Scaling Sparse Mixture-of-Experts Language Models

Taishi Nakamura

Institute of Science Tokyo

AI Research Summit 2026

About Me

- Taishi Nakamura
- PhD student, Institute of Science Tokyo
- Research Assistant at NII LLMC, working on the LLM-jp project
- Research interests: LLM scaling and efficient training/inference

Talk Outline: Two MoE Projects

Overall question:

How can we design and train high-performance MoE language models under fixed compute budgets?

Project 1: Drop-Upcycling (ICLR 2025)

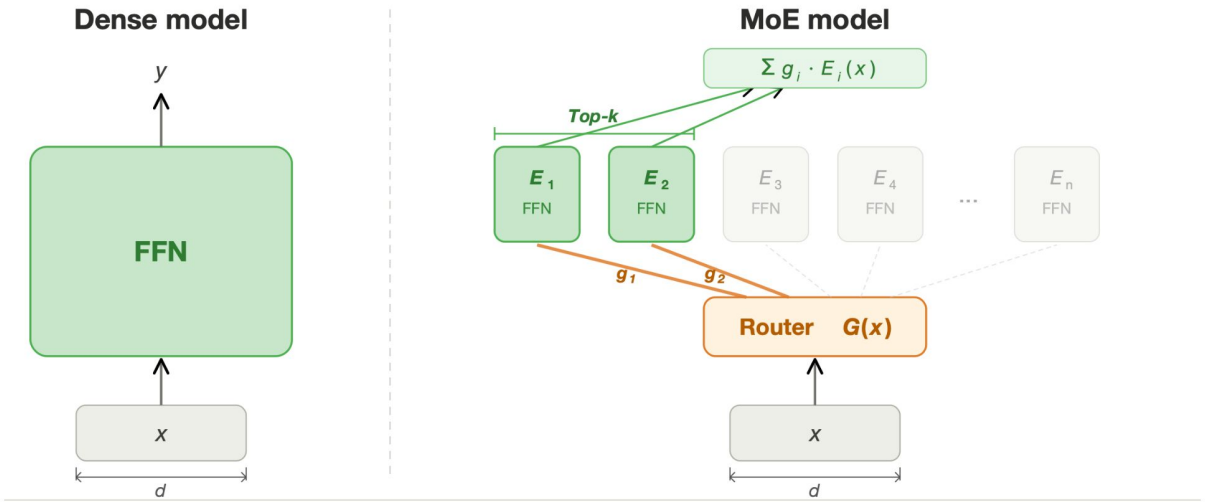
- Training MoE models from pretrained dense checkpoints
- How can we efficiently train strong MoE models from dense checkpoints?

Project 2: Optimal MoE Sparsity for Reasoning (ICLR 2026)

- Characterizing compute-optimal sparsity for reasoning tasks
- How sparse should MoE models be under fixed compute budgets?

Background: Mixture-of-Experts (MoE)

Activates only a subset of parameters per token, enabling larger models under the same compute budget.



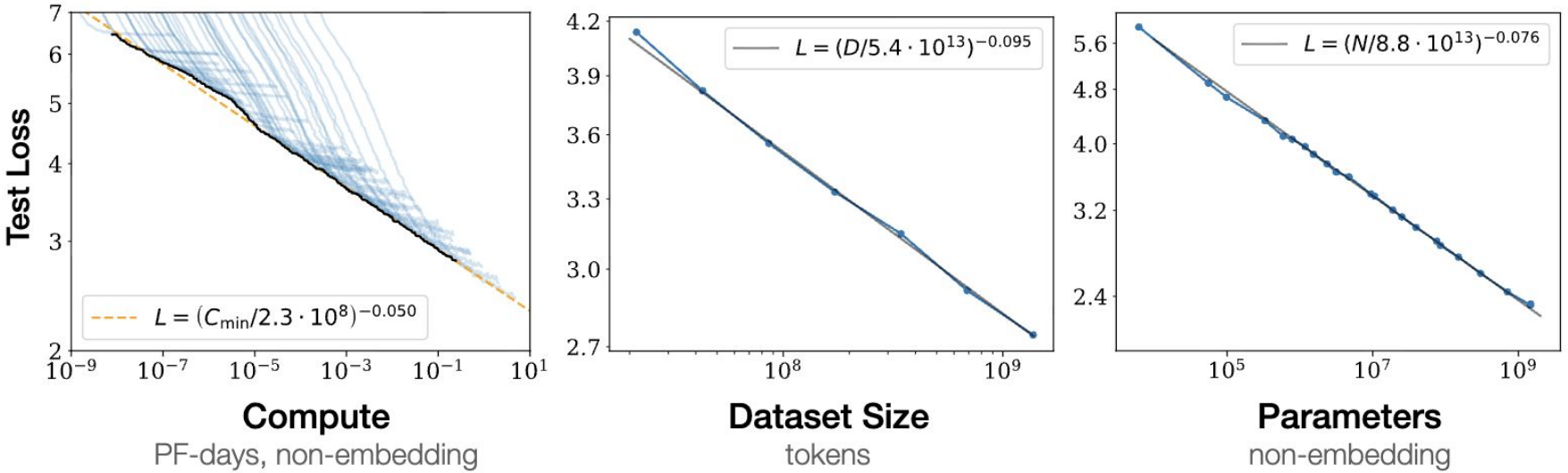
Top-k : Number of experts activated per token
E : Total number of experts in the model
d : Model width

Sparsity = *1 – (Top-k / E)*

MoE Density = *Top-k / E*

Background: Scaling Laws of Dense LLMs

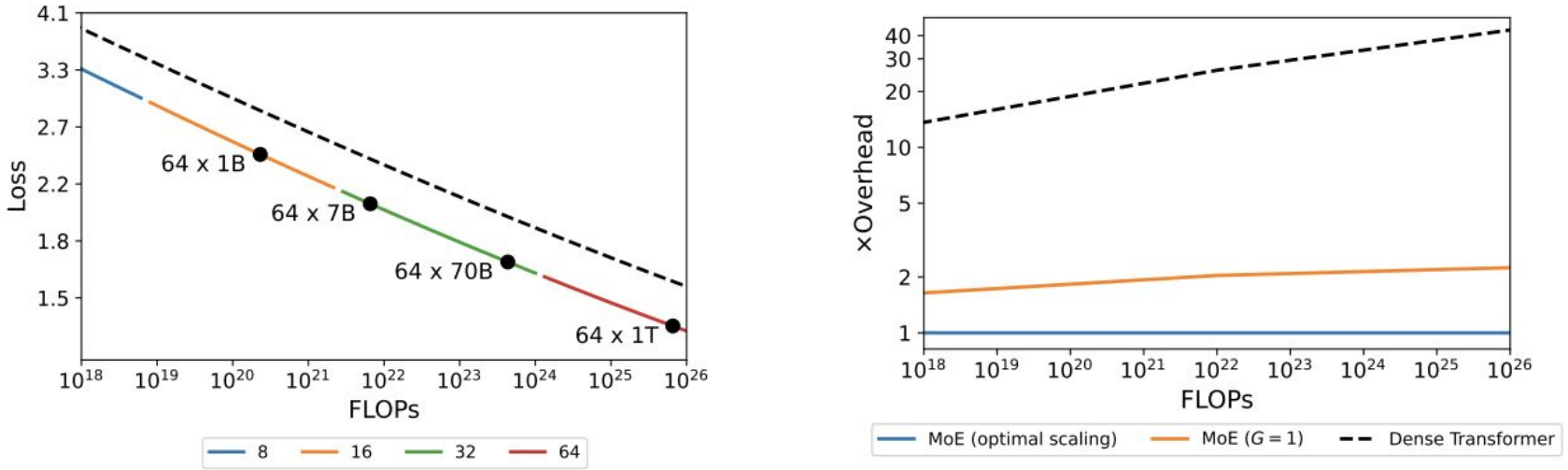
Pretraining loss follows power laws as a function of compute, training data, and model size. (Kaplan et al., 2020)



Source: Kaplan et al., "Scaling Laws for Neural Language Models" arXiv:2001.08361, Figure 1.

