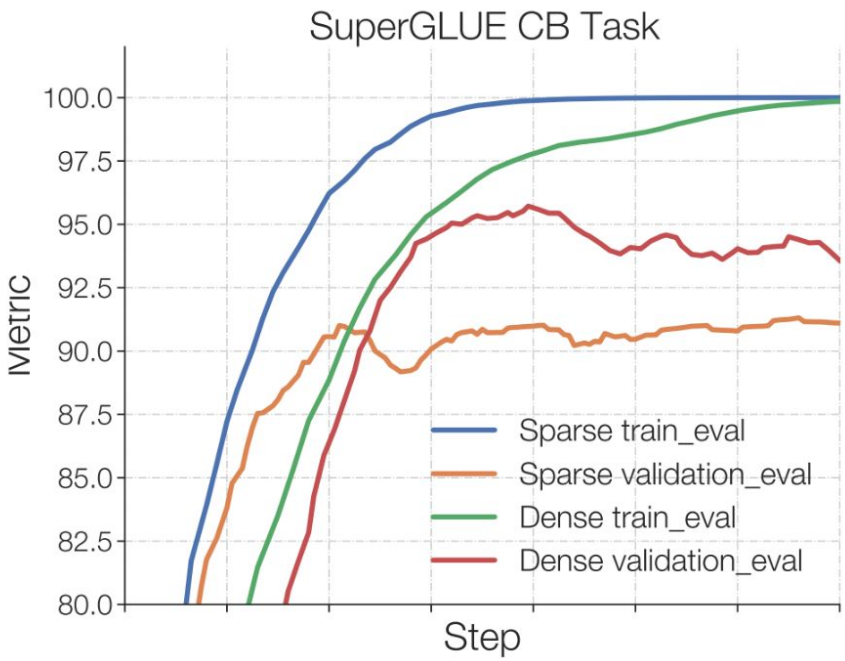
There may be advantages coming from higher granularity.

Overview

Sparse MoEs:

- What (origins & modern)
- How (to train)
- Scaling laws
- Best practices

Finetuning



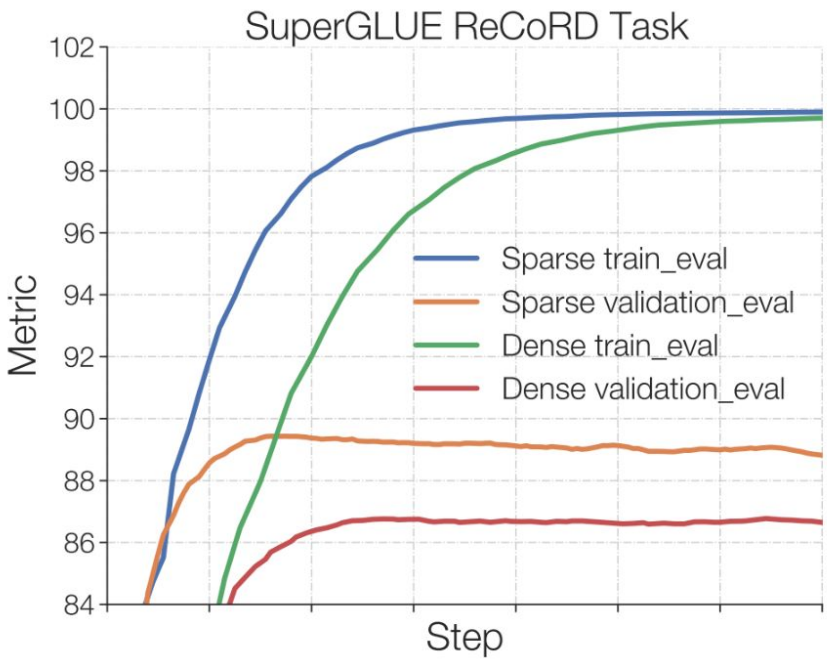


Figure: [ST-MoE: Designing Stable and Transferable Sparse Expert Models]

