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

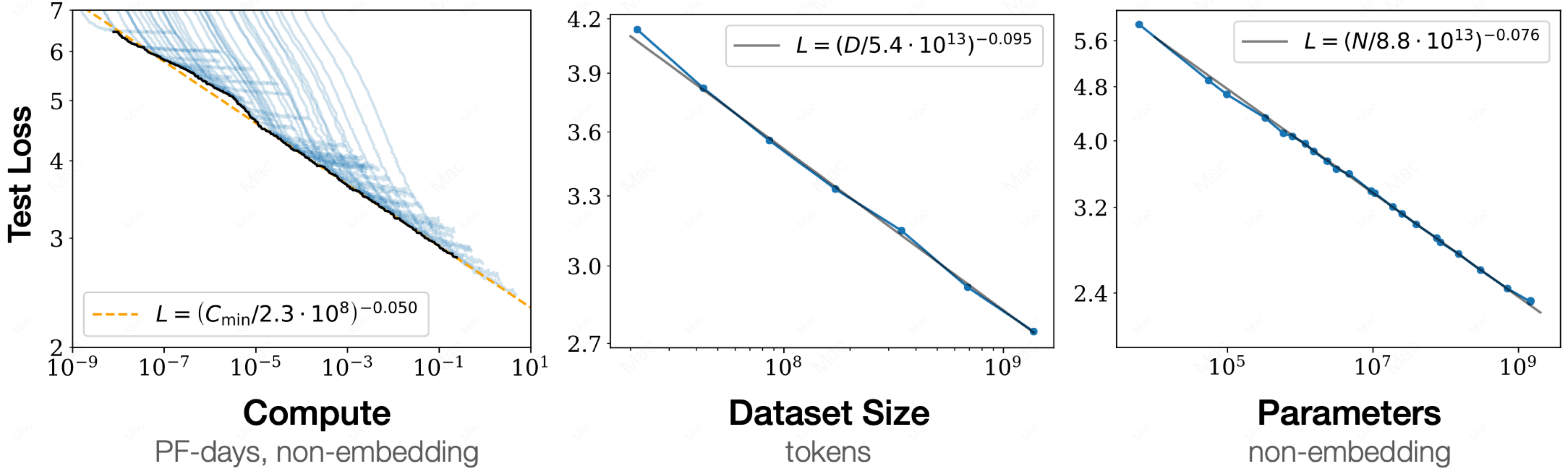
Final exam:

Dec 11 (Thurs) 8:30am-10:30am

LTC

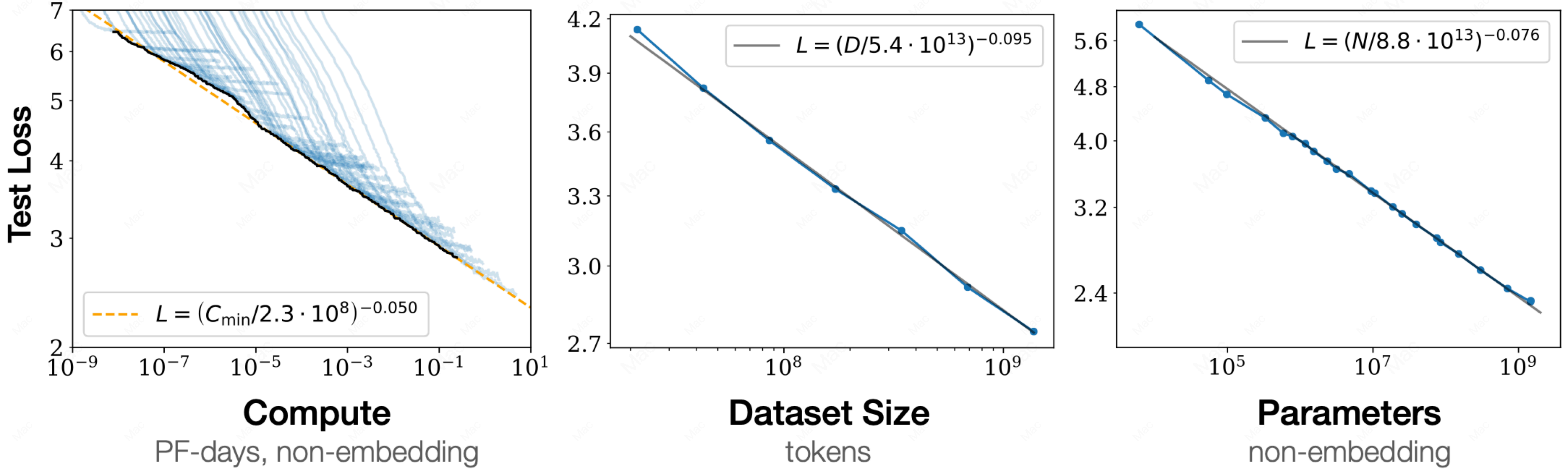
Challenges of Scaling Model Sizes

Scaling law tells us to scale up model sizes



Challenges of Scaling Model Sizes

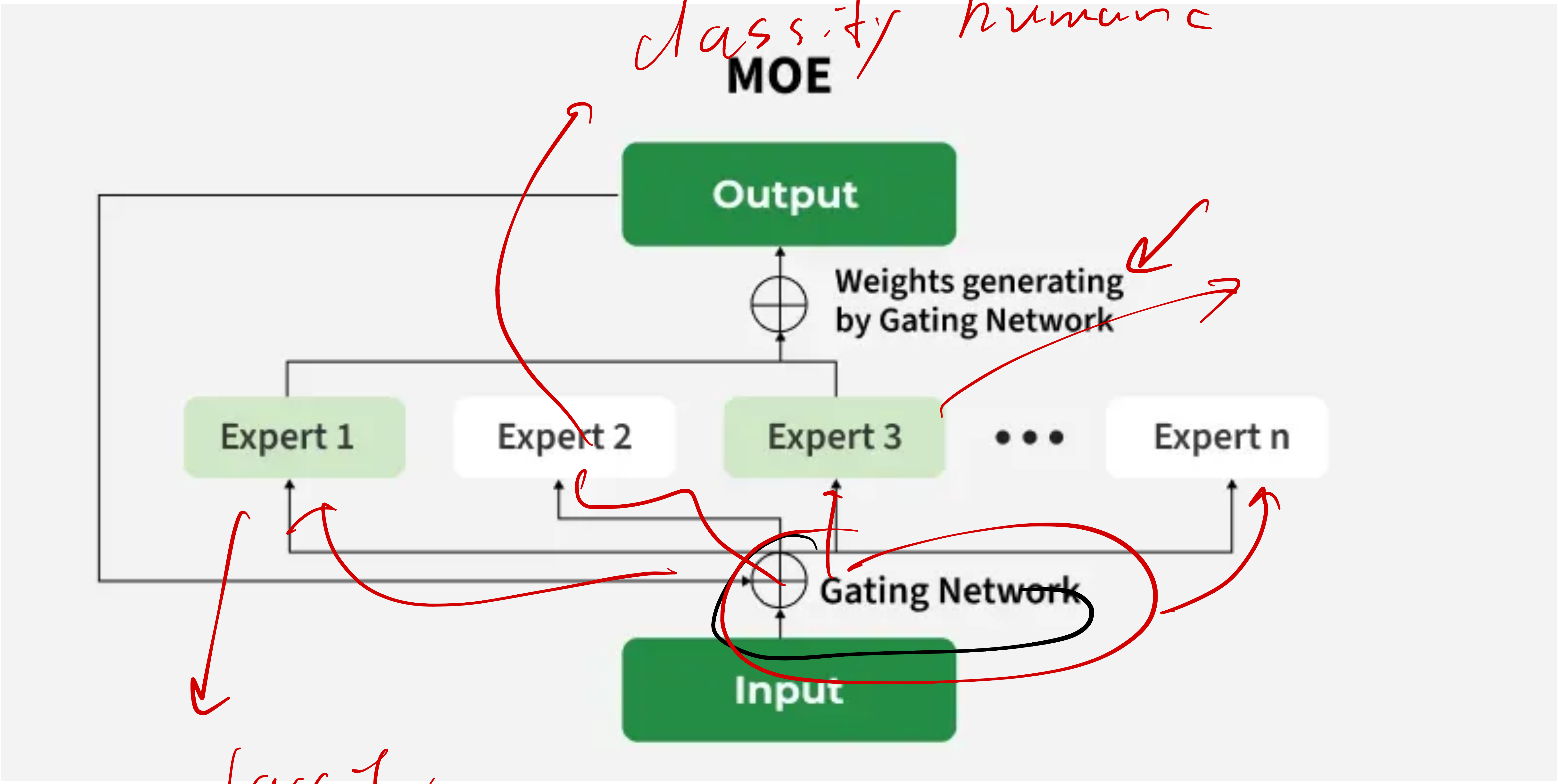
Scaling law tells us to scale up model sizes