Background: Scaling Laws for MoE

MoE achieves lower loss than dense models at the same compute.
(e.g., Clark et al., ICML 2022; Ludziejewski et al., ICML 2024)



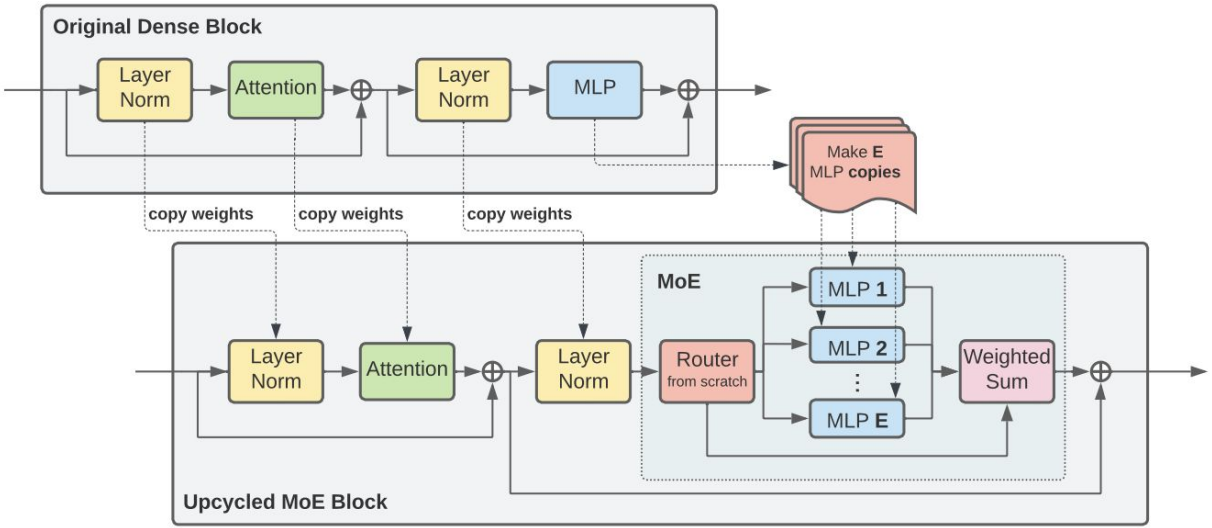
Source: Ludziejewski et al., "Scaling Laws for Fine-Grained Mixture of Experts" ICML 2024, Figure 1.

Compute Bottleneck

- Training MoE models from scratch remains computationally expensive
- **We therefore leverage pretrained dense models**

Background: naïve Upcycling

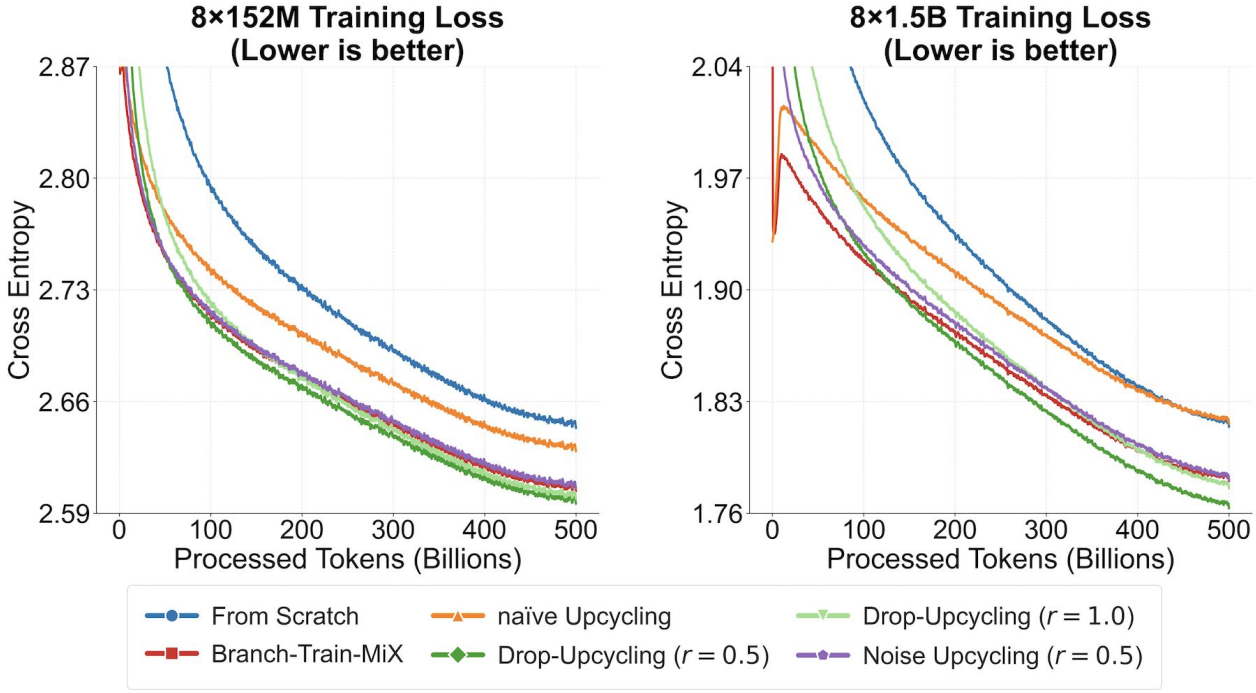
Leveraging pretrained dense models via sparse upcycling (Komatsuzaki et al., ICLR 2023)



Source: Komatsuzaki et al., "Sparse Upcycling: Training Mixture-of-Experts from Dense Checkpoints" ICLR 2023, Figure 1.

Limitation of naïve Upcycling

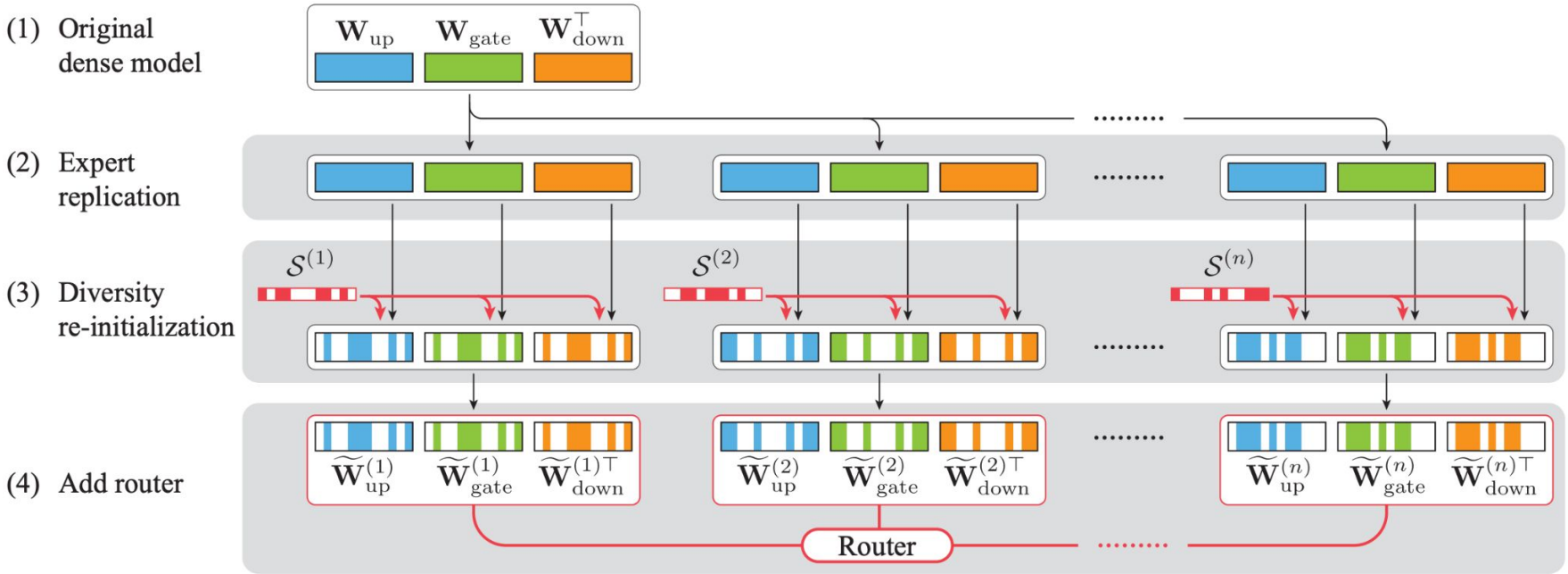
- However, all experts start from the same dense MLP
- **This leads to insufficient expert diversity**



Proposed Method: Drop-Upcycling

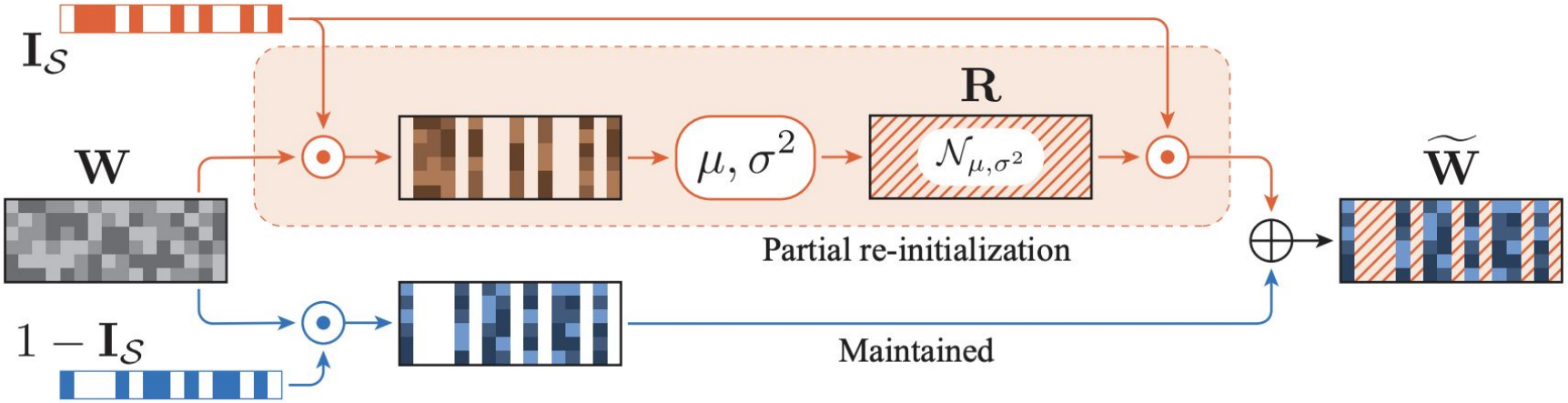
Expert replication – copy each dense FFN into *n* experts

