10-423/10-623 Generative AI

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Mixture of Experts

Matt Gormley & Henry Chai

Lecture 16

Oct. 28, 2024

Reminders

- **Homework 4: Visual Language Models**
  - Out: Fri, Oct 25
  - Due: Tue, Nov 5 at 11:59pm

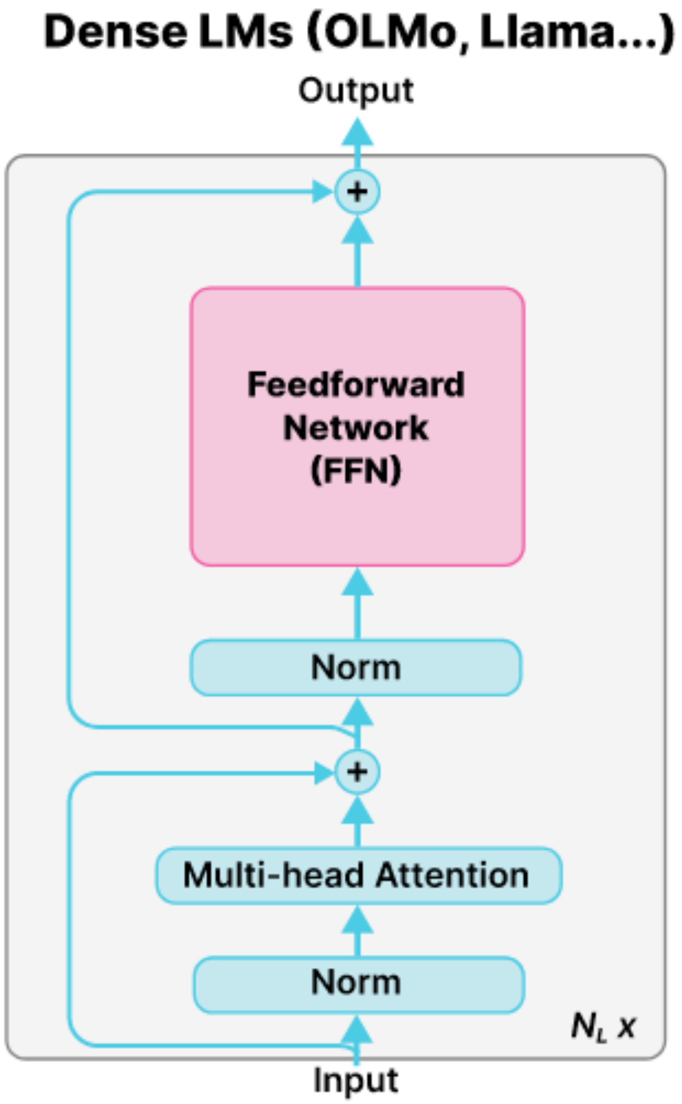
MIXTURE OF EXPERTS

Figure from http://arxiv.org/abs/2407.06204

MoE Timeline

Fig. 1. A chronological overview of several representative mixture-of-experts (MoE) models in recent years. The timeline is primarily structured according to the release dates of the models. MoE models located above the arrow are open-source, while those below the arrow are proprietary and closed-source. MoE models from various domains are marked with distinct colors: Natural Language Processing (NLP) in green, Computer Vision in yellow, Multimodal in pink, and Recommender Systems (RecSys) in cyan.

The Linear Layer in LLMs



It is common for more than half of the parameters in a Transformer LLM to reside within the feed-forward neural network layers

	GPT-3
vocab	50,257
d_model	12288
n_heads	96
n_layers	96
d_ff	49152
LayerNorm (B)	0.00
Embedding (B)	0.62
Attention (B)	57.99
Feed-forward (B)	115.97
Total (B)	**174.57**

Figure from http://arxiv.org/abs/2409.02060

parameters

$$d_{ff} = 4 \times d_{model}$$

$$\text{Attention} = (4 d_{model})^2 \times n_{layers}$$

$$\text{Embedding} = n_{vocab} \times d_{model}$$

$$\text{Feedforward} = (2(d_{model} \times d_{ff}) + d_{model} + d_{ff}) \times n_{layers}$$

