香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

COMP 4901B

Large Language Models

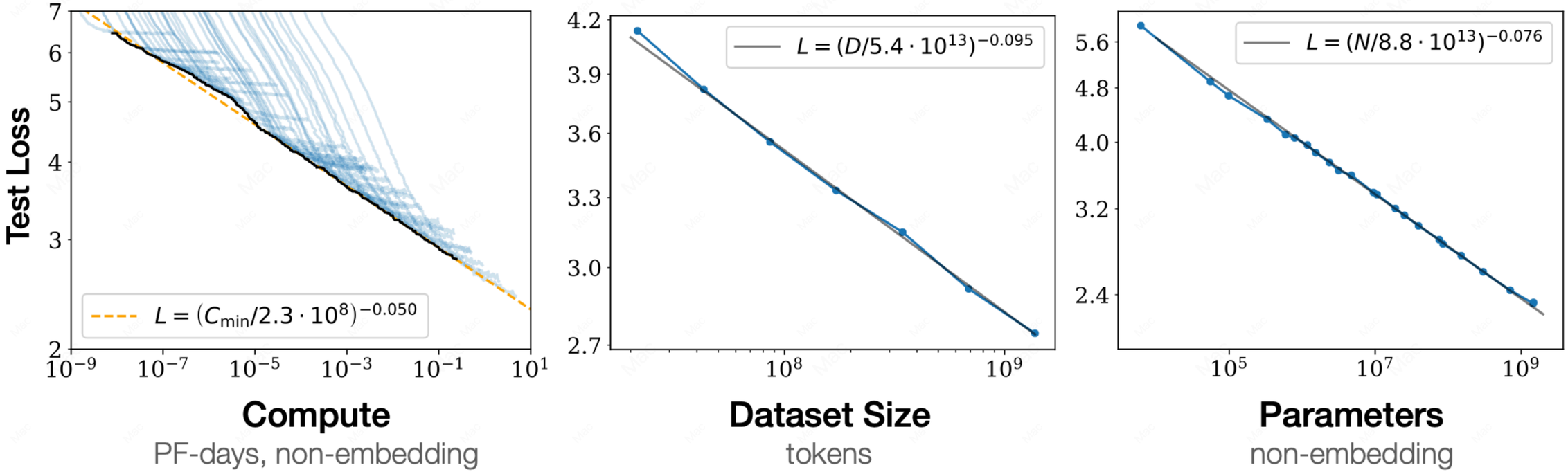
MoE LLMs and Review

Junxian He

Nov 26, 2025

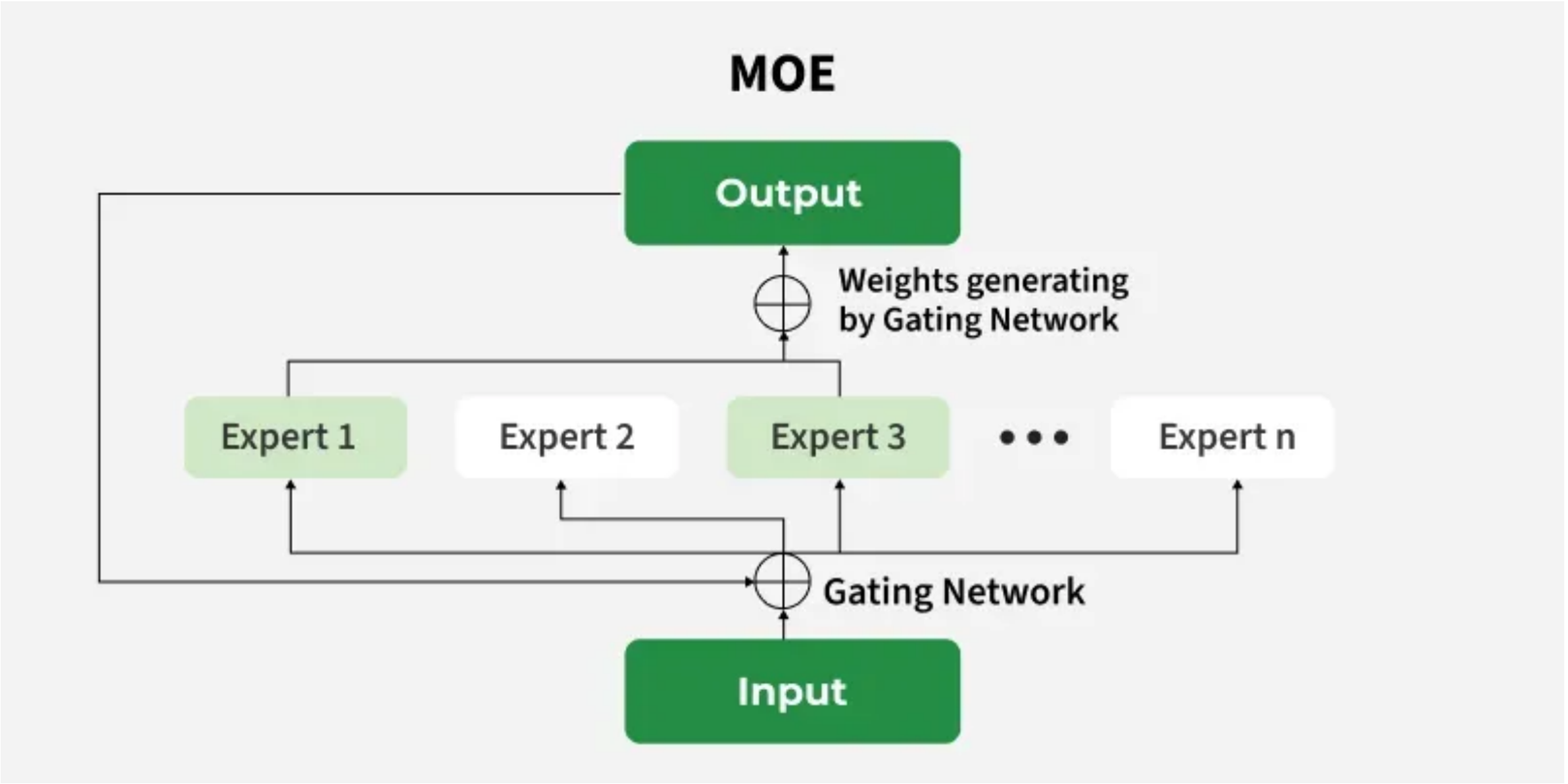
Challenges of Scaling Model Sizes

Scaling law tells us to scale up model sizes

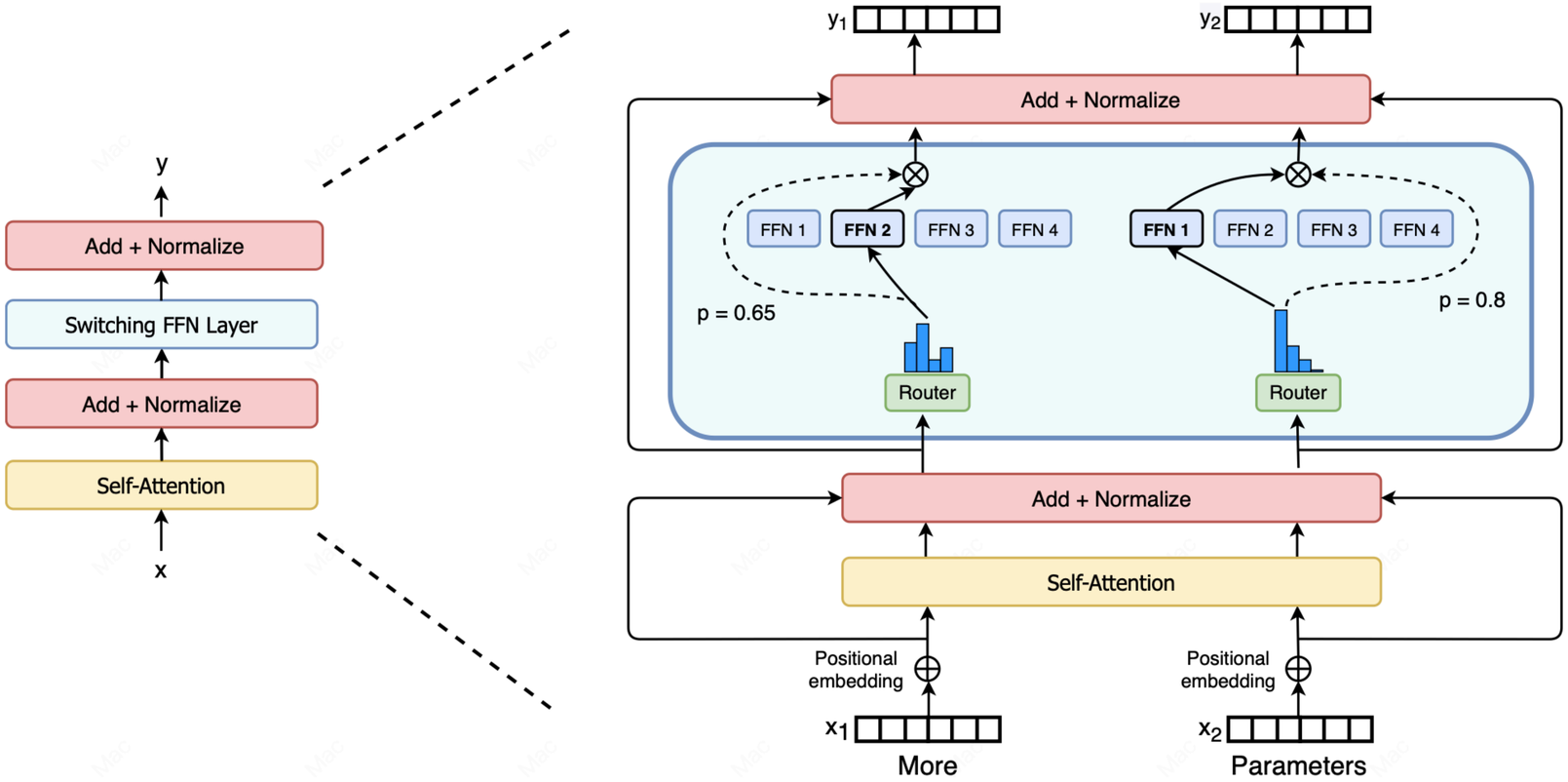


However, larger model sizes require more compute to train and causes higher latency

Mixture of Experts (MoE) in Traditional Machine Learning



MoE Transformer Language Models



Mixture of FFN Blocks