Diversity re-initialization – introduce random partial re-initialization per expert



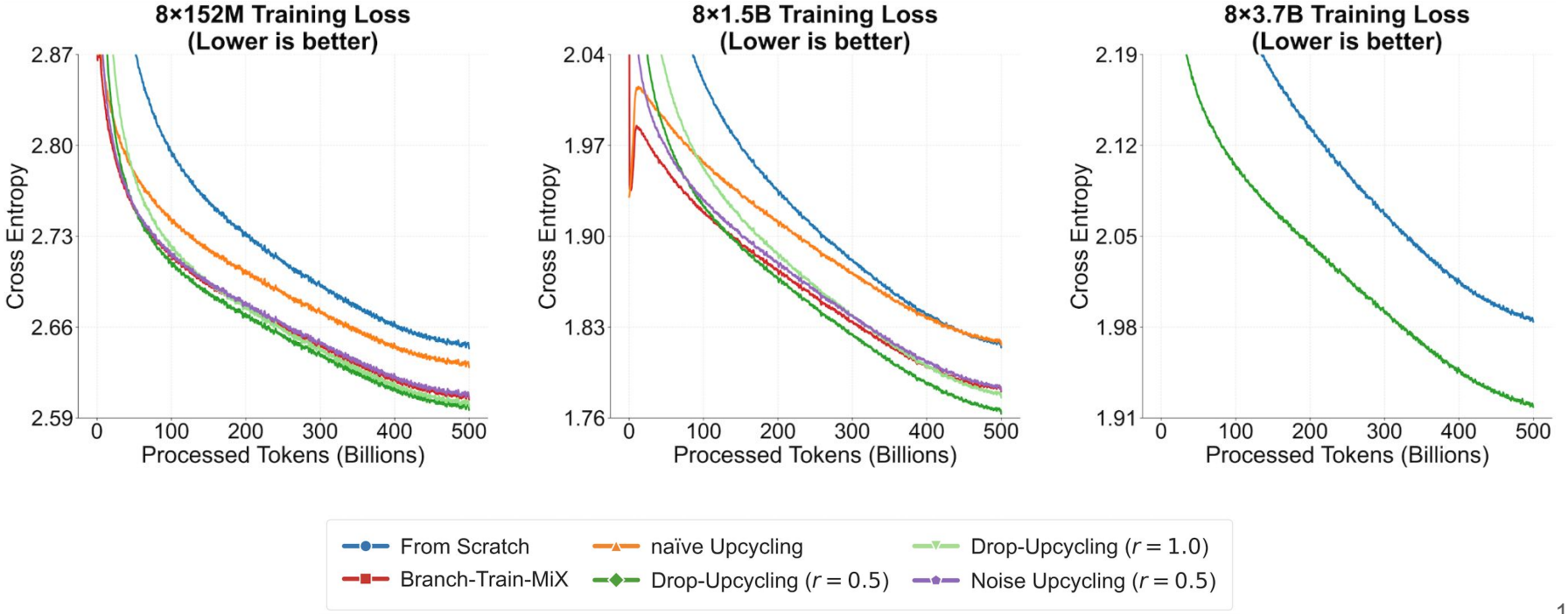
Proposed Method: Drop-Upcycling

- For each expert, randomly drop a fraction *r* of columns/rows and redraw them from $\mathcal{N}(\mu, \sigma^2)$.
- μ and σ are computed from the dropped weights themselves.