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

Mixture of Experts (MoE) in Traditional Machine Learning

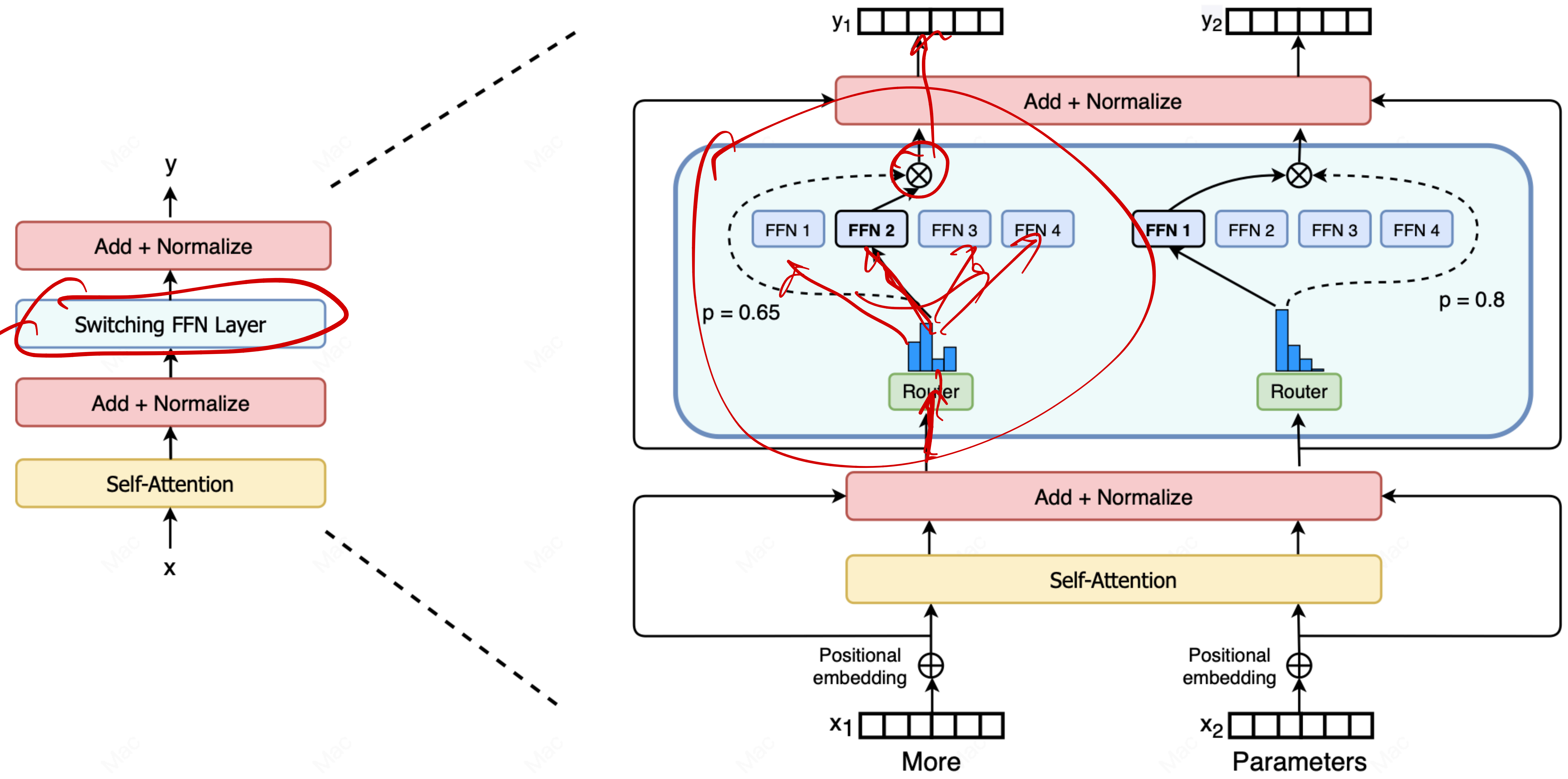
classify humans



classify animals

MoE Transformer Language Models

FFN

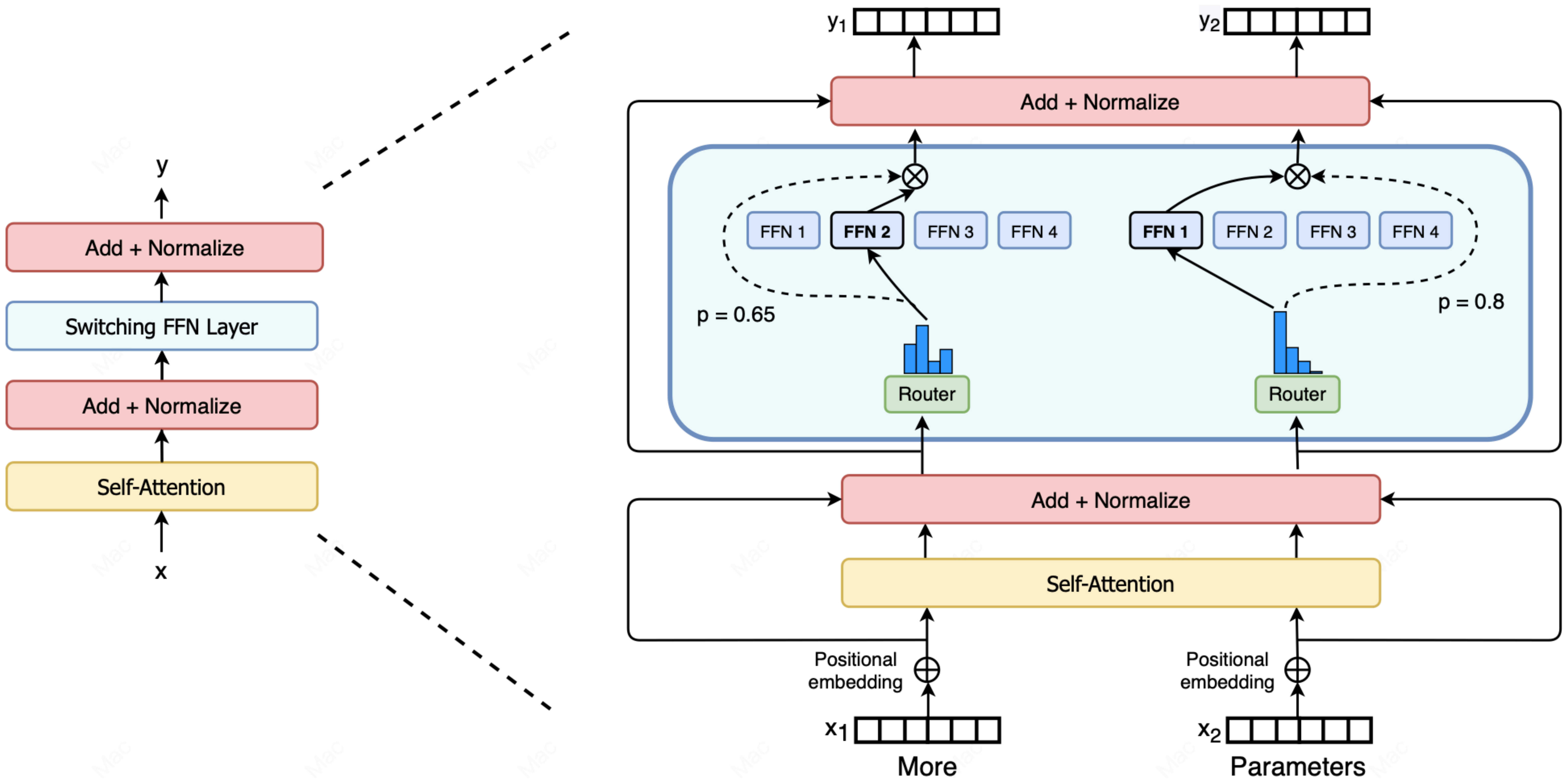


masked self attention

600B before GPT

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

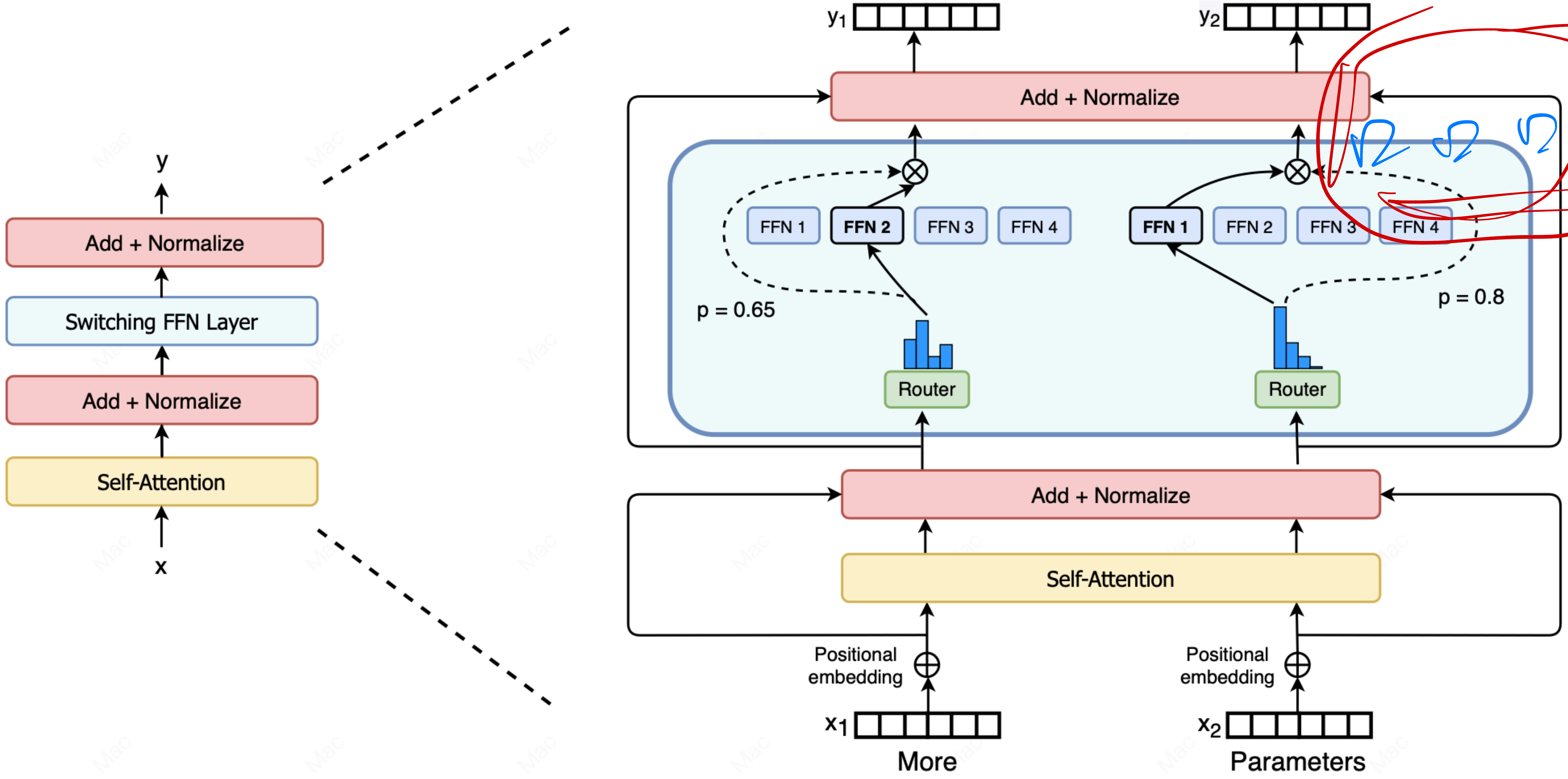
MoE Transformer Language Models



Mixture of FFN Blocks

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

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

P(experts)

Router

Sparse Routing

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

Sparse Routing

parameter

router

input

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

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

$p_i(x)$ prob x routing to expert i

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

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

Sparse Routing

Dense:

$E(x)$

$$h(x) = W_r \cdot x$$ 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

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

aggregation

$\mathcal{T}$ = top K

linear weighted addition

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

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

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

minimize

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]

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

B = batch

learnable

f_i P_i → optimize

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

ideal:

specific routing each token

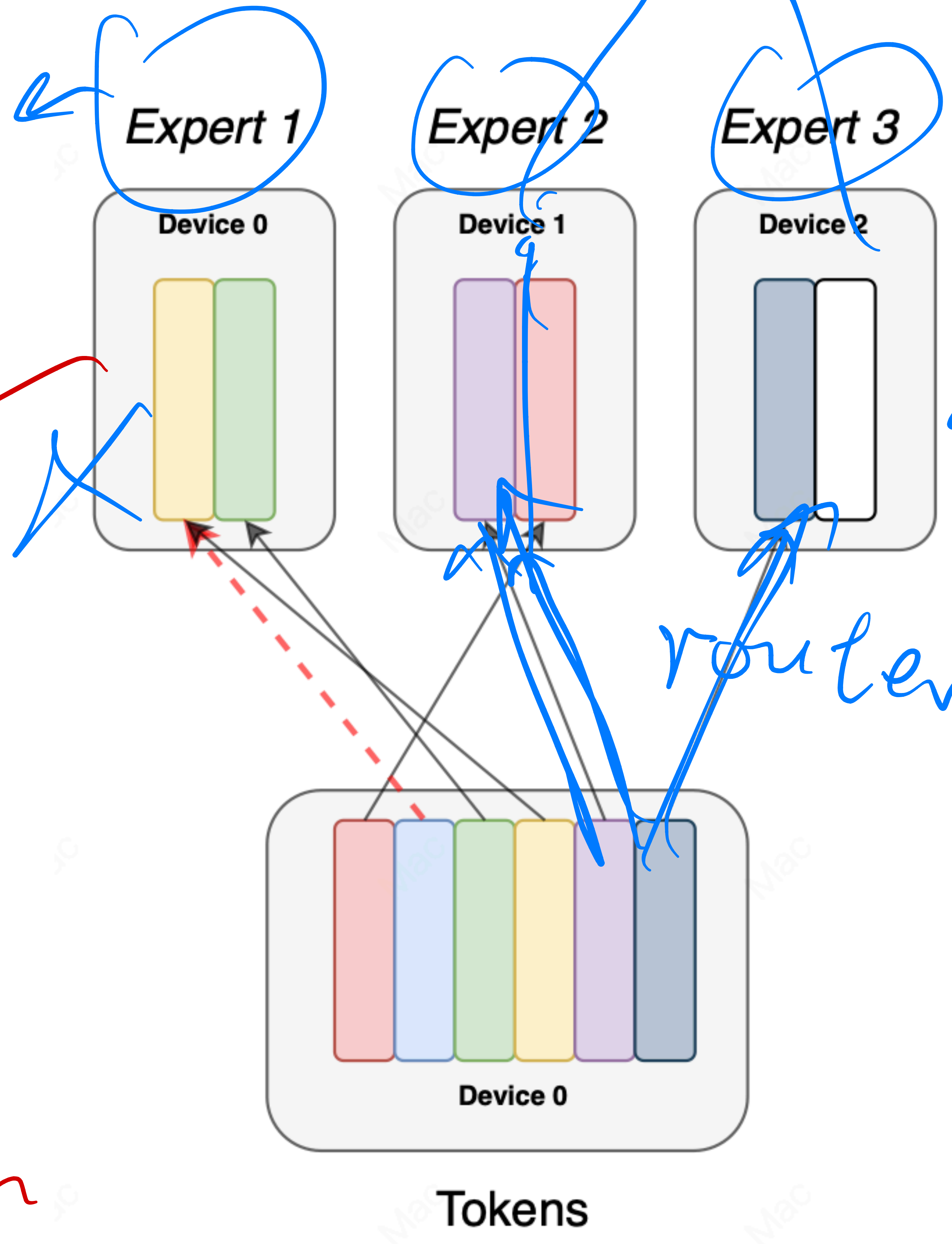
uniform load globally

Why MoE?

