IDEAS NCBR

Fine-Grained Mixture-of-Experts

LLM-Random Team @ IDEAS

- Jakub Krajewski
- Sebastian Jaszczur
- Jan Ludziejewski
- Maciej Pióro
- Szymon Antoniak
- Michał Krutul
- Tomasz Odrzygóźdź
- Marek Cygan

Outline

Outline

1. Introduction
2. Granularity
3. Experiments

Motivation: Neural Scaling Laws

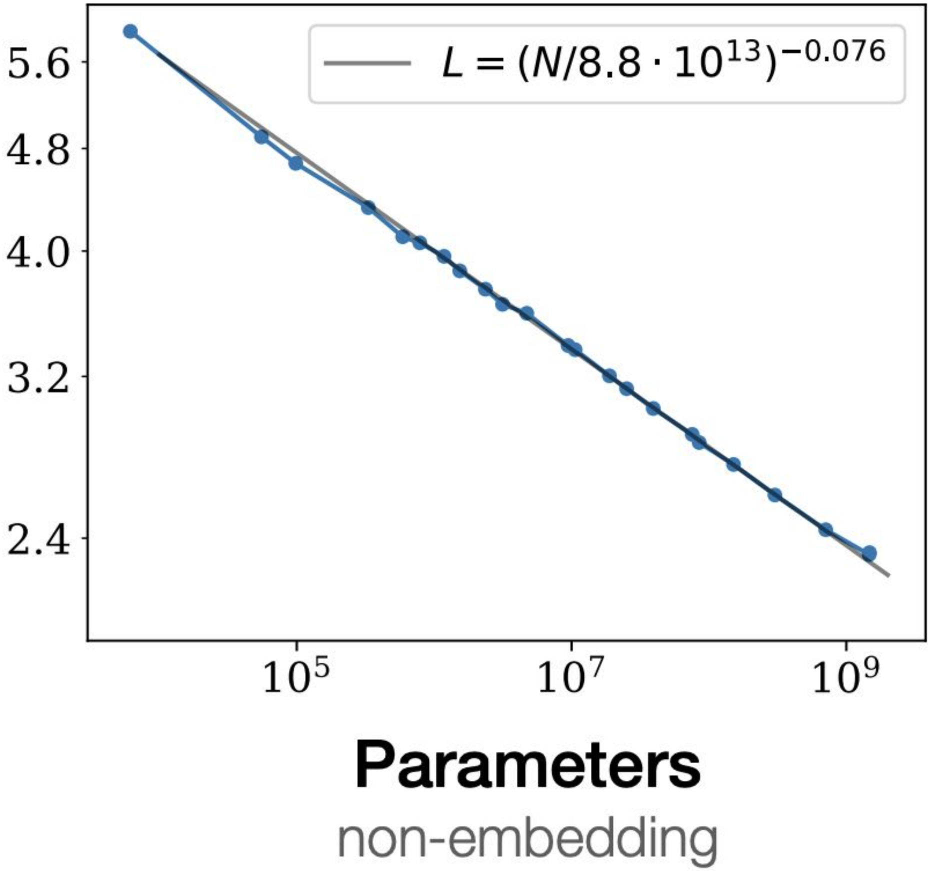
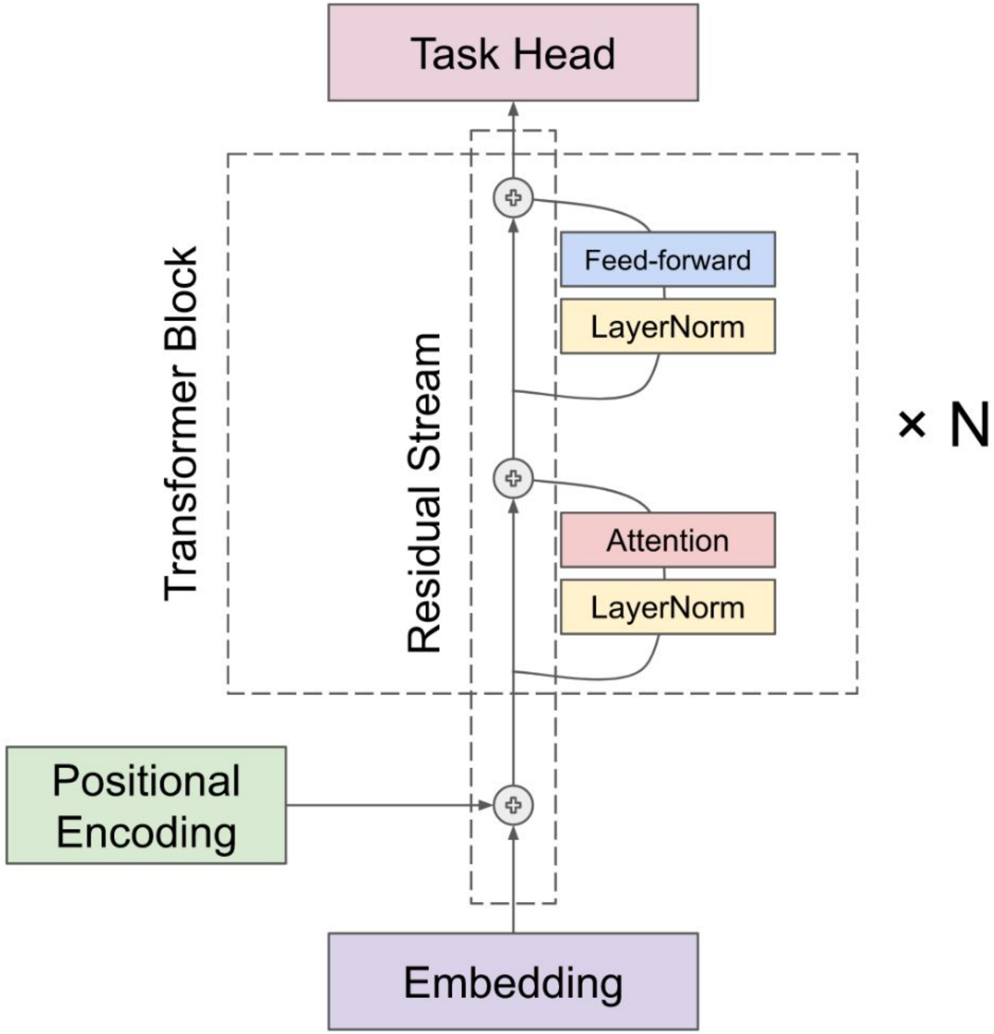