For each token at each layer, only a small fraction (e.g., 2 or 3) experts are activated by the router, thus this is also referred to as SPARSE models

Fedus et al. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. 2021

Sparse Routing

$$h(x) = W_r \cdot x$$ Logits of different experts

$$p_i(x) = \frac{e^{h(x)_i}}{\sum_j^N e^{h(x)_j}}.$$ Gate value, this is softmax

Sparse Routing: only top-k experts are used during both training and test time

$$y = \sum_{i \in \mathcal{T}} p_i(x) E_i(x).$$

Weighted addition of each expert's output

Fedus et al. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. 2021

Load Balancing

Only next-token prediction loss may learn to only use a few experts

Vicious cycle: When a certain subset experts are chosen, next-token prediction will optimize them to suit the input, then these experts are more likely to be chosen

Auxiliary Load Balancing Loss

$$\text{loss} = \alpha \cdot N \cdot \sum_{i=1}^{N} f_i \cdot P_i$$

where f_i is the fraction of tokens dispatched to expert i,

$$f_i = \frac{1}{T} \sum_{x \in \mathcal{B}} \mathbb{1}\{\text{argmax}\, p(x) = i\}$$

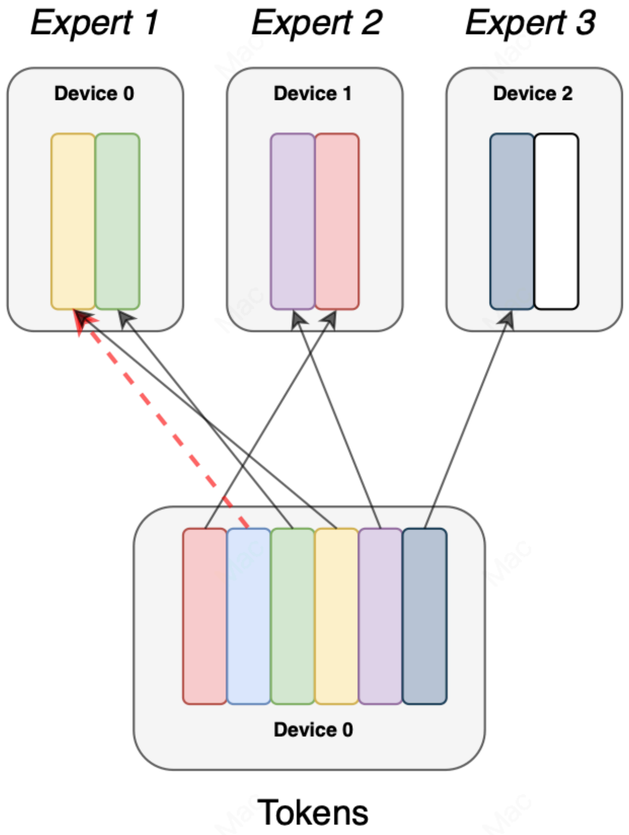
and P_i is the fraction of the router probability allocated for expert i, [2]