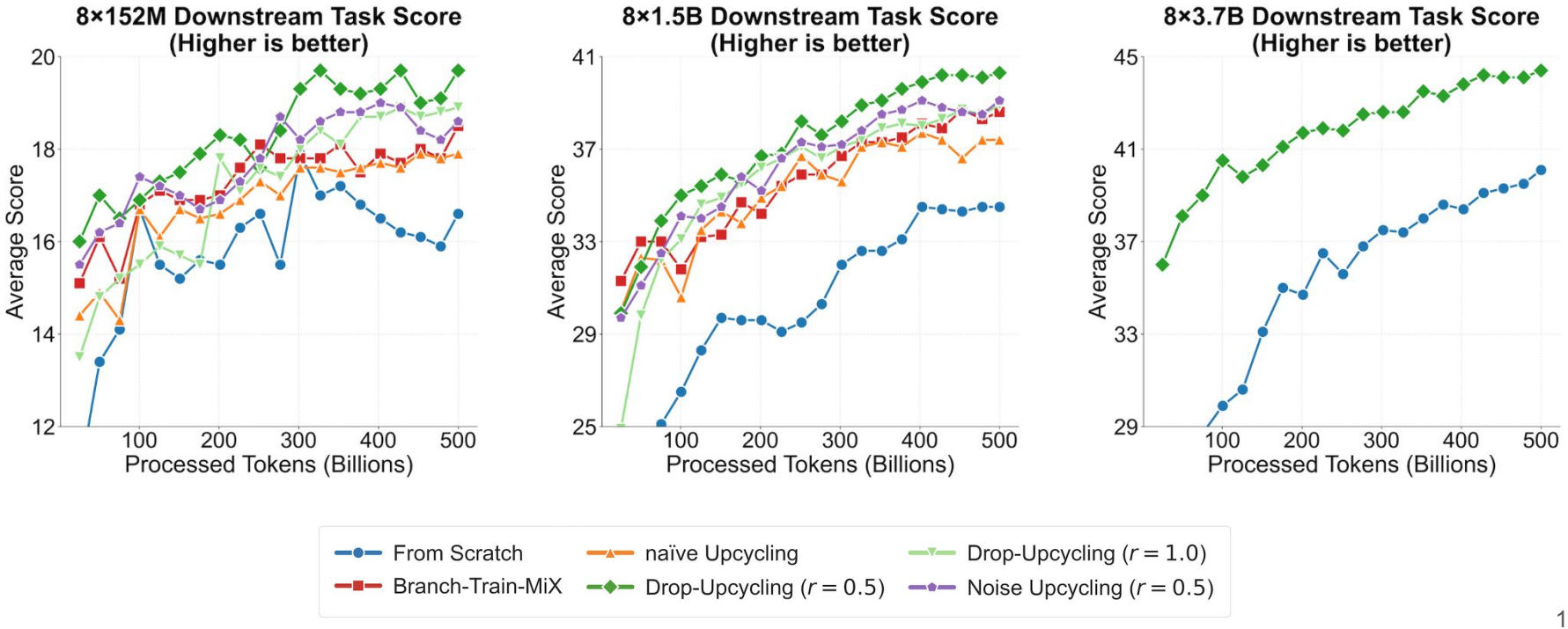
Training Convergence

Drop-Upcycling achieves **lower training loss.**



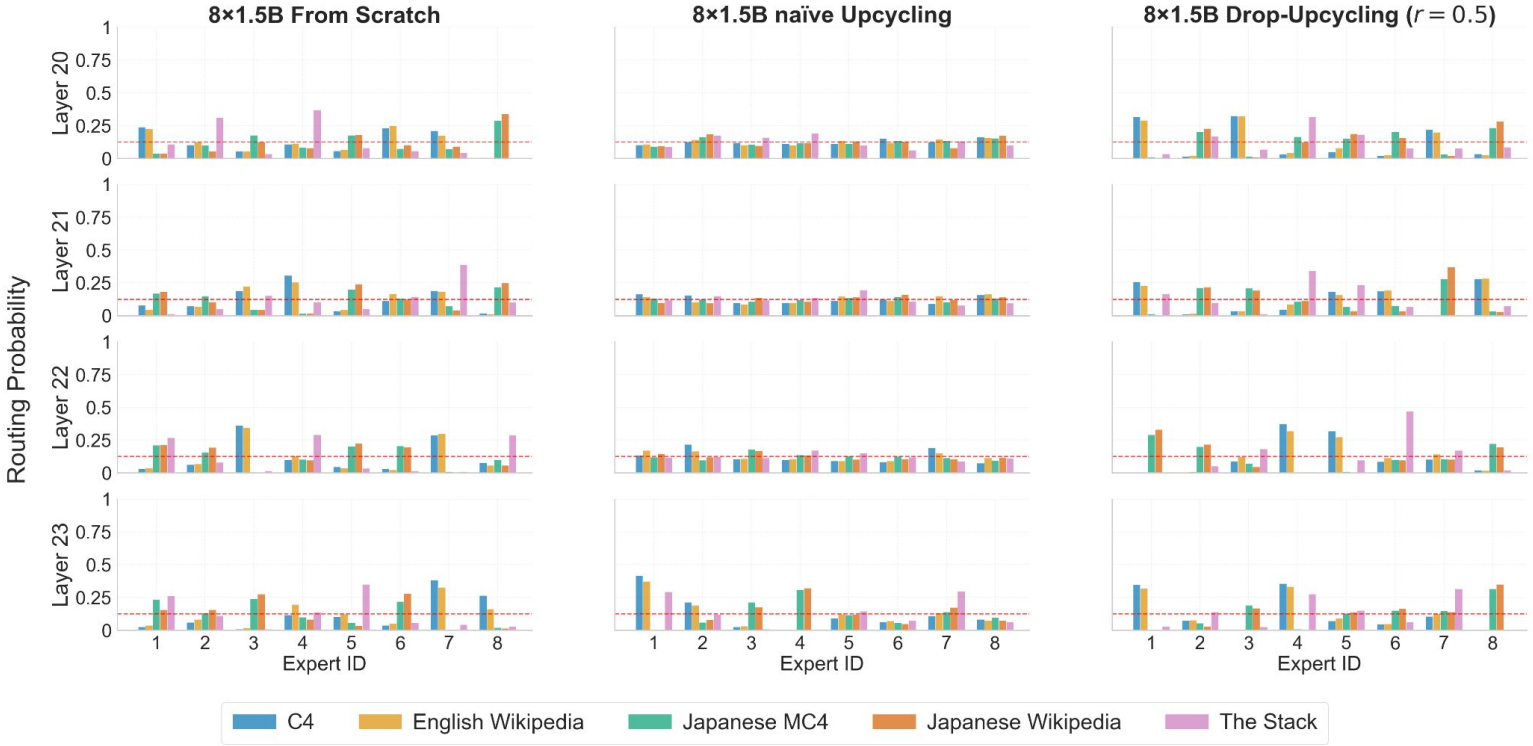
Downstream Performance

Drop-Upcycling achieves **strong downstream performance.**



Expert Routing Patterns

Partial re-initialization encourages specialized expert routing in **Drop-Upcycling.**

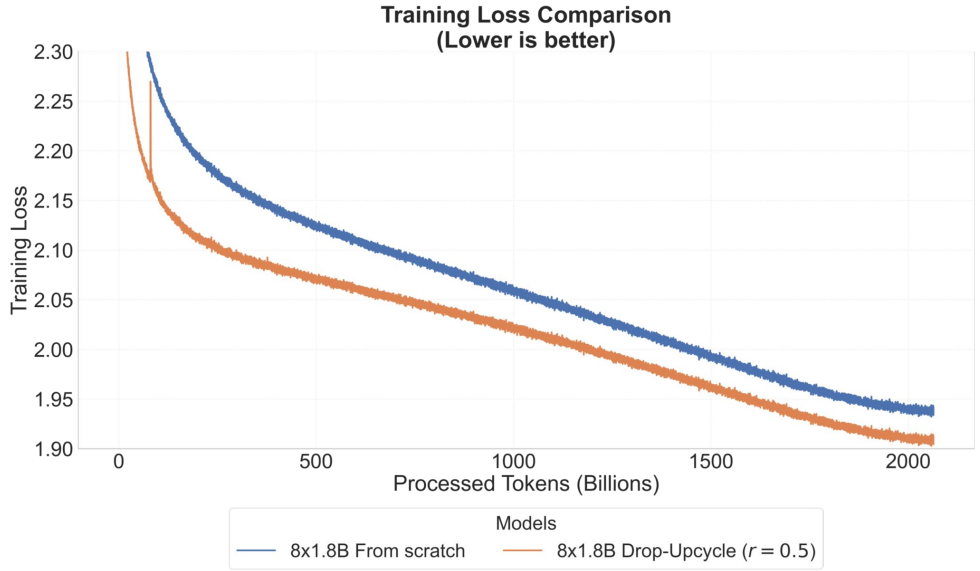


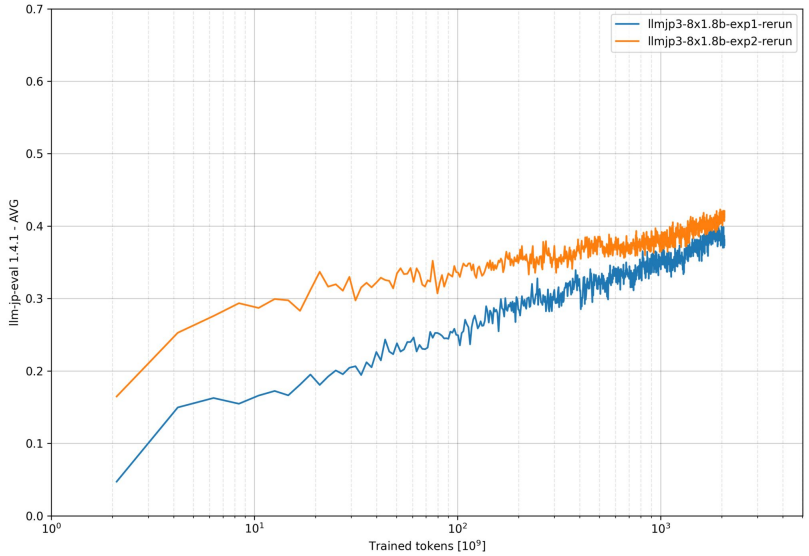
Scaling of Drop-Upcycling

- Drop-Upcycling has been used to train large-scale MoE models.
- We applied it to **8×1.8B** and **8×13B** MoE models initialized from pretrained dense checkpoints in the LLM-jp project

Scaling to 8×1.8B MoE

At a training scale of up to **~2.1T tokens,** 8×1.8B achieves **lower loss** and **better downstream performance** than training from scratch.

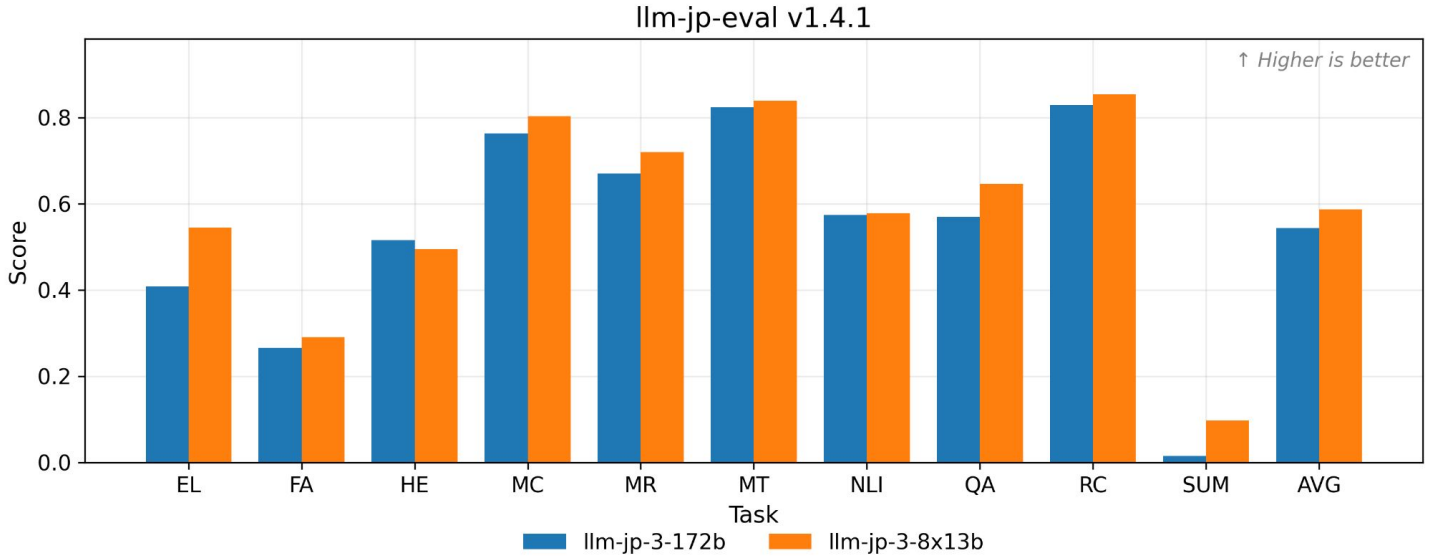




Source: LLM-jp Project, https://llm-jp.nii.ac.jp/ja/blog/blog-603/

Large-Scale Result: 8×13B MoE vs. 172B Dense

8×13B MoE Drop-Upcycled from 13B dense (22B active / 73B total) **matches or outperforms 172B dense** on llm-jp-eval v1.4.1.



Adoption of Drop-Upcycling in Recent MoE Models

Recent models adopt Drop-Upcycling to scale MoE models with fine-grained experts.

