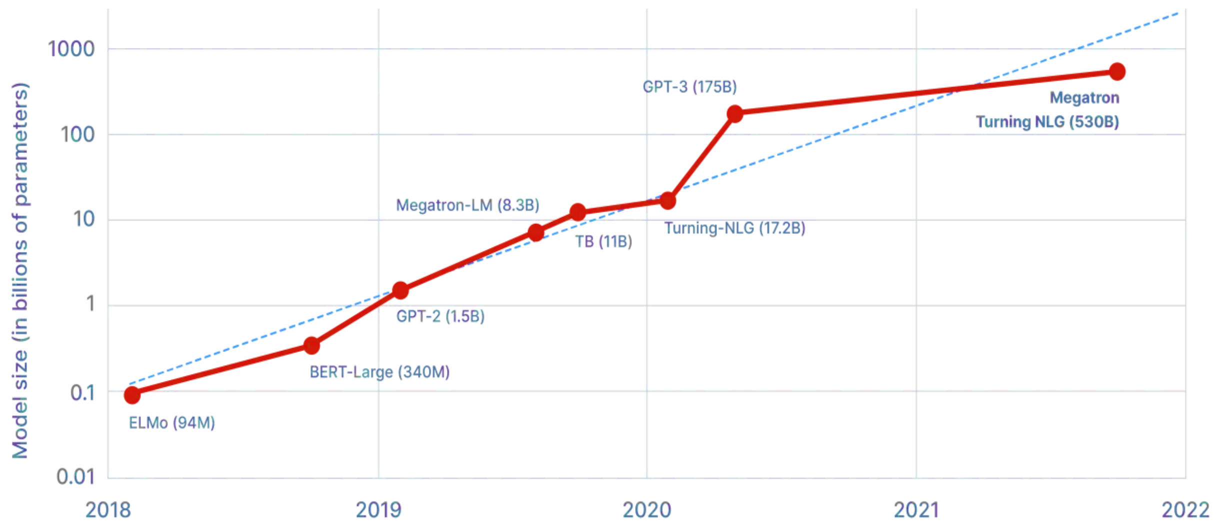
Finding the Right Optimization for Mixture-of-Experts

Zeynep Tandogan
Supervisor: Alexander Hägele – Martin Jaggi

EPFL MLO

LLMs Are Transforming the World

➢ The Need for Scale and Its Growing Costs



Language models are huge and getting bigger — but why?

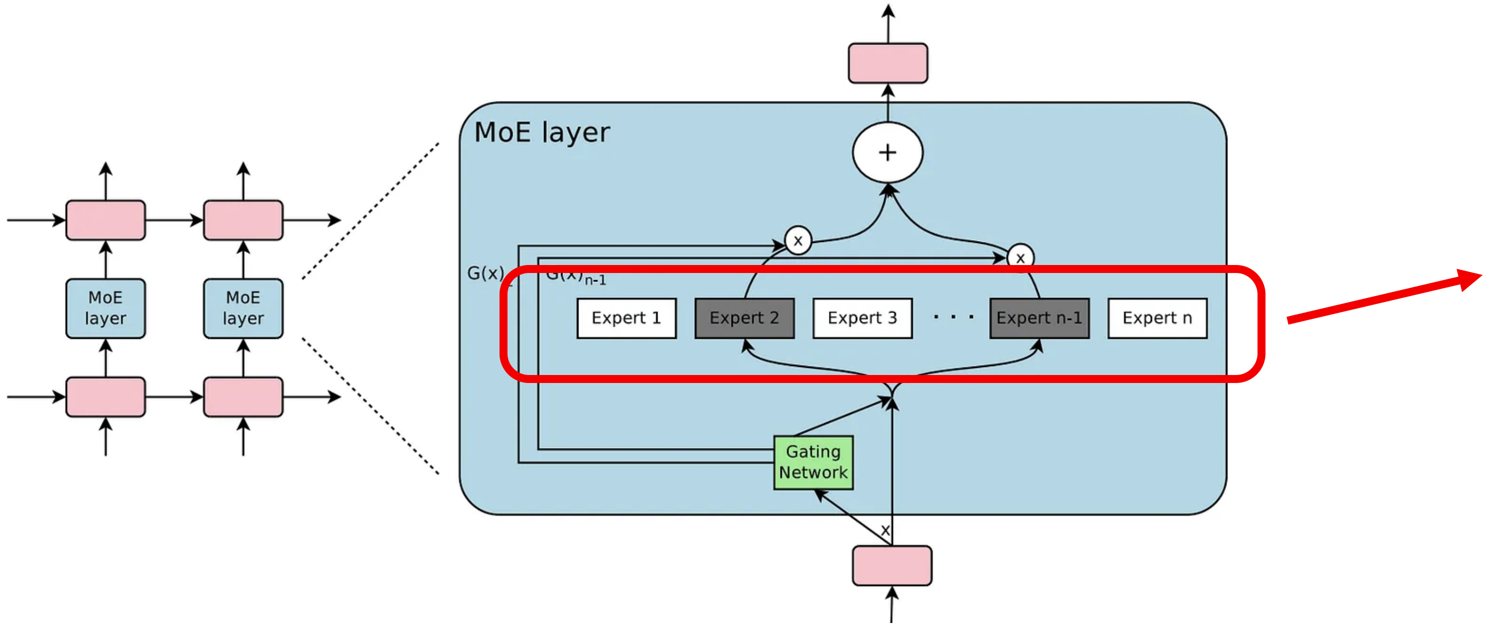
- *Bigger models understand language better, solve harder tasks.*
- *Emergent abilities (reasoning, coding, multi-step logic) appear with scale*

What about challenges?

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Mixture of Experts: Massive AI with Minimal Compute

➢ Can We Have Bigger Models Without Paying the Full Price?

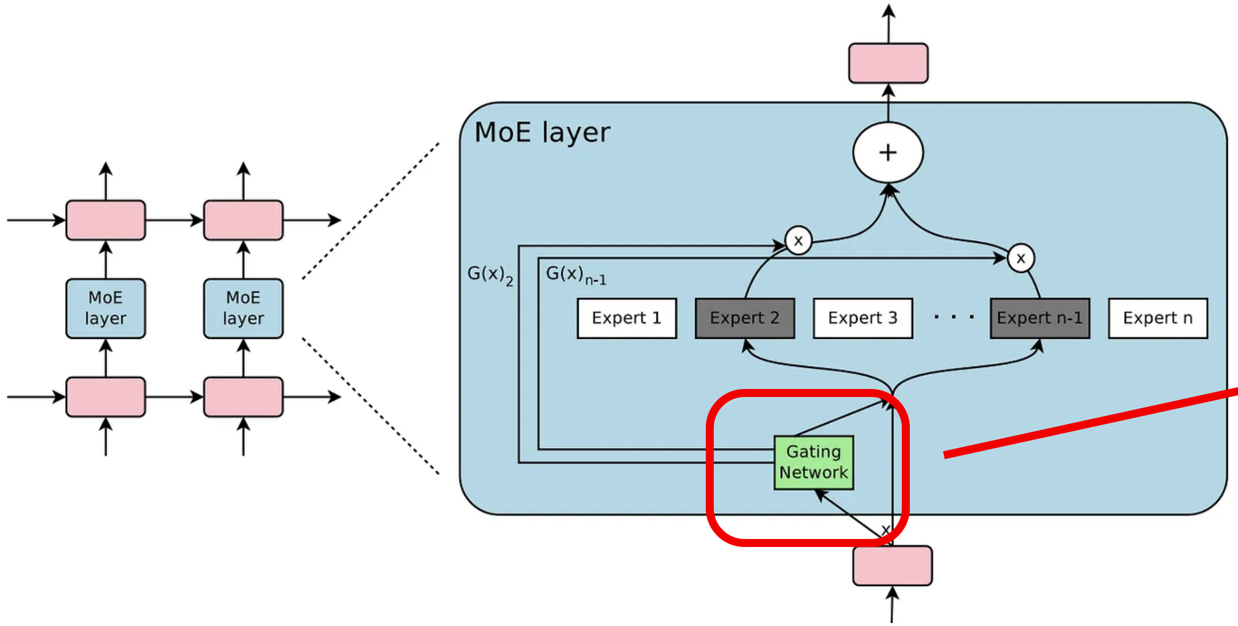


Two main components:

1) Expert Layers

Mixture of Experts: Massive AI with Minimal Compute

➢ Can We Have Bigger Models Without Paying the Full Price?