Flan-MoE Finetuning

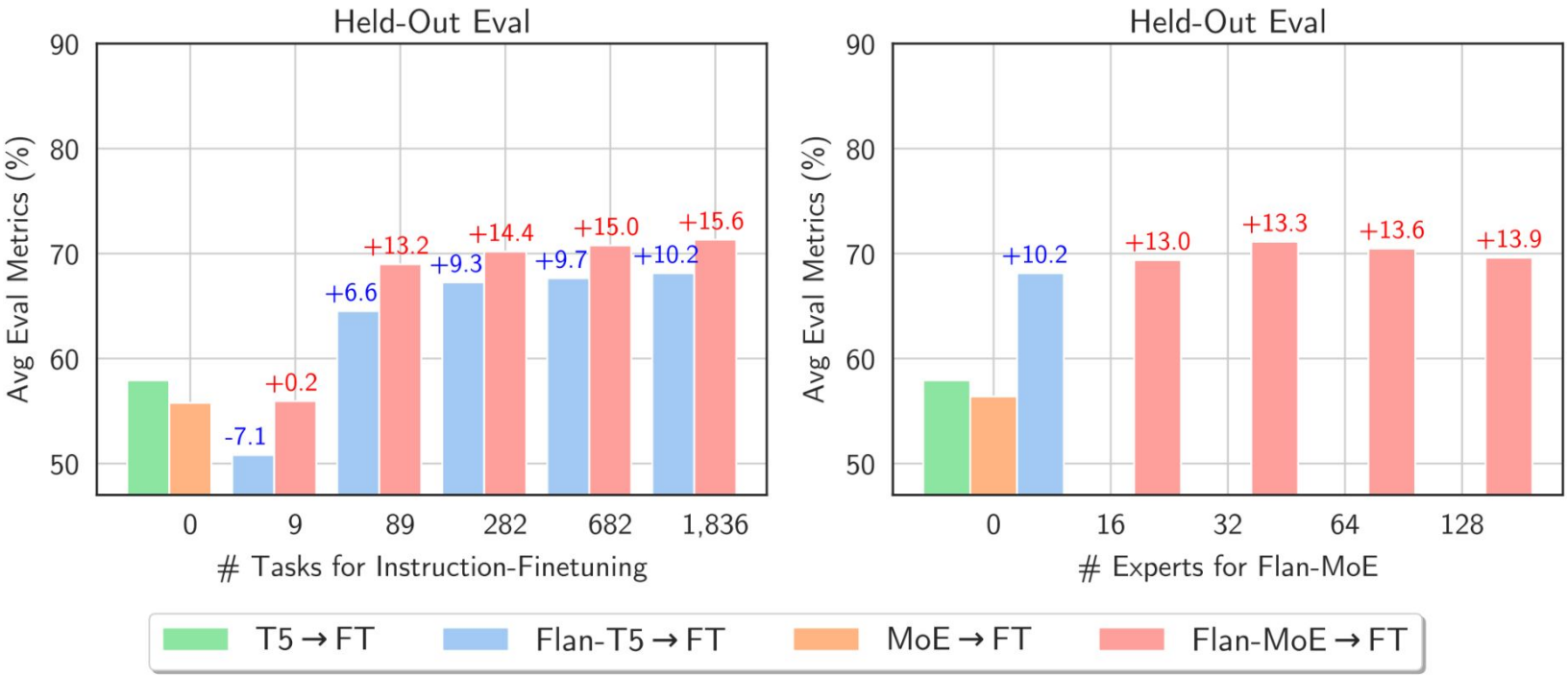