$$y = G(x)_1 \cdot E + G(x)_2 E + G(x)_3 E + G(x)_4 E$$

Sparsely-Gated Mixture of Experts

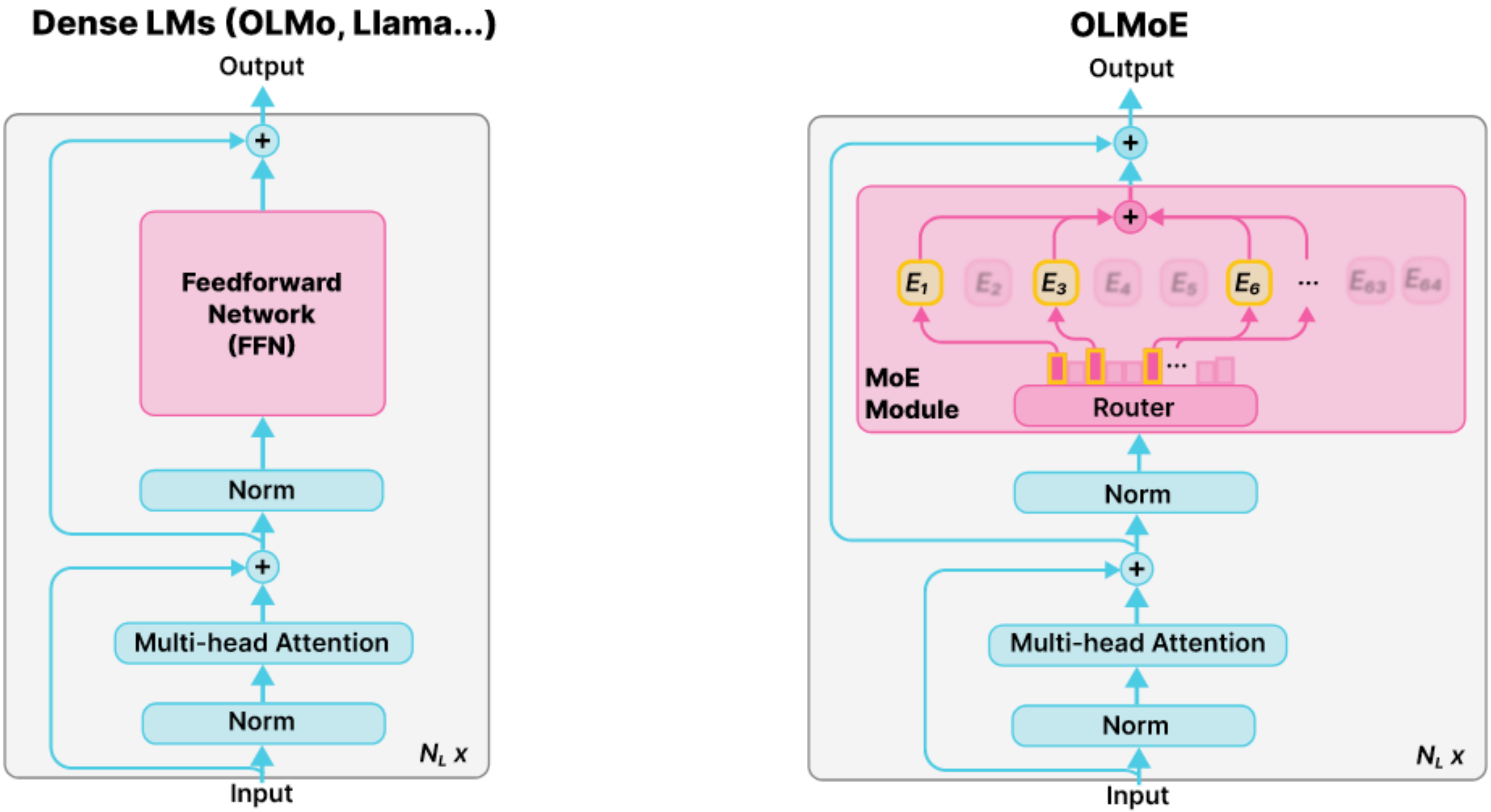


Figure from http://arxiv.org/abs/2409.02060

Dense Mixture of Experts

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

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

Figure from http://arxiv.org/abs/2401.04088

Sparsely-Gated Mixture of Experts

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

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

- Each expert is just a feed-forward network

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

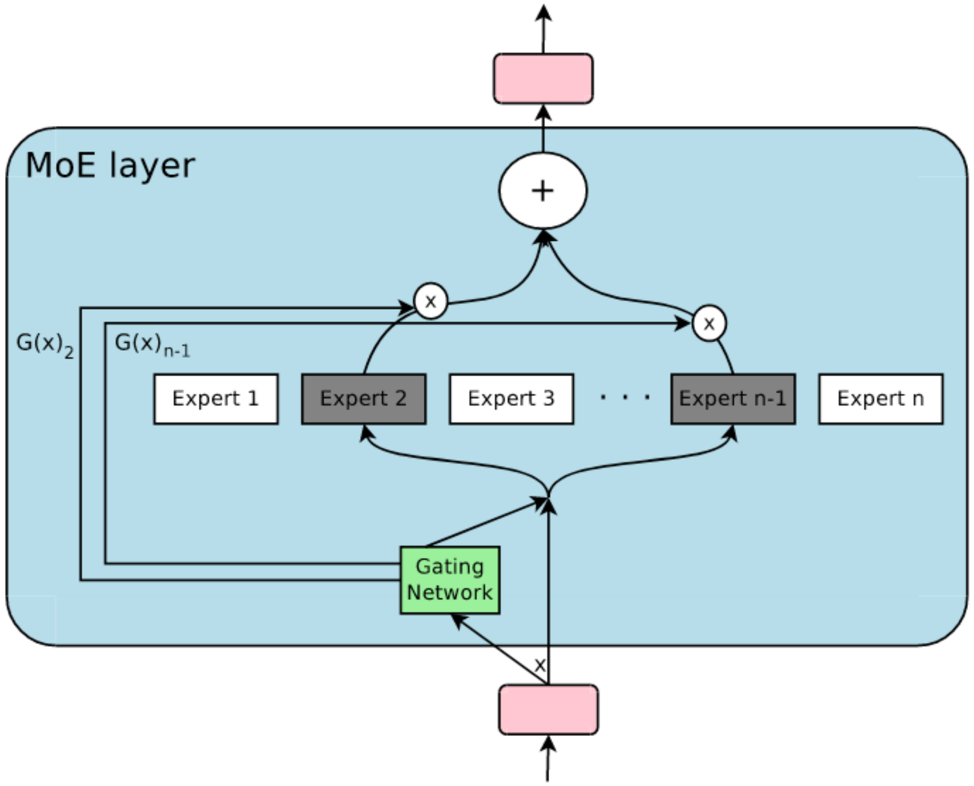


Figure from http://arxiv.org/abs/2401.04088

$$\vec{g} = \text{topk}(\vec{v}, k)$$

$$g_i = \begin{cases} v_i & \text{if } v_i \text{ is among top-k} \\ -\infty & \text{otherwise} \end{cases}$$