Two main components:

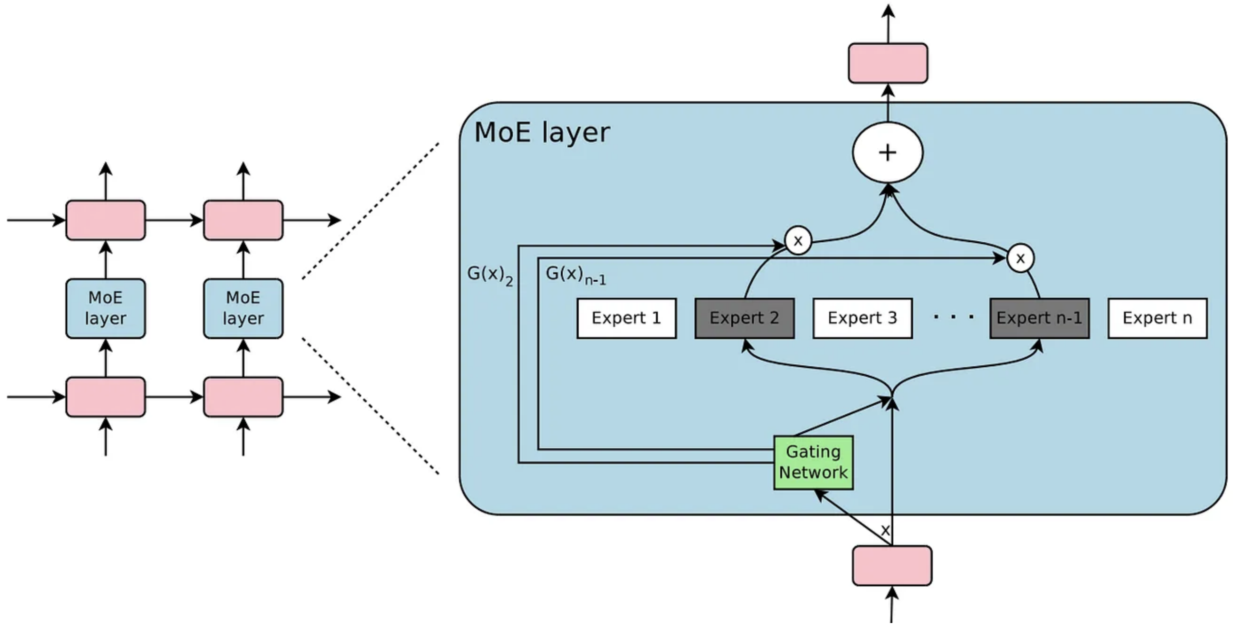
1) Expert Layers

2) Gating Network (Router)

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Mixture of Experts: Massive AI with Minimal Compute

➢ Can We Have Bigger Models Without Paying the Full Price?



Two main components:

1) Expert Layers

2) Gating Network (Router)

- **Sparse Activation:** Only *k* experts fire per token
- **Efficient Scaling:** Grow to billions or trillions of parameters without a matching rise in compute budget
- **Specialized Expertise:** Experts specialize in niche skills, boosting accuracy on diverse tasks

MoE Trade-Offs: Complexity Under the Hood

➢ What New Bottlenecks Come with Sparse Scaling?

- **Uneven data exposure** per expert - under-/over-utilization
- **Fluctuating expert batches**
- **Loss of expressivity** in seldom-activated experts
- **Trade-off** between **load balancing** and **overall performance**

OUR GOAL:
Ablate and understand optimization dynamics for MoE models to find better recipes for large scale training of MoEs.

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Experimental Setup

Model

- LLaMA-style decoder-only Transformer

Data

- **Fineweb Edu** sample 10Bt
- **Training tokens:** 9.95 B (≈ 9 949 090 040)
- **Validation tokens:** 4.90 M (4 899 304)

Core Hyperparameters

Component	Value
Layers	24 (decoder-only)
Hidden size	768
FFN dim (dense)	2048
Experts per layer	8 (Top-2 routing)
Expert FFN dim	2048
Sequence length	512
Training steps	50,000
Batch size	40 sequences (20 480 tokens/update)
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$)
weight decay	0.1
LR schedule	Cosine decay with 300-step warmup
Gradient clipping	1.0

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Exploring Optimization Strategies in Mixture-of-Experts Training

1. Learning Rate Strategies for Experts vs. Non-Experts
2. Effect of Auxiliary Loss Coefficients and Aux-Free-Loss Reproduction
3. Integration and Evaluation of the Shampoo Optimizer
4. Load Based Learning Rate Updates for Experts
5. Extending Experiments with Megatron LM

Learning Rate Strategies for Experts vs. Non-Experts

➢ **Why experts may need their own LRs?**

$$p = \frac{k}{E} \implies B_{\text{expert}} = p \times B_{\text{global}}$$

- **Experts see a smaller batch.**
  In Top-k routing (E experts, k per token), each expert is active only with probability

Signal-to-Noise Ratio of the Gradient

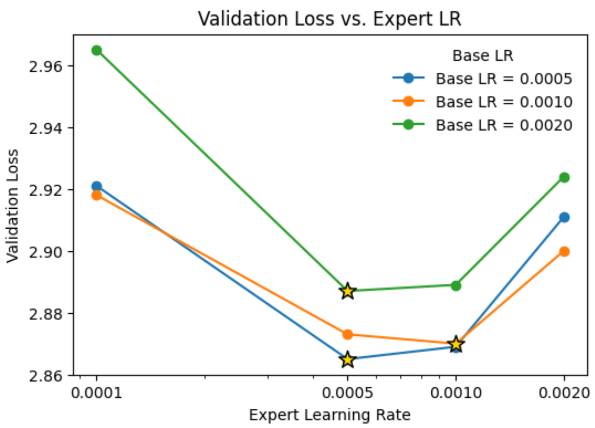
$$\begin{aligned}
\mathbb{E}[\hat{g}] &= \nabla L, && \text{(true gradient / signal)} \\
\|\mathbb{E}[\hat{g}]\| &= \|\nabla L\|, && \text{(signal magnitude)} \\
\mathrm{Var}(\hat{g}) &= \frac{\sigma^2}{B}, && \text{(variance } 1/B\text{)} \\
\mathrm{Std}(\hat{g}) &= \frac{\sigma}{\sqrt{B}}, && \text{(noise magnitude)} \\
\mathrm{SNR} &= \frac{\|\mathbb{E}[\hat{g}]\|}{\mathrm{Std}(\hat{g})} = \frac{\|\nabla L\|}{\sigma/\sqrt{B}} \propto \sqrt{B}. && \text{(signal-to-noise ratio)}
\end{aligned}$$

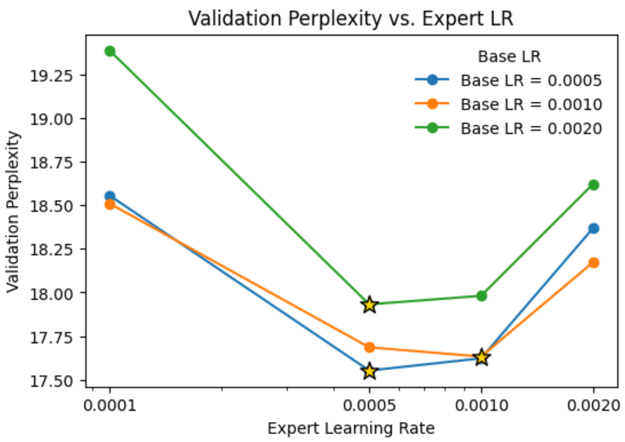
- **Each expert's effective batch is smaller**, its SNR is lower—so its gradient updates are **much noisier**.

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Learning Rate Strategies for Experts vs. Non-Experts

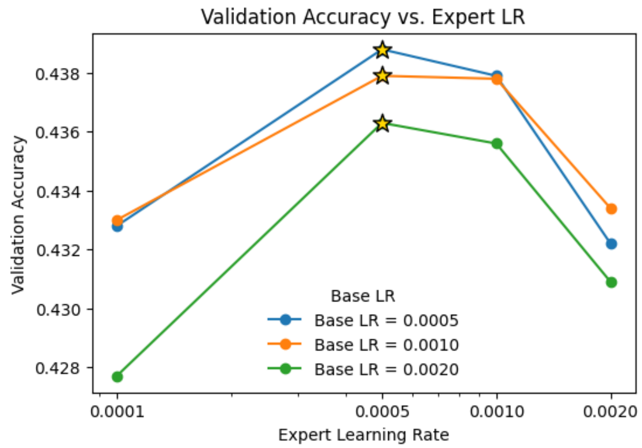
Component	Learning Rates
Non-Experts	0.0005, 0.001, 0.002
Experts	0.0001, 0.0005, 0.001, 0.002





Goal:

Investigate the impact of using different learning rates for expert (MLP layers) and non-expert parameters in MoE models.