Figure from *Kaplan et al. 2020,*
Scaling Laws for Neural Language Models

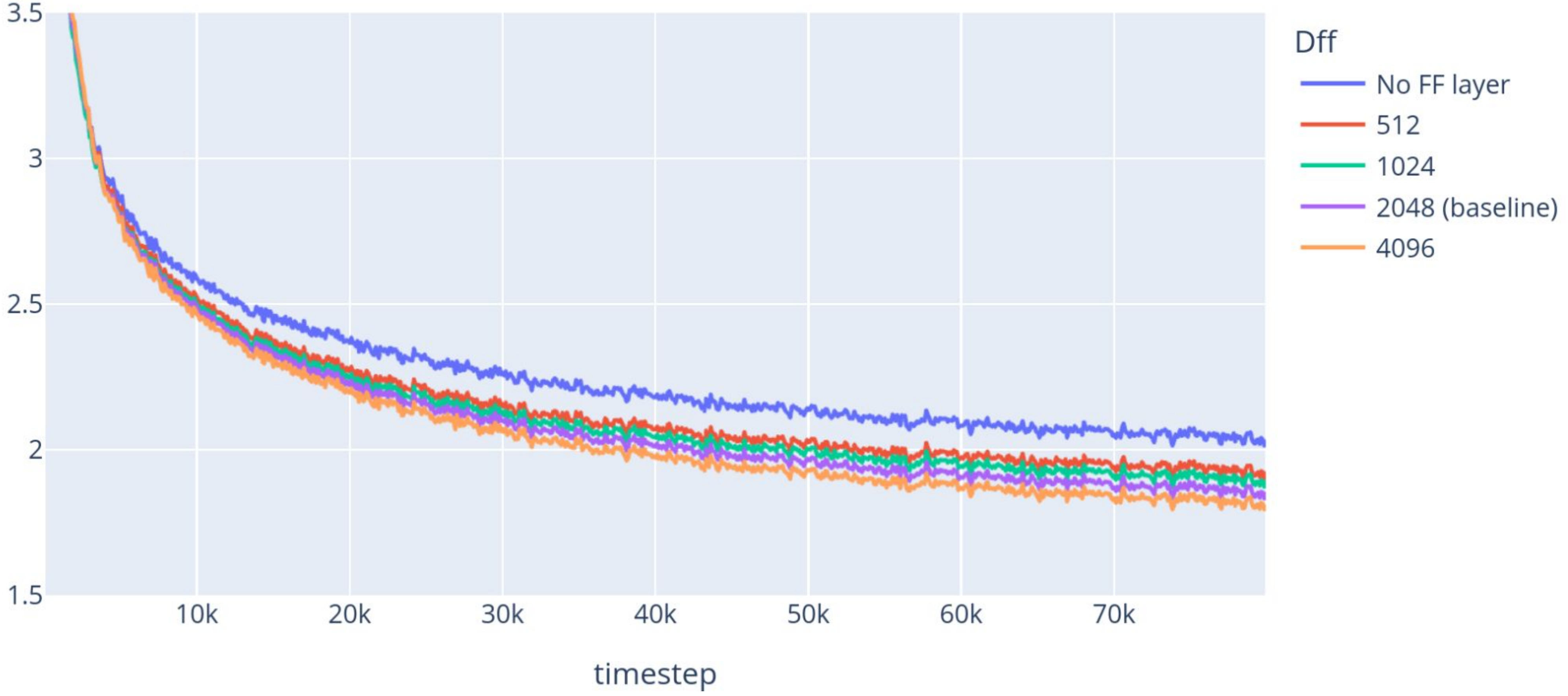
Motivation: Neural Scaling Laws

$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_c}{D} \right]^{\alpha_D}$$