not interpretable

aggre

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



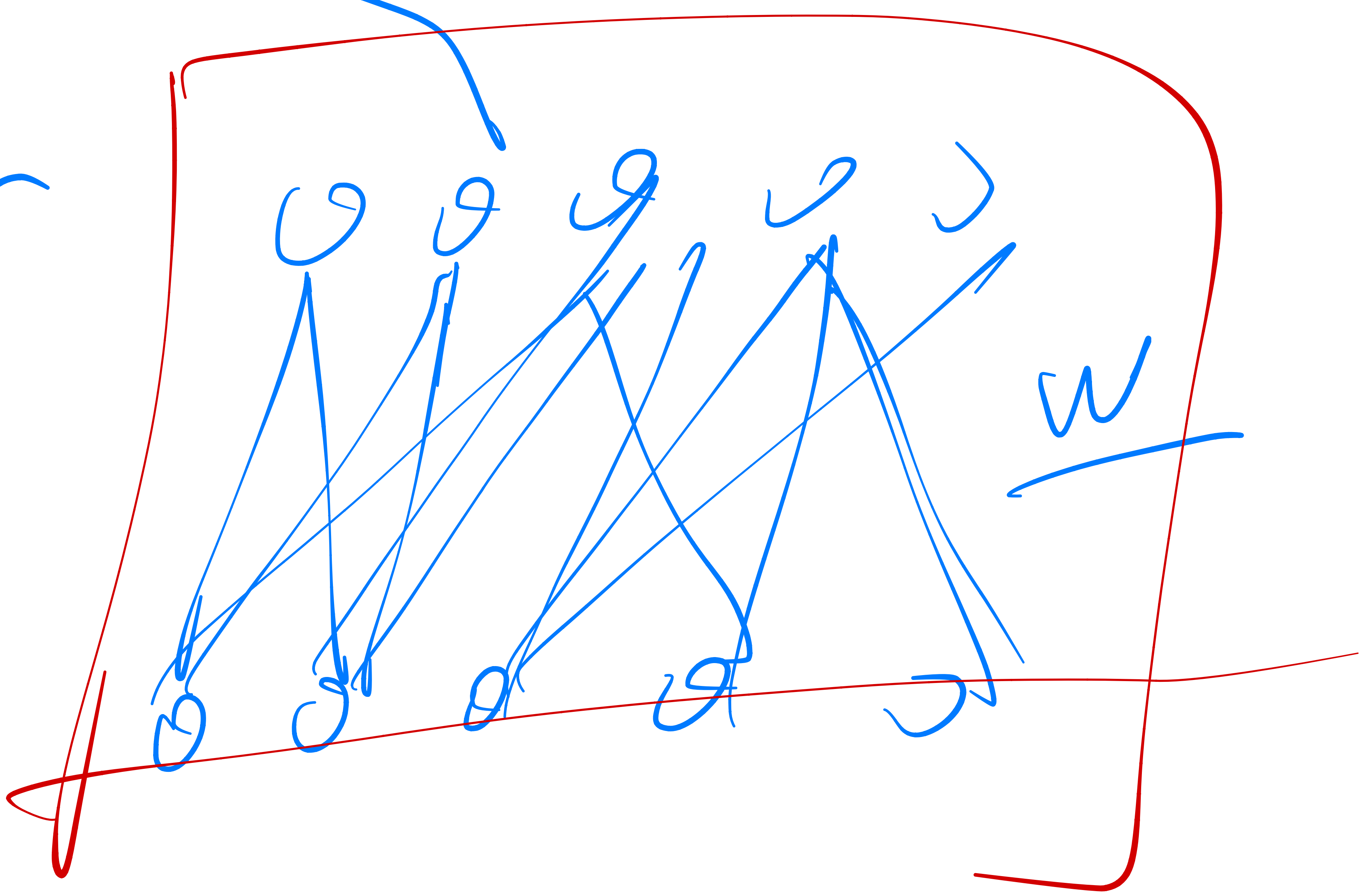
router

system infra

W

FFN

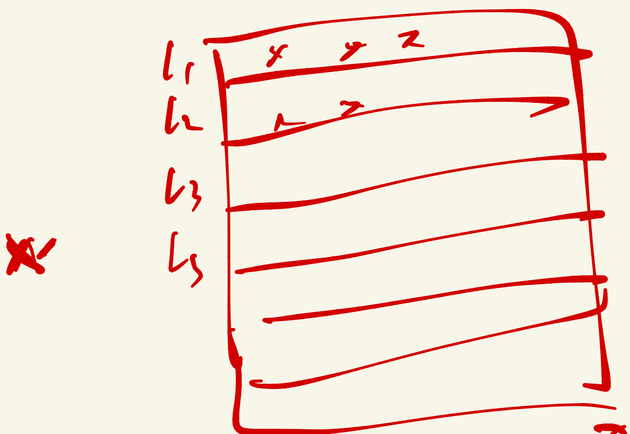
activation



W

Matrix multiplication

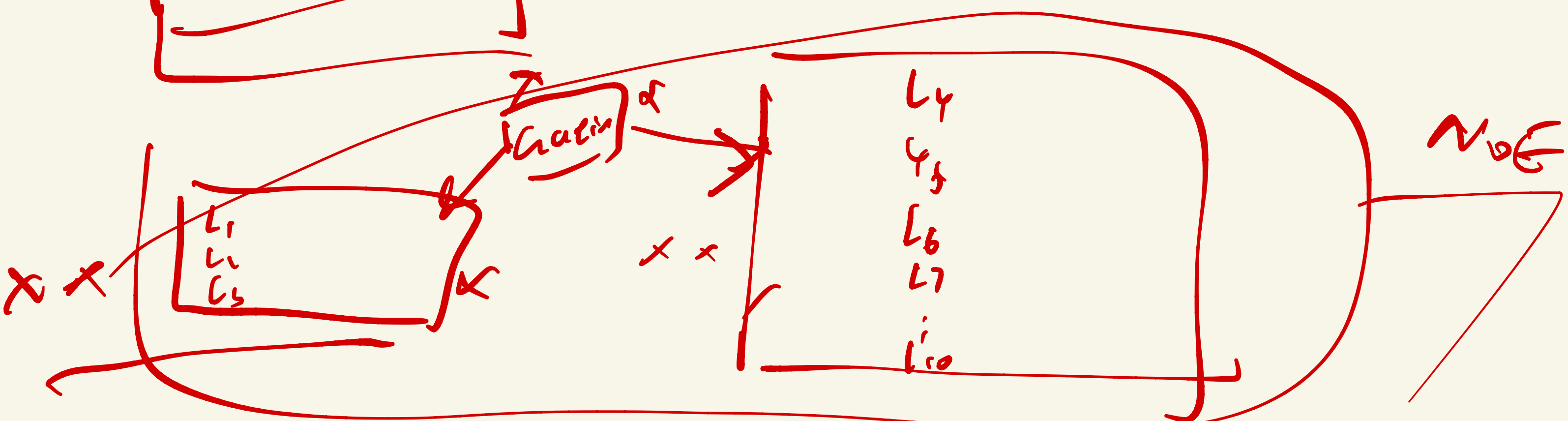
W



$$x^T l_1 = O_1$$

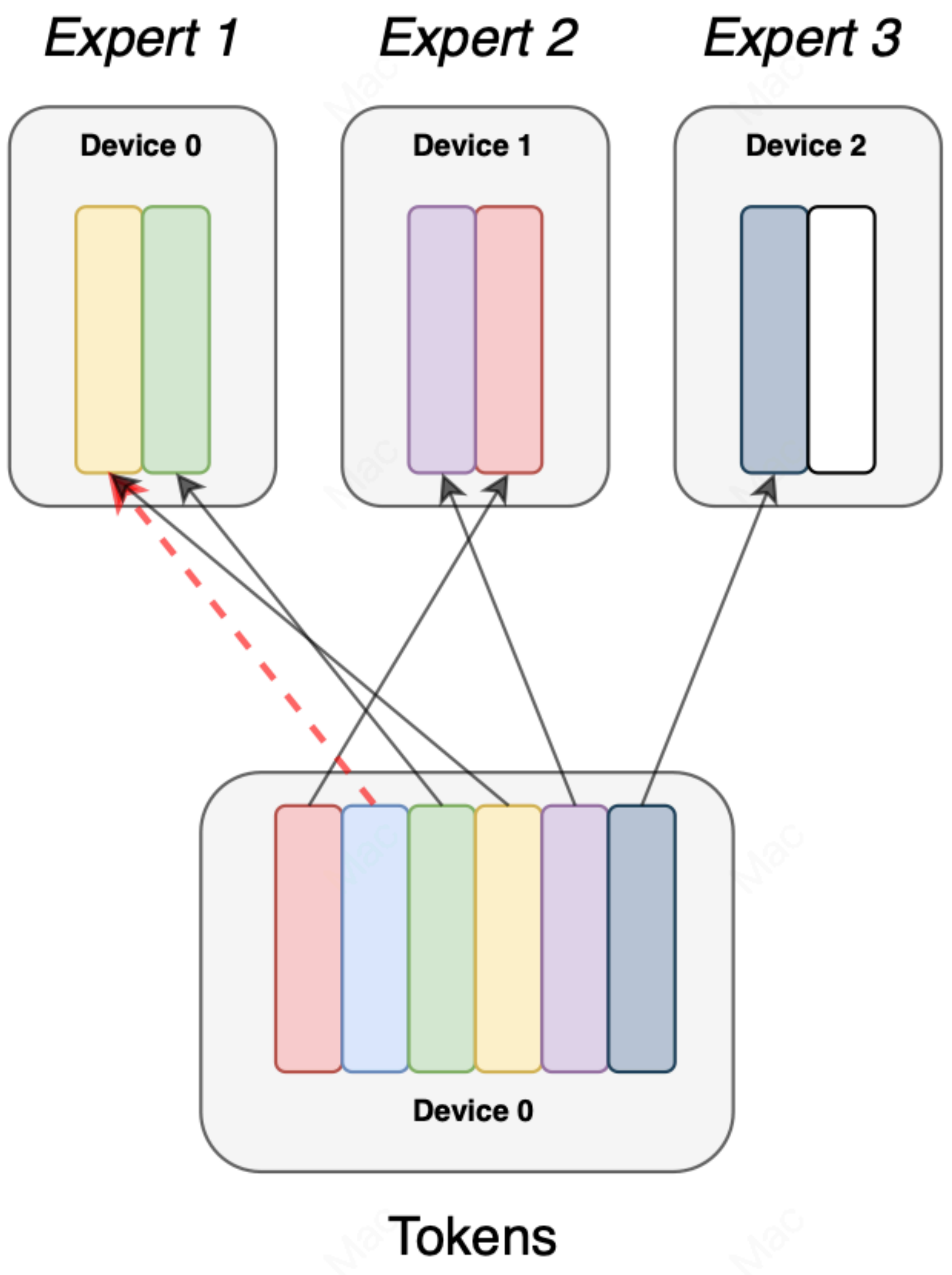
$$x^T l_2 = O_2$$

sparse



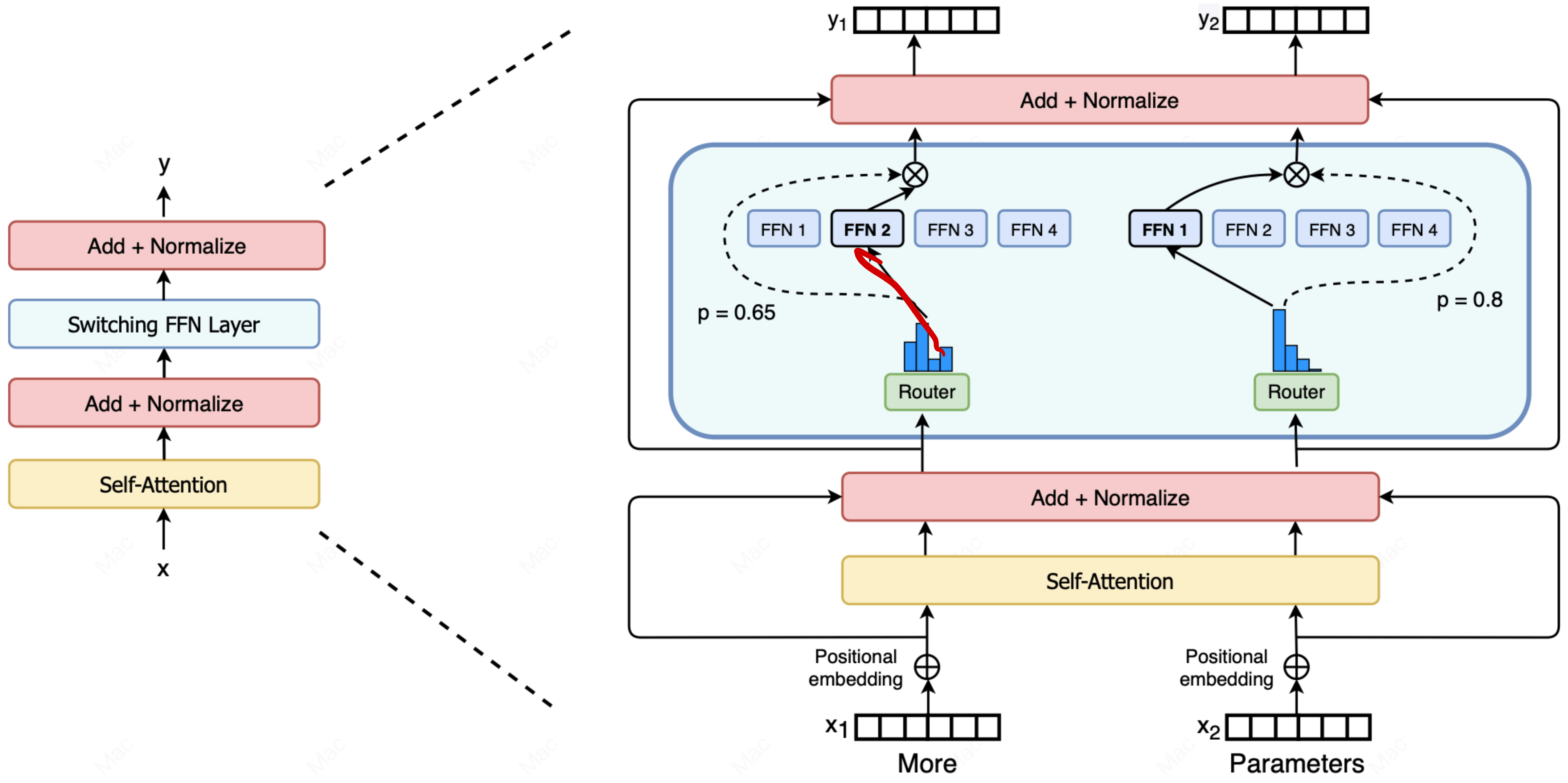
Why MoE?

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

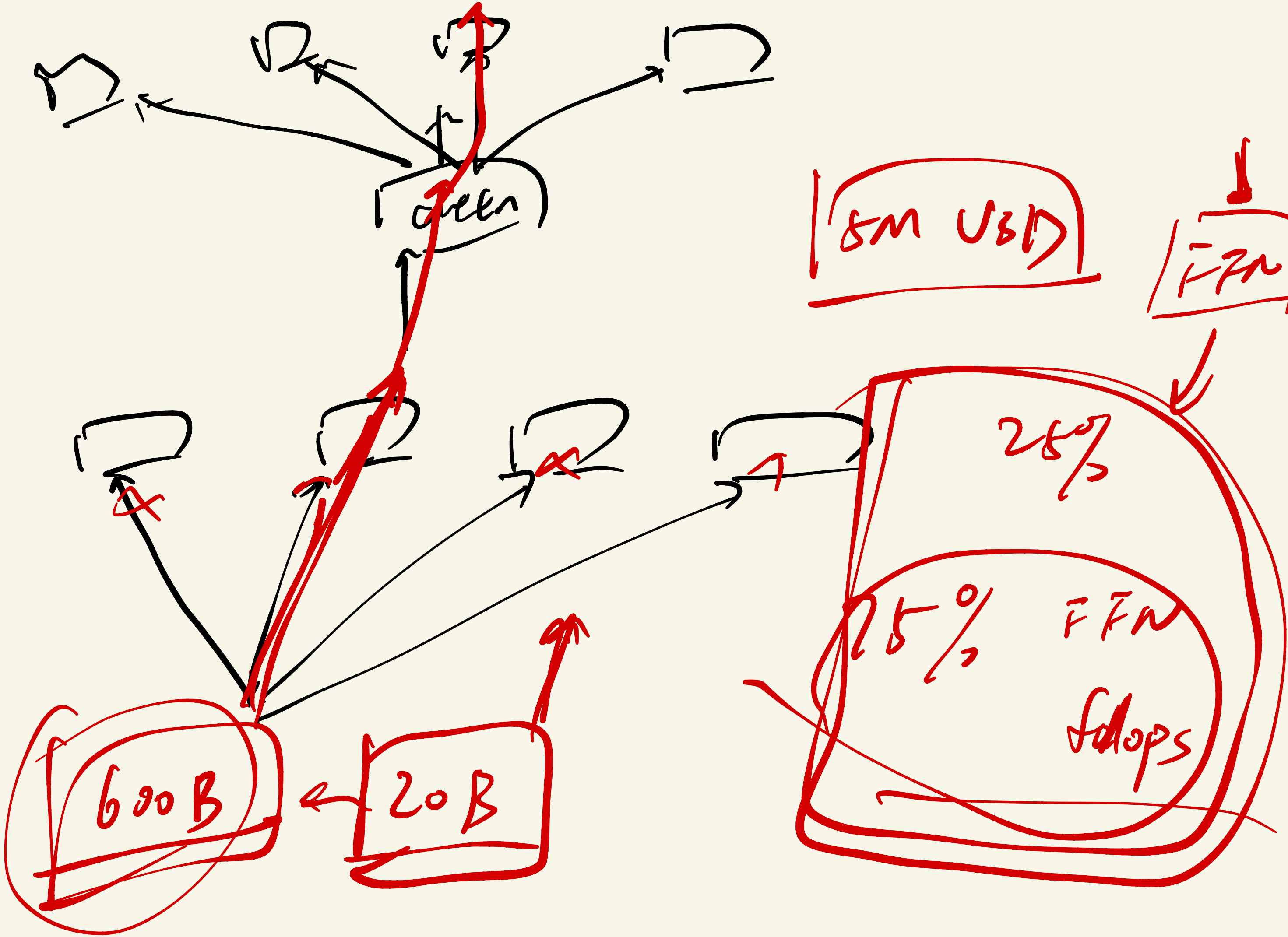
Why MoE?



forward backward

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

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



Qwen
600B
20B
5M USD
FFN
25%
25%
FFN
fdlops

attn

d: hidden size

n: seq length.

$\times W^q$ $\times W^k$ $\times W^v$

$O(nd^2)$

QK

$O(n^2 d)$

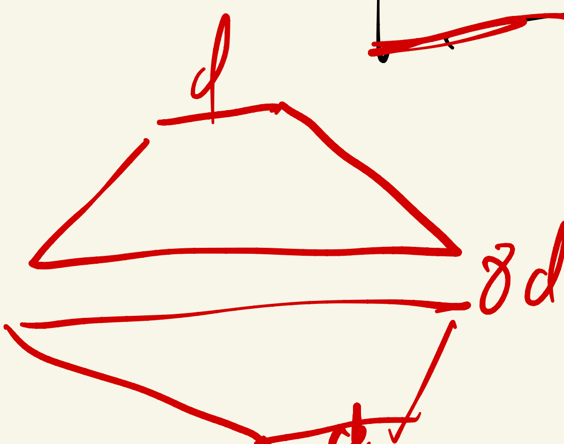
FFN

dominant parameter

dominate flops

$n^2 d = 16 n d^2$

$n \to \infty$

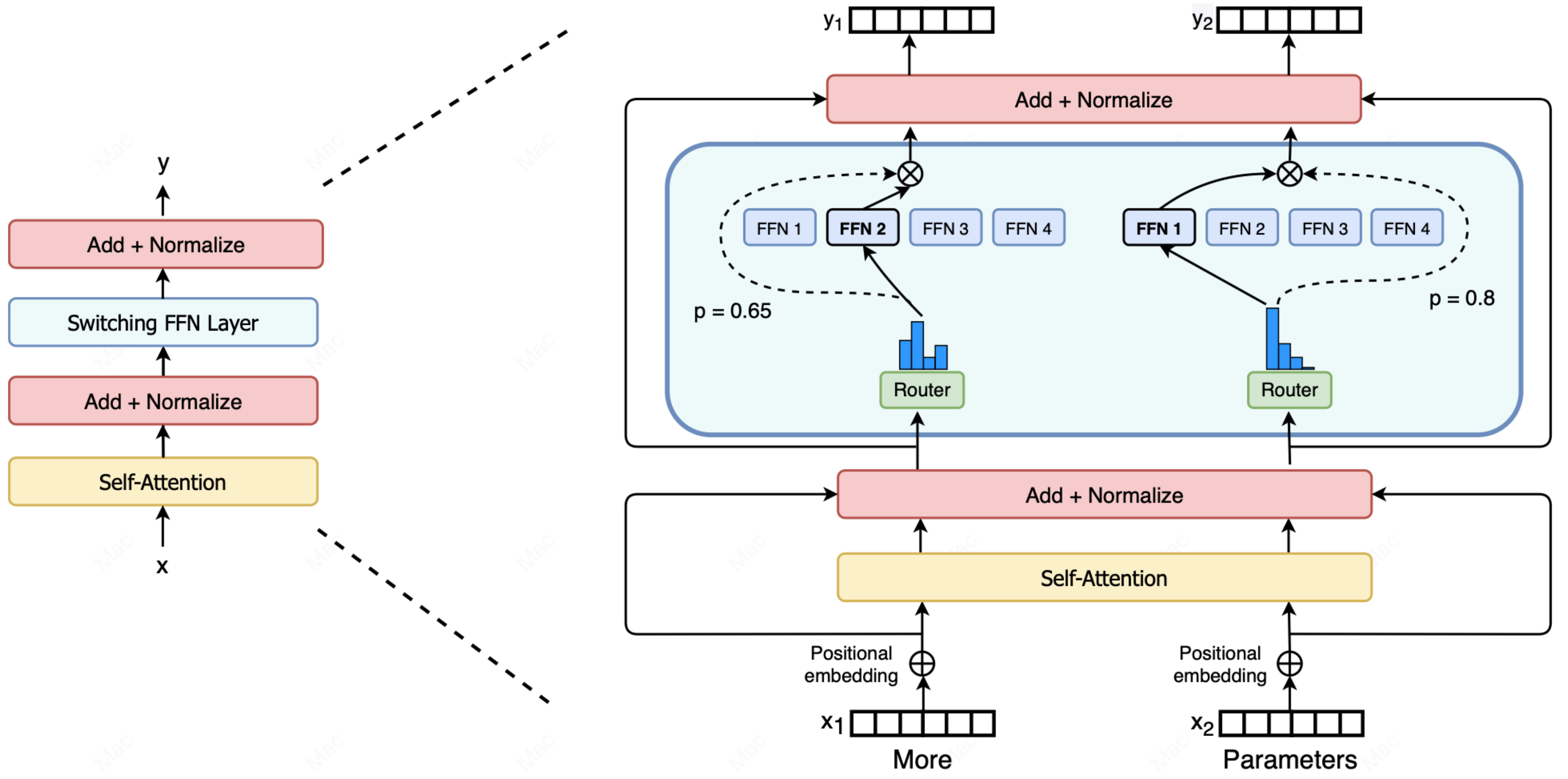


$O(8d^2 \times 2 \times n)$;

$+ 16 n d^2$

$d = 2048$

Why MoE?