$$P_i = \frac{1}{T} \sum_{x \in \mathcal{B}} p_i(x).$$

Fedus et al. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. 2021

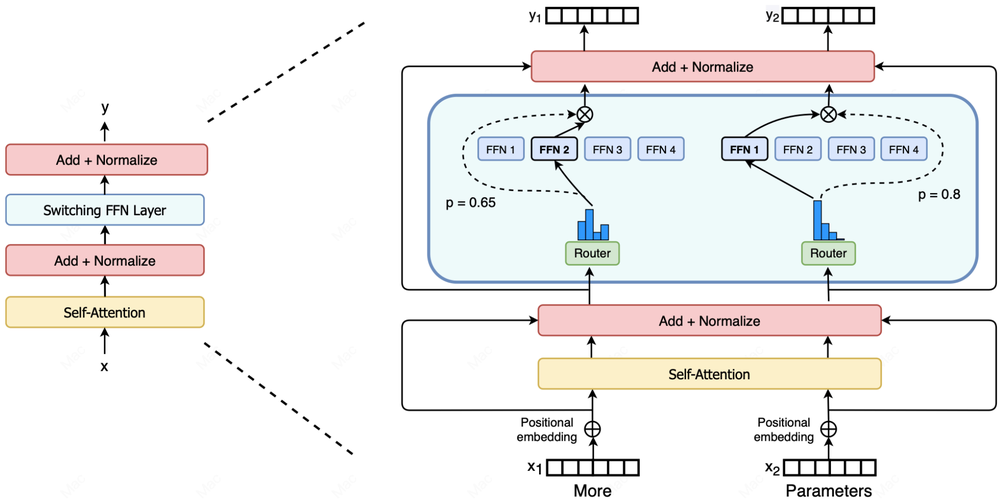
Why MoE?

MoE models support easier model parallel across different GPUs. It can easily split models



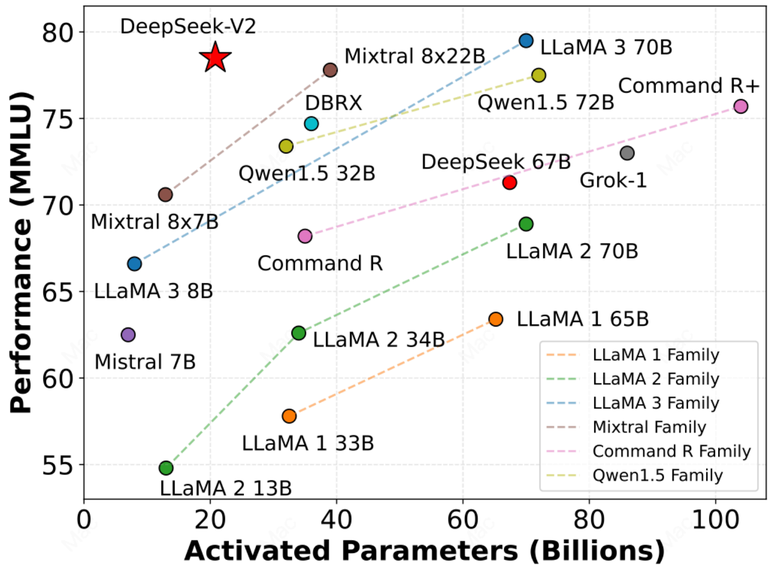
MoE models have major infra-wise benefits when scaling compared to dense models

Why MoE?