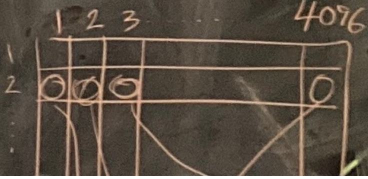
$$\vec{v} = \vec{x} \cdot W_g + b_g = [-47, 3, 8, -1]$$

$$\vec{g} = \text{topk}(\vec{v}, k=2) = [-\infty, 3, 8, -\infty]$$

$$\underline{G(x)} = \text{softmax}(\vec{g}) = \left[0, \frac{\exp(3)}{\exp(3)+\exp(8)}, \frac{\exp(8)}{\exp(3)+\exp(8)}, 0\right]$$

$$\begin{aligned} y = {} & \underbrace{G(x)_1}_{\to 0} E_1(\vec{x}) \\ & + G(x)_2 \boxed{E_2(\vec{x})} \\ & + G(x)_3 \boxed{E_3(\vec{x})} \\ & + \underbrace{G(x)_4}_{\to 0} E_4(x) \end{aligned}$$

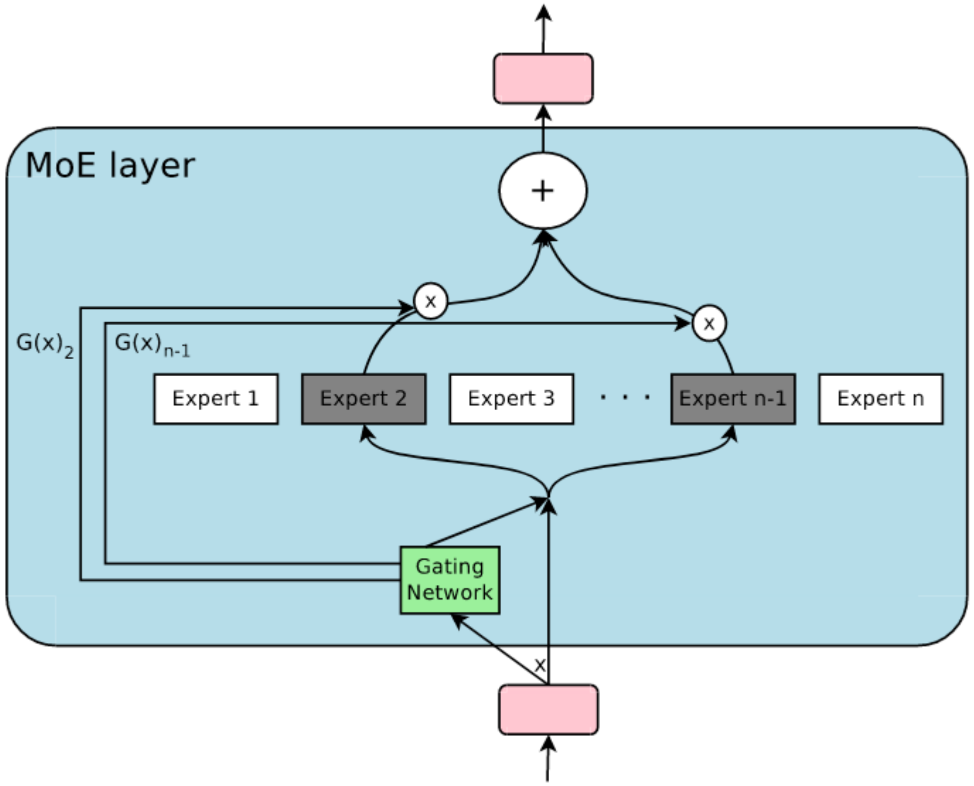
Computation of $E_i(x)$ is about $1/k$ of the usual FFN



Sparsely-Gated Mixture of Experts

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

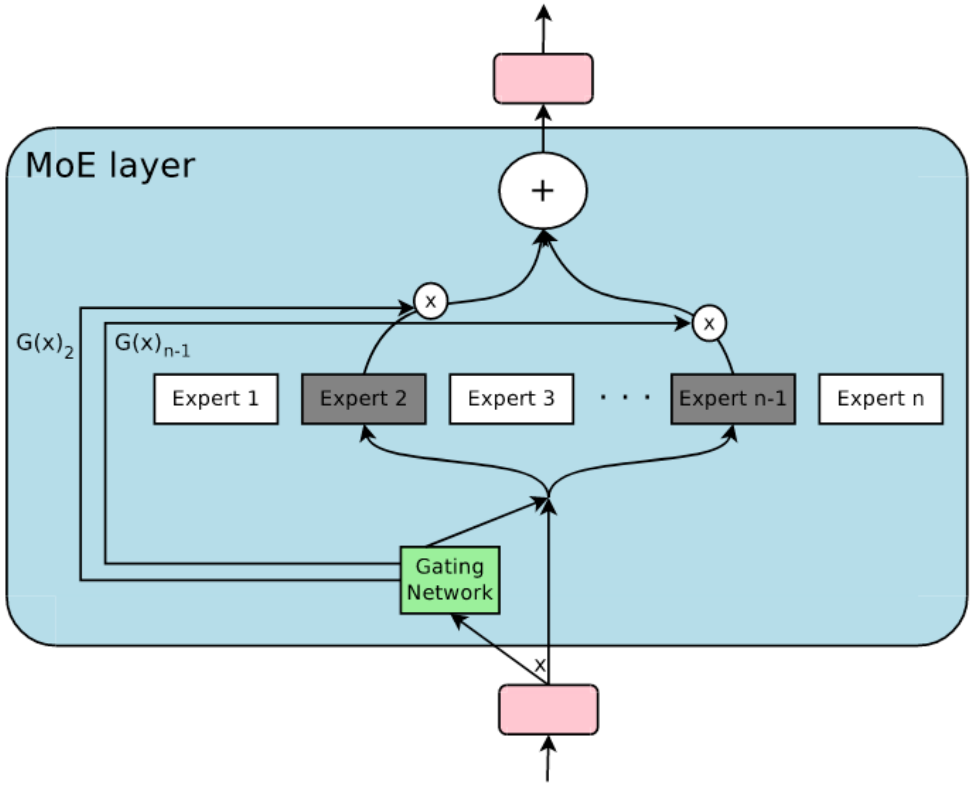
$$E_i(\mathbf{x}) = \text{FFN}_{\text{ReLU}}(\mathbf{x}) = \text{ReLU}(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$