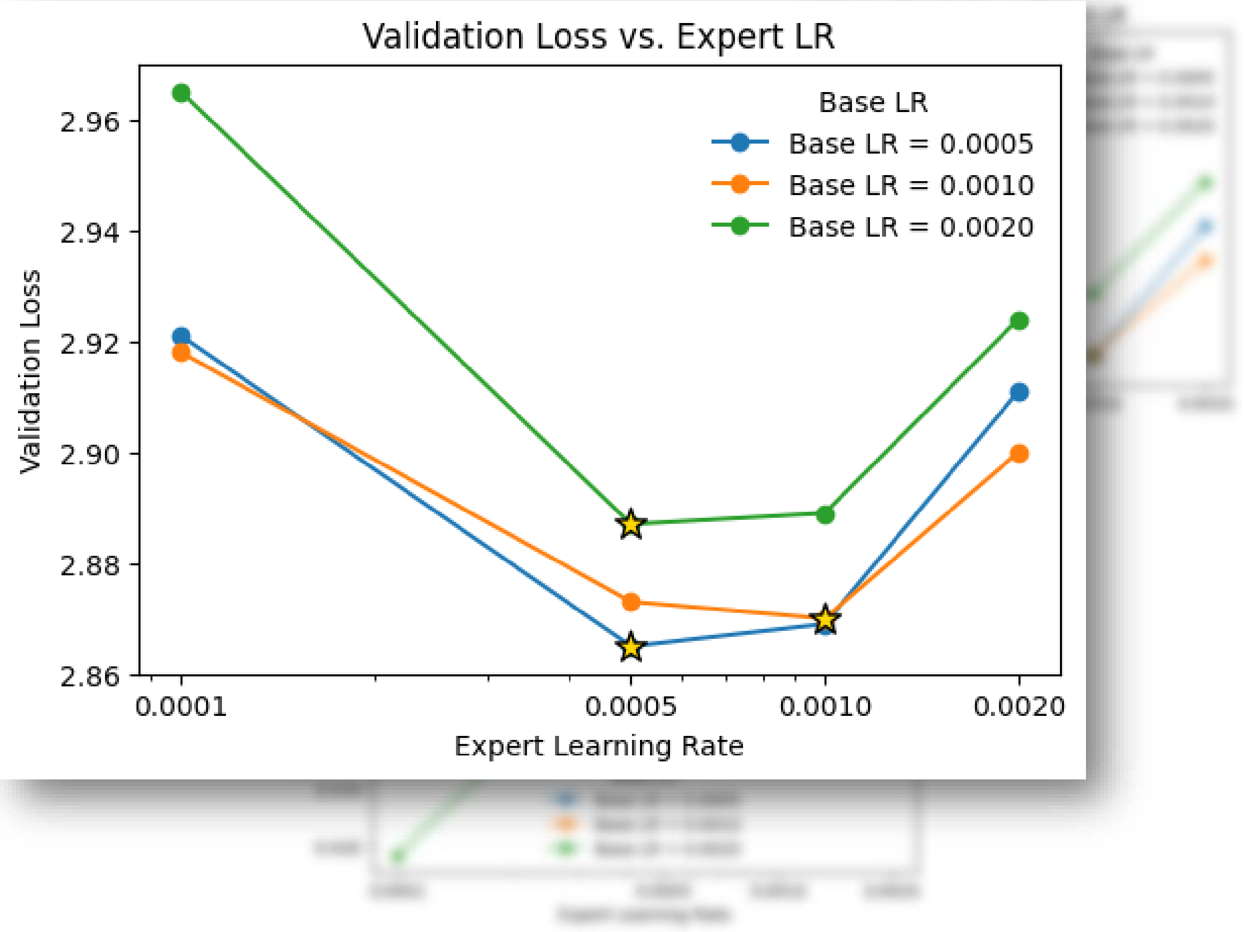


Zeynep Tandogan | zeynep.tandogan@epfl.ch

1 Learning Rate Strategies for Experts vs. Non-Experts

Component	Learning Rates
Non-Experts	0.0005, 0.001, 0.002
Experts	0.0001, 0.0005, 0.001, 0.002

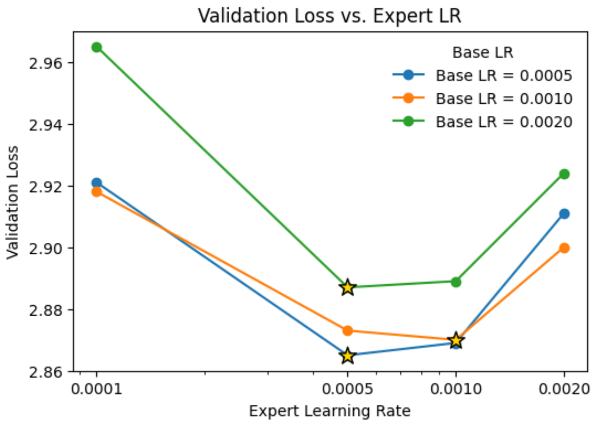
Goal:
Investigate the impact of using different learning rates for expert (MLP layers) and non-expert parameters in MoE models.

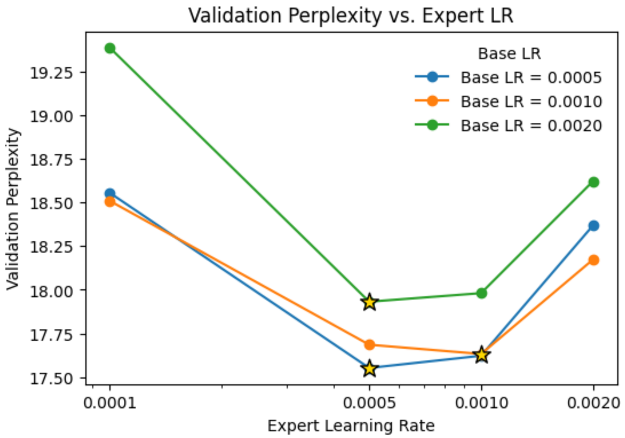


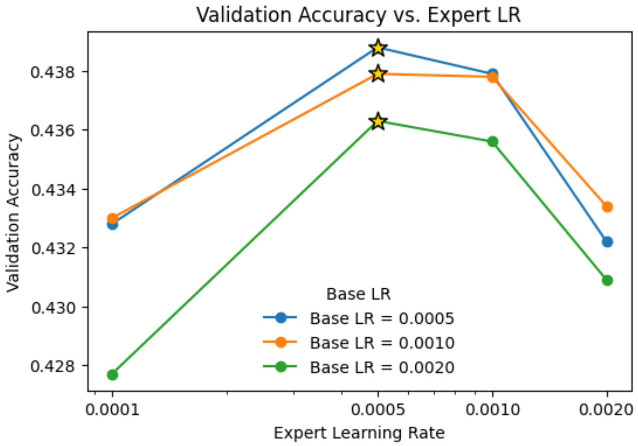
EPFL MLO

1 Learning Rate Strategies for Experts vs. Non-Experts

Component	Learning Rates
Non-Experts	0.0005, 0.001, 0.002
Experts	0.0001, 0.0005, 0.001, 0.002





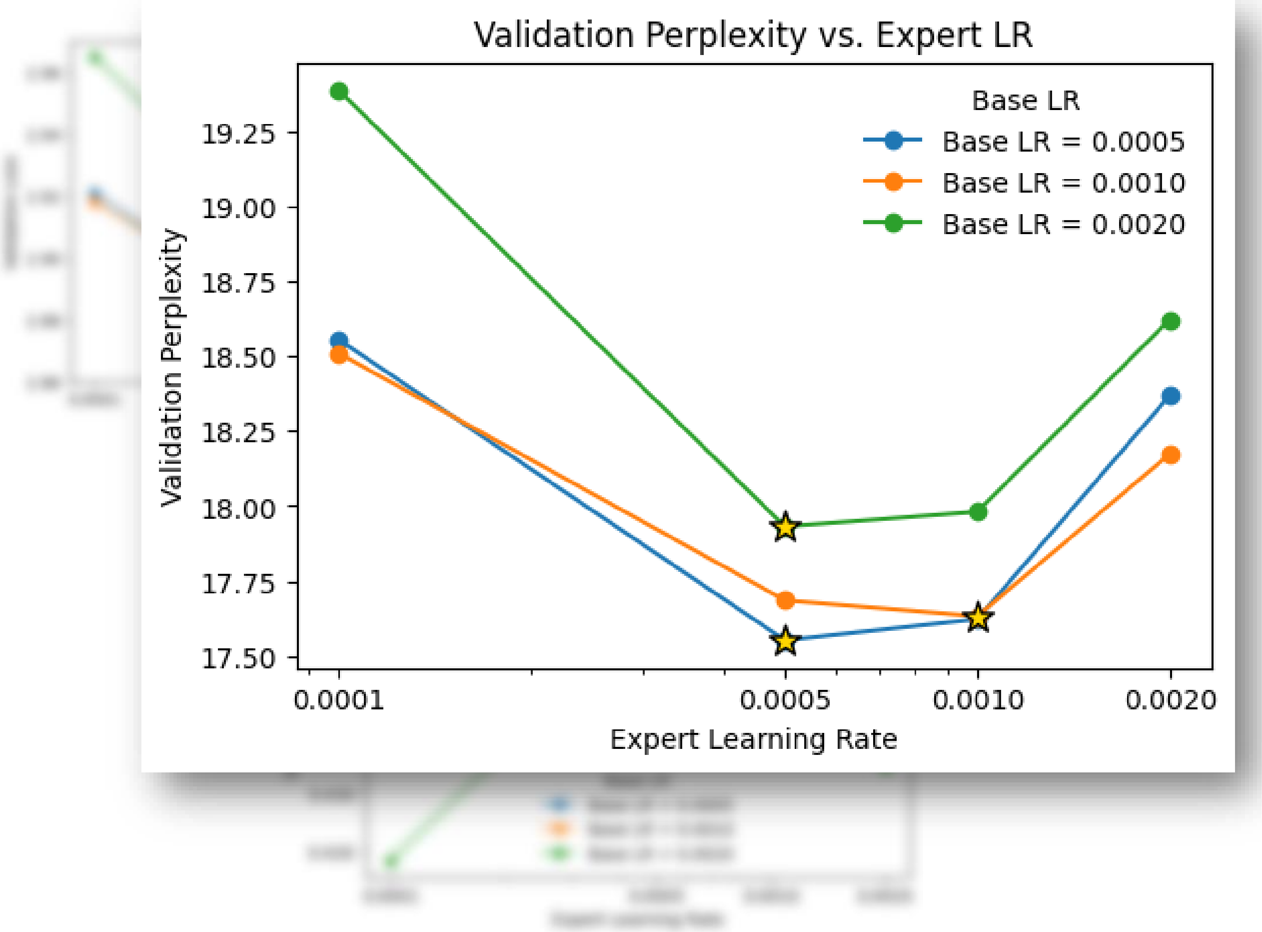


Goal:

Investigate the impact of using different learning rates for expert (MLP layers) and non-expert parameters in MoE models.