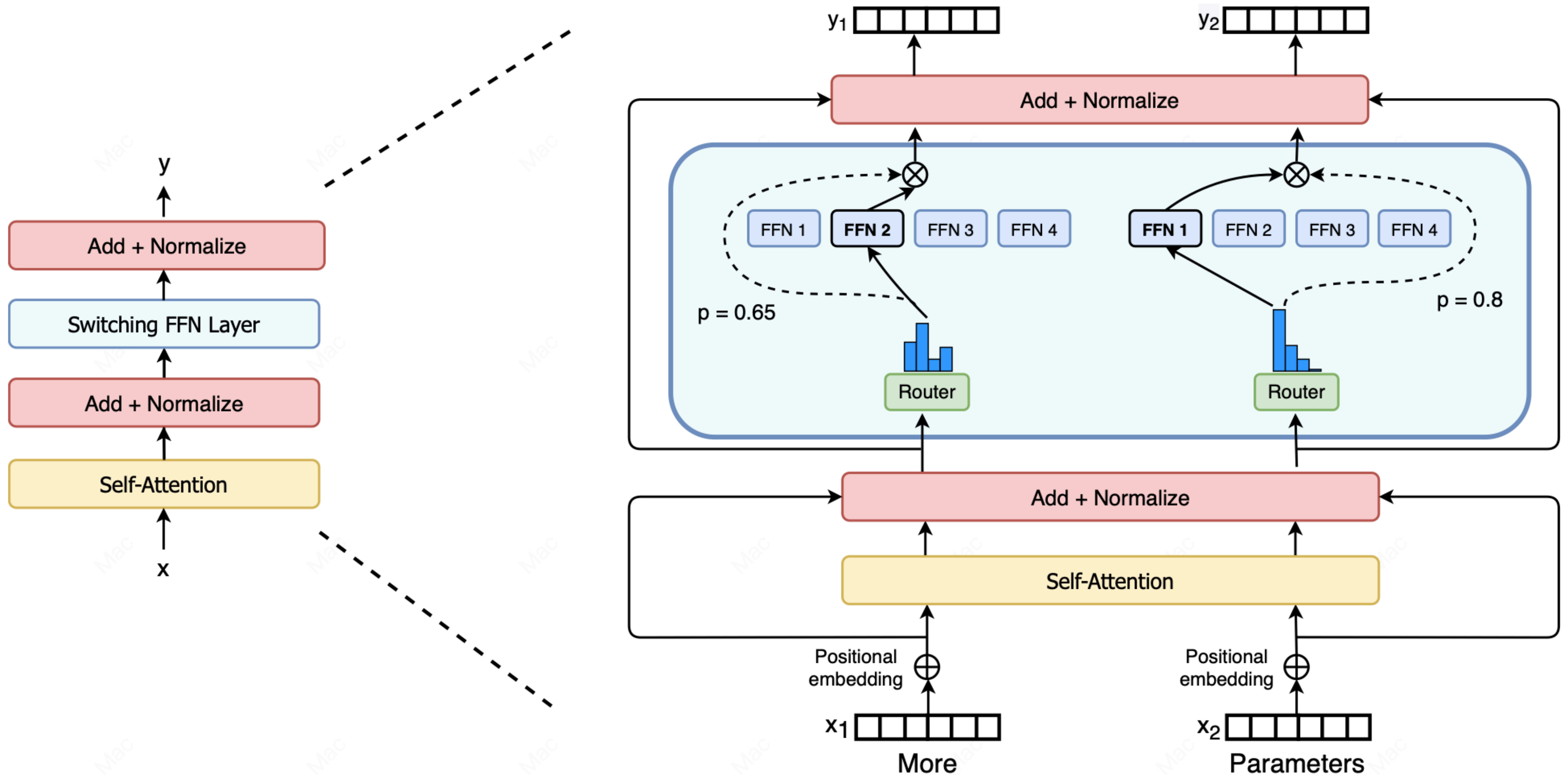


1. Training flops compared to dense model of the same size?
2. Inference flops compared to dense model of the same size?

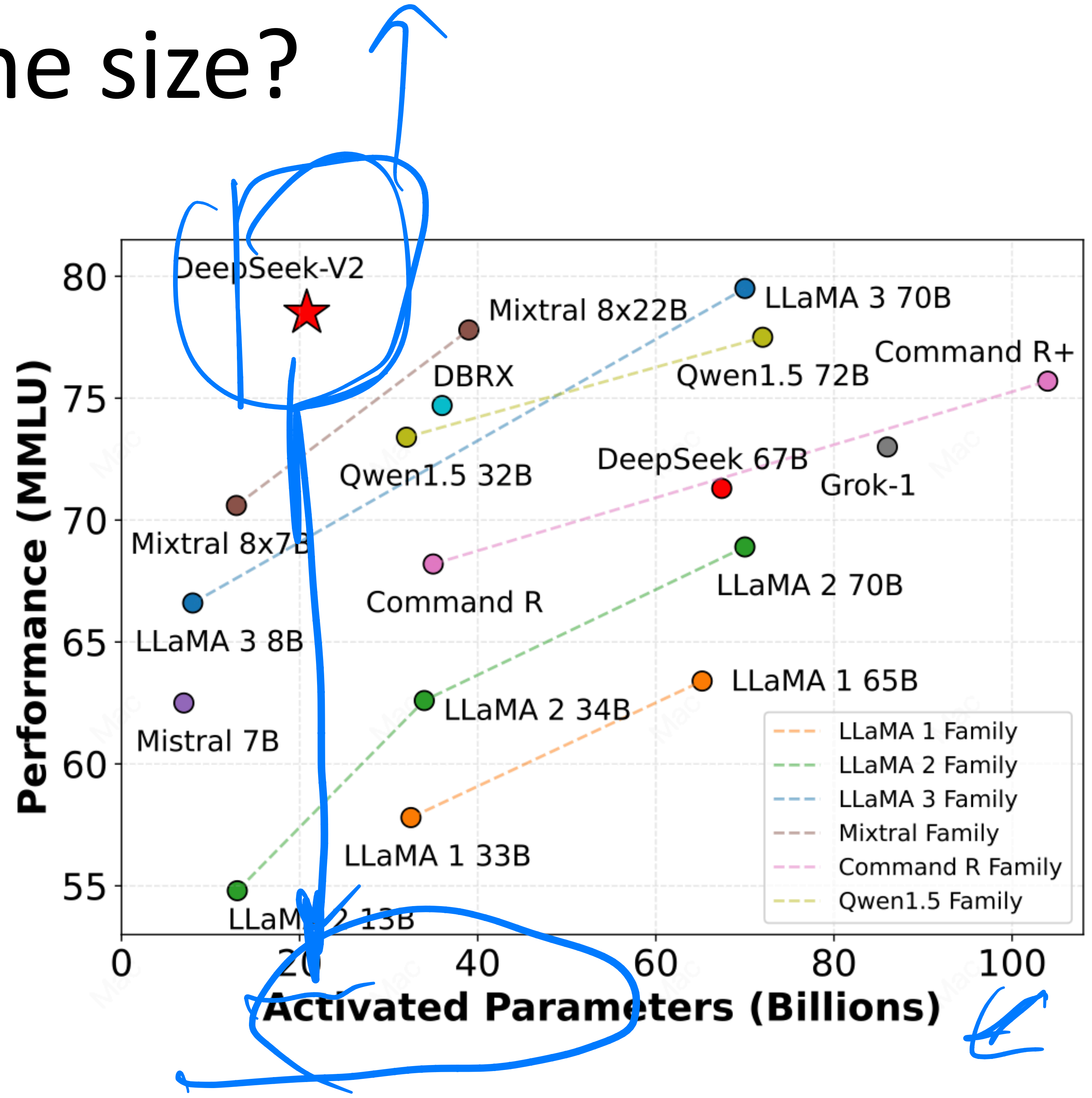
Theoretically, if the activated parameters are only 1B, then the inference latency can be optimized to be close to an 1B dense model

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

Why MoE?



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



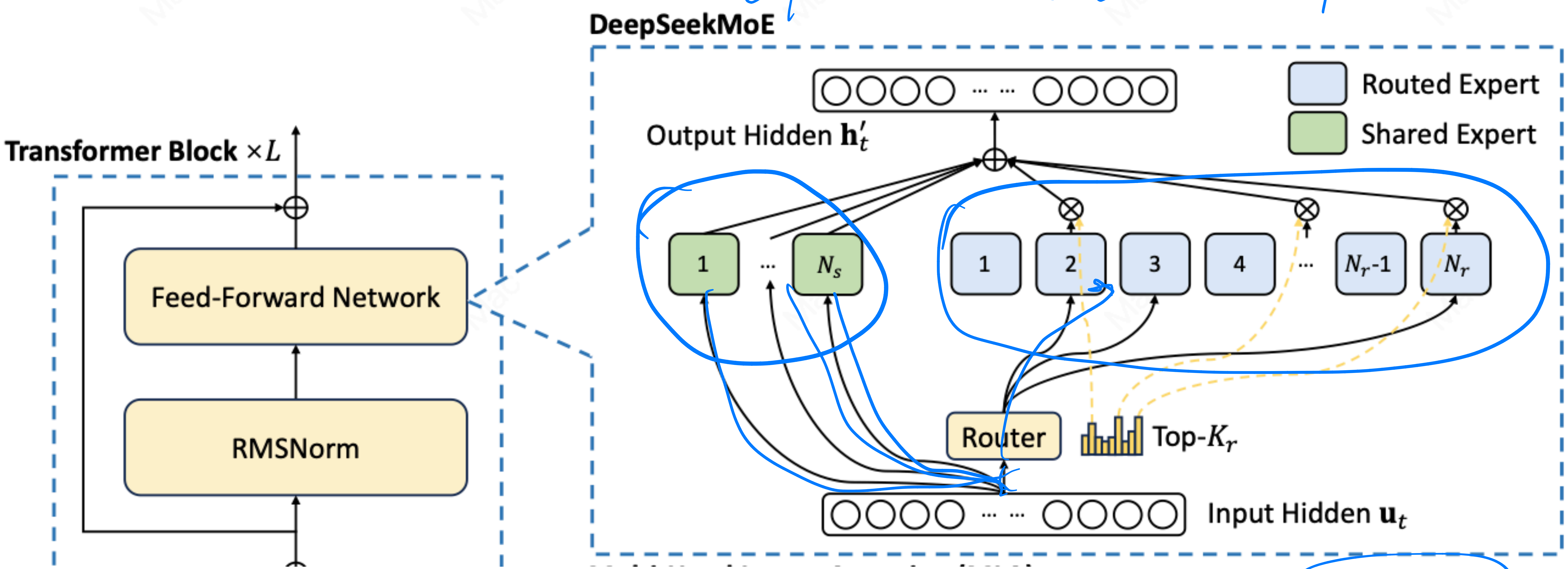
Theoretically, if the activated parameters are only 1B, then the inference latency can be optimized to be close to an 1B dense model

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

DeepSeek MoE

Deepseek V3 → post traing

deeseek R1



8

256

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

residual

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}

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

softmax

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

0

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

U input

e_i router parameter

Loss-Free Load Balancing in DeepSeek-v3

b_i bias $\uparrow [b_i]$

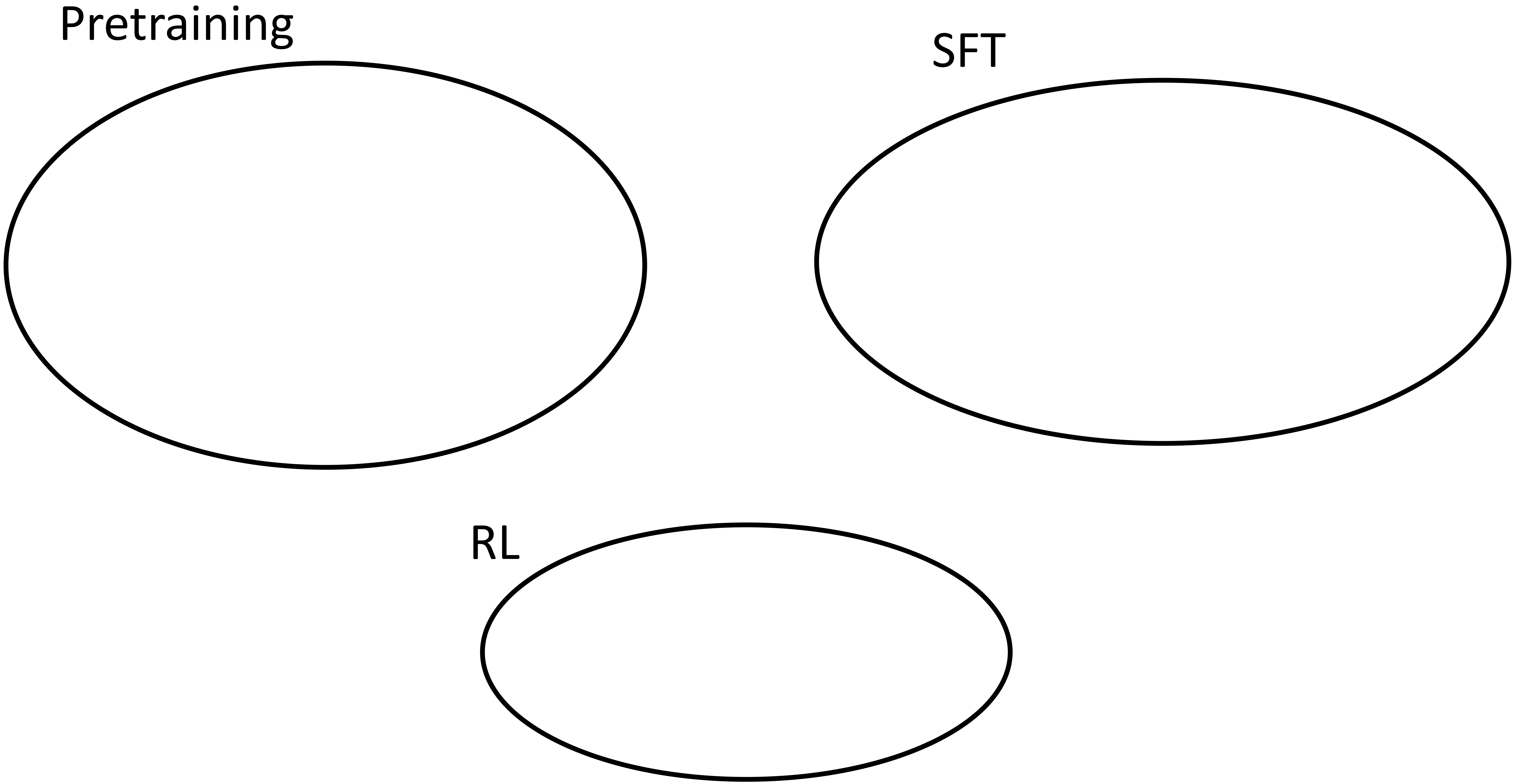
$$g'_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} + b_i \in \operatorname{Topk}(\{s_{j,t} + b_j | 1 \leqslant j \leqslant N_r\}, K_r), \\ 0, & \text{otherwise.} \end{cases}$$

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.