Figure from http://arxiv.org/abs/2401.04088

Sparsely-Gated Mixture of Experts

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

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



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

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

Figure from http://arxiv.org/abs/2401.04088

Sparsely-Gated Mixture of Experts

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

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

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

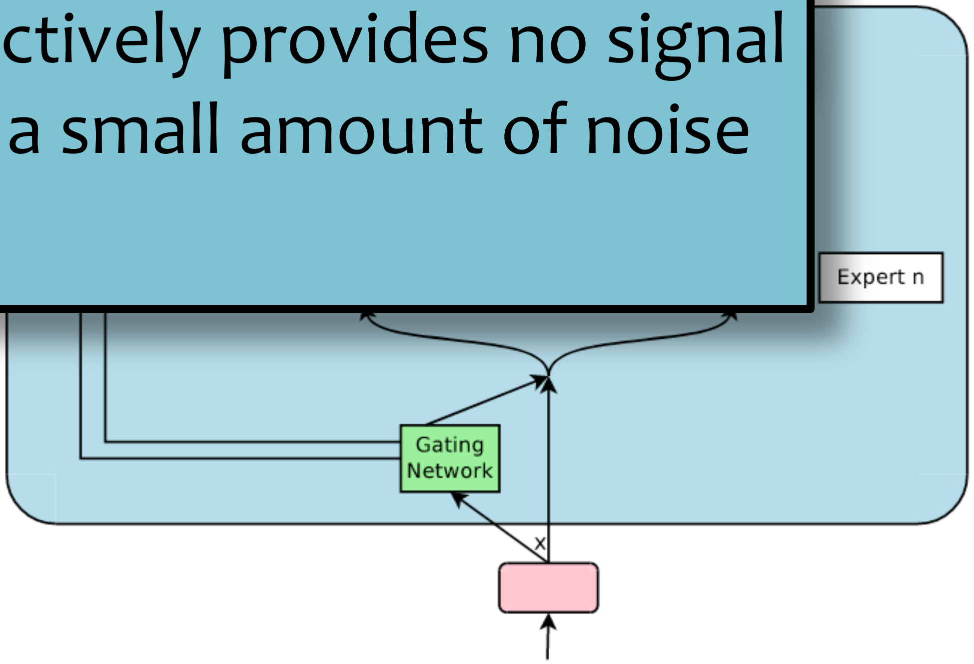


Figure from http://arxiv.org/abs/2401.04088

Balancing Expert Utilization in an MoE

- **Problem**: left unchecked, the expert gate tends to concentrate on a small number of experts that are popular early in training
- **Solution:**
  - Add two regularizers to the loss
  - *Load Balance Term:* encourages distributed load over the experts within each batch
  - *Router Z-loss:* penalizes large logits to the router to stabilize training

(this is the approach used by OlMoE, but lots of variants exist)

$$\mathcal{L} = \mathcal{L}_{CE} + \alpha \mathcal{L}_{LB} + \beta \mathcal{L}_{RZ}$$

$$\mathcal{L}_{LB} = N_e \sum_{i=1}^{N_e} f_i P_i$$

$f_i =$ fraction of tokens routed to expert i

$P_i =$ probability to E_i in current batch

$$\mathcal{L}_{RZ} = \frac{1}{N_b} \sum_{b=1}^{N_b} \left(\log \sum_{d=1}^{D} \exp(x_d^{(b)}) \right)$$

Active Parameters

- In a transformer with MoE layers, we typically choose the number k for the top-k to be rather small
  - Mixtral: $k = 2$, $N_e = 8$
  - OlMoE: $k=8$, $N_e = 64$
- The number of **active parameters** is the count of parameters that are selected for computation by the router
- For an MoE (roughly)
  - GPU memory requirement $\propto$ # of total parameters
  - FLOPS computation requirement $\propto$ # of active parameters