Learning Rate Strategies for Experts vs. Non-Experts

Component	Learning Rates
Non-Experts	0.0005, 0.001, 0.002
Experts	0.0001, 0.0005, 0.001, 0.002

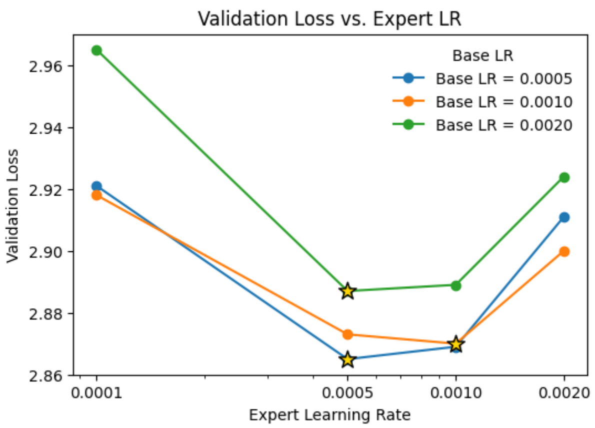


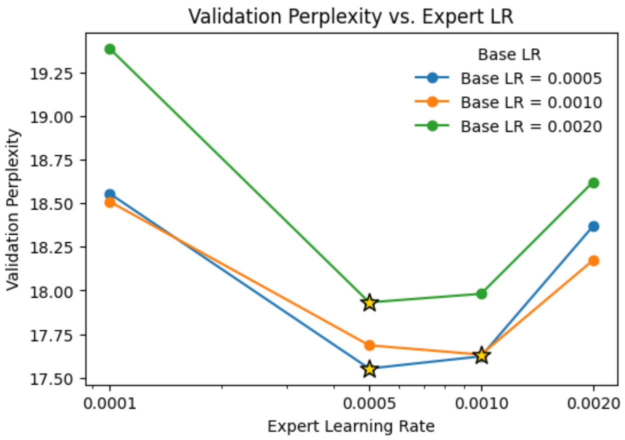
Goal:

Investigate the impact of using different learning rates for expert (MLP layers) and non-expert parameters in MoE models.

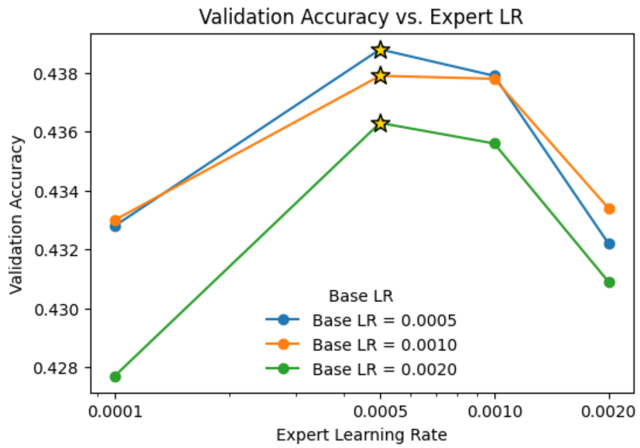
Learning Rate Strategies for Experts vs. Non-Experts

Component	Learning Rates
Non-Experts	0.0005, 0.001, 0.002
Experts	0.0001, 0.0005, 0.001, 0.002





Goal:
Investigate the impact of using different learning rates for expert (MLP layers) and non-expert parameters in MoE models.



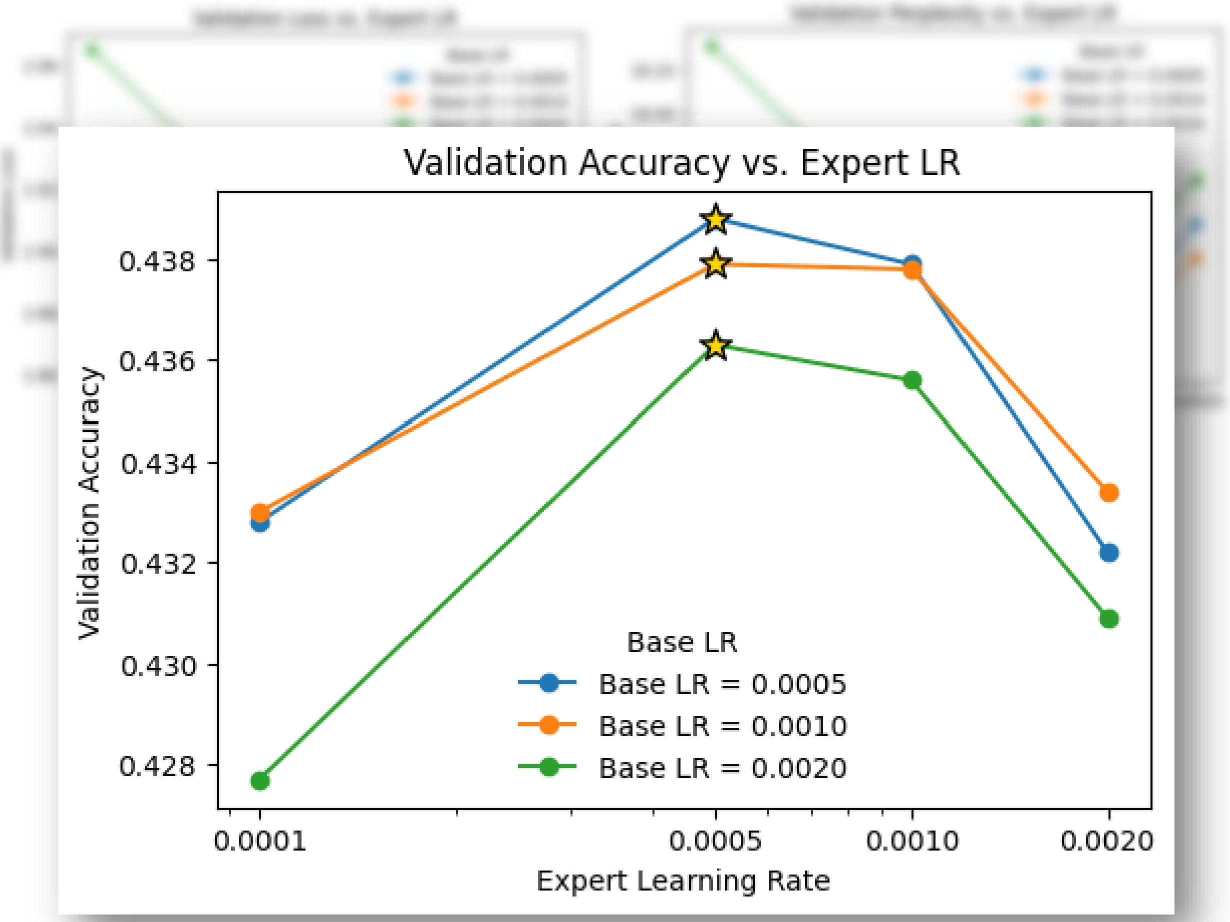
Zeynep Tandogan | zeynep.tandogan@epfl.ch

1 Learning Rate Strategies for Experts vs. Non-Experts

Component	Learning Rates
Non-Experts	0.0005, 0.001, 0.002
Experts	0.0001, 0.0005, 0.001, 0.002

Goal:

Investigate the impact of using different learning rates for expert (MLP layers) and non-expert parameters in MoE models.



1 Learning Rate Strategies for Experts vs. Non-Experts

➢ What should be different in terms of loss in MoEs?

- **Problem: Expert Imbalance**

  Some experts get most tokens; others are barely used.

- **Solution: Add auxiliary loss**

  Prevent expert collapse: stop the router from sending almost all tokens to a few experts

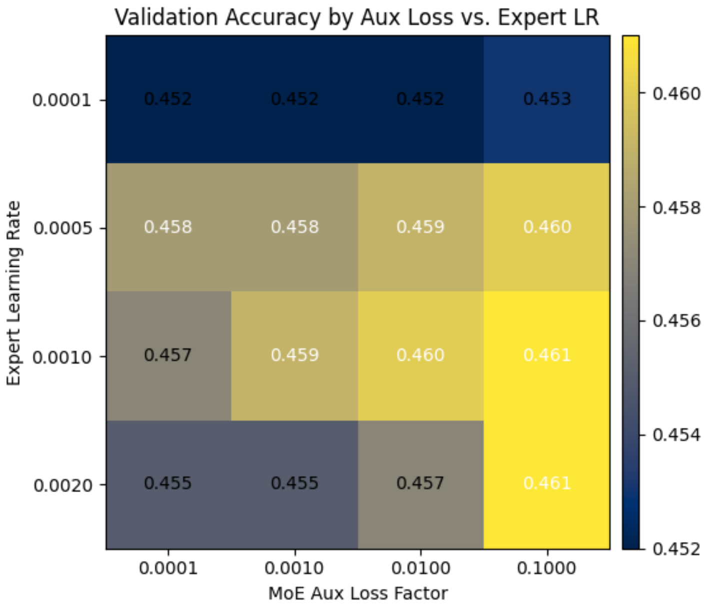
  Improve stability: keep router logits from growing too large

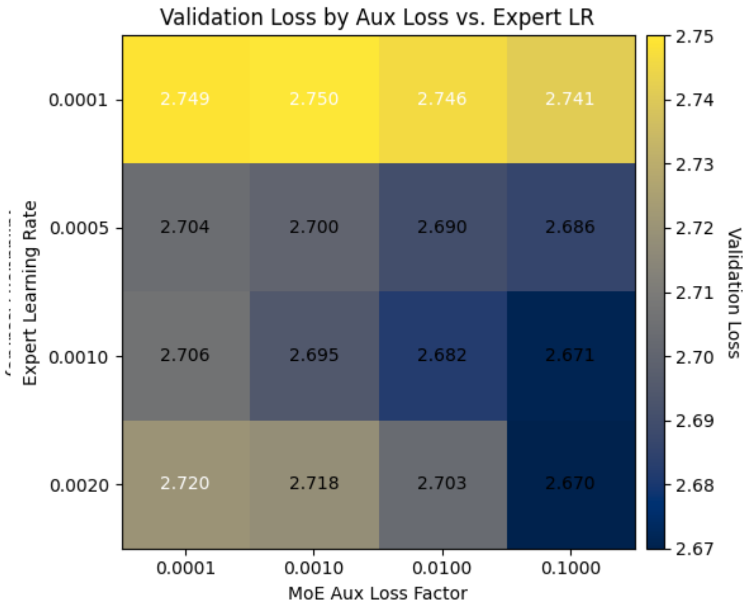
Scalar Weights

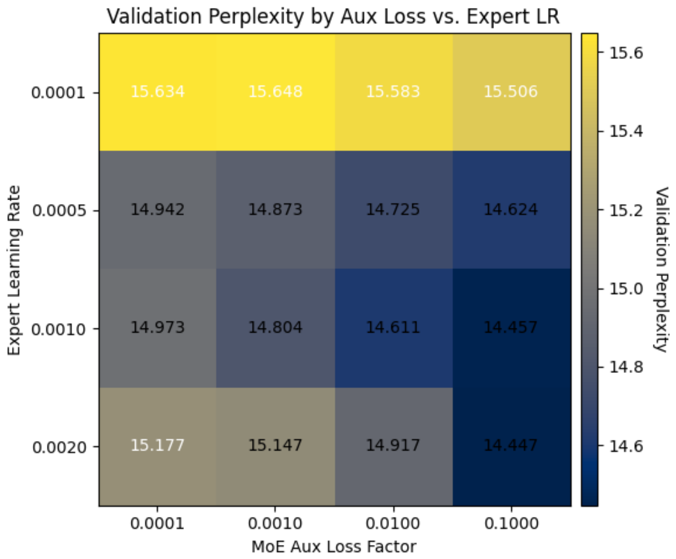
$$L_{tot} = \overbrace{L_{CE}}^{\text{Cross-Entropy Loss}} + c_B \underbrace{L_B}_{\substack{\text{Auxiliary Load}\\ \text{Balance Loss}}} + c_z \overbrace{L_Z}^{\text{Router Z-Loss}}$$