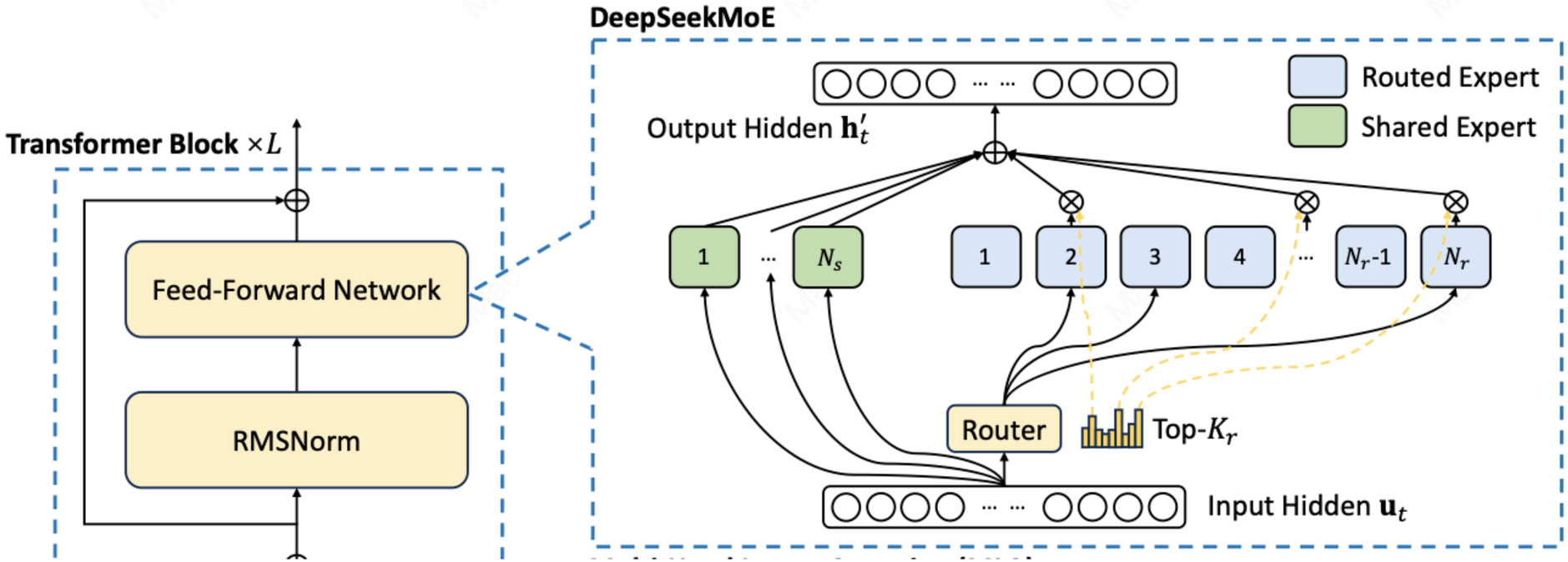
1. Training flops compared to dense model of the same size?
2. Inference flops compared to dense model of the same size?

Theoretically, if the activated parameters are only 1B, then the inference latency can be optimized to be close to an 1B dense model



DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model

DeepSeek MoE



Shared Experts + Routed Expert

For DS-V3, 1 shared expert + 256 routed experts, each token 8 experts are activated

DeepSeek V3 Technical Report

DeepSeek-v3

Shared experts

Routed experts

$$\mathbf{h}_t' = \mathbf{u}_t + \sum_{i=1}^{N_s} \mathrm{FFN}_i^{(s)}(\mathbf{u}_t) + \sum_{i=1}^{N_r} g_{i,t}\,\mathrm{FFN}_i^{(r)}(\mathbf{u}_t),$$

$$g_{i,t} = \frac{g'_{i,t}}{\sum_{j=1}^{N_r} g'_{j,t}},$$

$$g'_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \mathrm{Topk}(\{s_{j,t} | 1 \leqslant j \leqslant N_r\}, K_r), \\ 0, & \text{otherwise}, \end{cases}$$