Mixtral vs. Llama-2

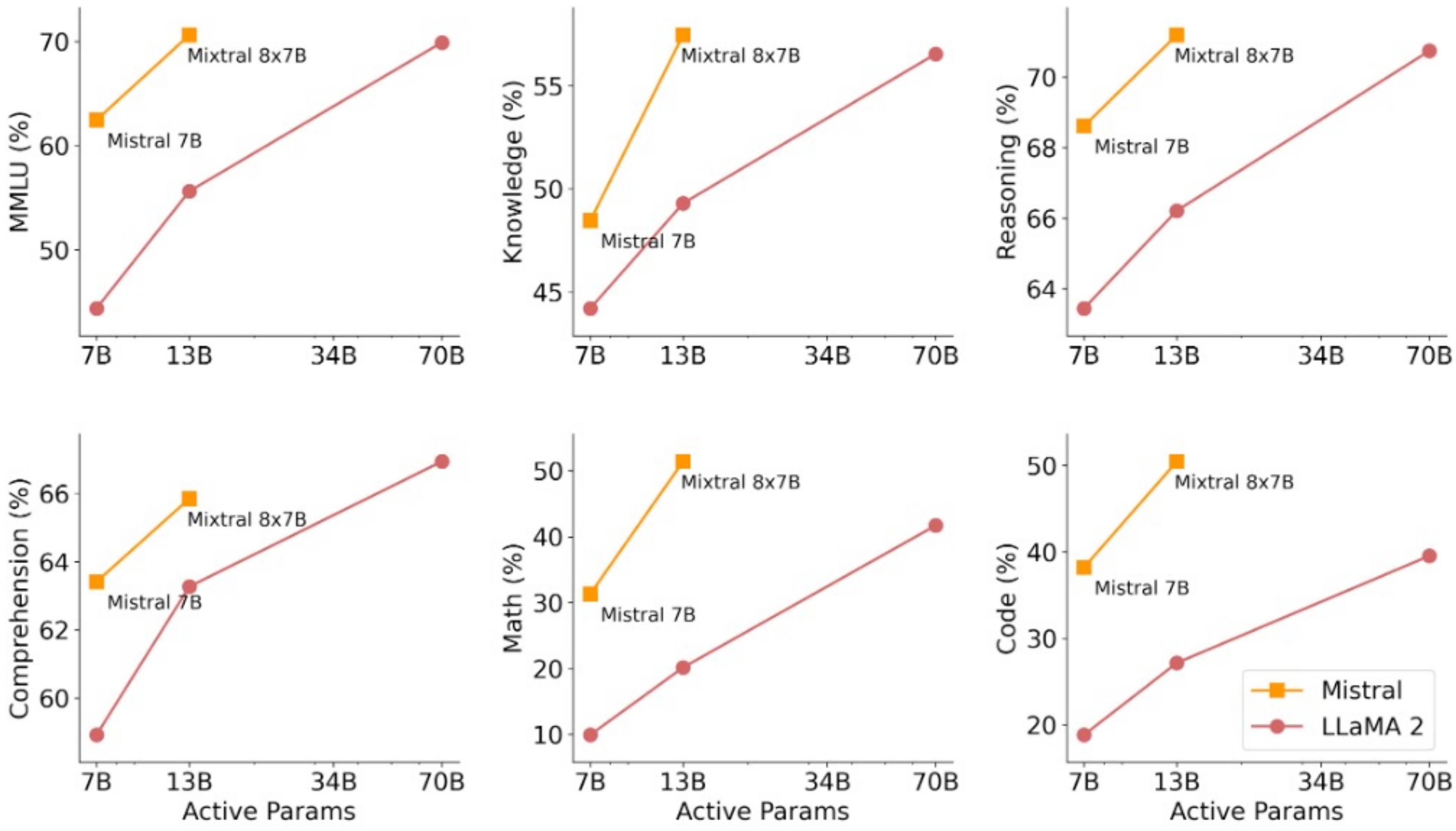


Figure from http://arxiv.org/abs/2401.04088

OlMoE Hyperparameters

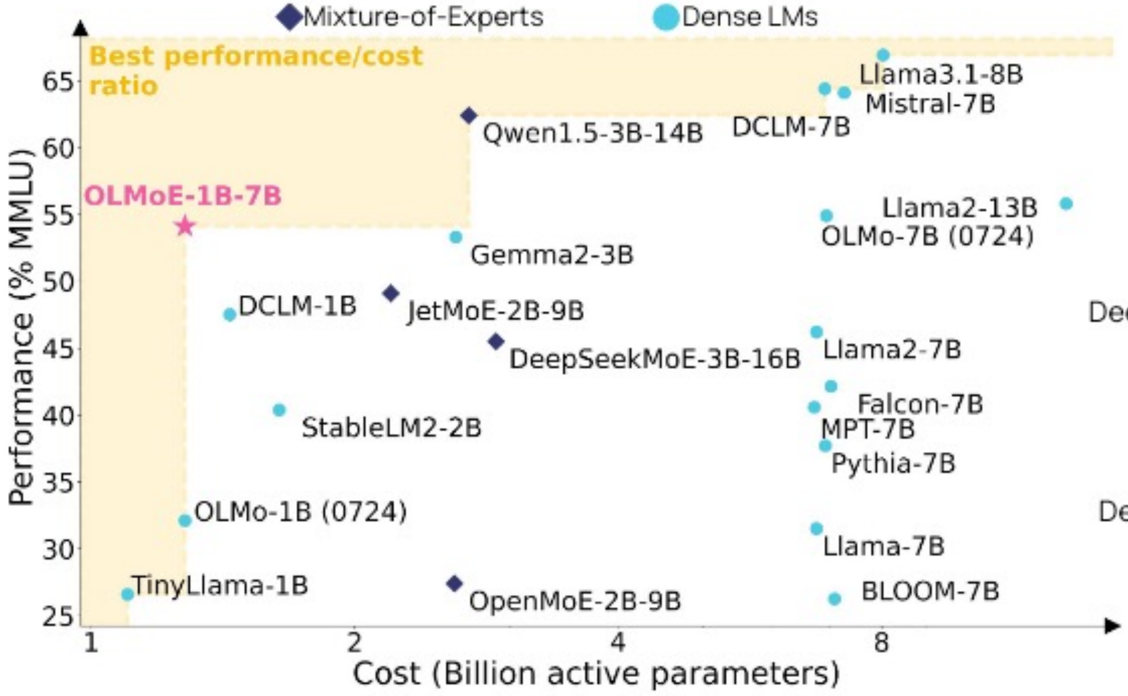
	OLMoE-1B-7B	JetMoE	OpenMoE	OLMo-1B (0724)
Dimension	2,048	2,048	2,048	2,048
Activation	SwiGLU	SwiGLU	SwiGLU	SwiGLU
FFN dimension	1,024	5,632	8,192	8,192
Vocab size	50,304	32,000	256,384	50,304
Attn heads	16	16	24	16
Num layers	16	24	32	16
Layer norm type	RMSNorm	RMSNorm	RMSNorm	non-parametric
Layer norm eps	1.0E-05	1.0E-05	1.0E-06	1.0E-05
QK-Norm	yes	no	no	no
Pos emb.	RoPE	RoPE	RoPE	RoPE
RoPE θ	10,000	10,000	10,000	10,000
Attention variant	full	MoA	full	full
Biases	-	MLP & Attn	-	-
Weight tying	no	yes	no	no
Init dist	trunc normal	?	?	normal
Init std	0.02	0.02	varies	varies
Init trunc	3×std	-	-	-
MoE layers	Every	Every	Every 6th	-
MoE layer type	dMoE	dMoE	ST-MoE	-
# Experts	64	8	32	1
# Activated	8	2	2	1
# Vocab params	103M	66M	525M	103M
# Active params	1.3B	2.2B	2.6B	1.3B
# Total params	6.9B	8.5B	8.7B	1.3B