Source: Cameron R. Wolfe, "Conditional Computation: The Birth of Mixture-of-Experts," Substack (Sep 2023), *https://cameronrwolfe.substack.com/p/conditional-computation-the-birth*

Learning Rate Strategies for Experts vs. Non-Experts







- ***Setup:*** Fixed **non-expert LR at 0.001**
- A strong aux-loss weight (**0.1**) consistently **boosts performance**
- With aux weight = 0.1, expert LRs from 5×10^{-4} to 2×10^{-3} all achieve nearly **identical results**
- Lowering expert LR to 1×10^{-4} causes a **clear decline**

Exploring Optimization Strategies in Mixture-of-Experts Training

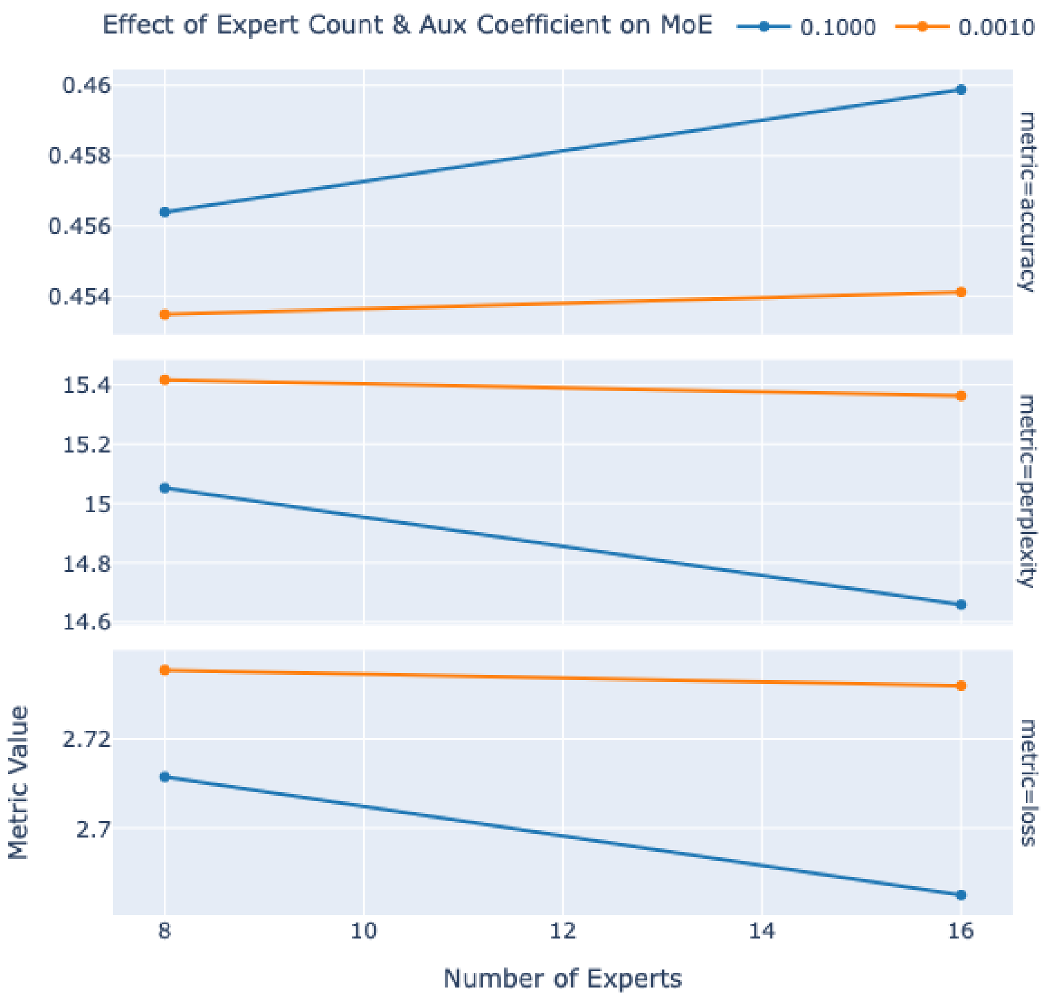
1. Learning Rate Strategies for Experts vs. Non-Experts
2. Effect of Auxiliary Loss Coefficients and Aux-Free-Loss Reproduction
3. Integration and Evaluation of the Shampoo Optimizer
4. Load Based Learning Rate Updates for Experts
5. Extending Experiments with Megatron LM

Zeynep Tandogan | zeynep.tandogan@epfl.ch

2 Effect of Auxiliary Loss Coefficients and Aux-Free-Loss Reproduction

➢ **How does the aux loss coefficient's impact change as we scale up the number of experts?**

- **Setup:** 8 and 16 experts with LR 0.0005
- Higher aux-loss factor (0.1) gives better baseline performance and **scales effectively** with more experts



Zeynep Tandogan | zeynep.tandogan@epfl.ch

➢ Aux-Free-Loss Reproduction

Problem: Performance Loss

- A large aux "load-balancing" loss injects extra gradients that compete with the main objective, hurting final performance

Solution: Aux-Free Balancing

Algorithm 1: Adjusting the per-expert bias b_i during training

Input: MoE model θ, training batch iterator B, bias update rate u.
1. Initialize $b_i = 0$ for each expert;
for *a batch* $\{(\mathbf{x}_k, \mathbf{y}_k)\}_k$ *in* B **do**
  2. Train MoE model θ on the batch data $\{(\mathbf{x}_k, \mathbf{y}_k)\}_k$, with gating scores calculated according to Eq. (3);
  3. Count the number of assigned tokens c_i for each expert, and the average number $\overline{c_i}$;
  4. Calculate the load violation error $e_i = \overline{c_i} - c_i$;
  4. Update $\mathbf{b}_i$ by $b_i = b_i + u * \mathrm{sign}(e_i)$;
end
Output: trained model θ, corresponding bias $\mathbf{b}_i$

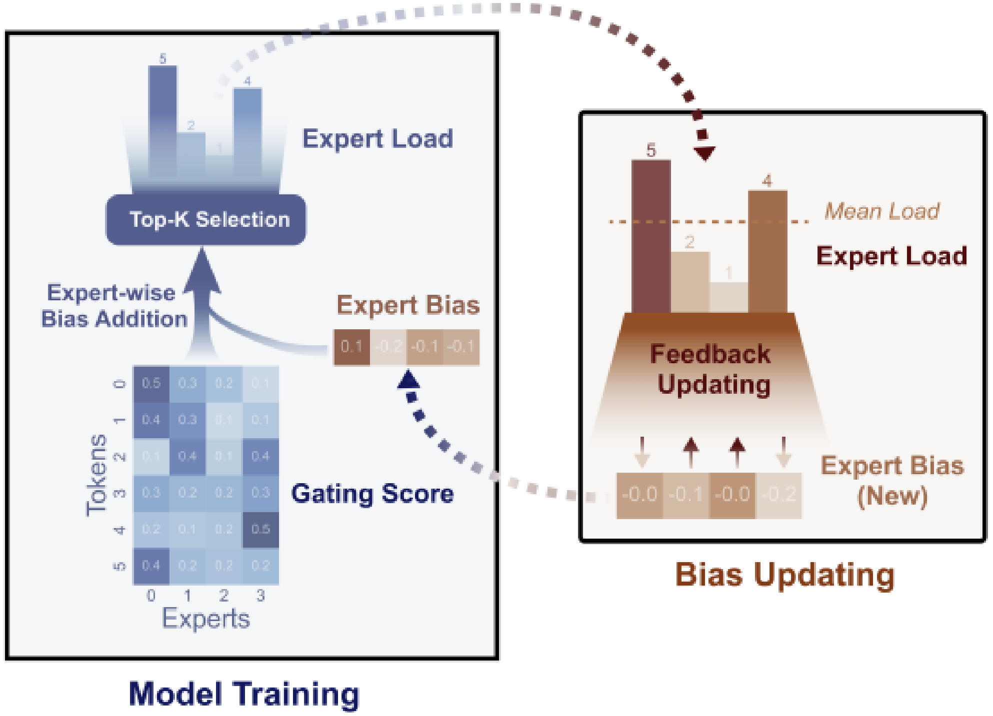
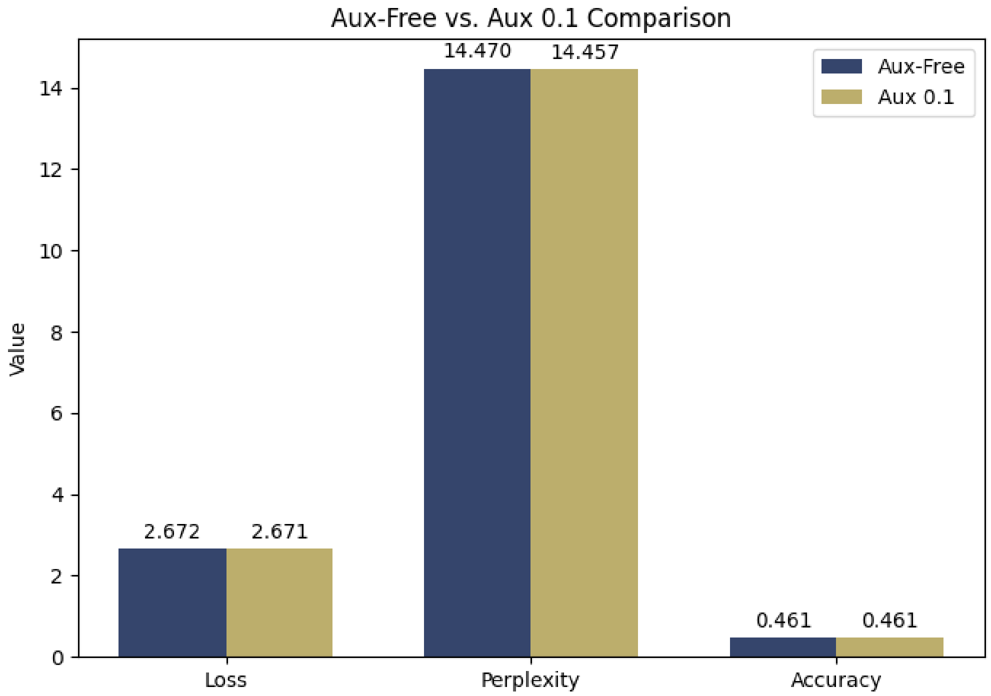