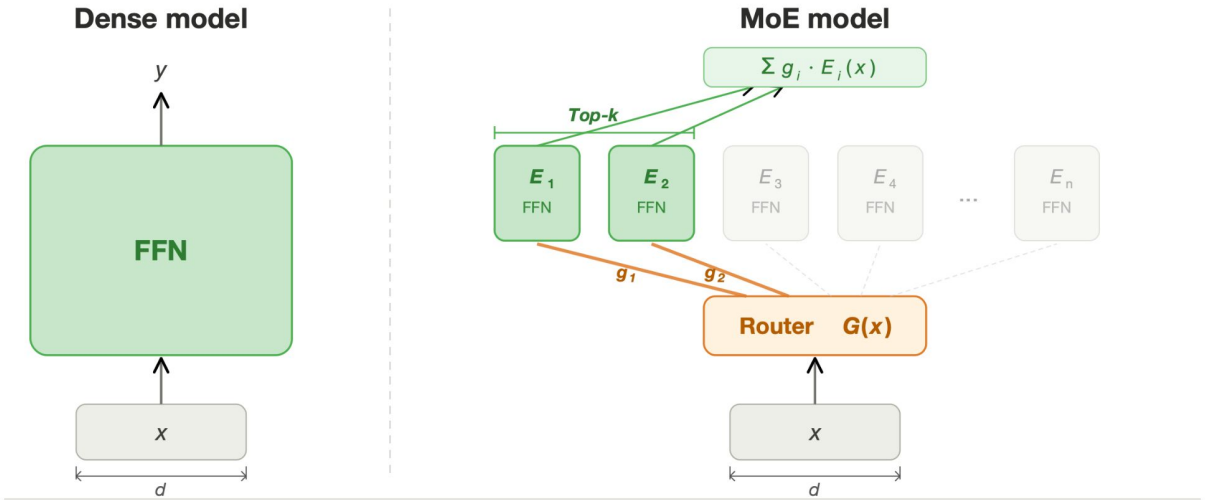
- Moondream 3 Preview
  - Initialized from Moondream 2 using Drop-Upcycling
  - 64 experts, top-8
- Marco-MoE (Jiang et al., 2026)
  - Uses Drop-Upcycling to initialize from a Qwen3 dense checkpoint
  - 256 experts, top-8, 5T tokens

How Sparse Should MoE Be?

So far, we have not carefully explored
how the number of experts and top-k affect
performance.
We study the optimal sparsity of MoE models.

Background: Mixture-of-Experts (MoE)

- MoE activates only a subset of experts per token, enabling a larger total parameter count at the same FLOPs.
- This introduces **sparsity** as a new design axis.

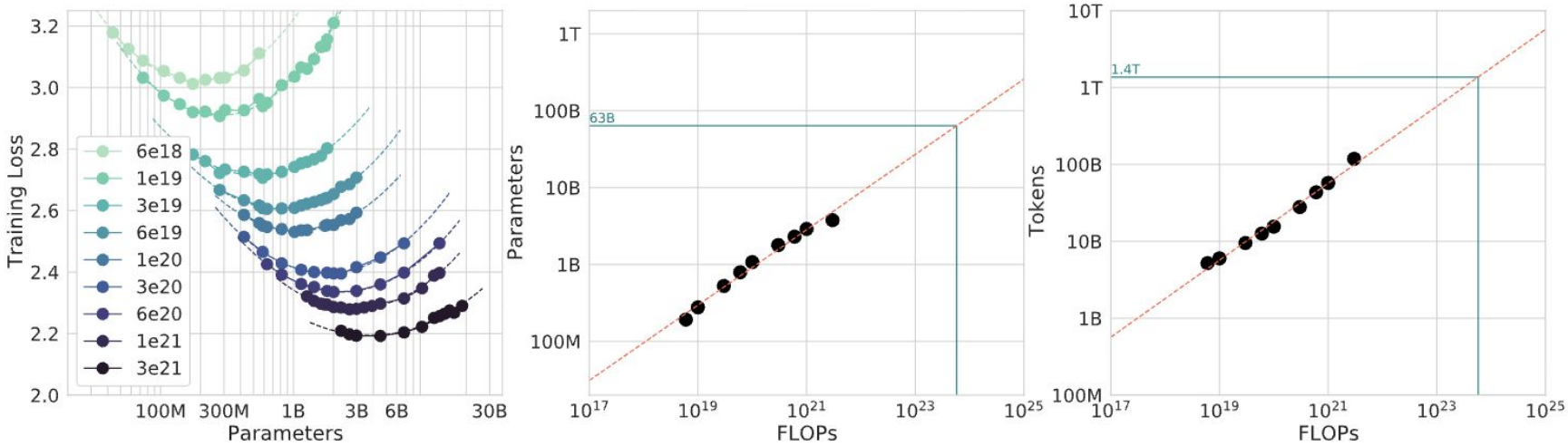


Top-k : Number of experts activated per token
E : Total number of experts in the model
d : Model width

Sparsity = *1 – (Top-k / E)*
MoE Density = *Top-k / E*

Background: Compute-Optimal Scaling (Dense)

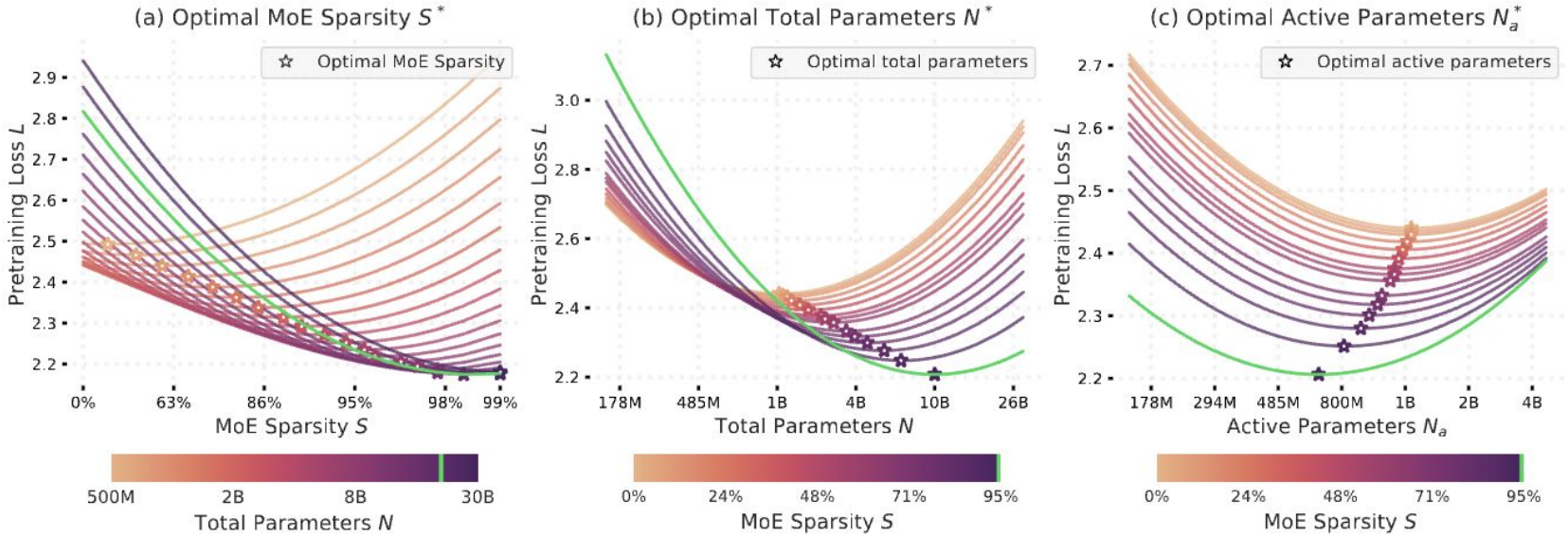
Under a fixed compute budget, model size and training data should be scaled proportionally. (Hoffmann et al., NeurIPS 2022)



Source: Hoffmann et al., "Training Compute-Optimal Large Language Models" NeurIPS 2022, Figure 3.

Background: Sparsity in MoE

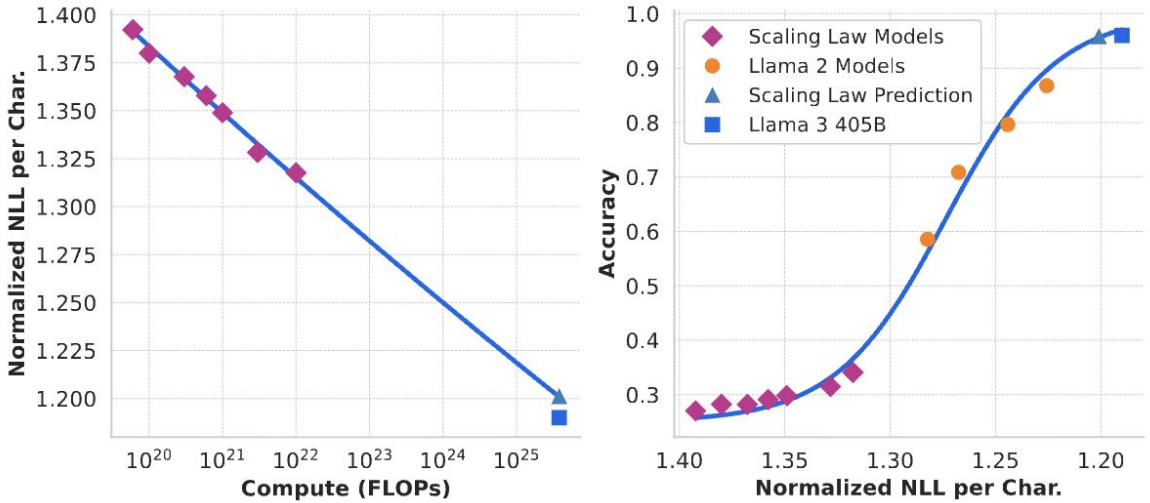
Under fixed compute, optimal sparsity increases with model size.
(Abnar et al., ICML 2025)



Source: Abnar et al., "Parameters vs FLOPs: Scaling Laws for Optimal Sparsity for Mixture-of-Experts Language Models" ICML 2025, Figure 2.