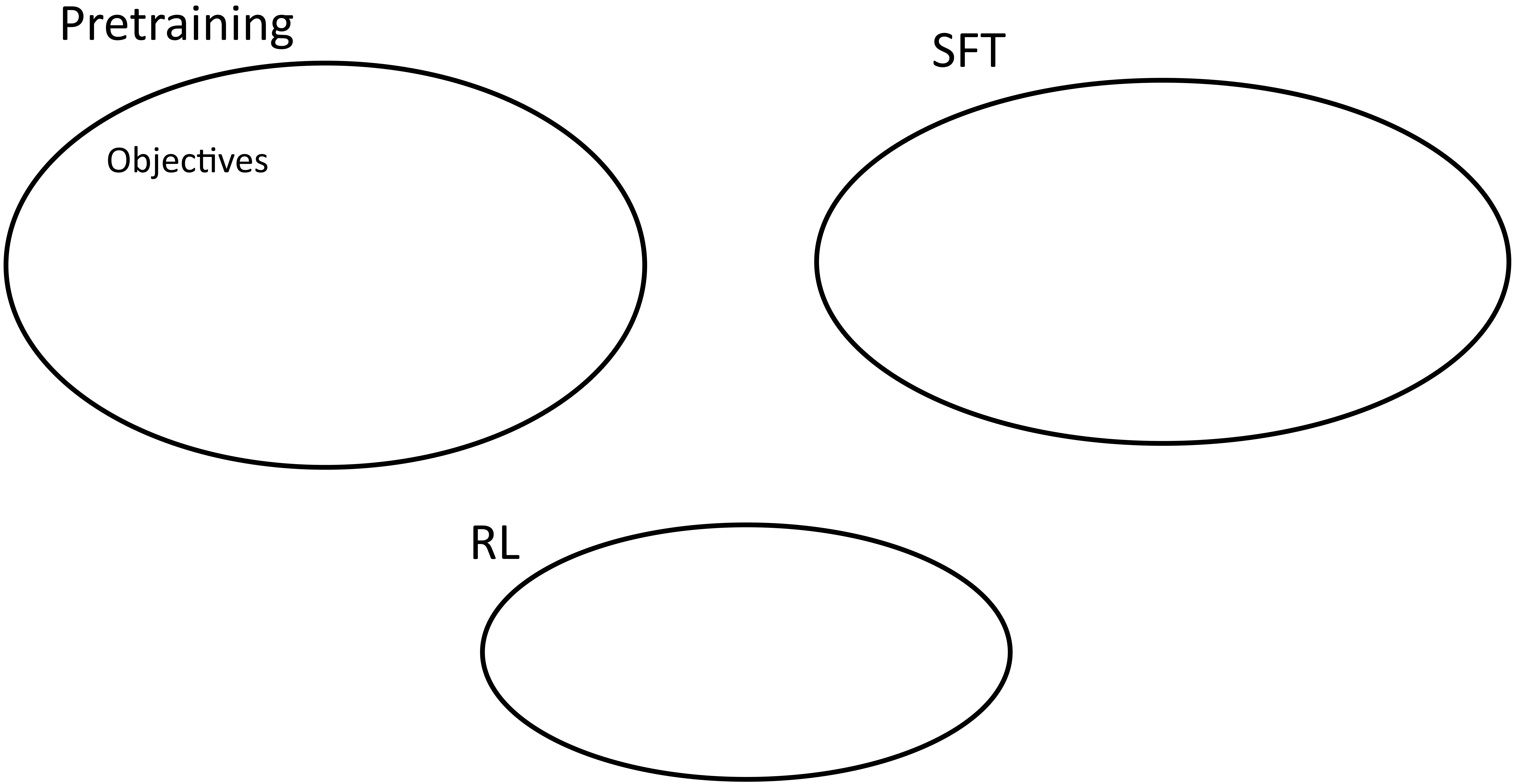
DeepSeek V3 Technical Report

Vibe Coding

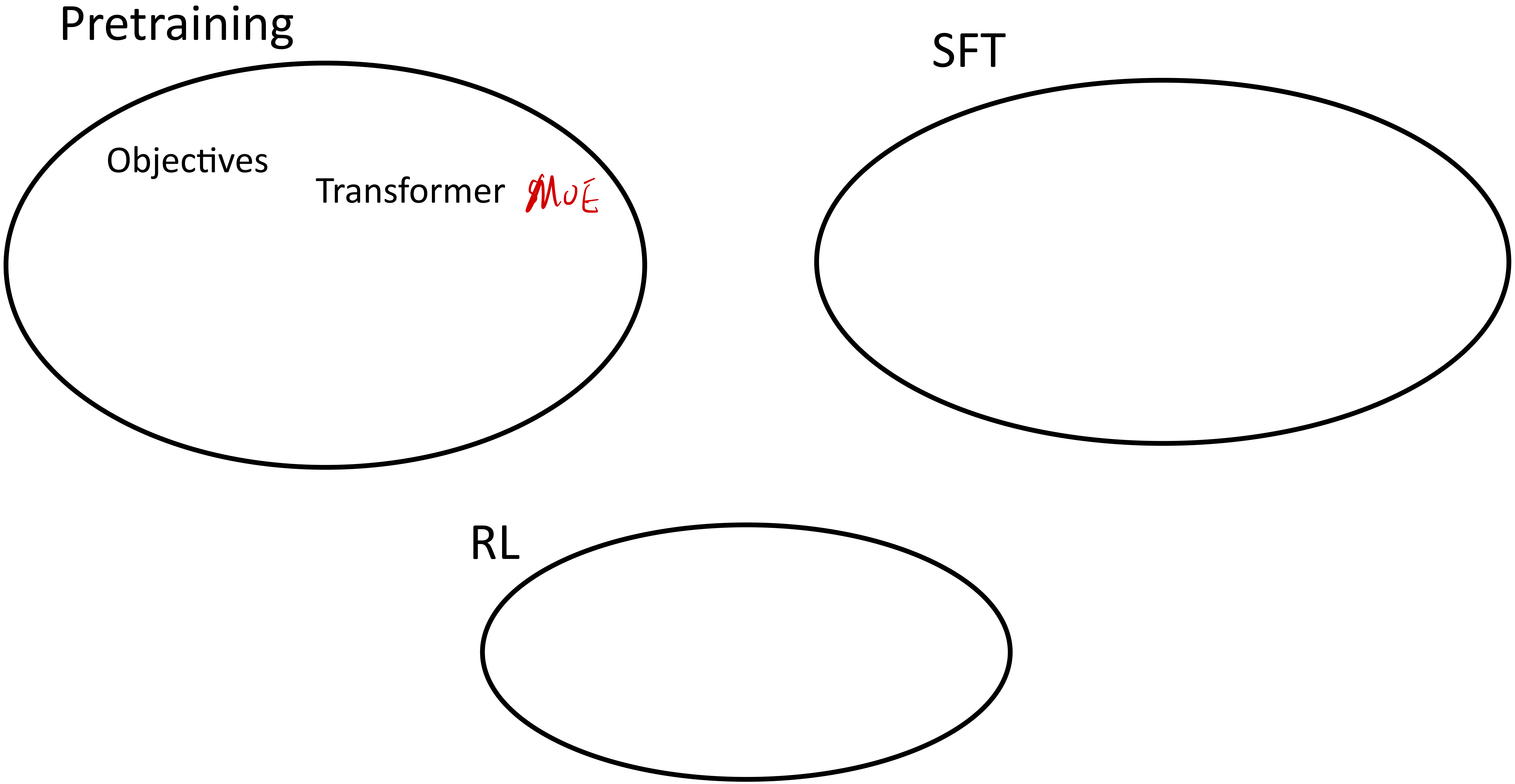
Review — Method



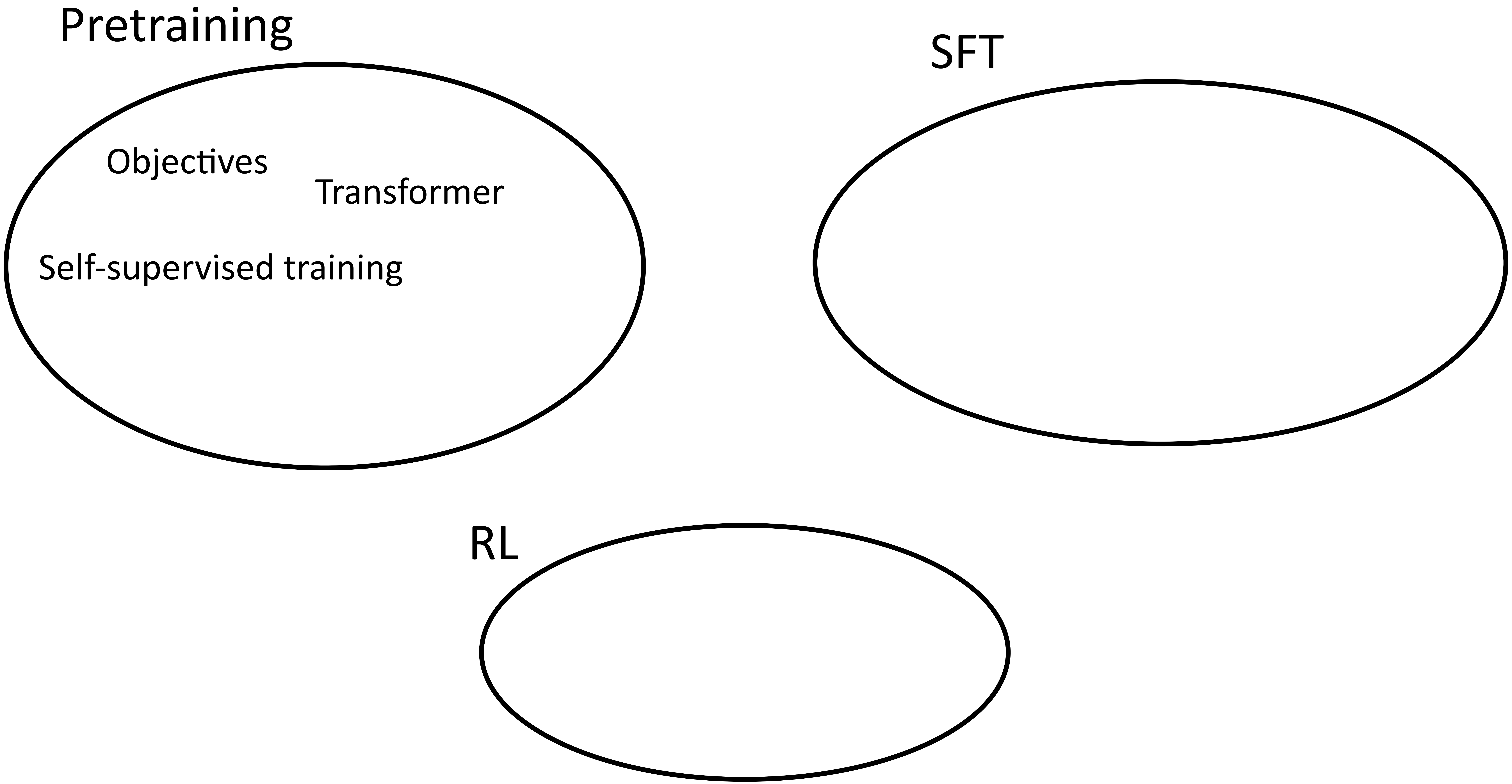
Review — Method



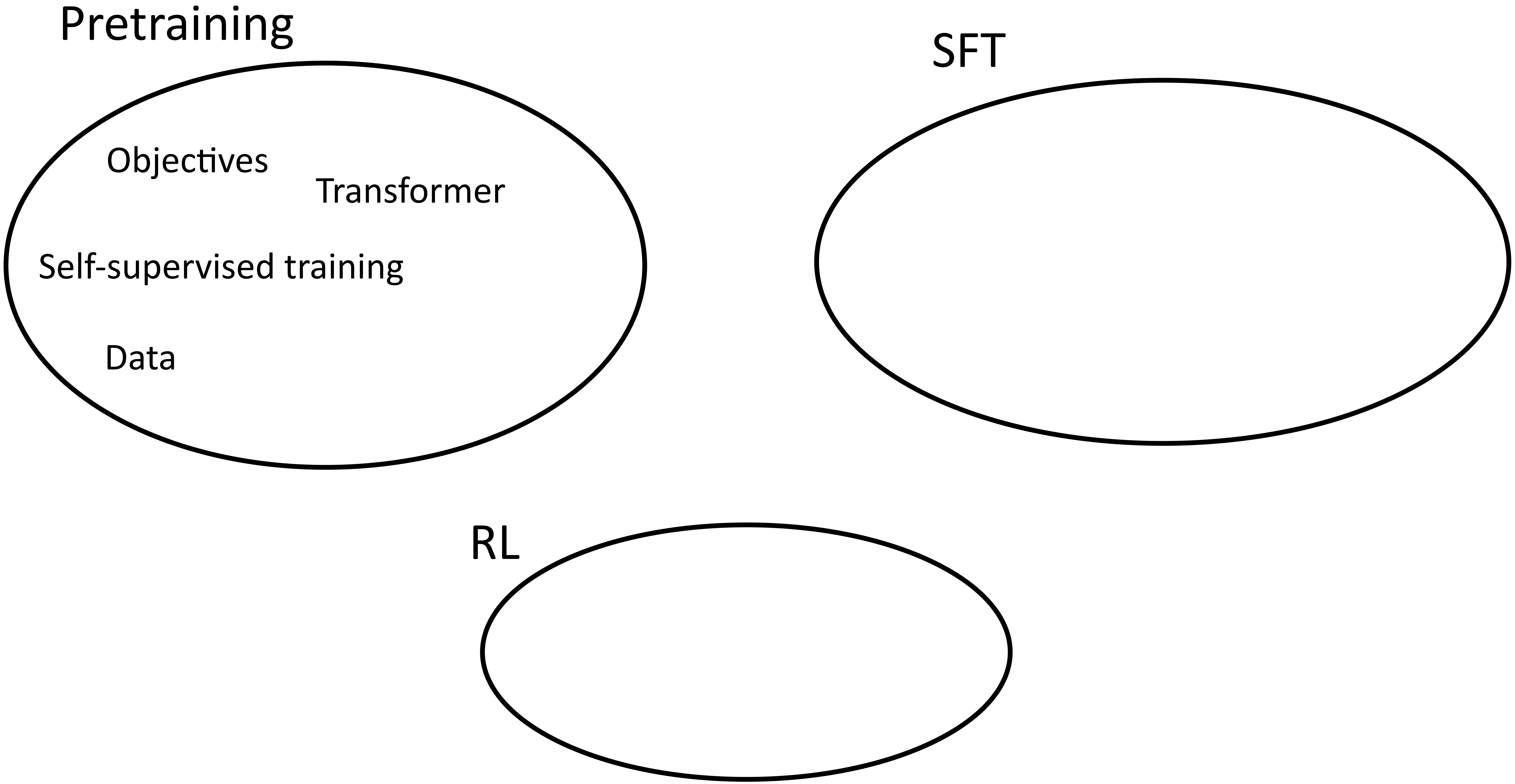
Review — Method



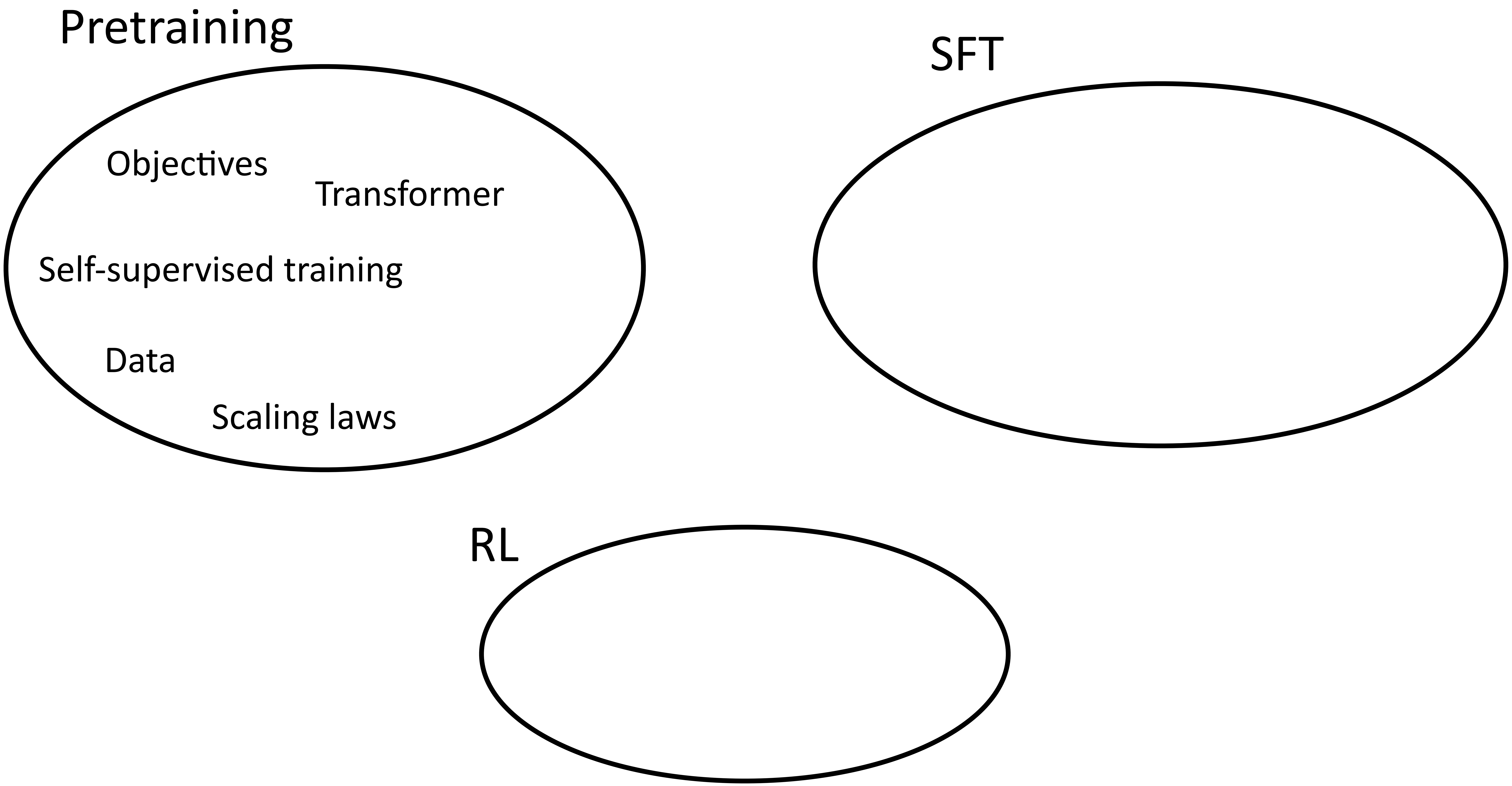
Review — Method



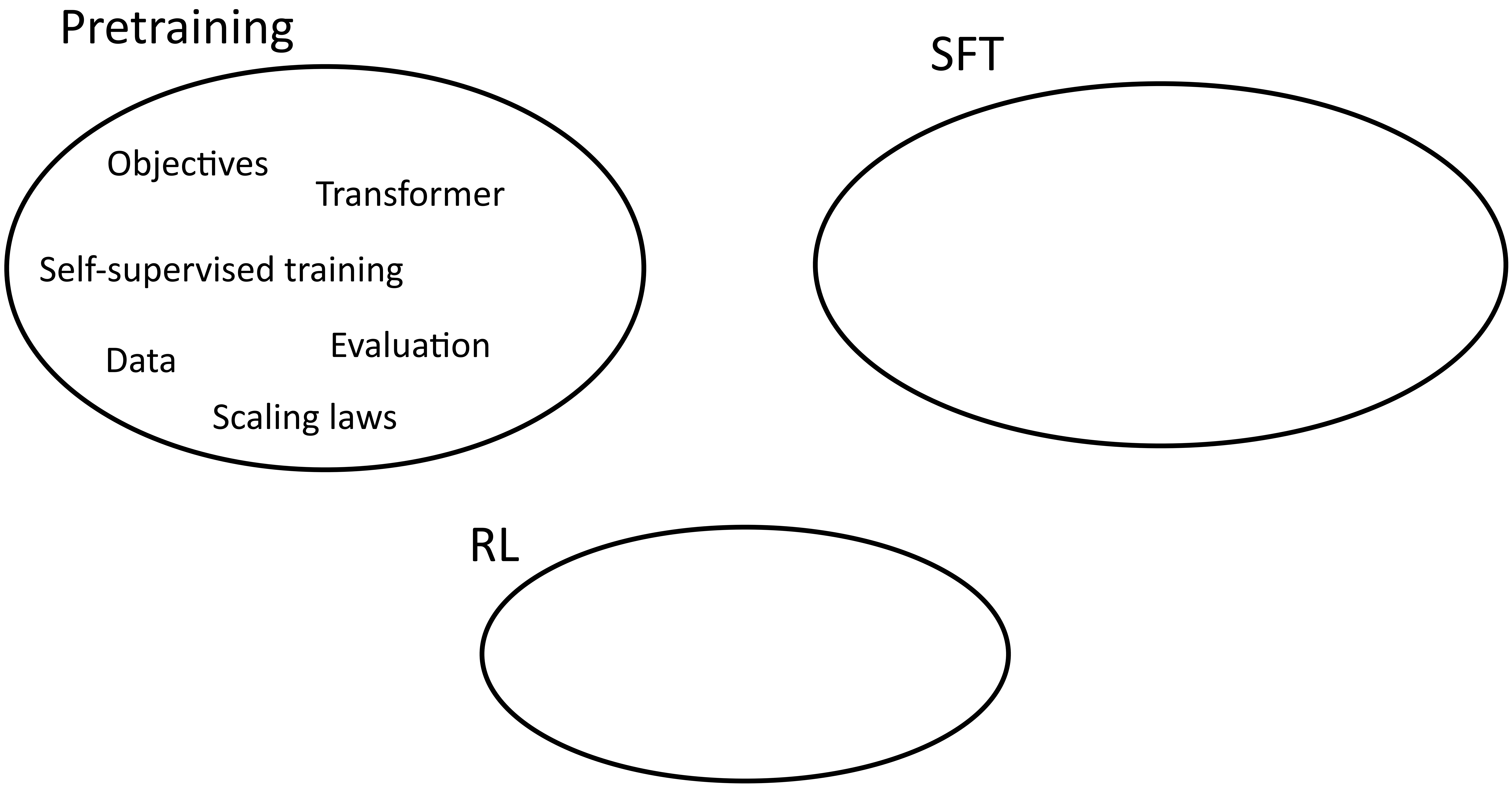
Review — Method



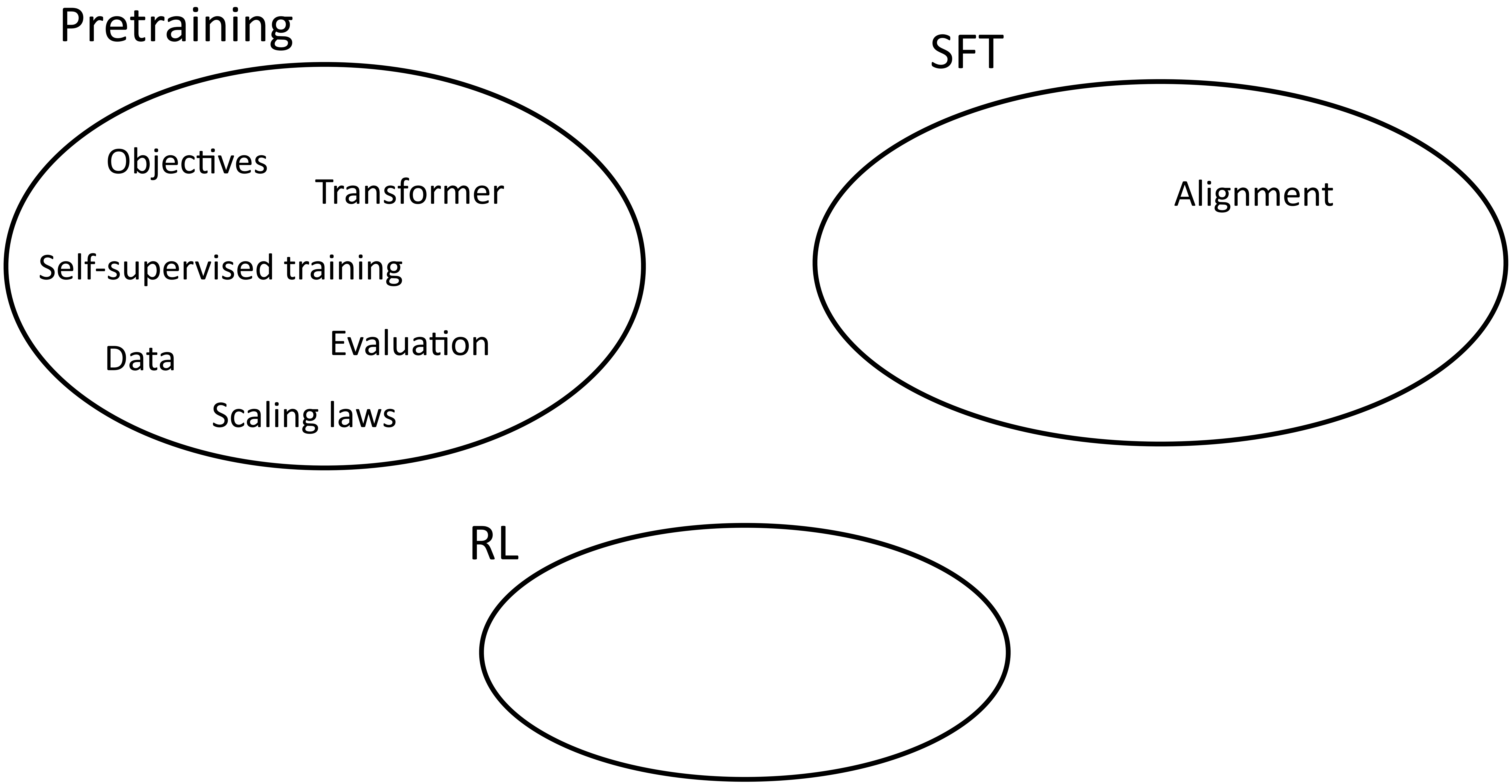