Sequence length	4,096	4,096	2,048	4,096
Batch size (samples)	1,024	1,024	2,048	512
Batch size (tokens)	~4M	~4M	~4M	~2M
warmup steps	2,000	2,500	10,000	2,000
peak LR	4.0E-04	5.0E-04	0.01	4.0E-04
minimum LR	5.0E-04	5.0E-05	-	5.0E-05
optimizer	AdamW	AdamW	Adafactor	AdamW
weight decay	0.1	0.1	0.0	0.1
beta1	0.9	?	0.9	0.9
beta2	0.95	?	-	0.95
AdamW epsilon	1.0E-08	?	-	1.0E-05
LR schedule	cosine	WSD	Inv Sq Root	cosine
gradient clipping	global 1.0	global 1.0	global 1.0	global 1.0
gradient reduce dtype	FP32	?	?	FP32
optimizer state dtype	FP32	?	?	FP32
LBL weight	0.01	0.01	0.01	-
Router z-loss weight	0.001	0.001	0.0001	-
Pretraining tokens	5,033B	1,000B	1,100B	2,000B
Annealing tokens	100B	250B	-	50B
Annealing schedule	linear	-	-	linear
Annealing min LR	0	-	-	0

Table 10: **Pretraining hyperparameters of OLMoE-1B-7B and comparable models trained from scratch.** We highlight rows where **OLMoE-1B-7B** differs from OLMo-1B. Active params include vocab params. "?" = undisclosed settings, FFN = feed-forward network, Attn = Attention, LR = learning rate, WSD = Weight-Stable-Decay [73], LBL = load balancing loss, Inv Sq Root = Inverse Square Root decay [153], trunc = truncation, std = standard deviation, "varies" = stds that are layer or weight-dependent.

Performance vs. Cost

MoEs provide a nice tradeoff between performance and FLOPS cost



How open are open MoEs?

Name	Model	Data	Code	Logs	#ckpts
Grok-86B-314B	✔	✘	✘	✘	1
Mixtral-39B-141B	✔	✘	✘	✘	1
DBRX-36B-132B	!!	✘	✘	✘	1
Skywork-22B-146B	!!	✘	✘	✘	1
DeepSeekV2-21B-236B	!!	✘	✘	✘	1
Arctic-17B-480B	✔	!!	✘	✘	1
Qwen2-14B-57B	✔	✘	✘	✘	1
Mixtral-13B-47B	✔	✘	✘	✘	1
Jamba-12B-52B	✔	✘	✘	✘	1
DeepSeekMoE-3B-16B	!!	✘	✘	✘	1
Qwen1.5-3B-14B	!!	✘	✘	✘	1
OpenMoE-3B-9B	✔	✔	✔	✘	6
JetMoE-2B-9B	✔	!!	!!	✘	1
OLMoE-1B-7B	✔	✔	✔	✔	**244**

Figure 1: **Performance, cost, and degree of openness of open MoE and dense LMs.** Model names contain rounded parameter counts: `model-active-total` for MoEs and `model-total` for dense LMs. `#ckpts` is the number of intermediate checkpoints available. We highlight MMLU as a summary of overall performance; see §3 for more results. **OLMoE-1B-7B** performs best among models with similar active parameter counts and is the most open MoE.