Background: Predicting Performance from Loss

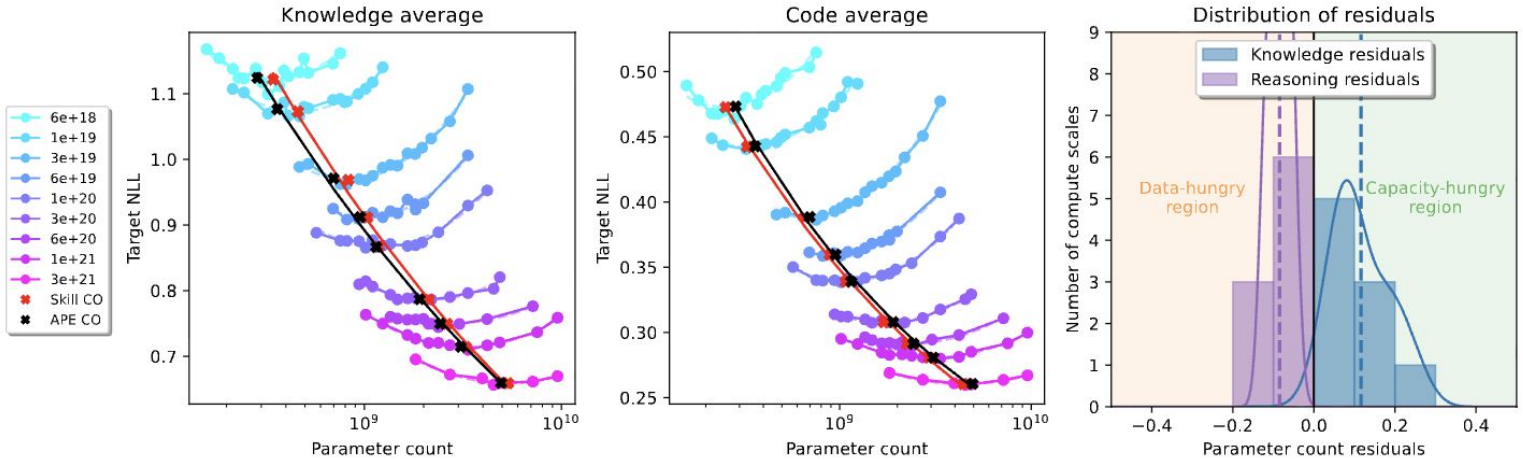
- Scaling laws to predict downstream performance from loss
- However, this relationship is task-dependent and unreliable for reasoning. (Lourie et al., 2025; Jelassi et al., ICLR 2025)



Source: Grattafiori et al., "The Llama 3 Herd of Models" arXiv:2407.21783, Figure 4.

Background: Skill-Dependent Scaling

Scaling behaviour is not uniform across tasks (Roberts et al., 2025)
Reasoning and knowledge exhibit different compute-optimal trends.



(a) Knowledge QA. (b) Code. (c) Data- vs capacity-hunger.

Source: Roberts et al., "Compute Optimal Scaling of Skills: Knowledge vs Reasoning" arXiv:2503.10061, Figure 1.

Research Question: Sparsity Design for Reasoning

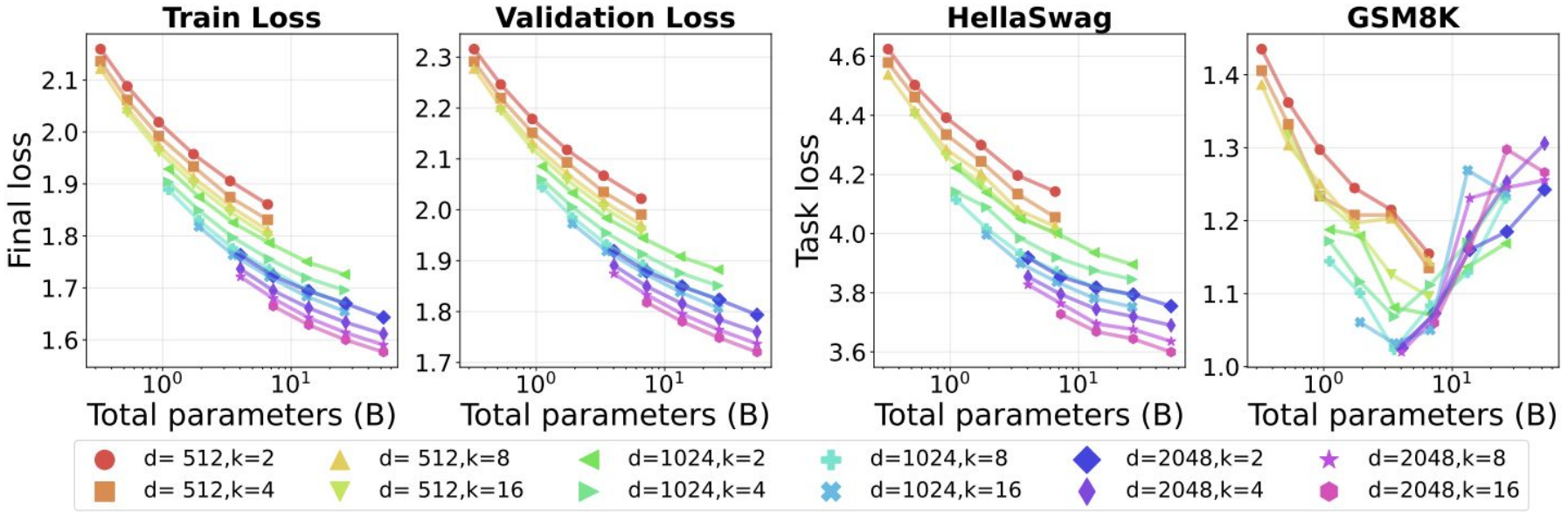
- What is the compute-optimal sparsity for MoE models on reasoning tasks under a fixed compute budget?
- How do total and active parameter counts affect reasoning performance?
- Can techniques such as self-consistency and GRPO mitigate the negative effects of overly sparse MoE models?

Experimental Setup

- Swept Dimensions
  - Model width $d \in \{512, 1024, 2048\}$
  - Number of experts $E \in \{8, 16, 32, 64, 128, 256\}$ ($E \leq 128$ for $d = 2048$)
  - Top-$k \in \{2, 4, 8, 16\}$
  - Total params: 0.32B–52B
  - Active params: 0.17B–7.1B
- Training
  - 125B tokens
  - Web (~43B), Math (~32B), STEM + Wikipedia (~49B)
- Evaluation Tasks
  - Memorization: TriviaQA, HellaSwag
  - Reasoning: GSM8K, GSM-Plus

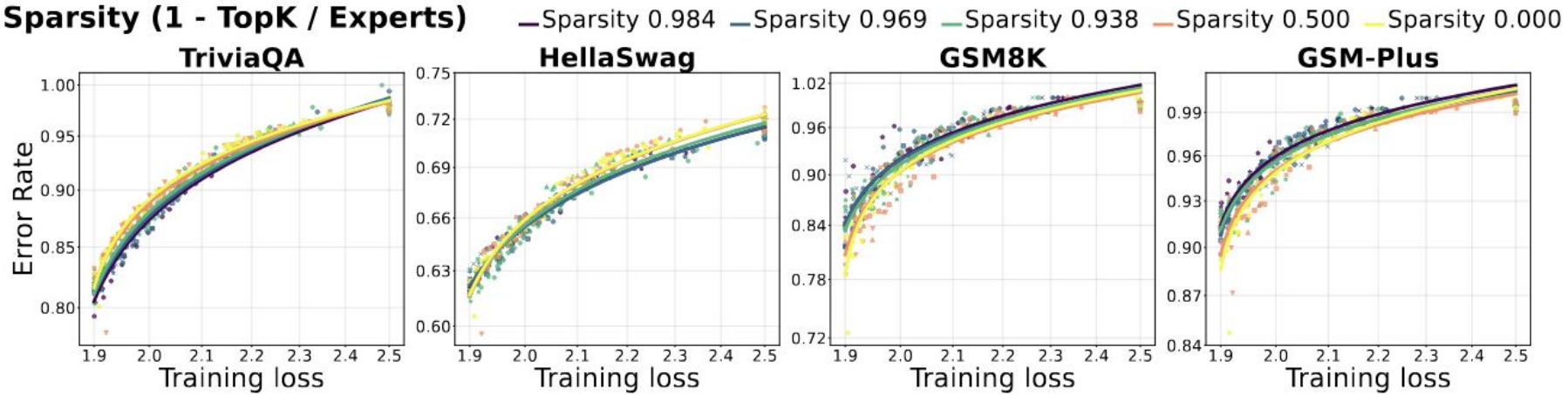
Validation Loss Does Not Always Predict Task Loss

- Cross-entropy loss decreases monotonically with model size.
- But task loss for reasoning is non-monotonic.