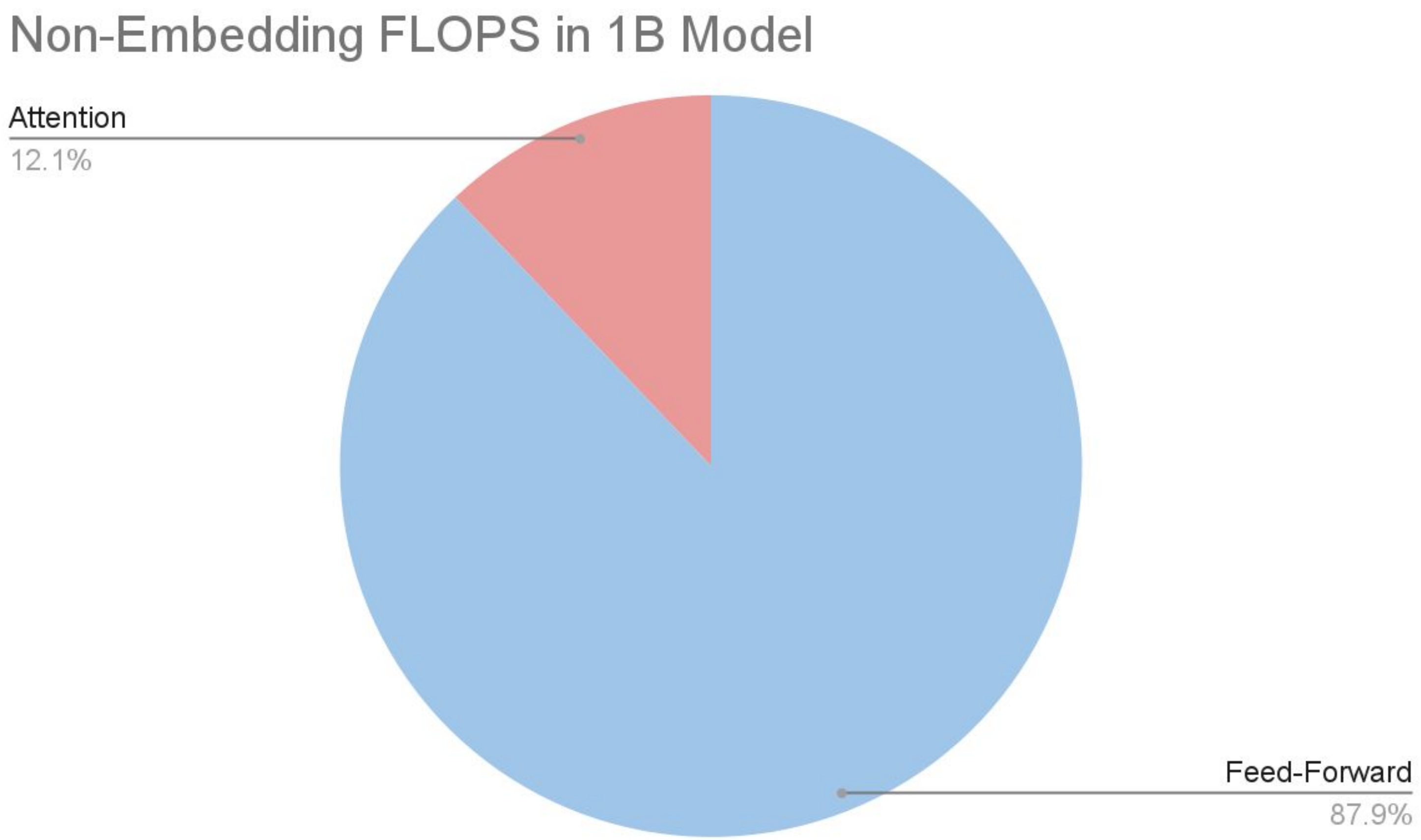
The Transformer



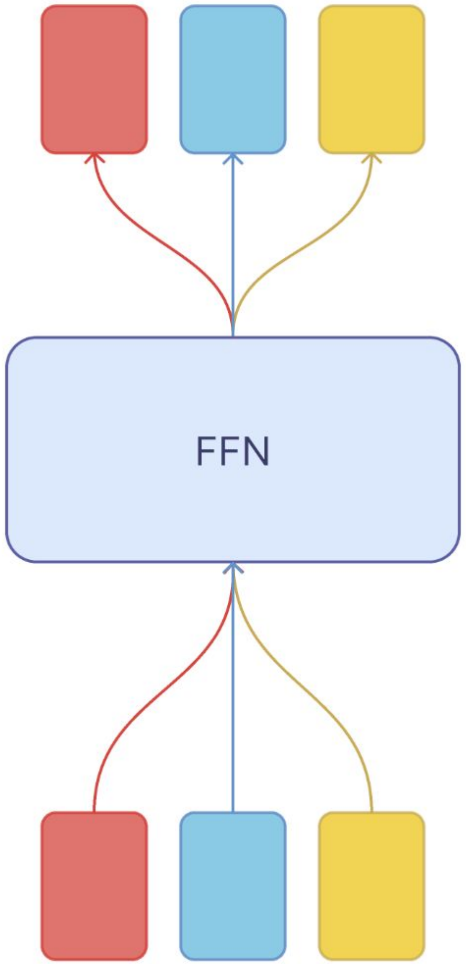
Feed-Forward Layer Width



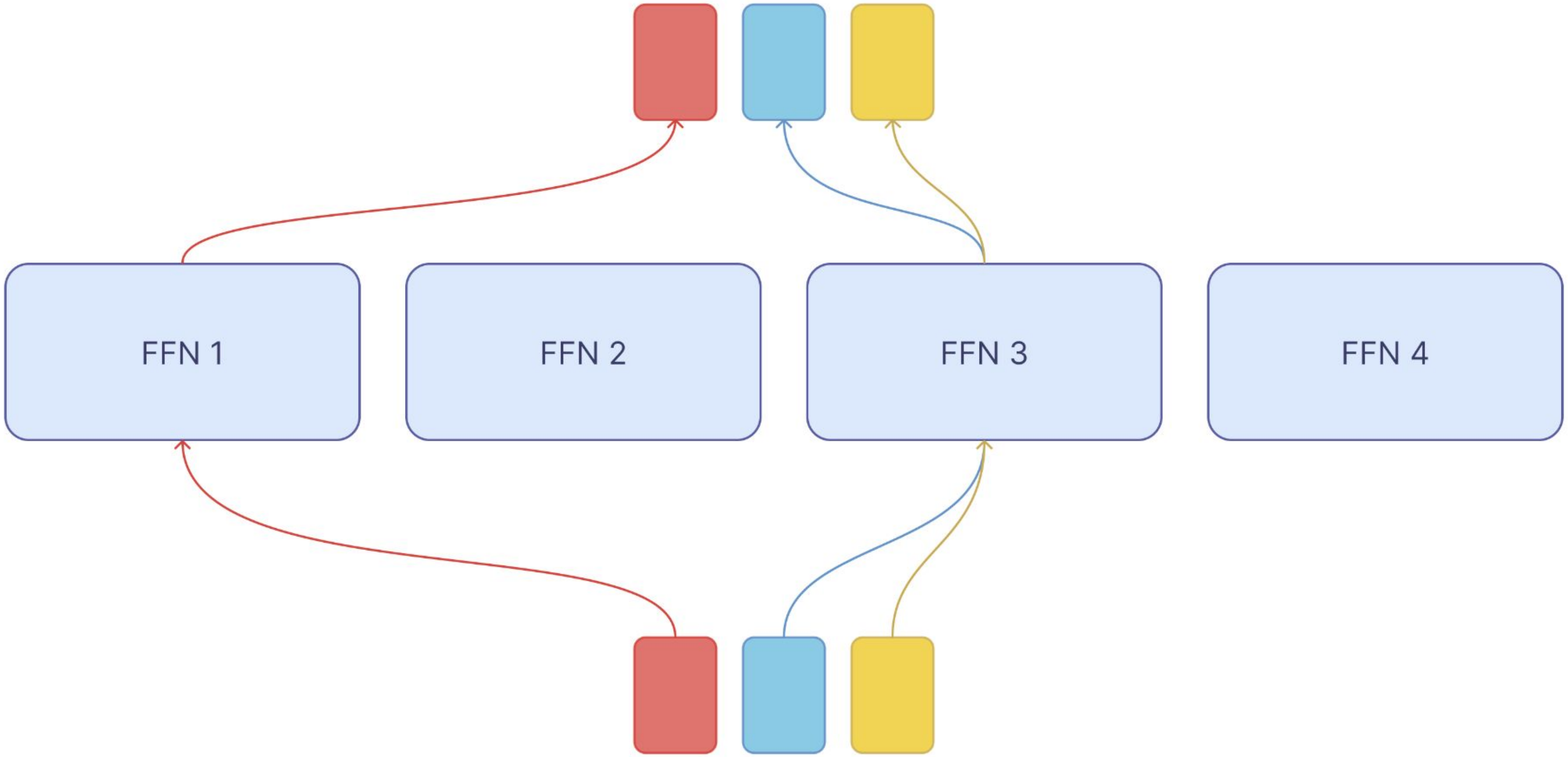
Computation in Feed-Forward Layer



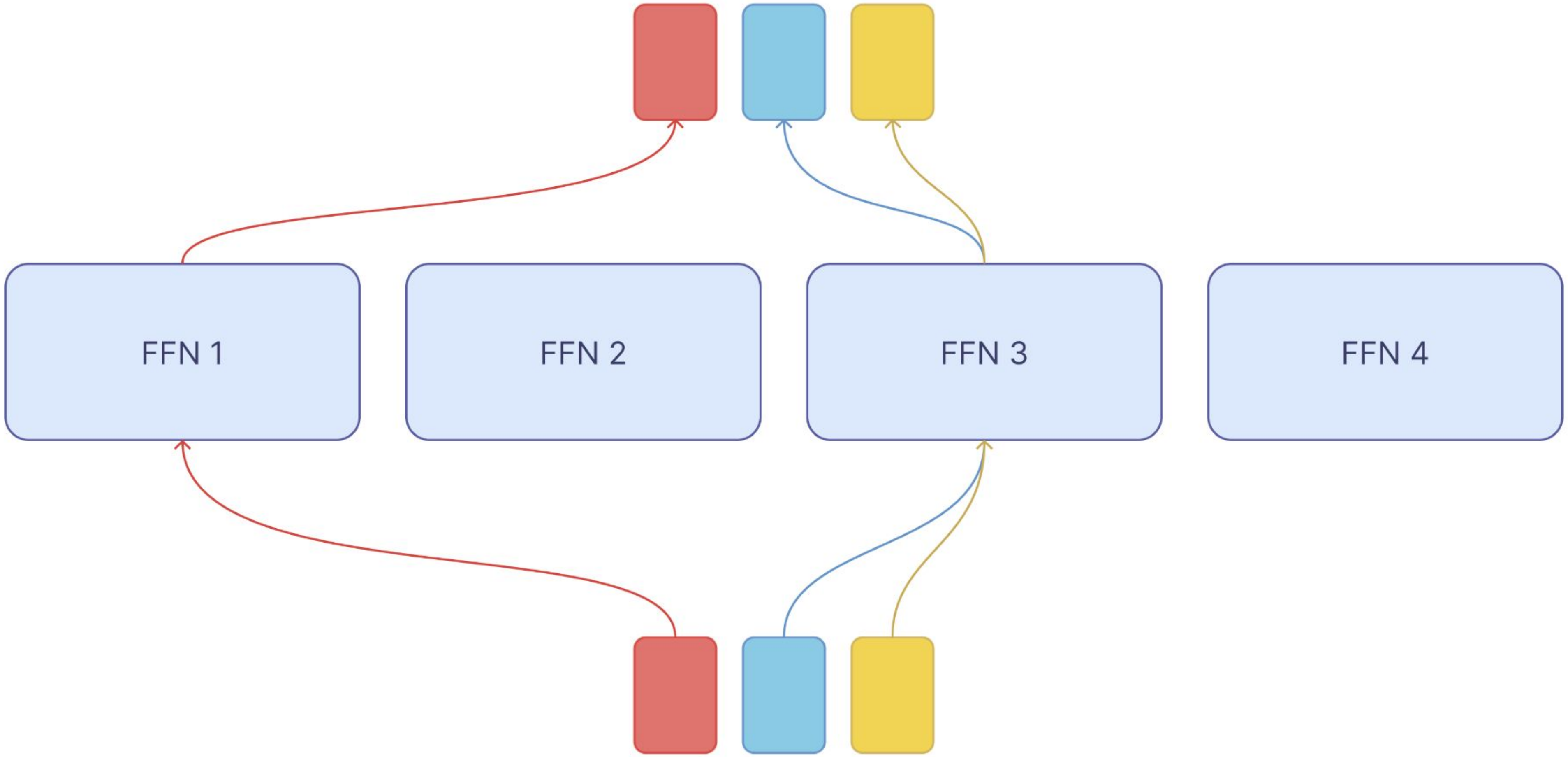
Standard Feed-Forward Layer



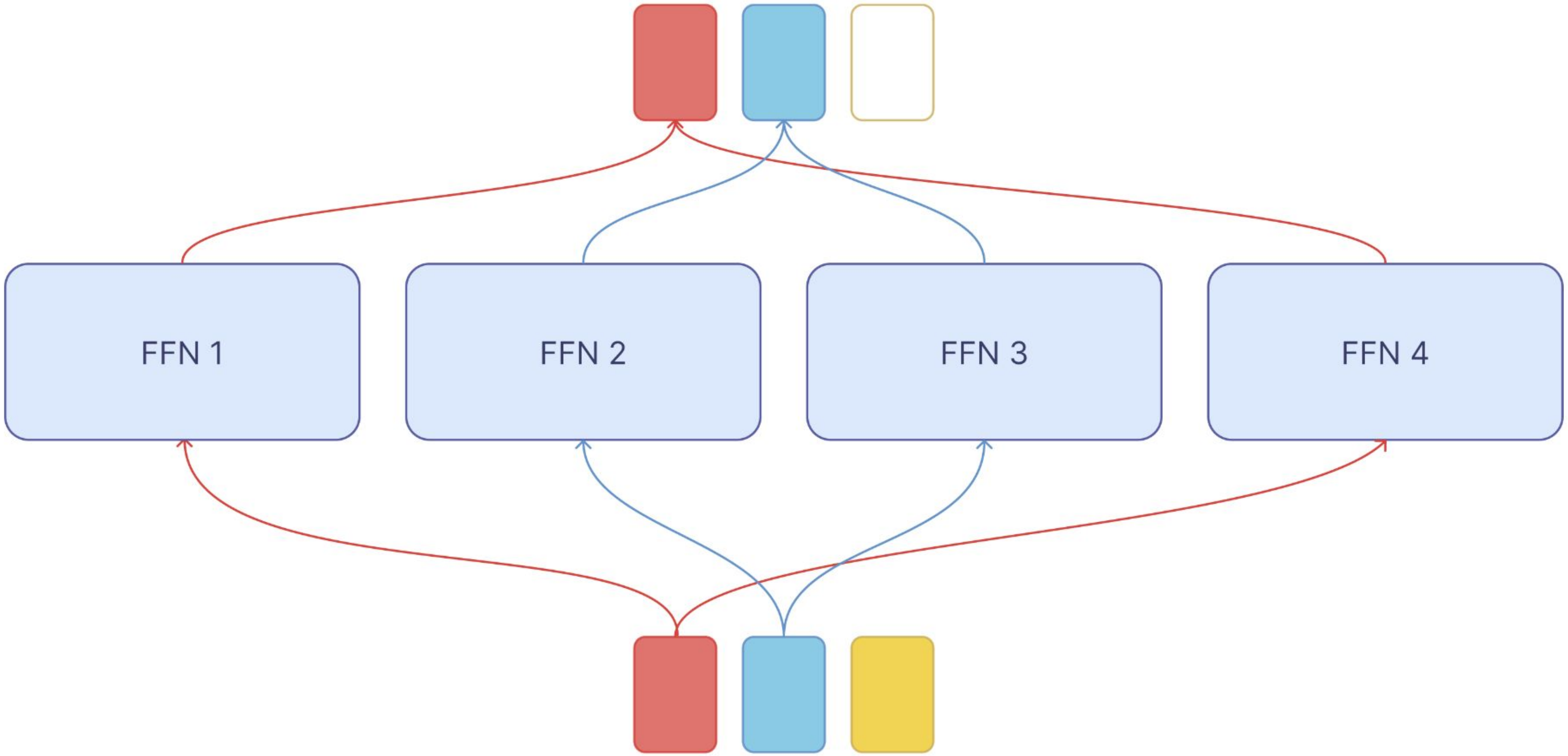