Figure from http://arxiv.org/abs/2409.02060

Performance vs. Cost

MoEs provide a nice tradeoff between performance and FLOPS cost

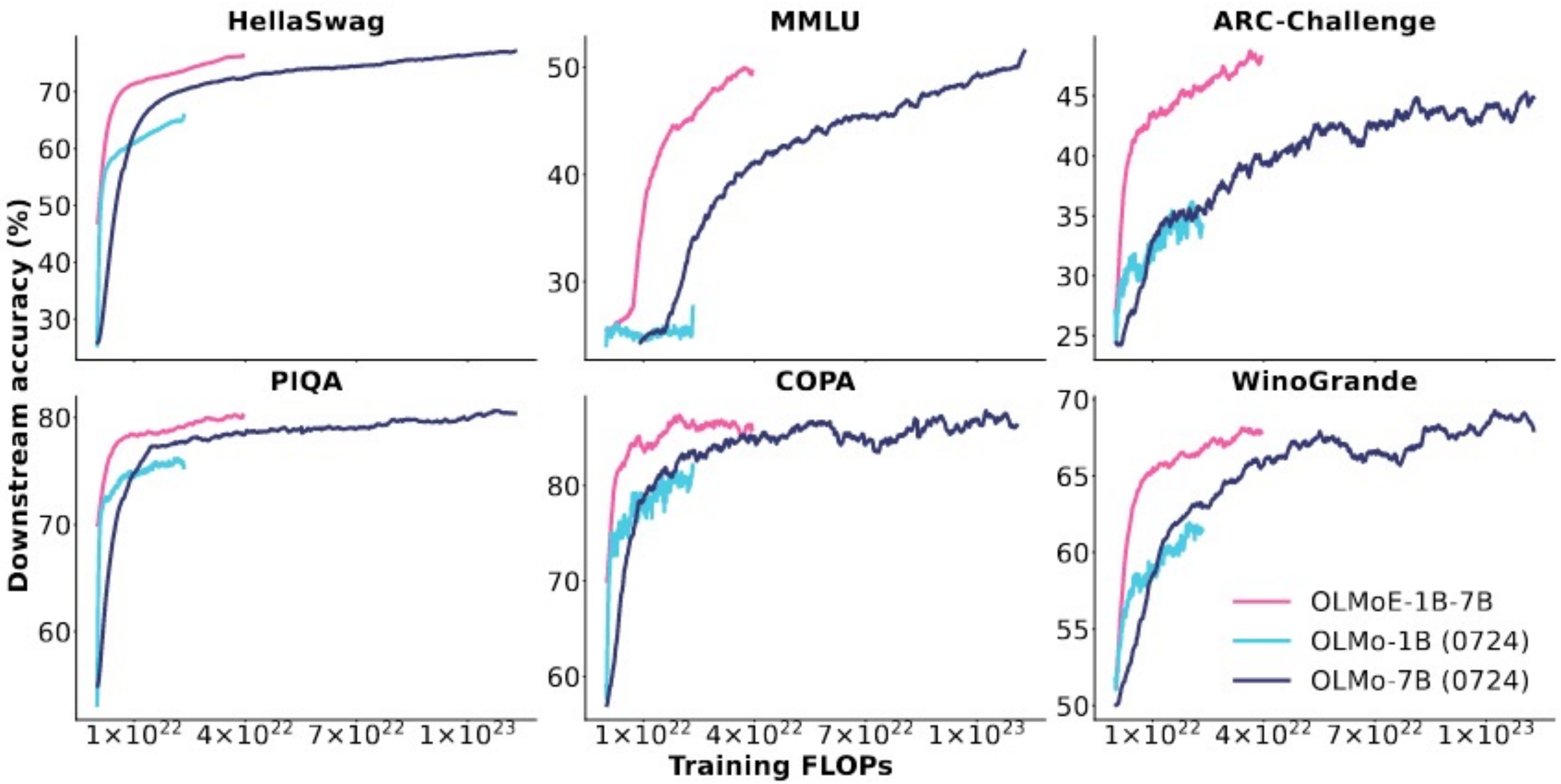
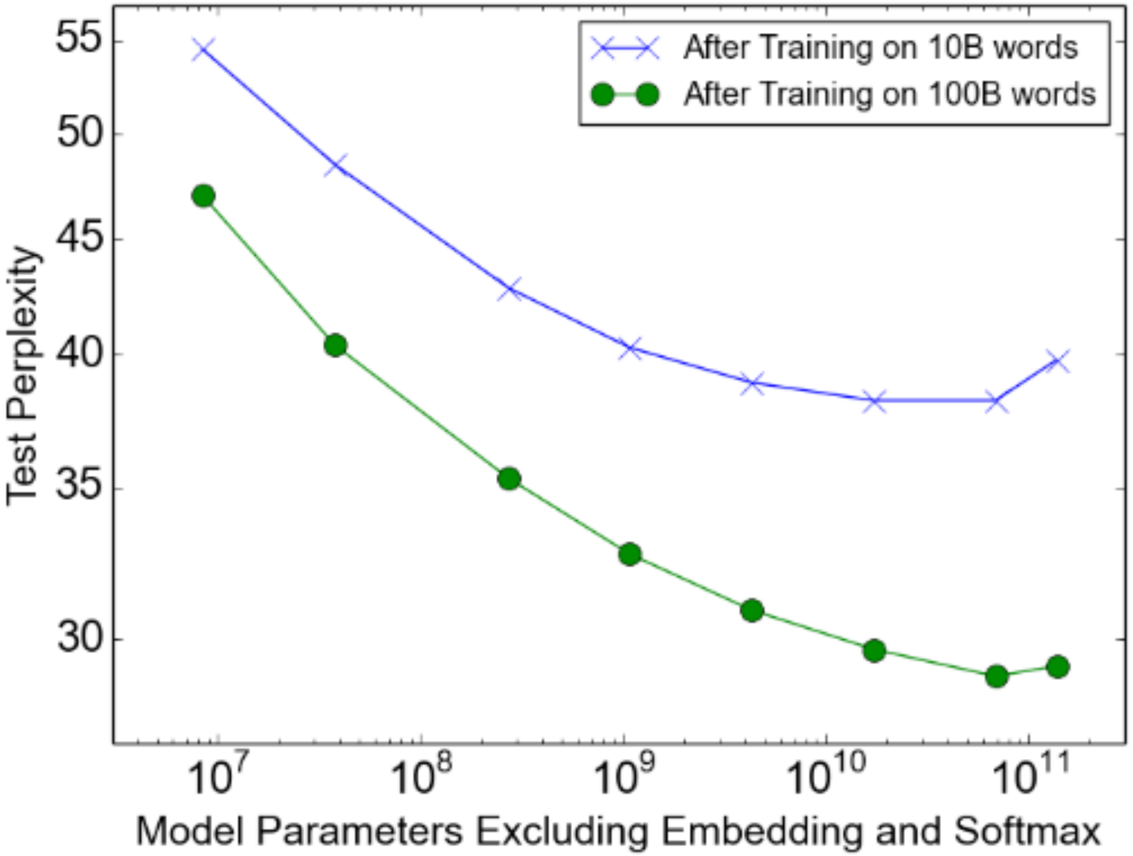


Figure 3: **Evaluation of OLMoE-1B-7B and the current best OLMo models during pretraining.** OLMoE-1B-7B differs from the OLMo models in its MoE architecture, several training hyperparameters, and its training dataset, see §2. A version of this plot with tokens as the x-axis and markers where annealing starts is in Appendix E. More results, logs, and configurations: `https://wandb.ai/ai2-llm/olmoe/reports/Plot-OLMoE-1B-7B-vs-OLMo-7B-vs-OLMo-1B--Vmlldzo4OTcyMjEz`

Figure from http://arxiv.org/abs/2409.02060

How many experts to choose?