Review — Method



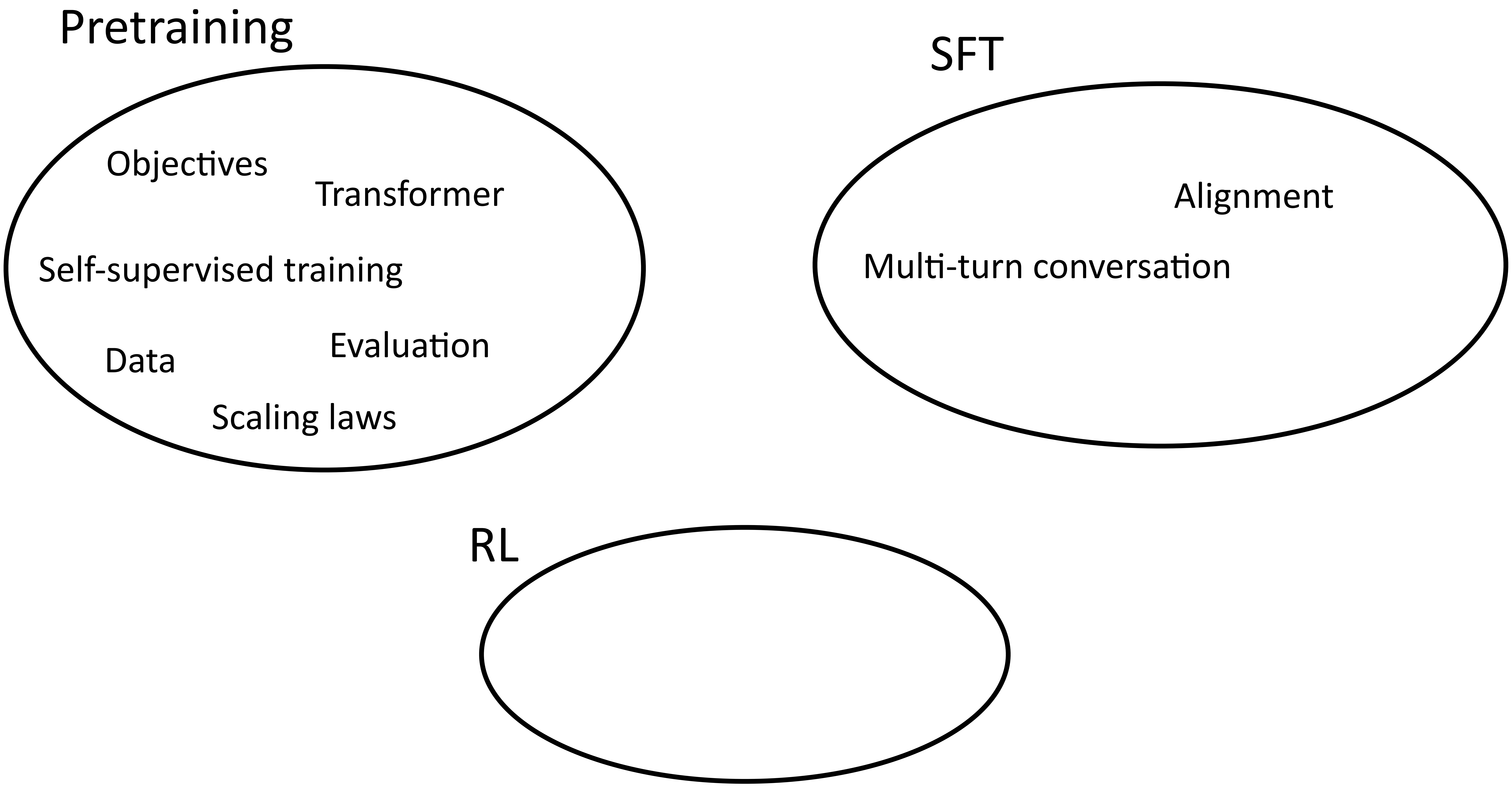
Review — Method



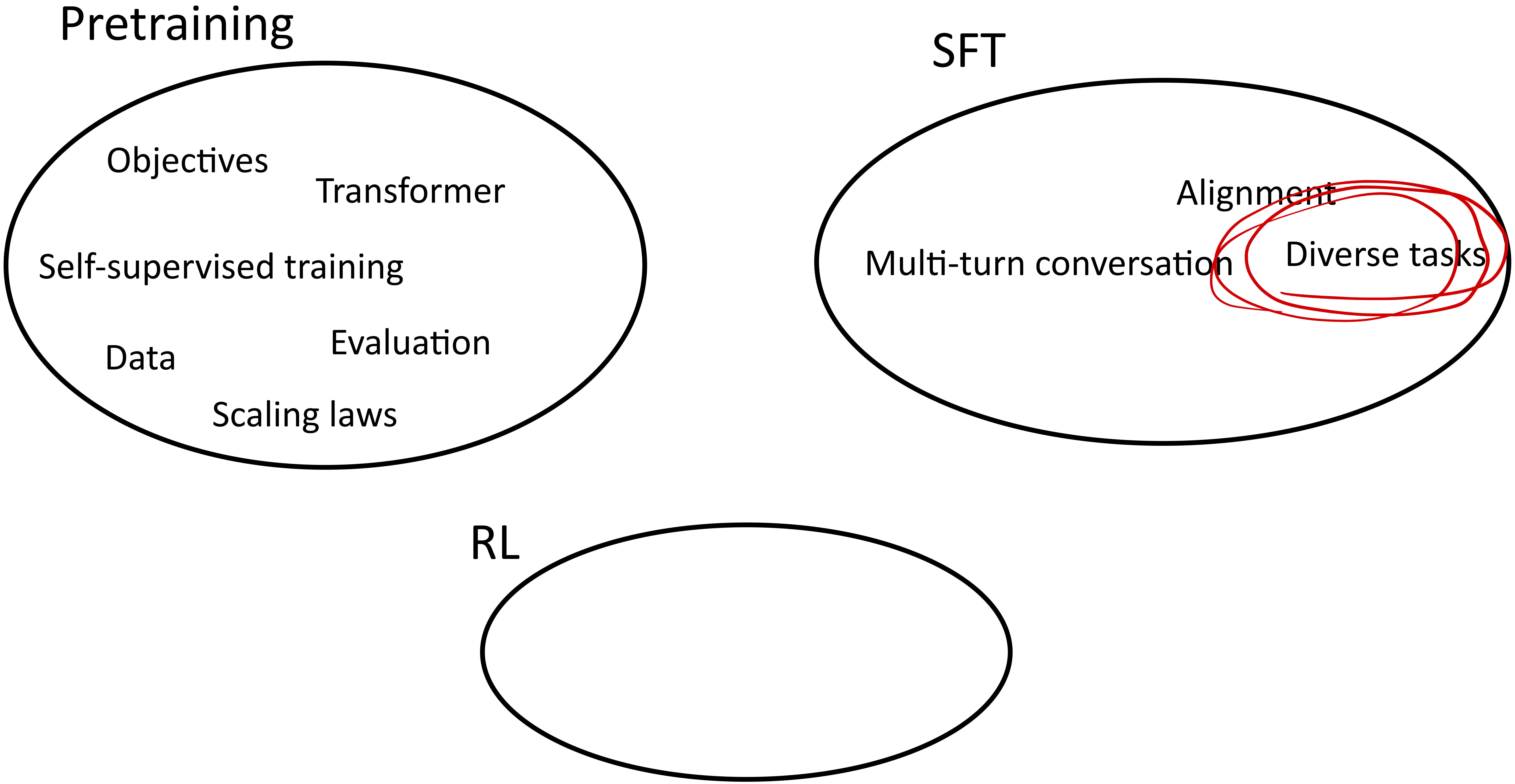
Review — Method



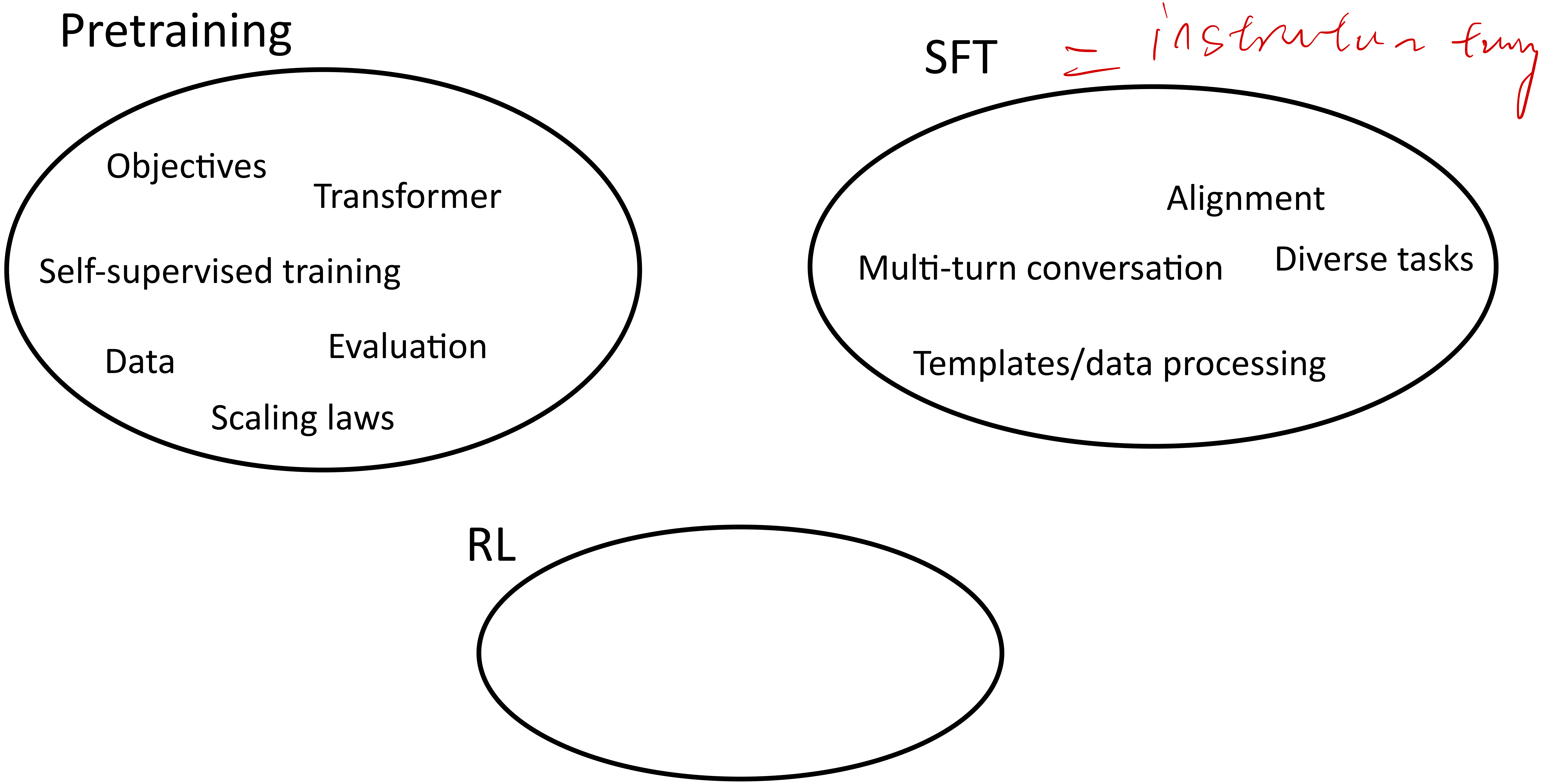
Review — Method



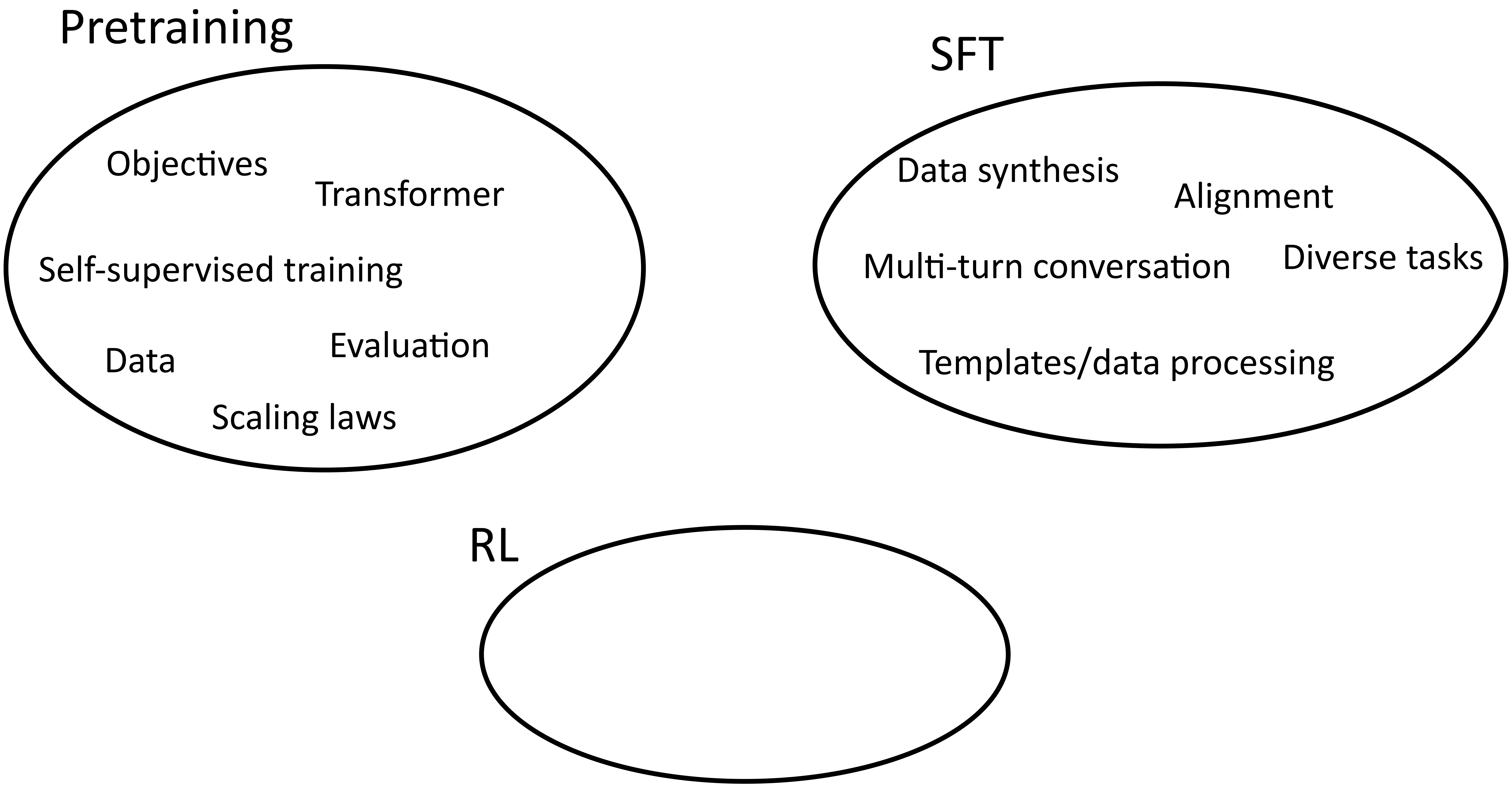
Review — Method



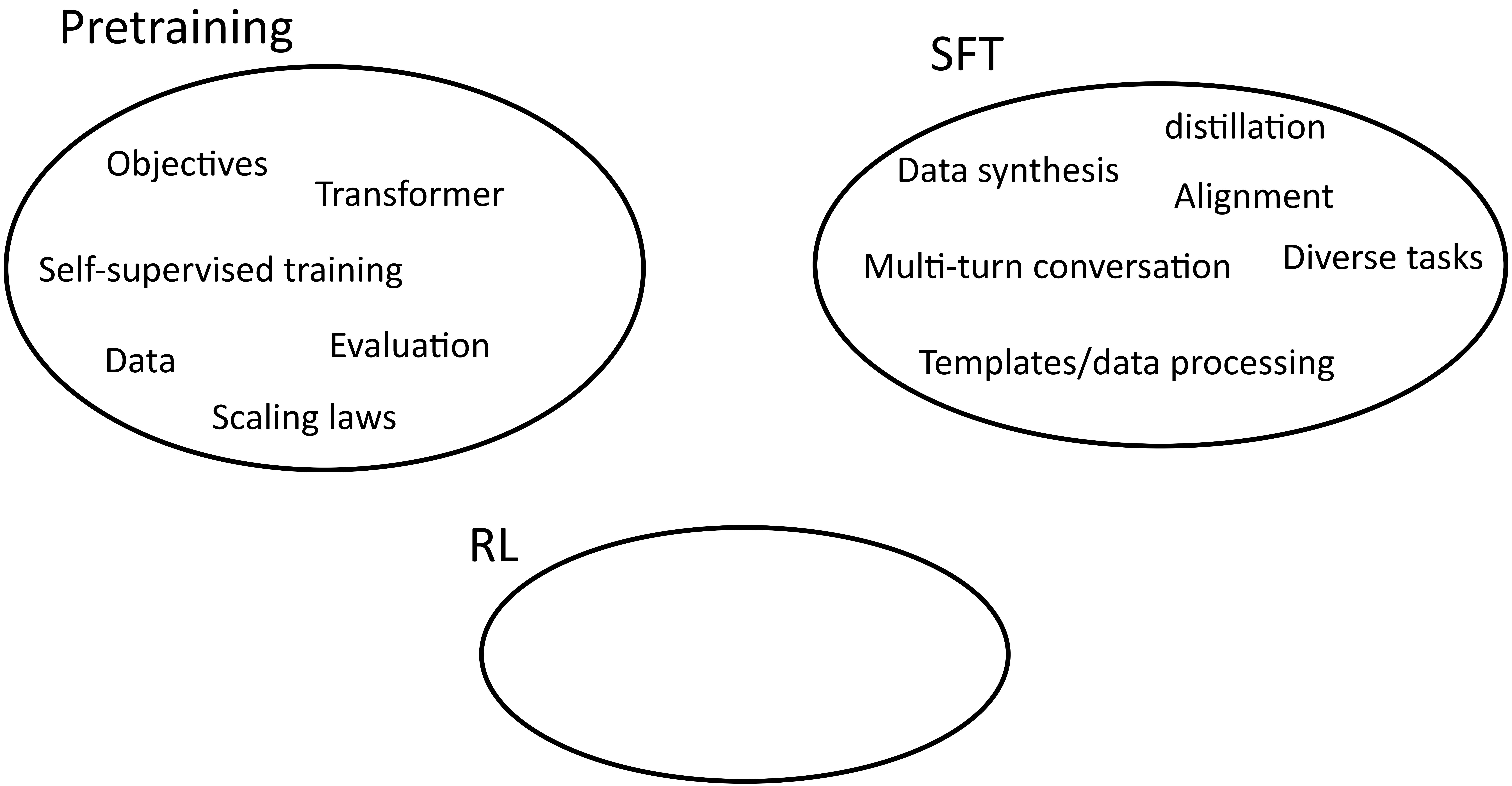
Review — Method



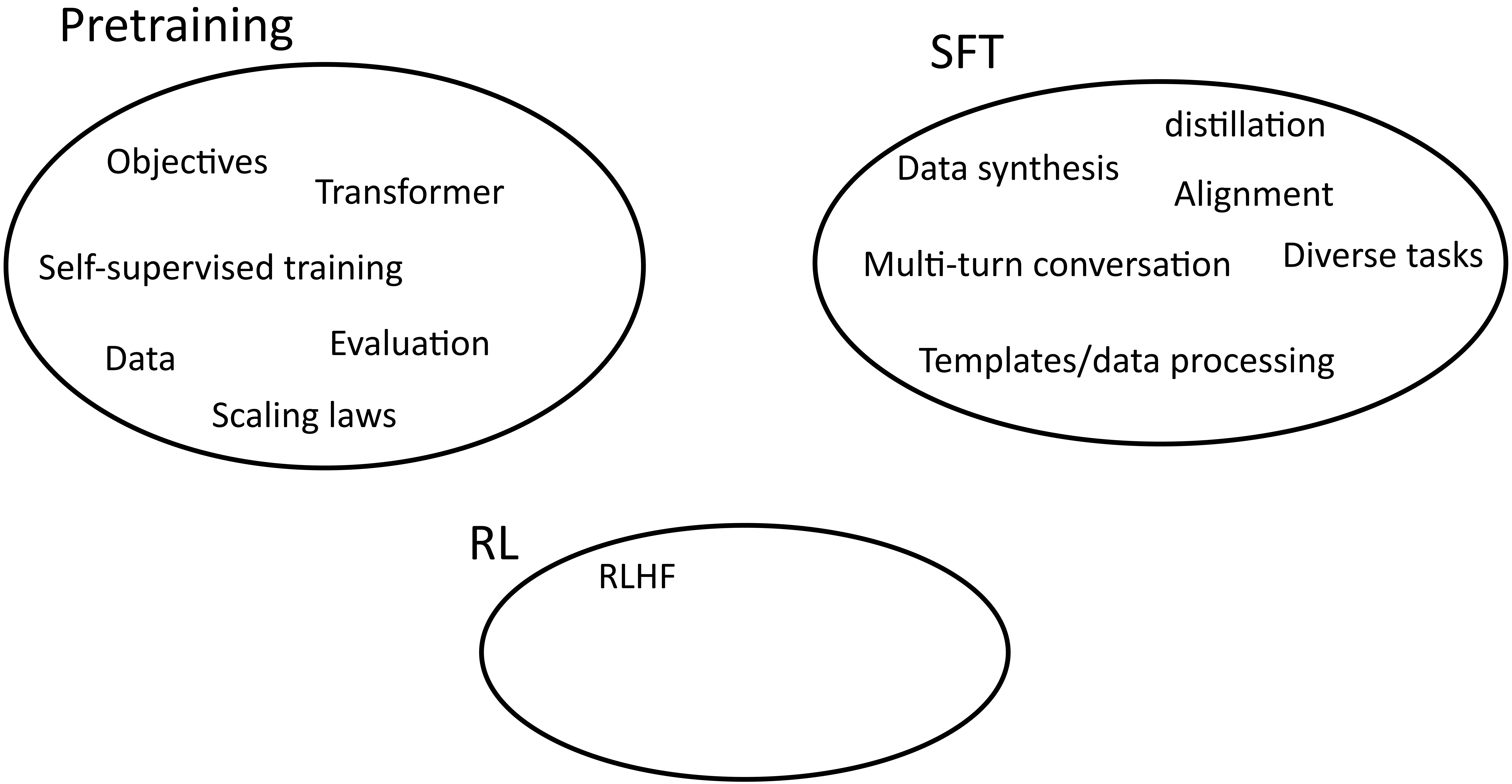
Review — Method



Review — Method



Review — Method



Review — Method

Pretraining

Objectives
Transformer
Self-supervised training
Data
Evaluation
Scaling laws

SFT

distillation
Data synthesis
Alignment
Multi-turn conversation
Diverse tasks
Templates/data processing

policy gradient

RL

RLHF
RLVR
reasoning

Review — Method

Pretraining

- Objectives
- Transformer
- Self-supervised training
- Data
- Evaluation
- Scaling laws