Mixture-of-Experts Layer



Mixture-of-Experts: Token Choice



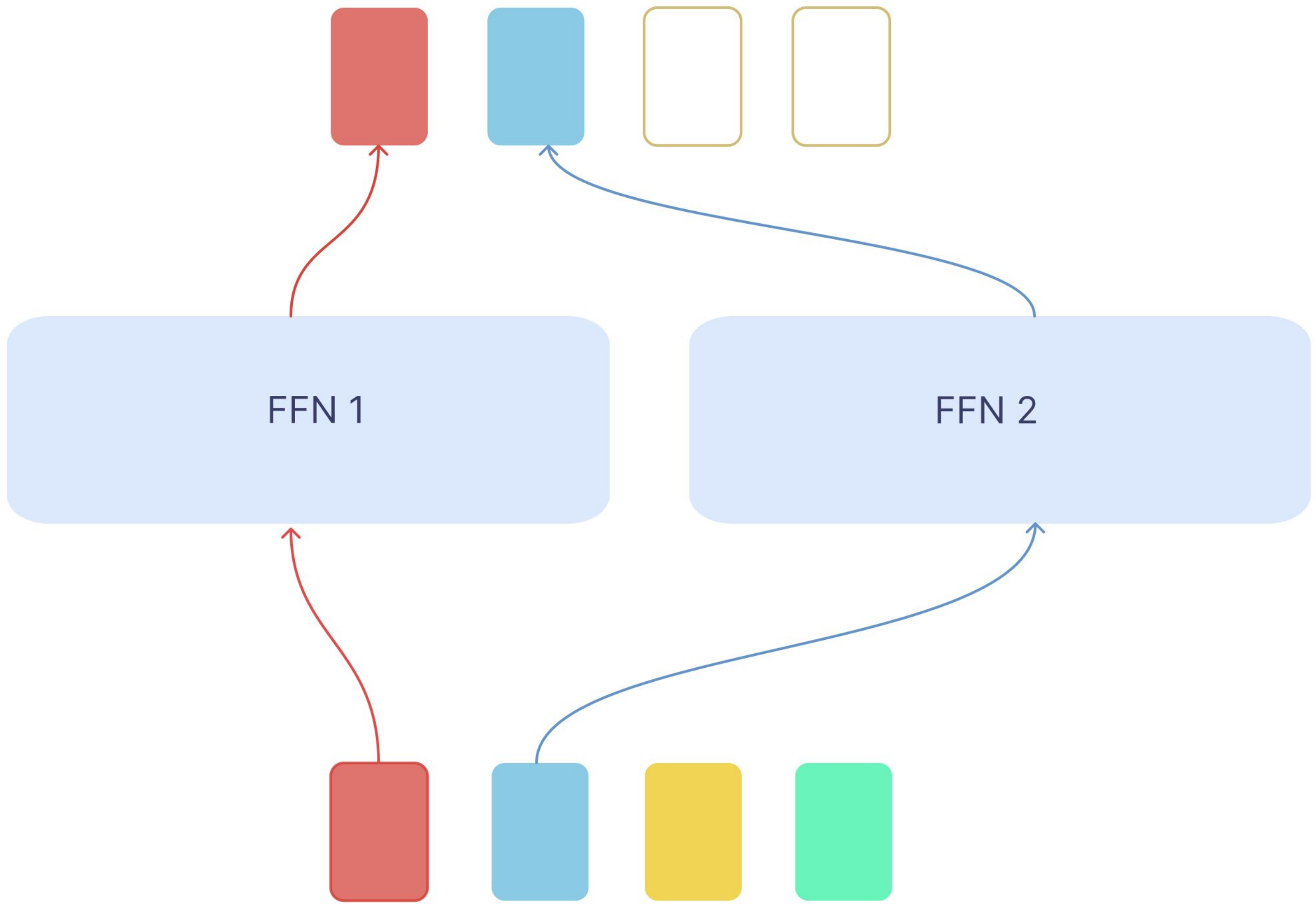
Mixture-of-Experts: Expert Choice



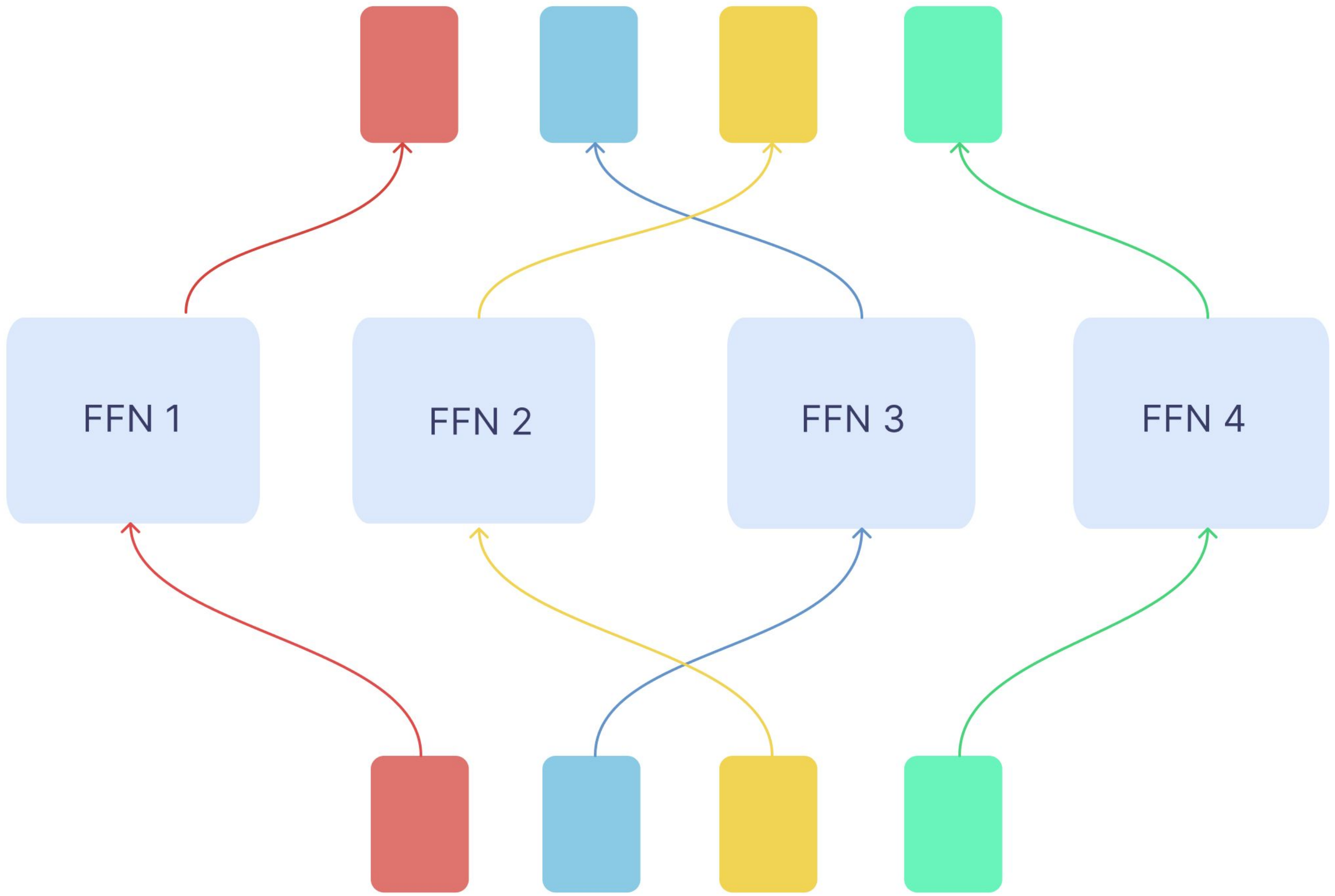
Outline

1. Introduction
2. Granularity
3. Experiments

Standard Mixture-of-Experts



Granular Mixture-of-Experts