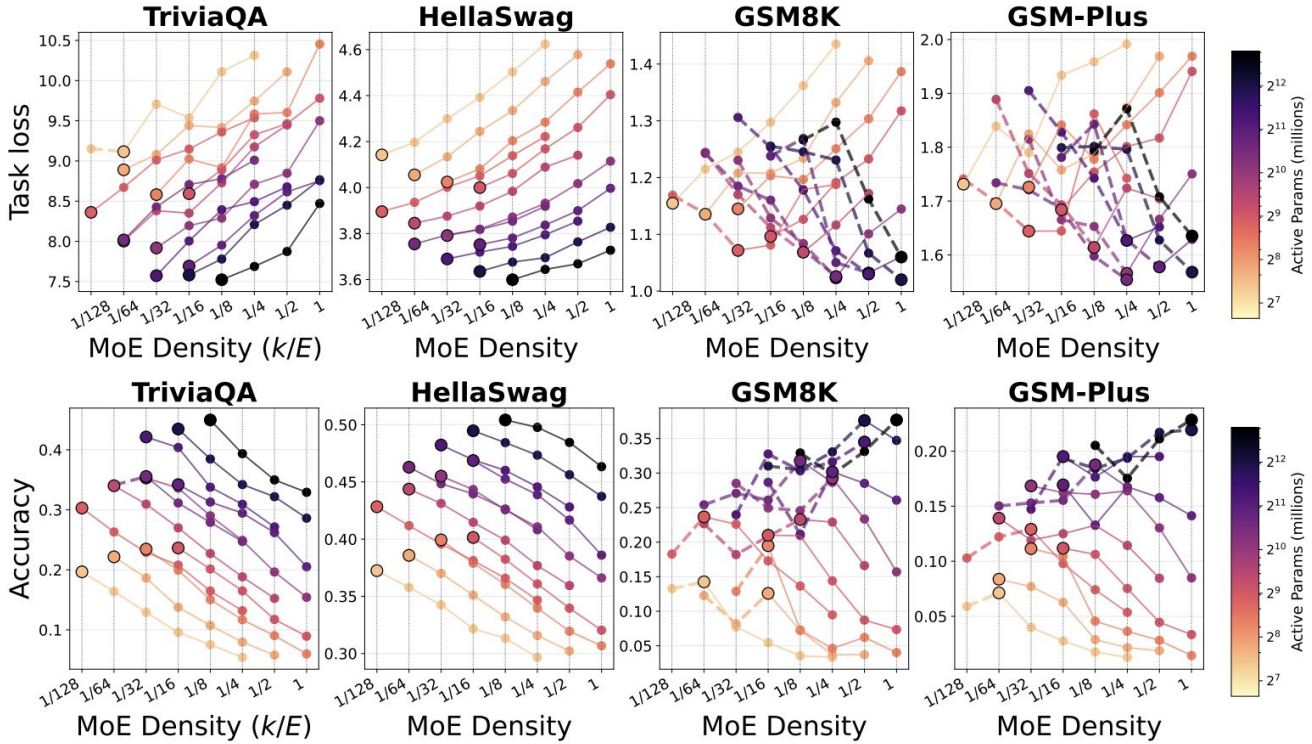
Same Training Loss, Different Reasoning Performance

- For the same cross-entropy loss, higher sparsity leads to worse reasoning performance.
- At the same loss, denser MoE shows better reasoning performance.



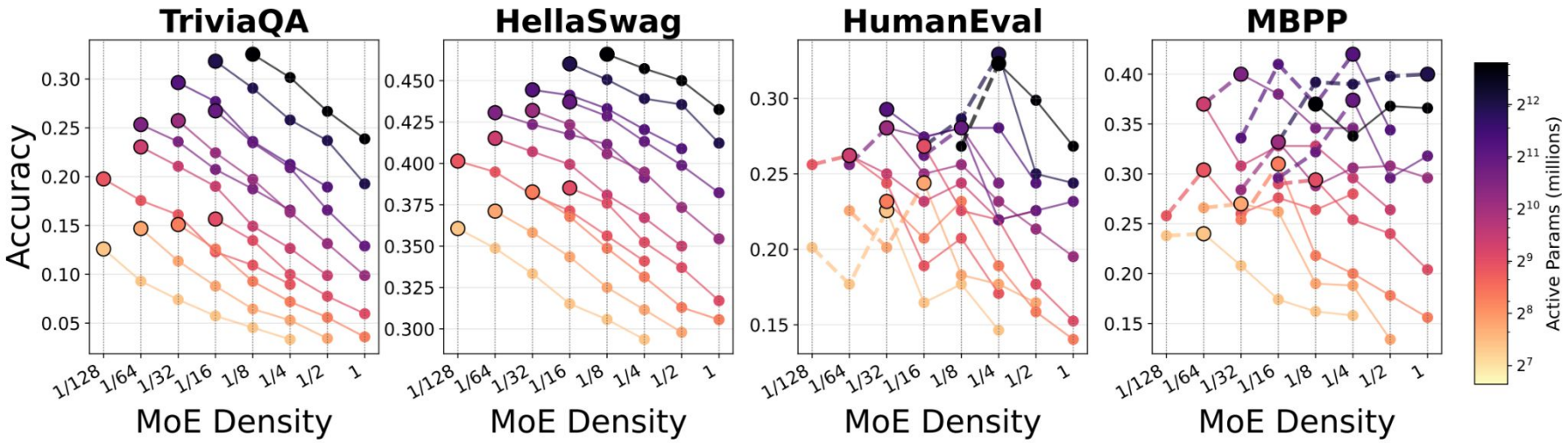
Sparsity Matters for Reasoning in MoE

For reasoning tasks (GSM8K, GSM-Plus), sparsity (1-Top-*k*/*E*) matters.