$$s_{i,t} = \mathrm{Sigmoid}\left(\mathbf{u}_t{}^T \mathbf{e}_i\right),$$

Loss-Free Load Balancing in DeepSeek-v3

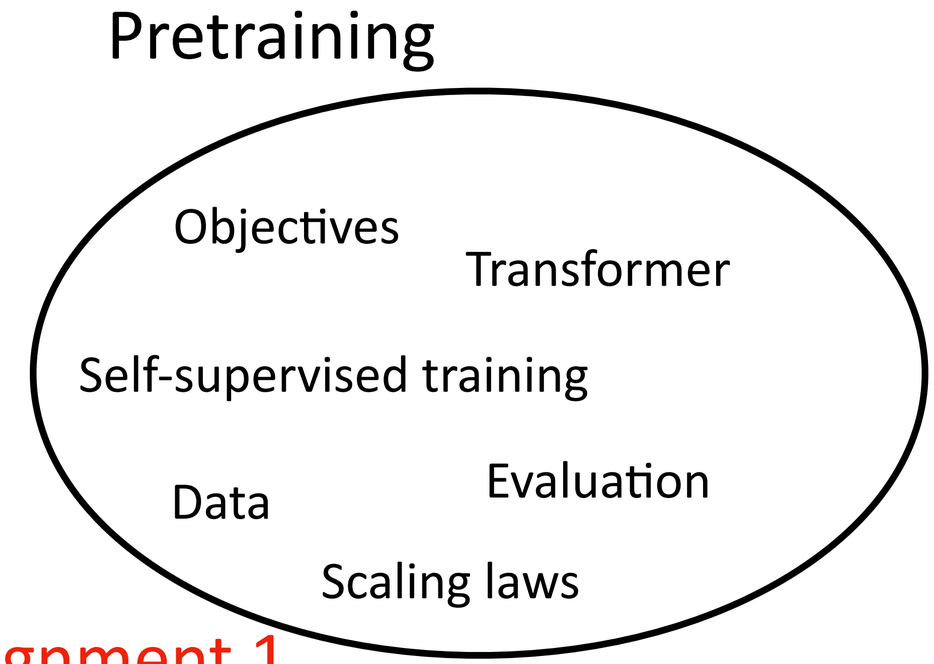
$$g'_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} + b_i \in \text{Topk}(\{s_{j,t} + b_j | 1 \leqslant j \leqslant N_r\}, K_r), \\ 0, & \text{otherwise.} \end{cases}$$

Note that the bias term is only used for routing. The gating value, which will be multiplied with the FFN output, is still derived from the original affinity score $s_{i,t}$. During training, we keep monitoring the expert load on the whole batch of each training step. At the end of each step, we will decrease the bias term by γ if its corresponding expert is overloaded, and increase it by γ if its corresponding expert is underloaded, where γ is a hyper-parameter called bias update speed. Through the dynamic adjustment, DeepSeek-V3 keeps balanced expert load during training, and achieves better performance than models that encourage load balance through pure auxiliary losses.