Definition

Suppose we fix the number of parameters and computational budget in the MoE model.

By *granularity* we will understand

$$g = \frac{d_{ff}}{d_{expert}}.$$

Main question

+ Mixture-of-Experts (*small* granularity): studied
+ Sparse model (*extreme* granularity): studied
+ What’s in between?

Main question

+ Mixture-of-Experts (*small* granularity): studied
+ Sparse model (*extreme* granularity): studied
+ What's in between?

Main question

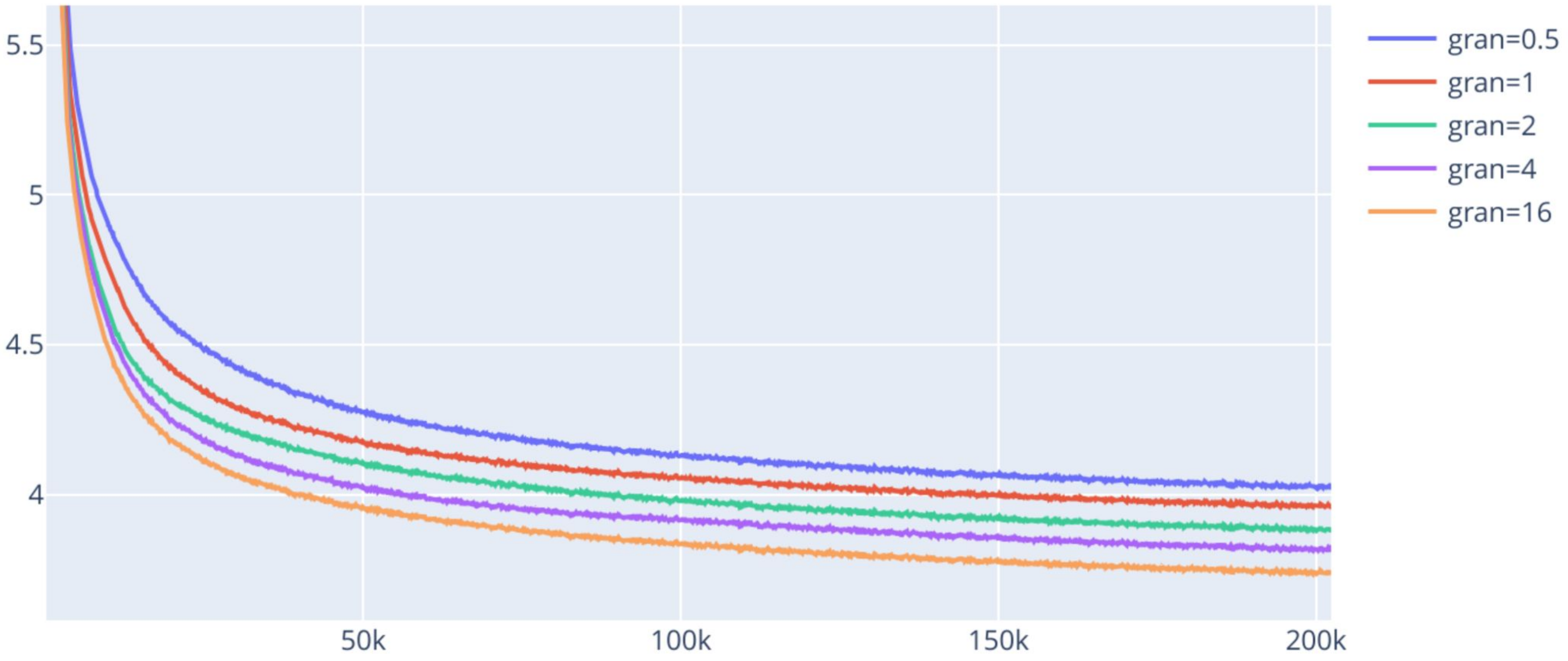
+ Mixture-of-Experts (*small* granularity): studied
+ Sparse model (*extreme* granularity): studied
+ What’s in between?