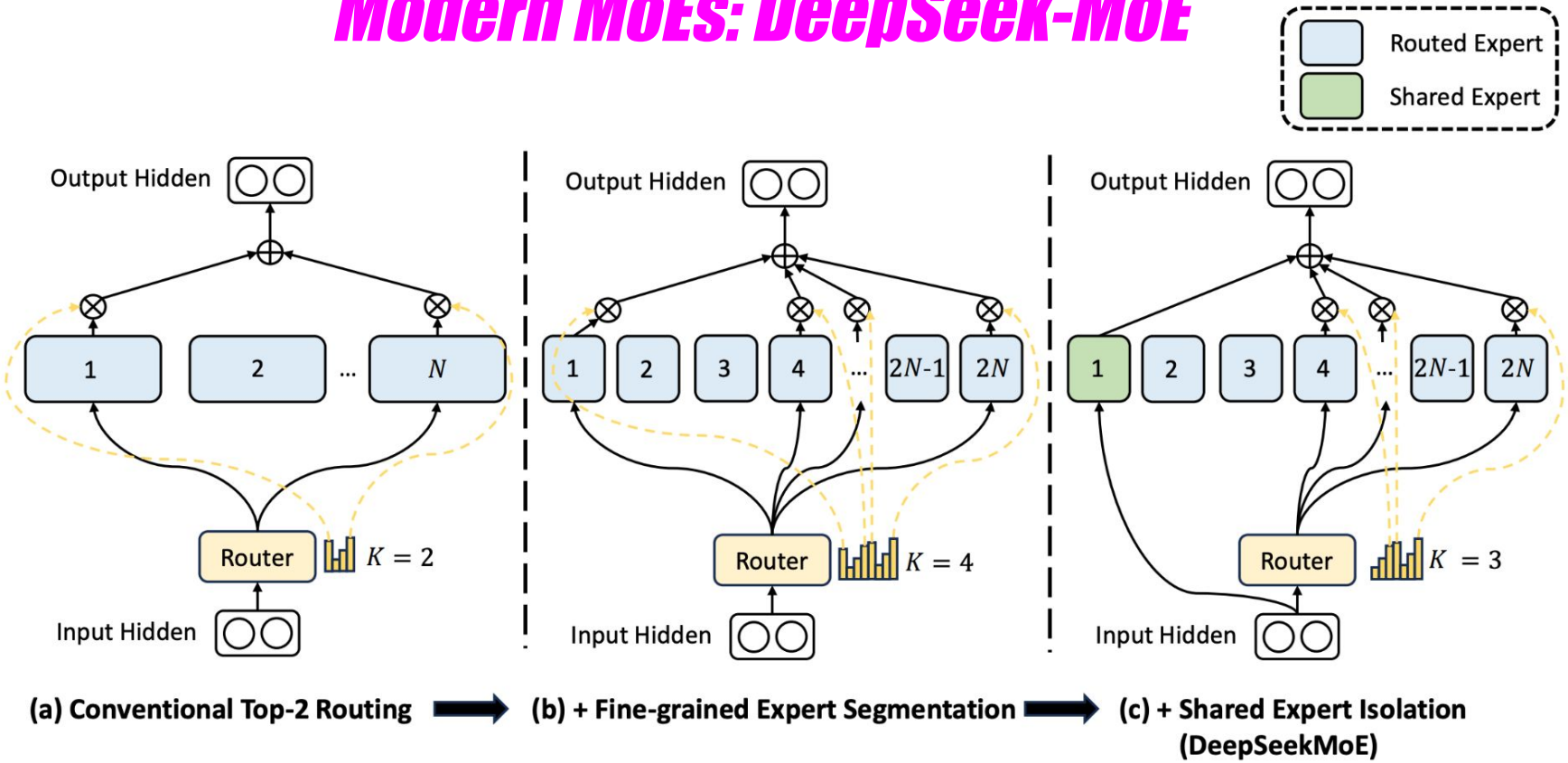