Early work with MoEs in LSTM-LMs favored a very large number of experts



32, 256, 1024, 4096, 16384, 65536, and 131072 experts.

up to 137 billion parameters in the MoE layer.

How many experts to choose?

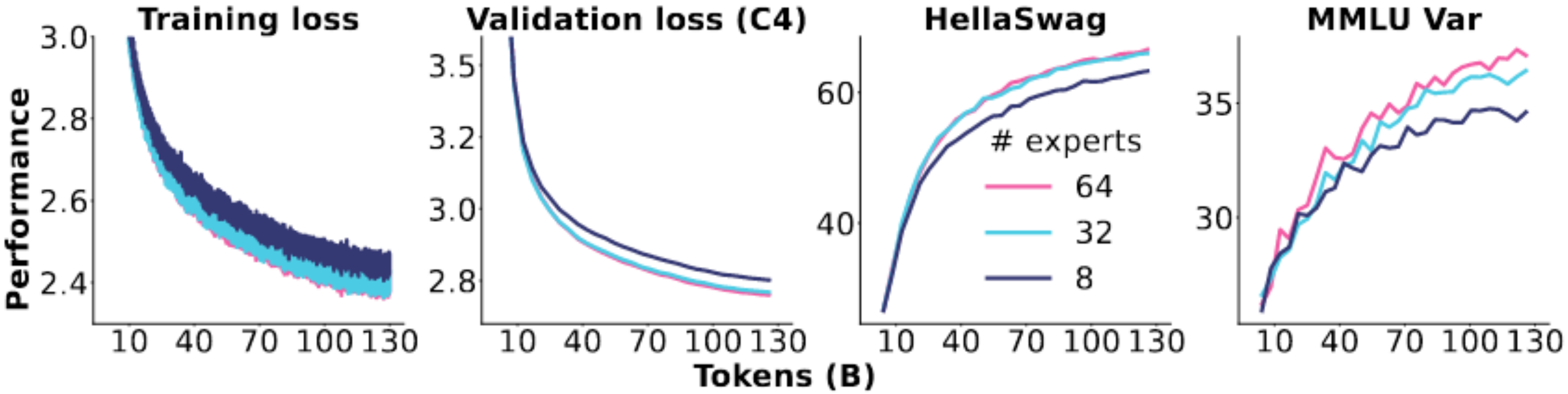


Figure 5: **Expert granularity.** We vary the number of experts in tandem with the FFN dimension to ensure that active and total parameters and thus compute cost remain the same. For example, for 64 experts, the FFN dimension is 1,024 and 8 experts are activated, while for 32 experts it is 2,048 with 4 activated experts. More results, logs, and configurations: https://wandb.ai/ai2-llm/olmoe/reports/Plot-Granularity--Vmlldzo4OTIxOTE4

More recent work on Transformer LMs has favored a comparatively small number of experts

Figure from http://arxiv.org/abs/2401.04088

MoEs and Parallelism

MoEs and Parallelism

Scaling Laws for Routed LMs

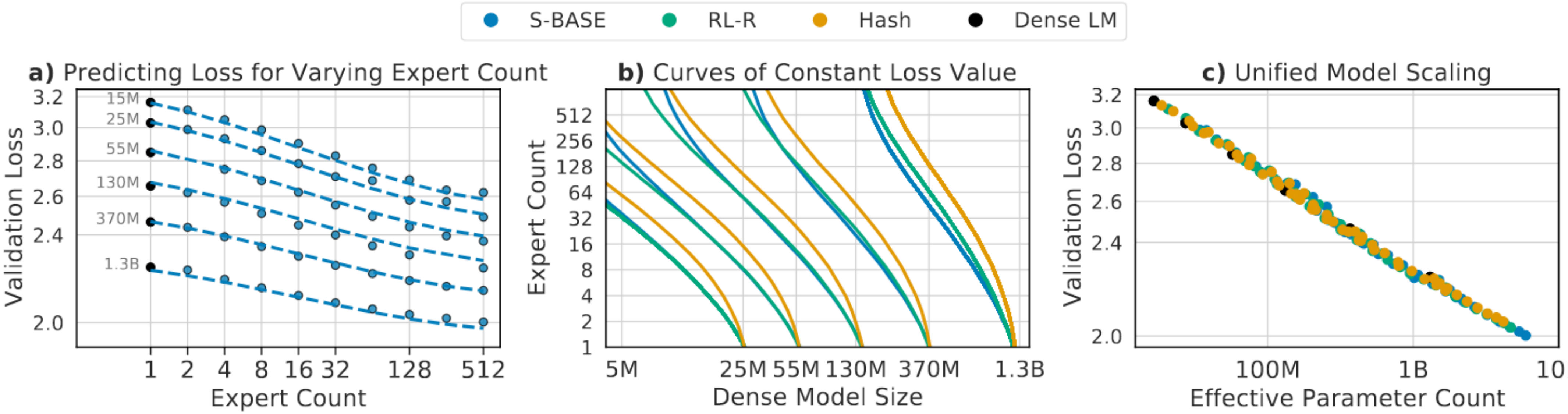


Figure 1. **(a)** The performance achieved by Routing Networks when varying the number of experts for a fixed dense model size is described by a bilinear function (Eq. 1), **(b)** whose level curves indicate how to trade model size with expert count to maintain a fixed performance, **(c)** and which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

Figure from https://proceedings.mlr.press/v162/clark22a.html