Main question

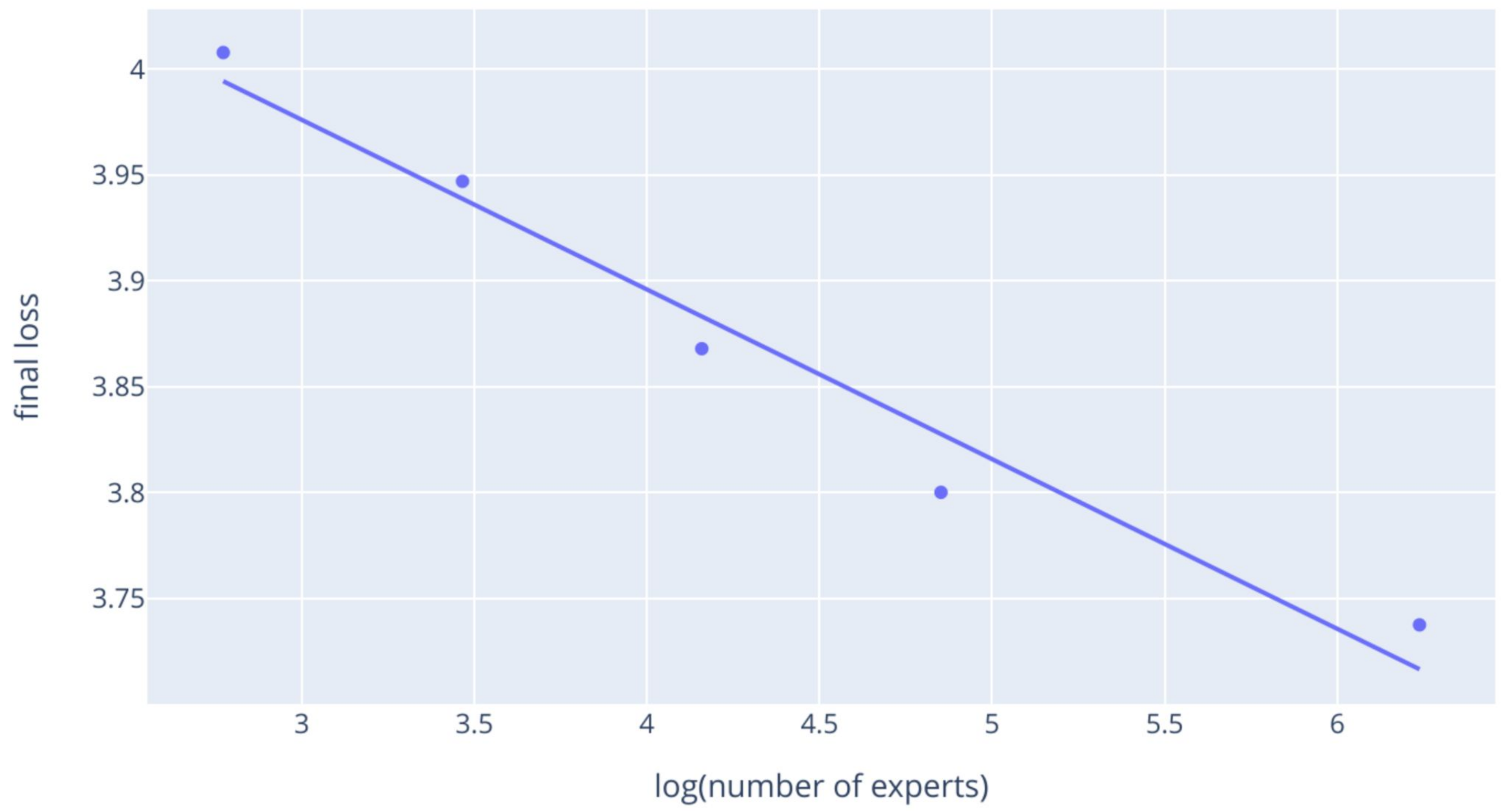
We want to study the relation between **granularity** and **the final model performance**.

Outline

90M Model: Granularity vs Loss



90M Model: Granularity vs Loss



The cost

How do we pay for the lower loss?

- More expensive shuffle operation
- Higher communication cost
- The exact gains depend on the implementation and hardware

The cost

How do we pay for the lower loss?

+ More expensive shuffle operation
+ Higher communication cost
+ The exact gains depend on the implementation and hardware

The cost

How do we pay for the lower loss?

+ More expensive shuffle operation
+ Higher communication cost
+ The exact gains depend on the implementation and hardware

The cost

How to make the benefit practical?

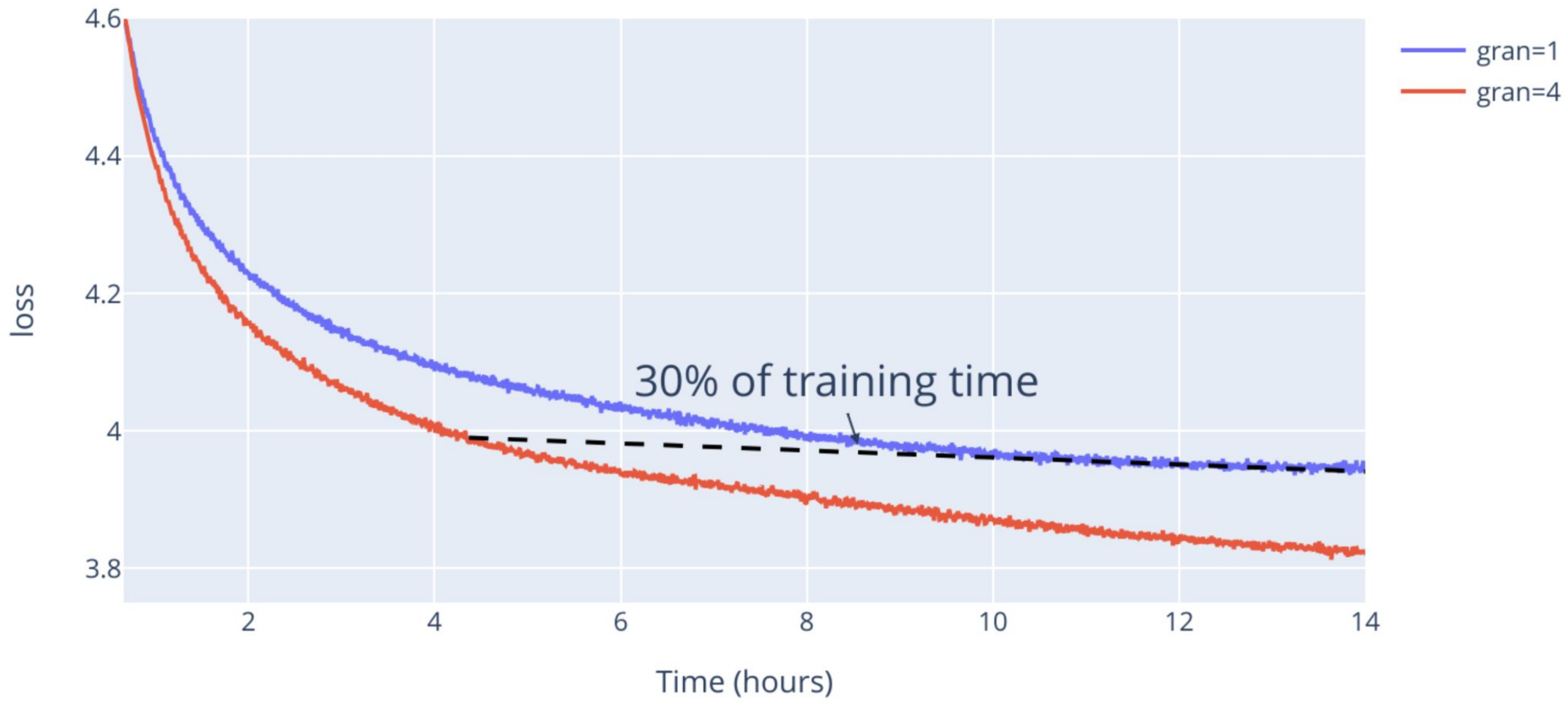
- If we understand the per-step relation between granularity and time, we only need to measure step time for the granular model, which is cheap
- We can also design our training in such a way to make the use of granularity