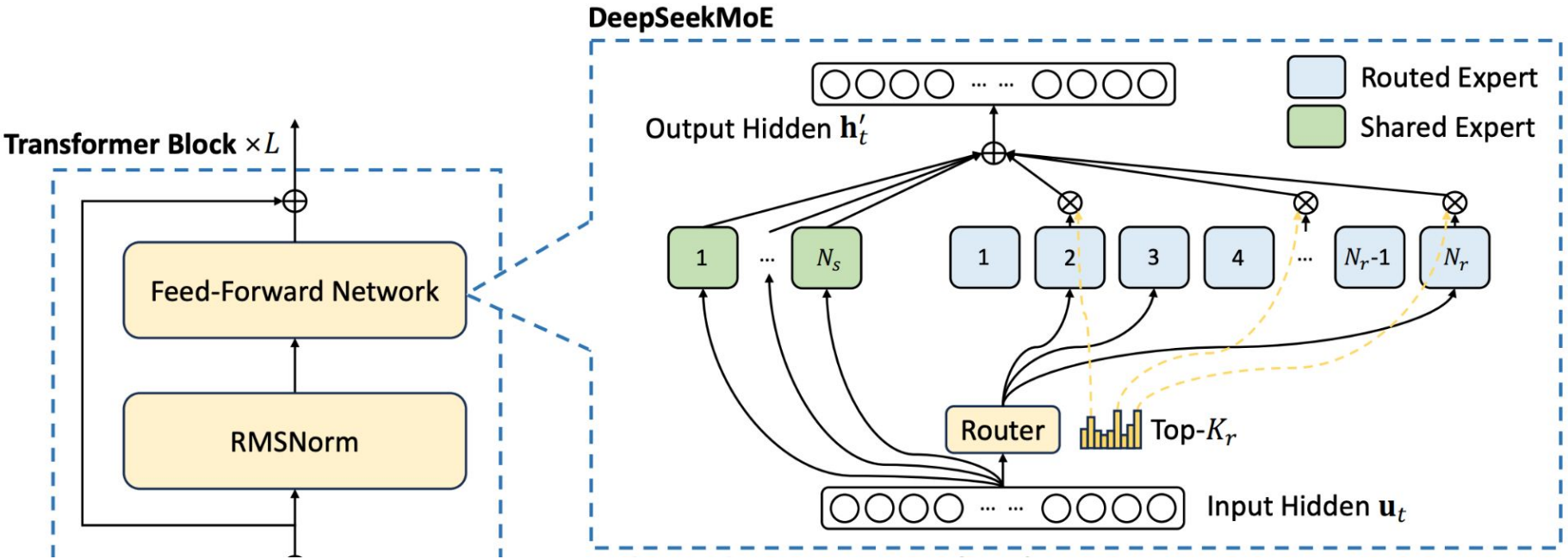
Figure: [Mixture-of-Experts Meets Instruction Tuning:A Winning Combination for Large Language Models]