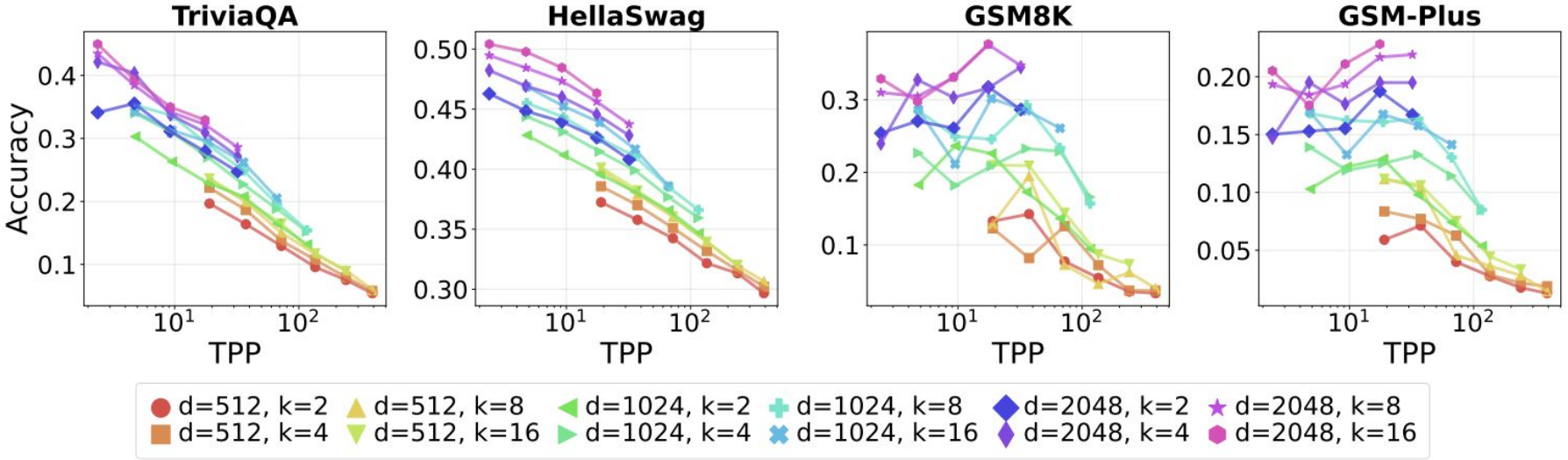
Code-Trained Models Show the Same Trend

Code benchmarks show a similar trend to math



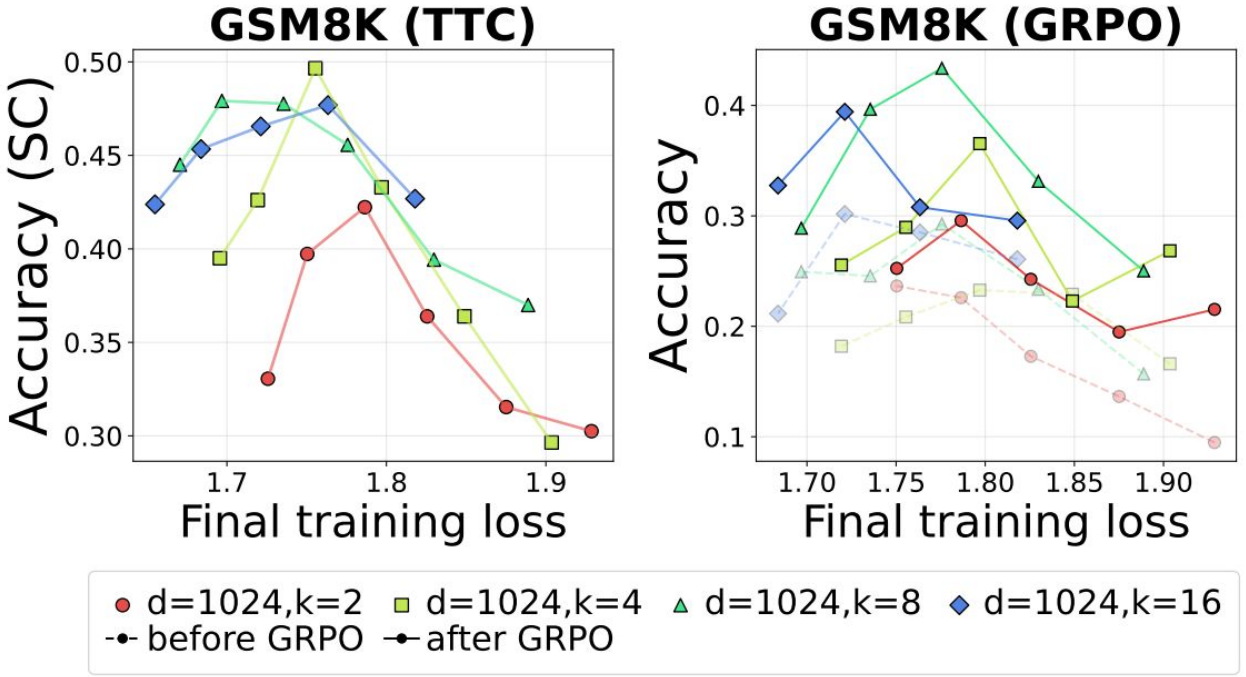
Tokens Per Parameter Matter for Reasoning in MoE

- total tokens per parameter (TPP) = training tokens / total parameters
- For reasoning tasks (GSM8K, GSM-Plus), TPP is a key factor.
- Top-k also matters.



The inverted U-shaped trend persists even after RL

- Test-time compute (Self-Consistency) and GRPO improve accuracy.
- But the effect of sparsity remains.



Conclusion

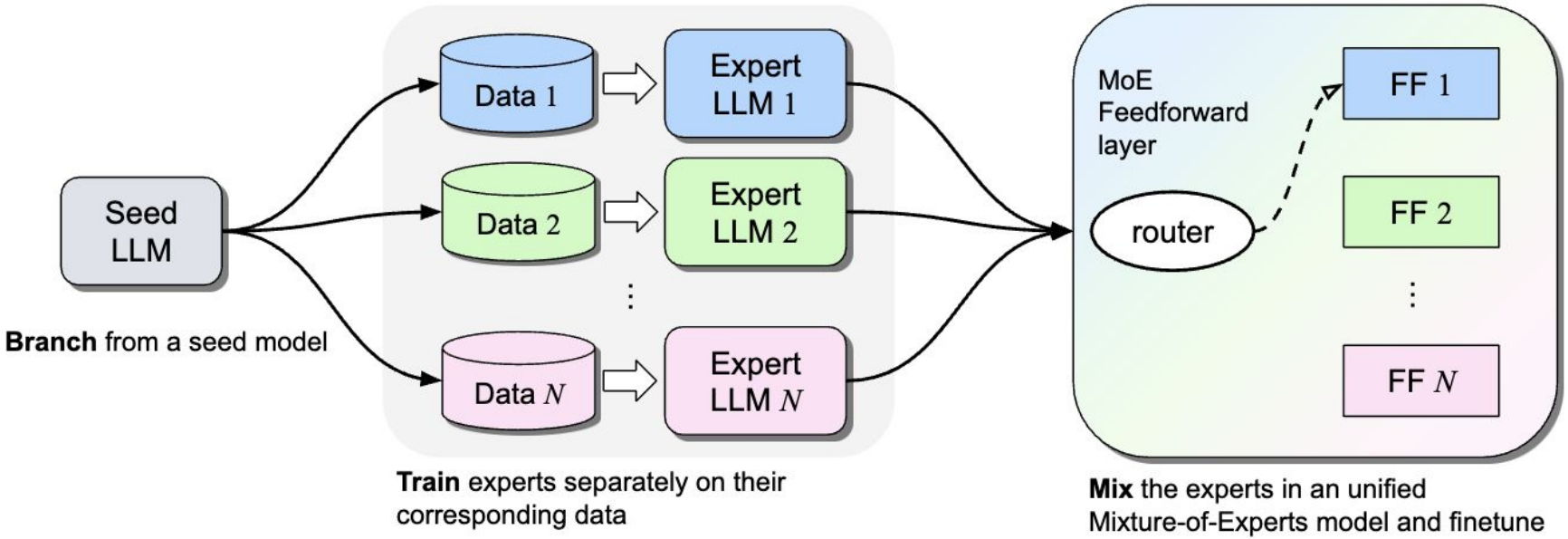
- We propose **Drop-Upcycling**, enabling strong MoE performance from pretrained dense models.

- We characterize the optimal sparsity of MoE models under a fixed compute budget.
  - **Total tokens per parameter and the number of active parameters are key factors in reasoning performance.**

Thank you

LinkedIn: Taishi Nakamura

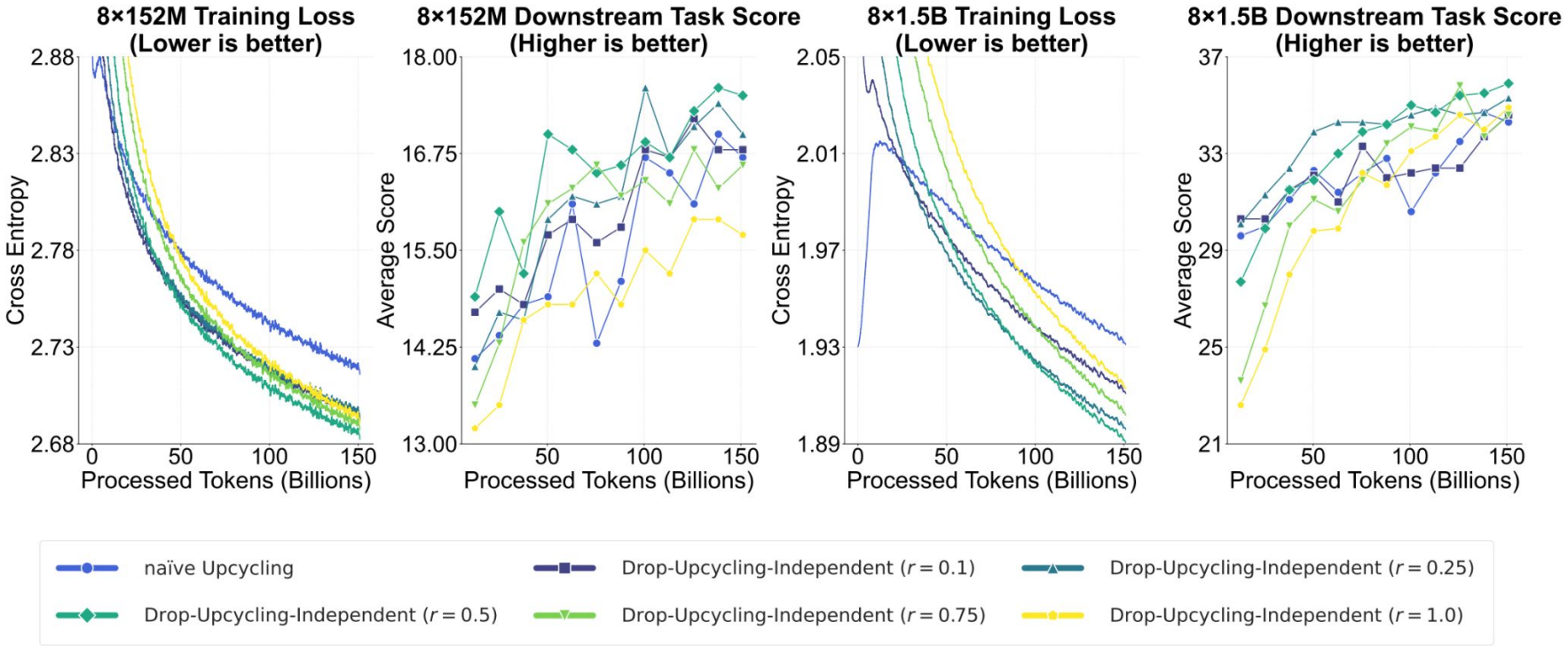
Appendix: Branch-Train-MiX



Source: Sukhbaatar et al., "Branch-Train-MiX: Mixing Expert LLMs into a Mixture-of-Experts LLM" COLM 2024, Figure 1.

Appendix: Impact of Re-initialization Ratio *r*

***r* = 0.5** best balances transfer and specialization.



Appendix: How Does Model Depth Affect Reasoning?

Adding 32-layer models does not change the U-shaped relationship between performance and TPP.