The cost

How to make the benefit practical?

+ If we understand the per-step relation between granularity and time, we only need to measure step time for the granular model, which is cheap
+ We can also design our training in such a way to make the use of granularity

90M Model: Loss vs Time



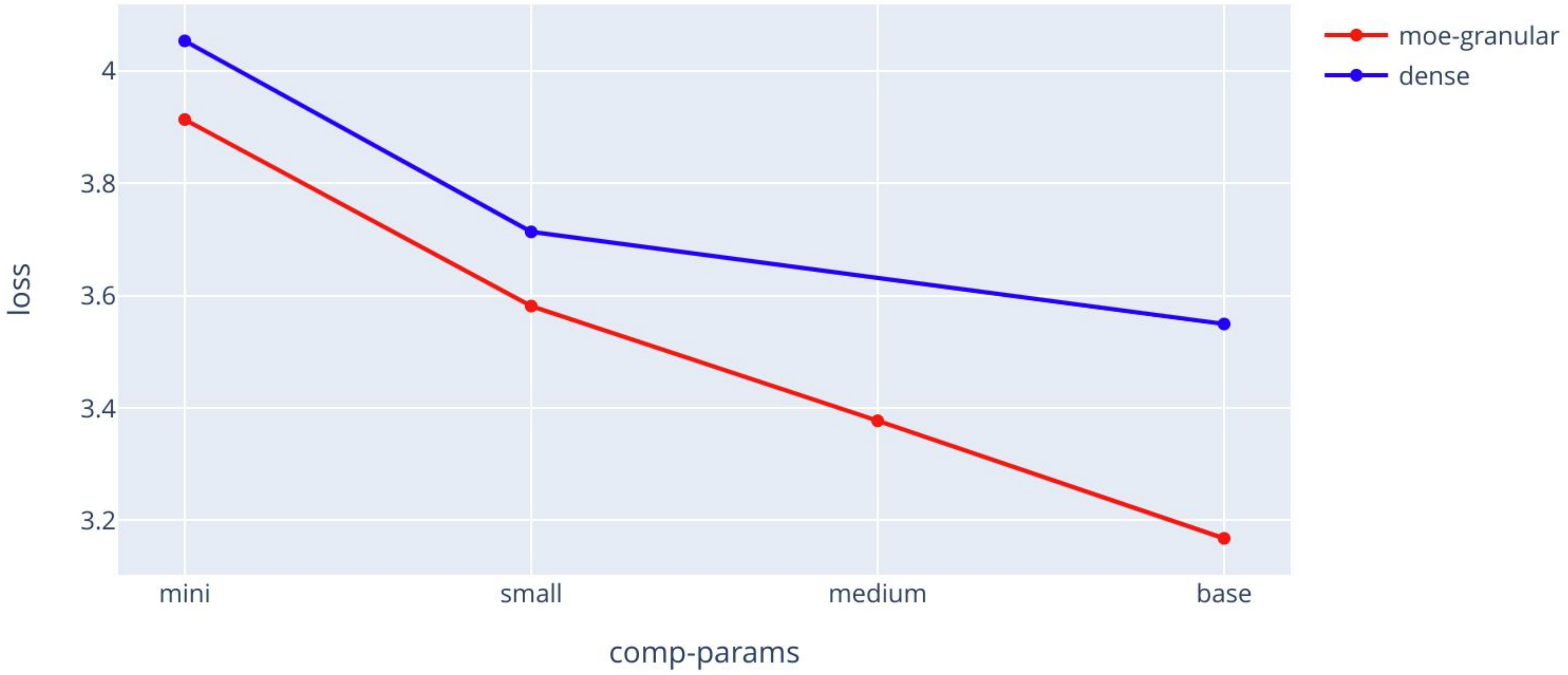
Scaling Up

In the next part of our project we wanted to find out how do these results transfer when **scaling up** the number of parameters. We examined models on four sizes:

+ MoE-mini: **90M** parameters
+ MoE-small: **300M** parameters
+ MoE-medium: **500M** parameters
+ MoE-base: **1.9B** parameters

We compared **granular models** against their **dense counterparts** and **baseline MoE**.

Loss Scaling: Dense vs Granular MoE



The Trend Continues

For our biggest MoE model (1.9B), we need:

+ **28% less steps** to reach the final loss - when training on **10B** tokens
+ **39% less steps** to reach the final loss - when training on **20B** tokens

Bonus

As an addition, we observed other advantages of the granular model:

+ Better scaling with MoE on **every layer** (allows for uniform architecture)
+ **Lower** amounts of token dropping

Final Remarks

+ We present and study a new dimension in scaling MoE Language Models
+ We are currently working on larger-scale experiments
+ Our code is open-sourced at **github.com/llm-random**
+ Feel free to contact us with any questions
+ The paper will be out soon!

Thank you for your attention!