Modern MoEs: DeepSeek-MoE



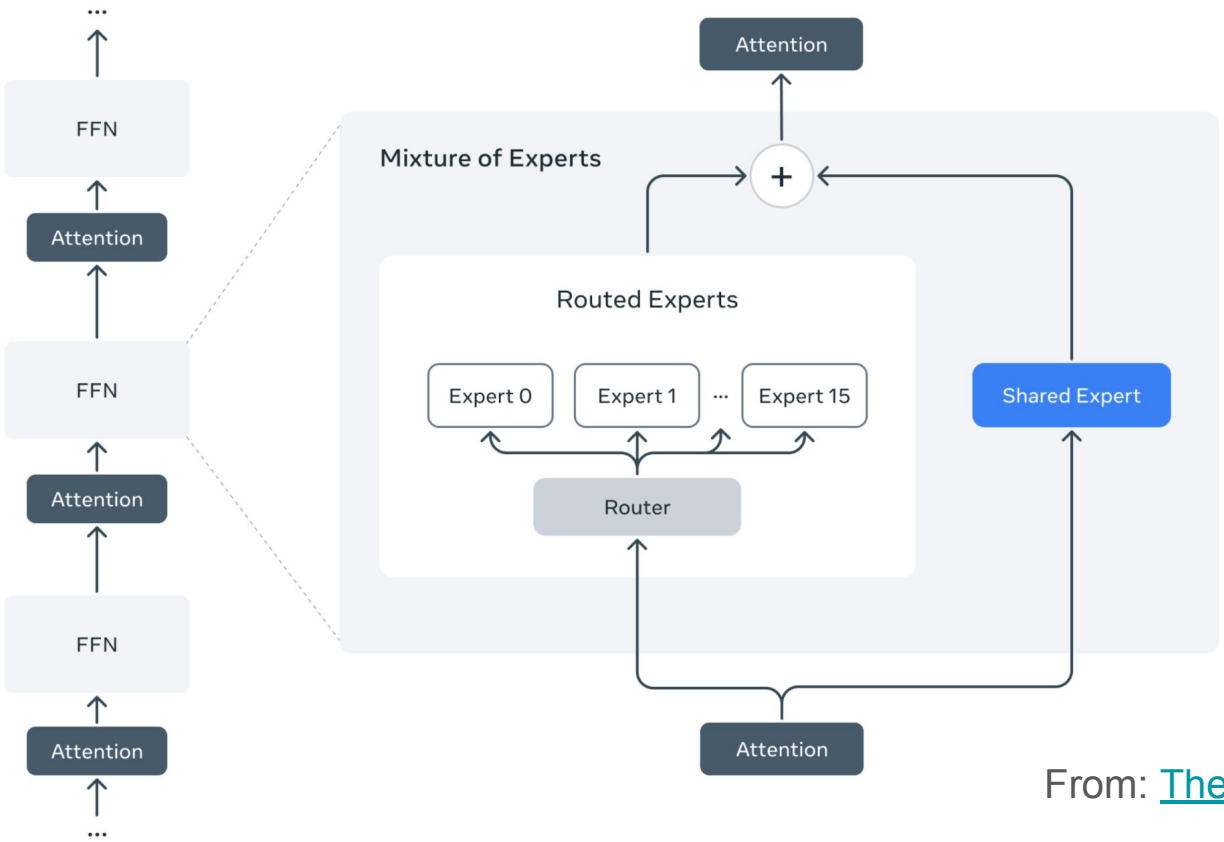
From: DeepSeek-V3 Technical Report

Modern MoEs: DeepSeek-v3

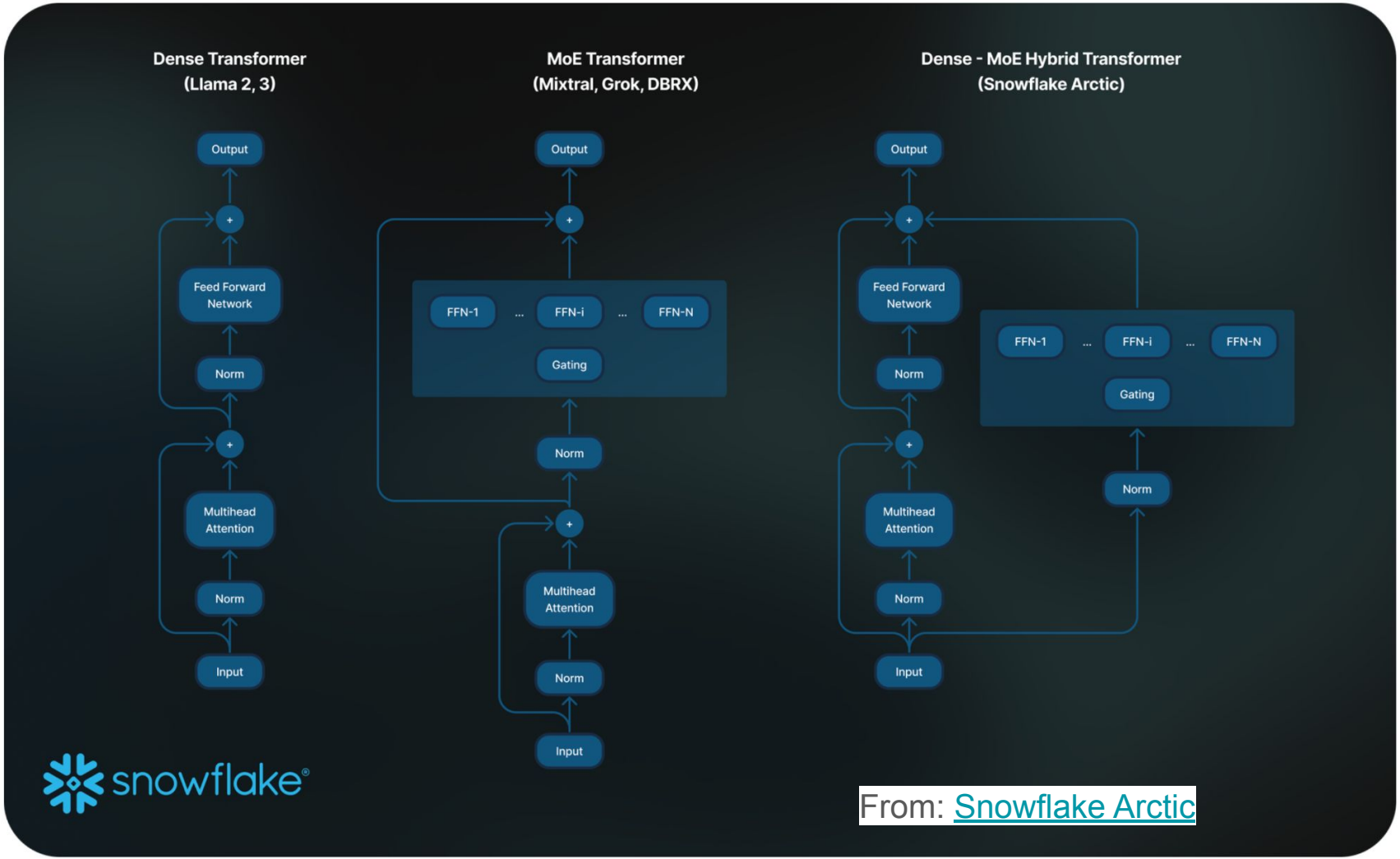


From: DeepSeek-V3 Technical Report

Modern MoEs: Llama 4



From: [The Llama 4 herd]


Dense Transformer
(Llama 2, 3)
Output
+
Feed Forward Network
Norm
+
Multihead Attention
Norm
Input
MoE Transformer
(Mixtral, Grok, DBRX)
Output
+
FFN-1
...
FFN-i
...
FFN-N
Gating
Norm
+
Multihead Attention
Norm
Input
Dense - MoE Hybrid Transformer
(Snowflake Arctic)
Output
+
Feed Forward Network
Norm
+
Multihead Attention
Norm
Input
FFN-1
...
FFN-i
...
FFN-N
Gating
Norm
snowflake
From: Snowflake Arctic

Lecture Recap

MoEs increase capacity without increasing computation.

We payed for this by accepting additional model complexity.

Learning and downstream performance are both superior.

Sources and References

- Adaptive Mixtures of Local Experts (1991)
- Hierarchical Mixtures Of Experts And The Em Algorithm (1993)
- Learning Factored Representations in a Deep Mixture of Experts
- Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer
- GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding
- Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity
- A Review of Sparse Expert Models in Deep Learning
- ST-MoE: Designing Stable and Transferable Sparse Expert Models
- Mixture-of-Experts Meets Instruction Tuning:A Winning Combination for Large Language Models
- Unified Scaling Laws for Routed Language Models
- Scaling Laws for Fine-Grained Mixture of Experts
- OLMoE: Open Mixture-of-Experts Language Models
- Mixture of Experts Explained
- Mixture-of-Experts (MoE): The Birth and Rise of Conditional Computation
- Mixture of Experts CMU course
- Mixture-of-Experts (MoE) LLMs