Figure 1: Loss-Free Balancing selects experts according to a "biased gating score" in each training step and updates this expert-wise bias after each training step.

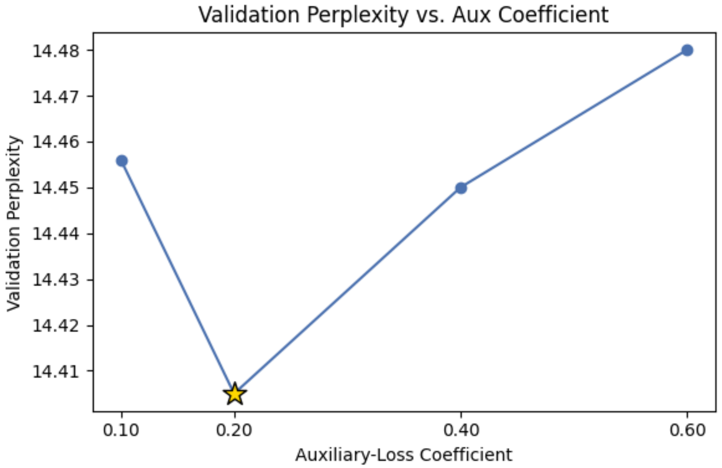
➢ Aux-Free-Loss Reproduction

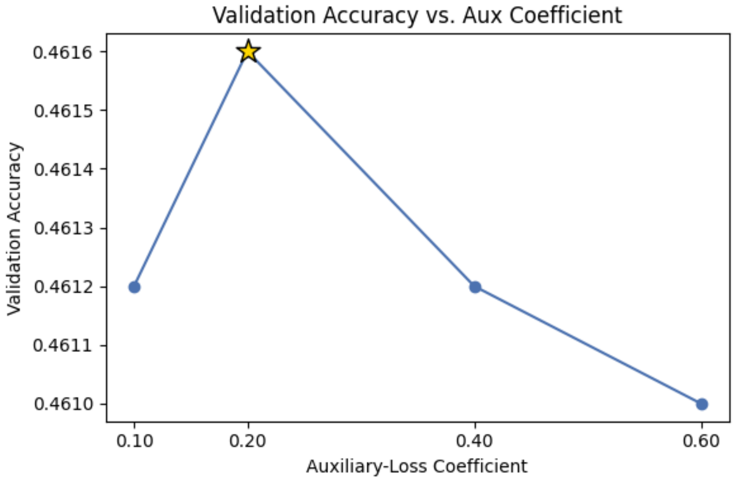


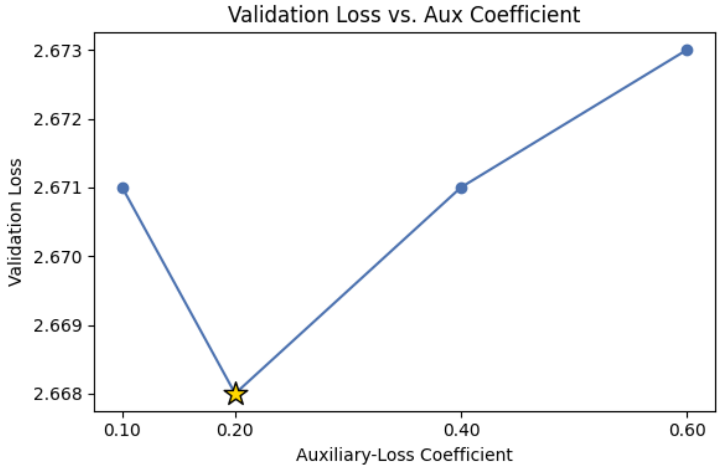
- Loss Free vs Aux 0.1 : **equivalent performance** across all metrics.

2 Effect of Auxiliary Loss Coefficients and Aux-Free-Loss Reproduction

➢ Up to which aux loss coefficient do we still observe performance improvements?







- Swept the aux-loss weight from **0.1 up to 0.6** to identify stability limits.
- **No degradation** in any metric for coefficients up to **0.2**.

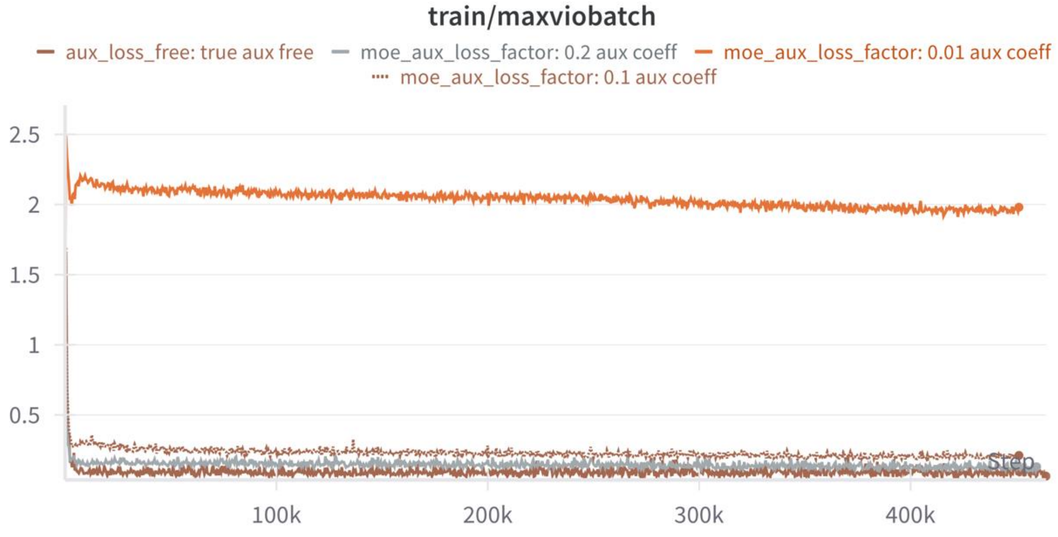
➢ Aux-Free-Loss Reproduction

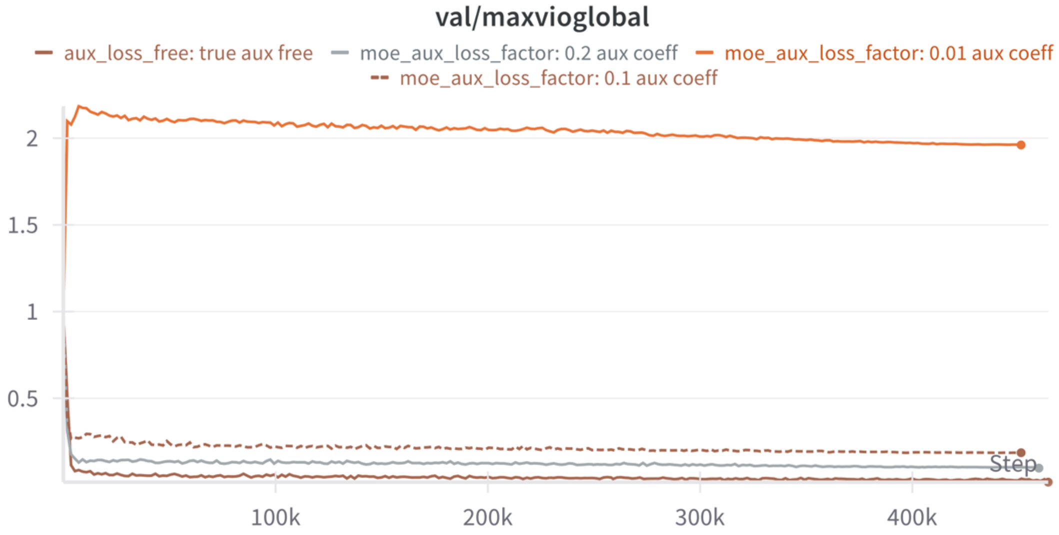
Maximal Violation (MaxVio) Analysis

- 1) MaxVio Global
- 2) MaxVio Batch

$$\text{MaxVio} = \frac{\max_i \text{Load}_i - \overline{\text{Load}_i}}{\overline{\text{Load}_i}}$$