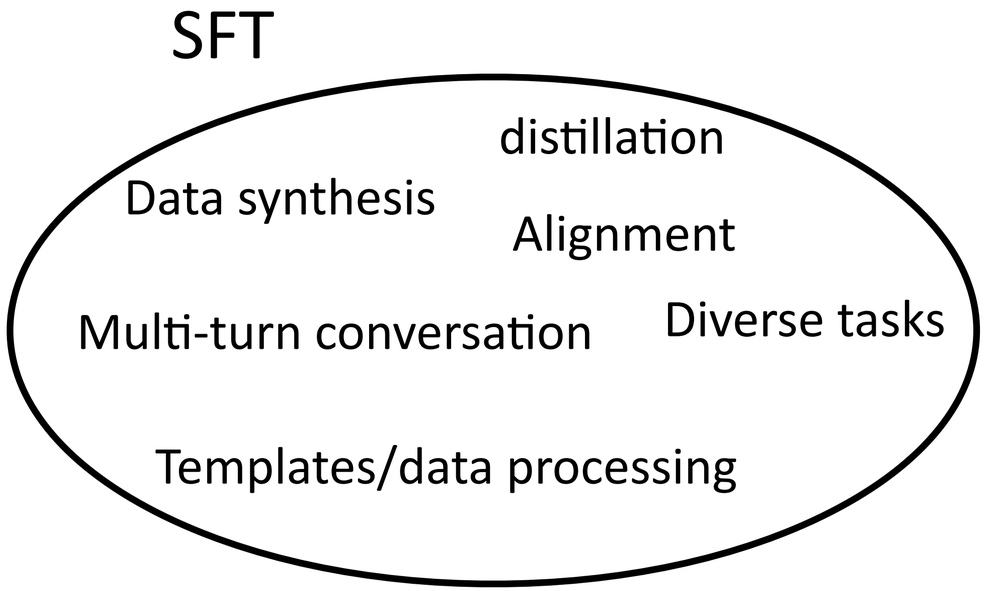
DeepSeek V3 Technical Report

Vibe Coding for HW3

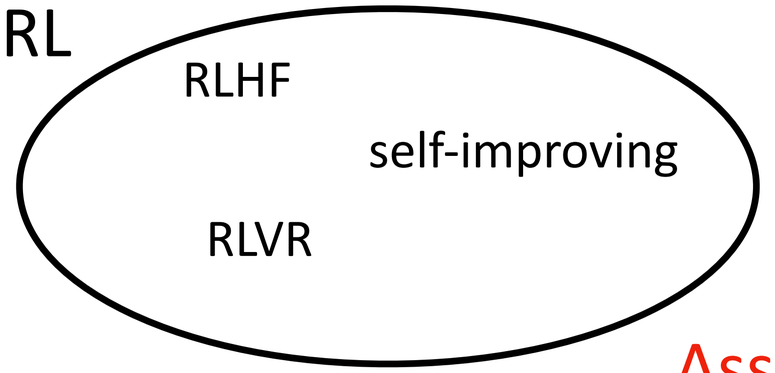
Review — Method



Assignment 1



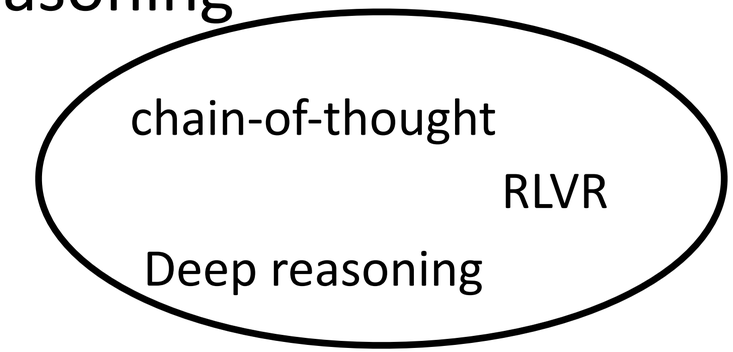
Assignment 2



Assignment 3

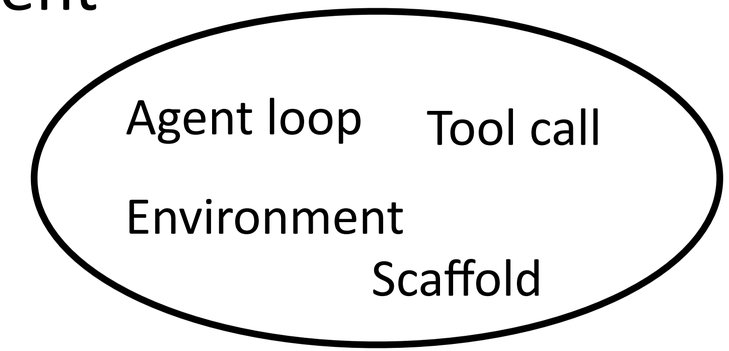
Review — Applications

Reasoning



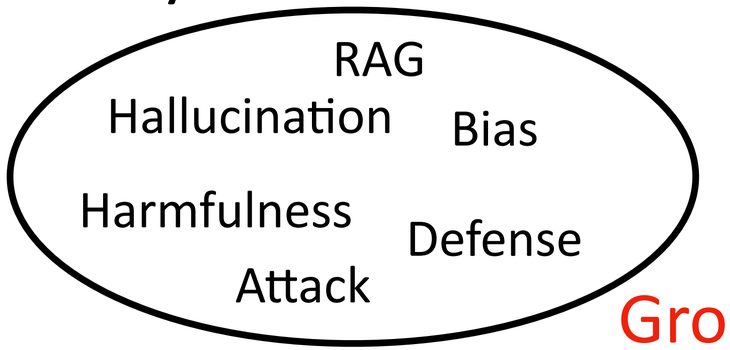
Assignment 3, 4

Agent



Group project + Assignment 4

Ethics/Safety



Group project

About Large Language Models

1. Learning from practice (e.g., programming assignments, projects) is the most important

2. This field is evolving very fast (like many things change every 6 months), some of the course contents may get out-of-date, need to catch up things quickly

3. Always embrace new technologies / applications

4. LLMs allow you to build amazing stuff quickly that can make actual impacts

Student Feedback Questionnaire (SFQ)

Anonymous and not mandatory, ddl tomorrow (Nov 29)

https://sfq-survey.ust.hk

Thank You for taking the course!