- Loss-Free reduces both Global and Batch MaxVio below those of aux=0.1-0.2, confirming **its superior load balance.**





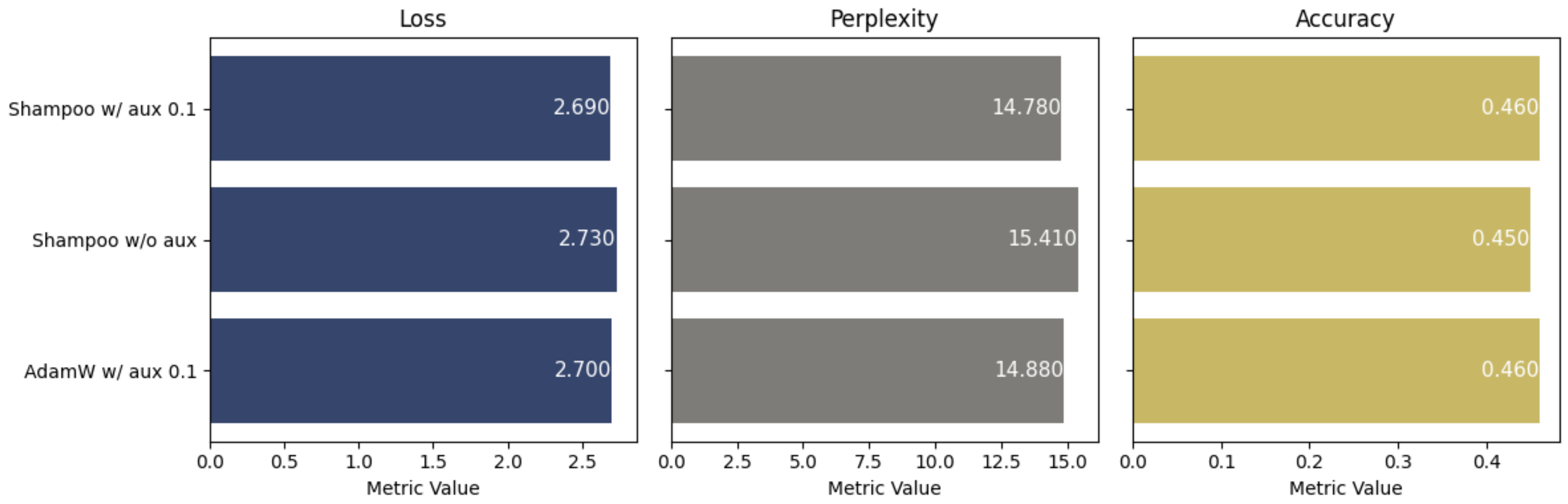
Exploring Optimization Strategies in Mixture-of-Experts Training

1. Learning Rate Strategies for Experts vs. Non-Experts
2. Effect of Auxiliary Loss Coefficients and Aux-Free-Loss Reproduction
3. Integration and Evaluation of the Shampoo Optimizer
4. Load Based Learning Rate Updates for Experts
5. Extending Experiments with Megatron LM

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Integration and Evaluation of the Shampoo Optimizer

➢ Can using the Shampoo optimizer naturally lead to more balanced expert selection?



Until now: AdamW

New optimizer: Meta's Distributed Shampoo
- Fall back to AdamW on very large matrices - every step
- Shampoo - in each 100 steps

Hao-Jun Michael Shi, Tsung-Hsien Lee, Shintaro Iwasaki, Jose Gallego-Posada, Zhijing Li, Kaushik Rangadurai, Dheevatsa Mudigere & Michael Rabbat. "A Distributed Data-Parallel PyTorch Implementation of the Distributed Shampoo Optimizer for Training Neural Networks At-Scale." arXiv preprint arXiv:2309.06497, Sep 12 2023

Integration and Evaluation of the Shampoo Optimizer

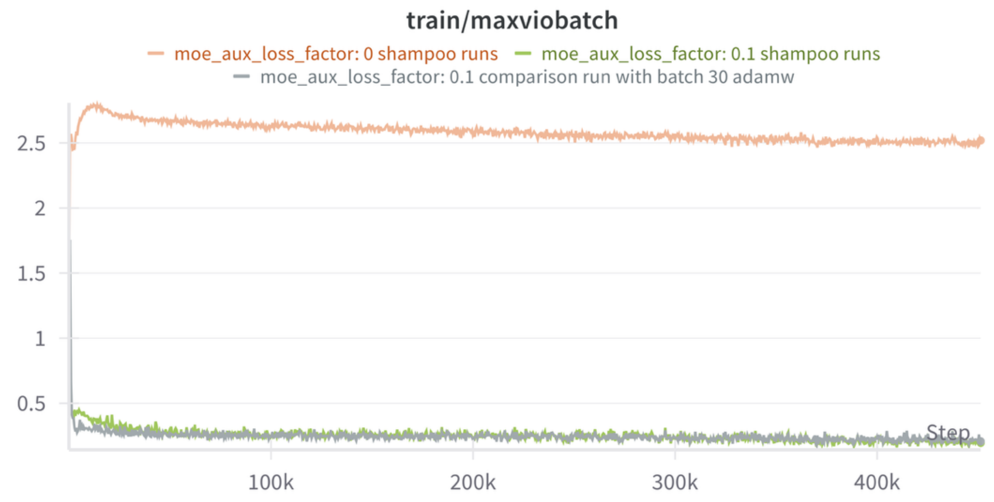
- Shampoo optimizer achieves slightly lower MaxVio in both batch and global settings, indicating **improved load balance**.

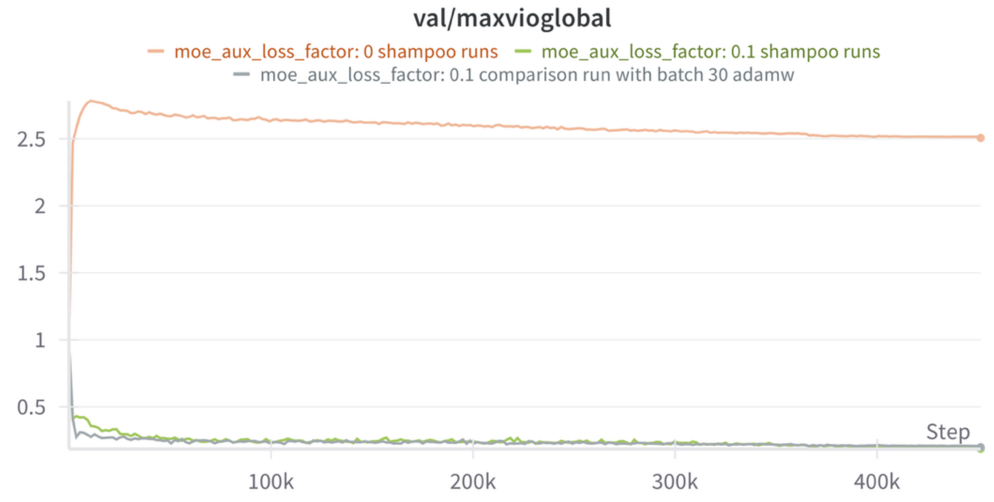
However,

using Shampoo takes longer training times…

Wall-clock time for 50 K steps: AdamW vs. Shampoo.

Optimizer	Wall-clock Time	Relative Cost
AdamW	1 d 16 h (40 h)	1×
Shampoo	2 d 2 h (50 h)	1.25×





Exploring Optimization Strategies in Mixture-of-Experts Training

1. Learning Rate Strategies for Experts vs. Non-Experts
2. Effect of Auxiliary Loss Coefficients and Aux-Free-Loss Reproduction
3. Integration and Evaluation of the Shampoo Optimizer
4. Load Based Learning Rate Updates for Experts
5. Extending Experiments with Megatron LM

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Load Based Learning Rate Updates for Experts

➢ Batch-fraction load ratio

$$r_i = \frac{L_i}{T},$$

$$\sum_i L_i = kT \implies \sum_i r_i = k,$$

$$r_i \in [0, 1].$$

- **Always** scales each expert's learning rate **downward** relative to the base
- **Lightly loaded** experts experience the **largest LR reduction**

➢ Ideal-Normalized load ratio

$$r_i = \frac{L_i}{\bar{L}},$$

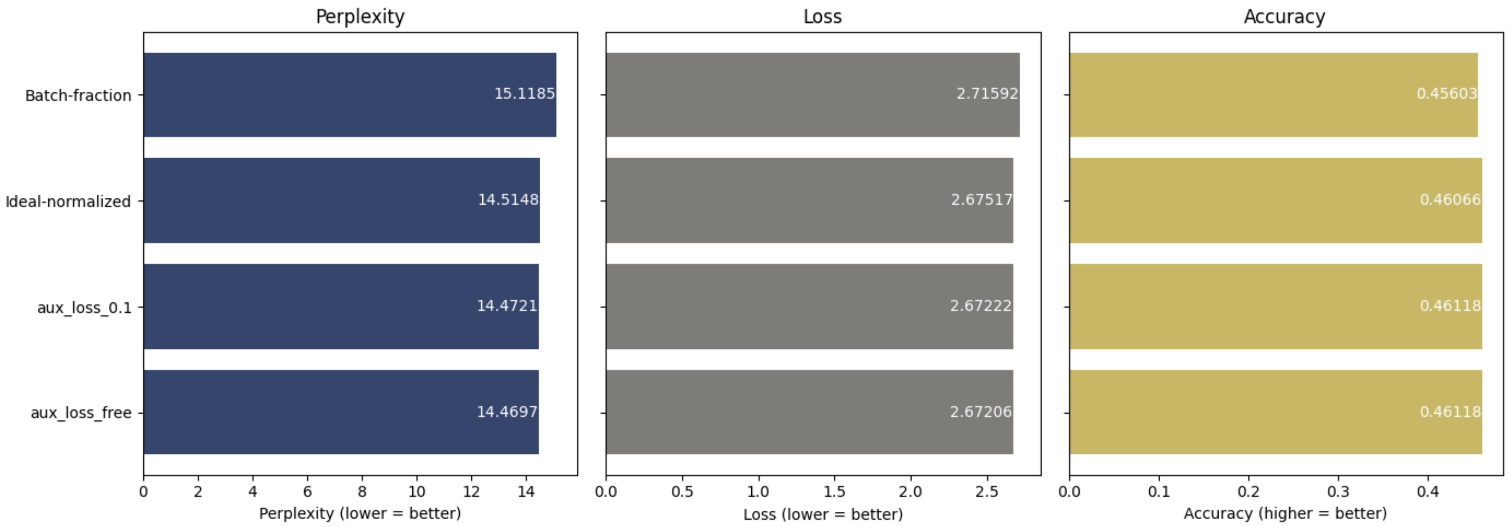
$$\bar{L} = \frac{\sum_i L_i}{E} = \frac{kT}{E}.$$

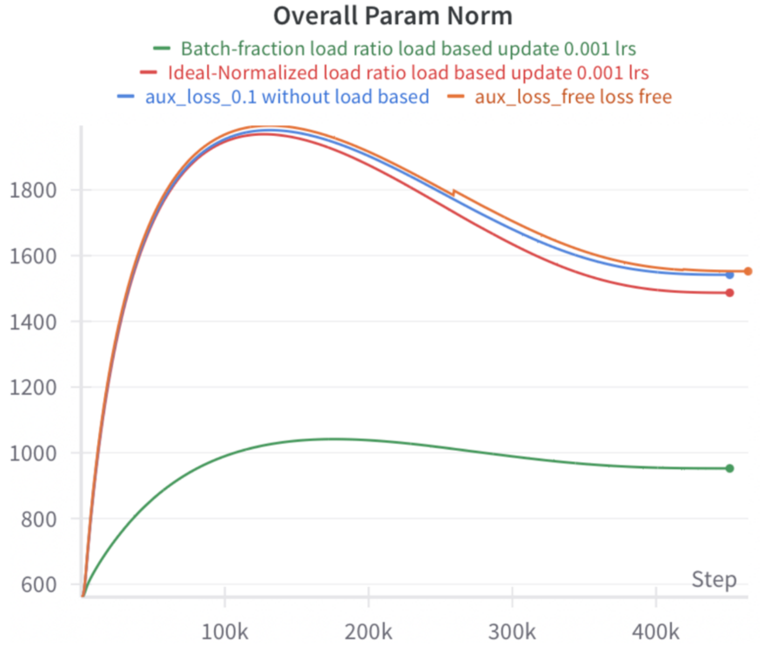
- Bounded below by 0; no fixed upper bound.
- **Overloaded** experts receive an **LR boost**, **underloaded** ones are **dampened**
- Mean scaling remains exactly 1 across all experts

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Load Based Learning Rate Updates for Experts

- **Setup:** 8 experts with LR 0.001, top-k 2
- Both batch-fraction and ideal-normalized methods **underperform** relative to the aux-loss 0.1 and aux-free approaches.



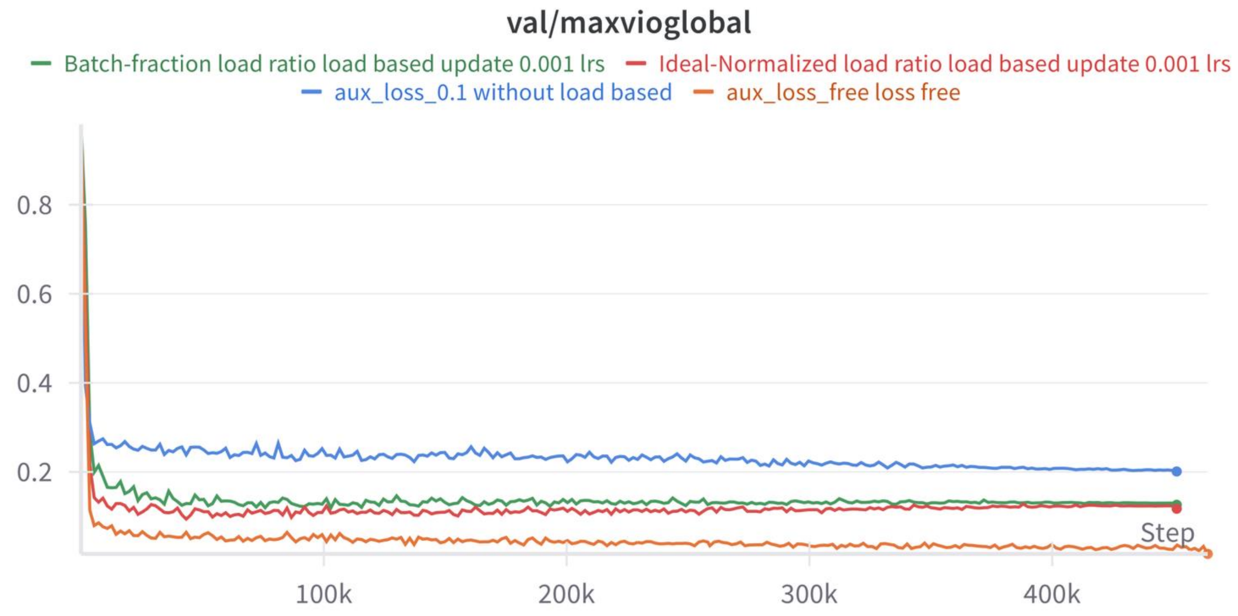


Underfitting risk in batch fraction?

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Load Based Learning Rate Updates for Experts

- Ideal-normalized most closely matches the aux-free MaxVio trend, followed by batch-fraction, with aux-loss 0.1 trailing behind.



Zeynep Tandogan | zeynep.tandogan@epfl.ch

Exploring Optimization Strategies in Mixture-of-Experts Training

1. Learning Rate Strategies for Experts vs. Non-Experts
2. Effect of Auxiliary Loss Coefficients and Aux-Free-Loss Reproduction
3. Integration and Evaluation of the Shampoo Optimizer
4. Load Based Learning Rate Updates for Experts
5. Extending Experiments with Megatron LM

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Extending Experiments with Megatron LM

Setup: Fixed learning rate at 0.001; compared **sigmoid and softmax** gating with **different bias update rates**.

- With softmax aux-free and a bias update rate of 1×10^{-3}, maxvio stayed around 1.5—significantly higher than the sigmoid variant.
- Increasing the bias update rate to 1×10^{-2} was necessary to bring softmax's maxvio down to acceptable levels.

Configuration	lm loss	maxvio	z_loss
0.001-lr, sigmoid e-3, loss free	**2.871**	0.086	0.003
0.001-lr, sigmoid e-2, loss free	2.893	0.313	0.003
0.001-lr, softmax e-3, loss free	3.068	1.574	0.003
0.001-lr, softmax e-2, loss free	2.890	0.081	0.003
0.001-lr, sigmoid aux 0.1	3.223	0.072	0.010
0.001-lr, softmax aux 0.1	2.896	**0.032**	0.002

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Extending Experiments with Megatron LM

➢ ***How does expert learning rates impact MoE performance as we scale up the number of experts?***

Configuration	lm loss	maxvio	z_loss
0.001 expert lr,	**2.896**	**0.032**	0.002
0.0005 expert lr	2.960	0.041	0.003

Average loss metrics for 8-expert configurations
(non expert LR 0.001 with softmax aux coefficient 0.1)

Configuration	lm loss	maxvio	load_balancing	z_loss
0.0005 expert lr	**3.090**	0.587	0.999	**0.007**
0.001 expert lr	3.109	**0.439**	**0.998**	0.007

Average loss metrics for 128-expert configurations (non expert LR 0.001)

8-Expert Setup: Matched LRs boost both accuracy and balance over a lower expert LR.

128-Expert Setup: Lower expert LR yields minor accuracy gains but worsens load balance.

Note: 128-expert runs use 12 layers & 1024-dim FFNs (down from 24/2048).

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Summary of Our Findings

Hyper-parameter	Recommended Setting	Rationale
Base LR (non-experts)	0.001	Stable across **model & expert counts**.
Expert LR	0.0005–0.001	Clear U-shape (minimum at 5×10^{-4}); avoid 1×10^{-4}; up to 2×10^{-3} safe when aux-loss = 0.1; splitting schedules yields no gain.
Aux-loss weight	0.05–0.1	Boosts accuracy, perplexity & MaxVio by **balancing traffic**; **plateaus beyond 0.2** and reverses if it dominates.
Gating	Sigmoid, top-k = 2	Encourages **balance** with aux-loss 0.1; softmax needs 10× higher **bias update rate** to match.
Load-balance regularizer	Aux-free (or ideal-normalized with clipping)	Yields **lowest MaxVio**; batch-fraction **under-fits**, ideal-normalized can be **unstable** w/o clipping.
Load-based LR scaling	Not recommended	Underperforms both aux-based ,and aux-free: batch-fraction **under-fits**; ideal-normalized risks **instability**.
# Experts	8–16 (128 if FFN dims ↓)	Aux-loss 0.1 scales effectively; impact of aux coef grows with more experts; lower expert LR gives **minor accuracy gains** at 128 but **worsens balance**.
Optimizer	AdamW (Distributed Shampoo optional)	Shampoo lowers **loss/perplexity & MaxVio** but adds **≈25%** training time—use when **balance is critical**.

Zeynep Tandogan | zeynep.tandogan@epfl.ch

Thanks for listening!

Any questions?

Zeynep Tandogan | zeynep.tandogan@epfl.ch