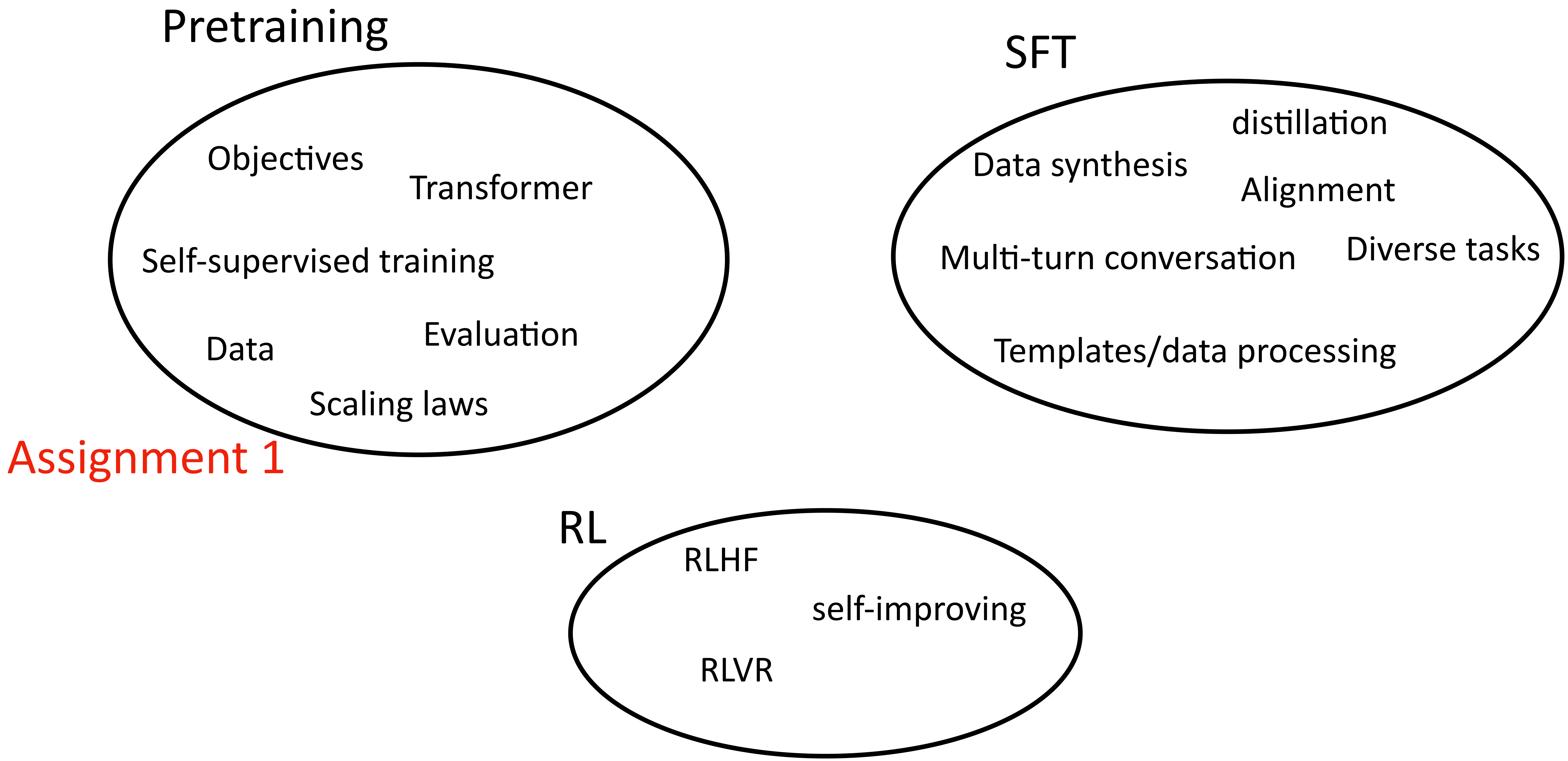
SFT

- distillation
- Data synthesis
- Alignment
- Multi-turn conversation
- Diverse tasks
- Templates/data processing

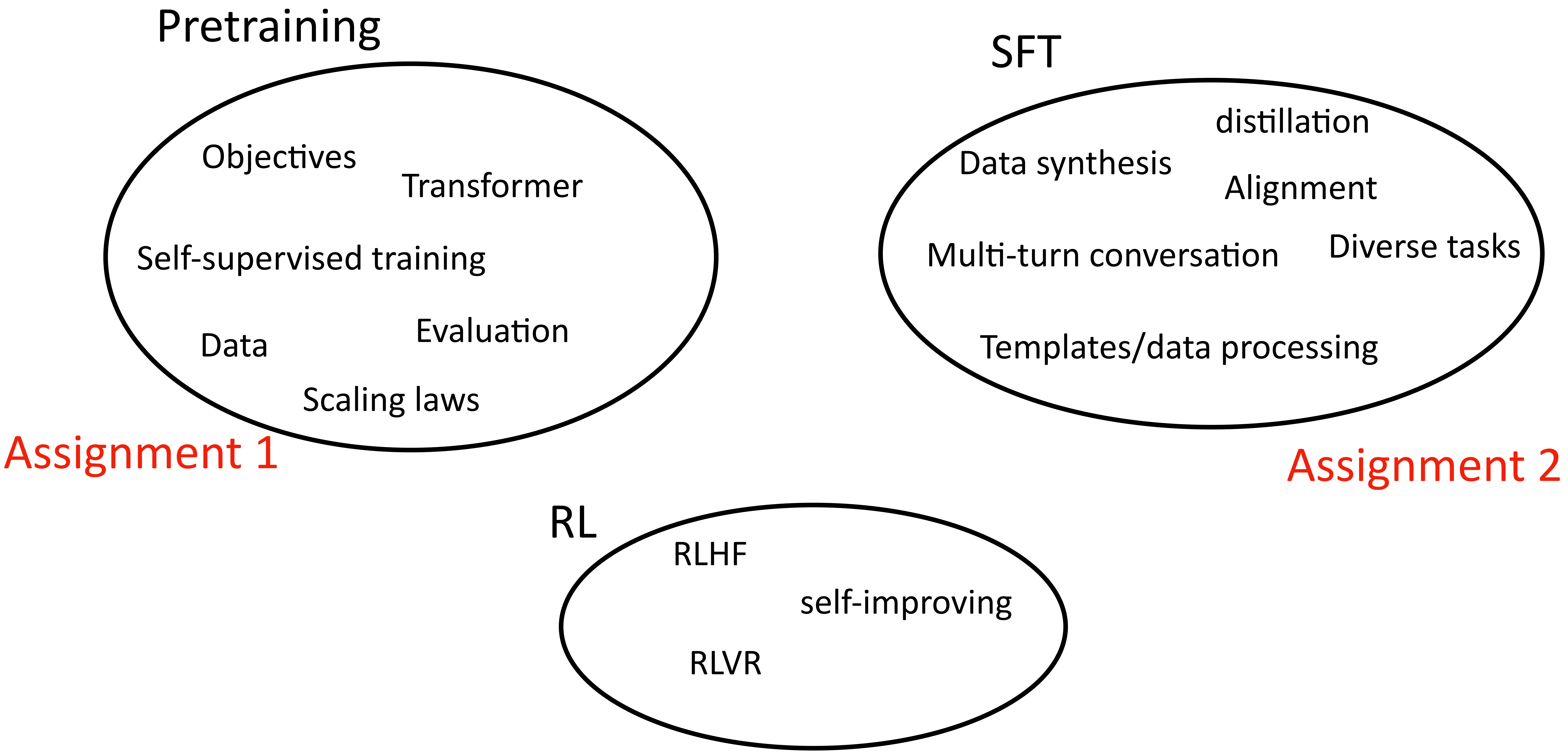
RL

- RLHF
- self-improving — no imitation learning
- RLVR

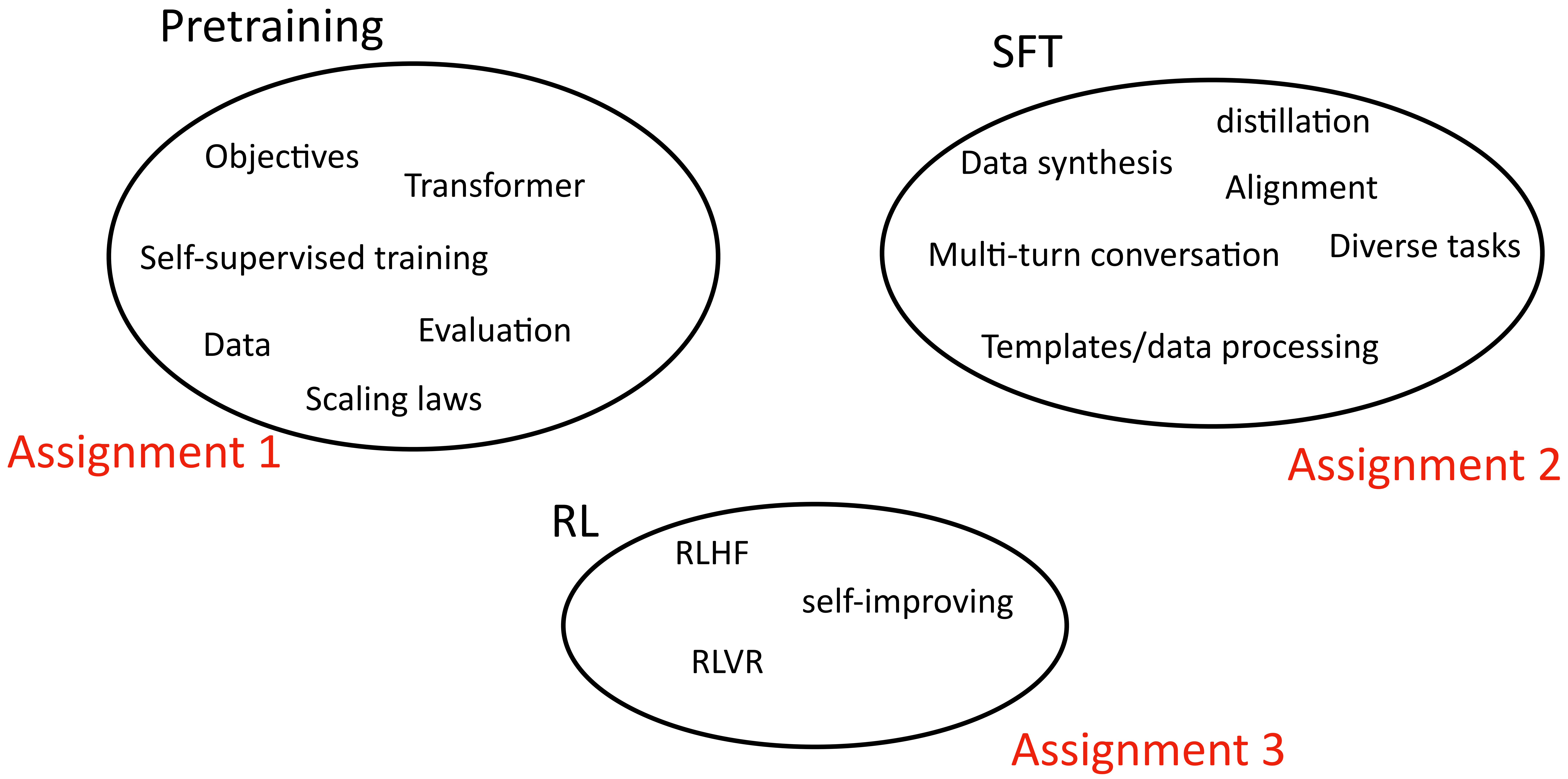
Review — Method



Review — Method



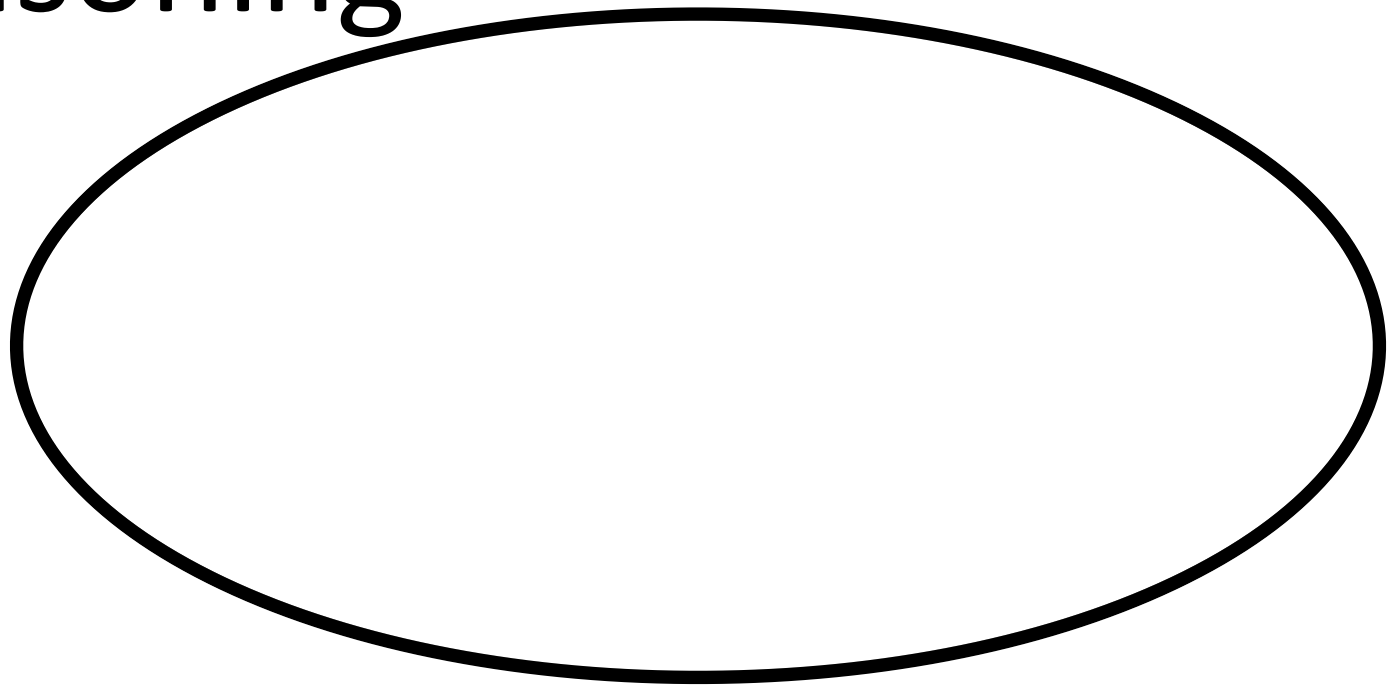
Review — Method



Review — Applications

Review — Applications

Reasoning



Review — Applications

Reasoning

chain-of-thought

Review — Applications

Reasoning

chain-of-thought

Deep reasoning

Review — Applications

Reasoning

chain-of-thought

RLVR

Deep reasoning

Review — Applications

Reasoning

chain-of-thought

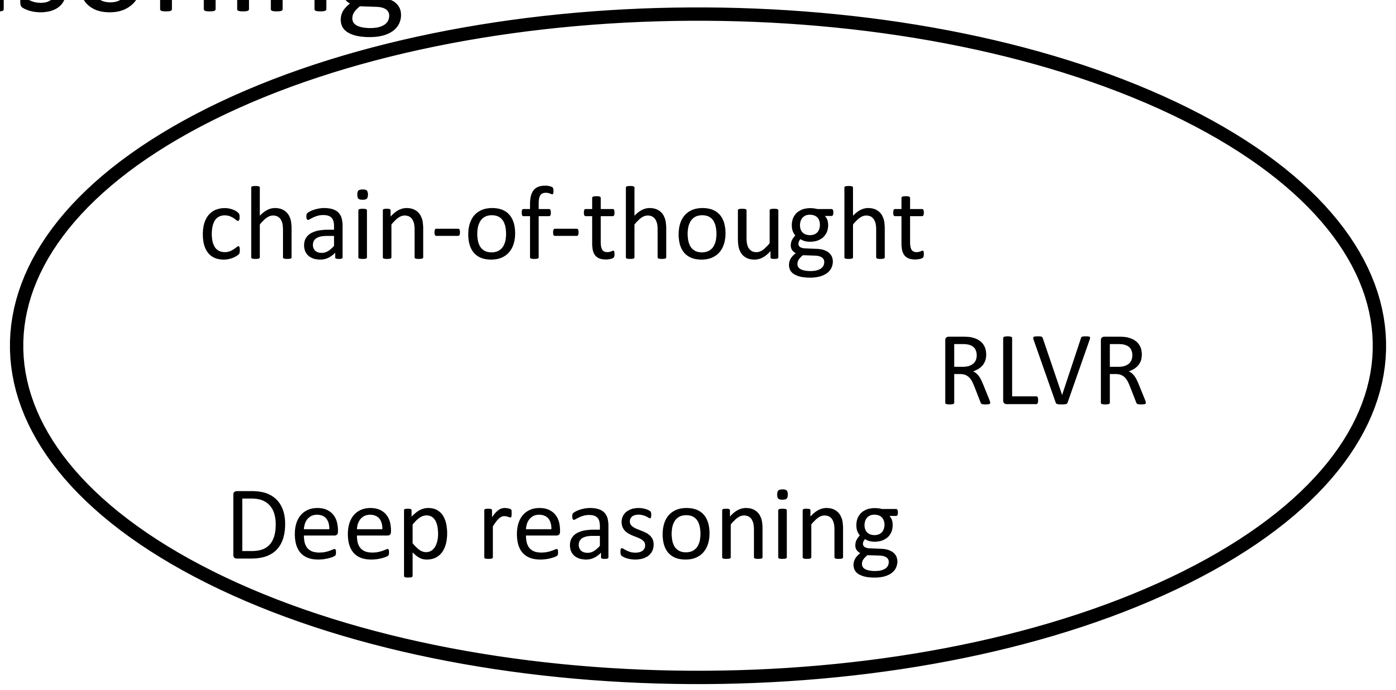
RLVR

Deep reasoning

Assignment 3, 4

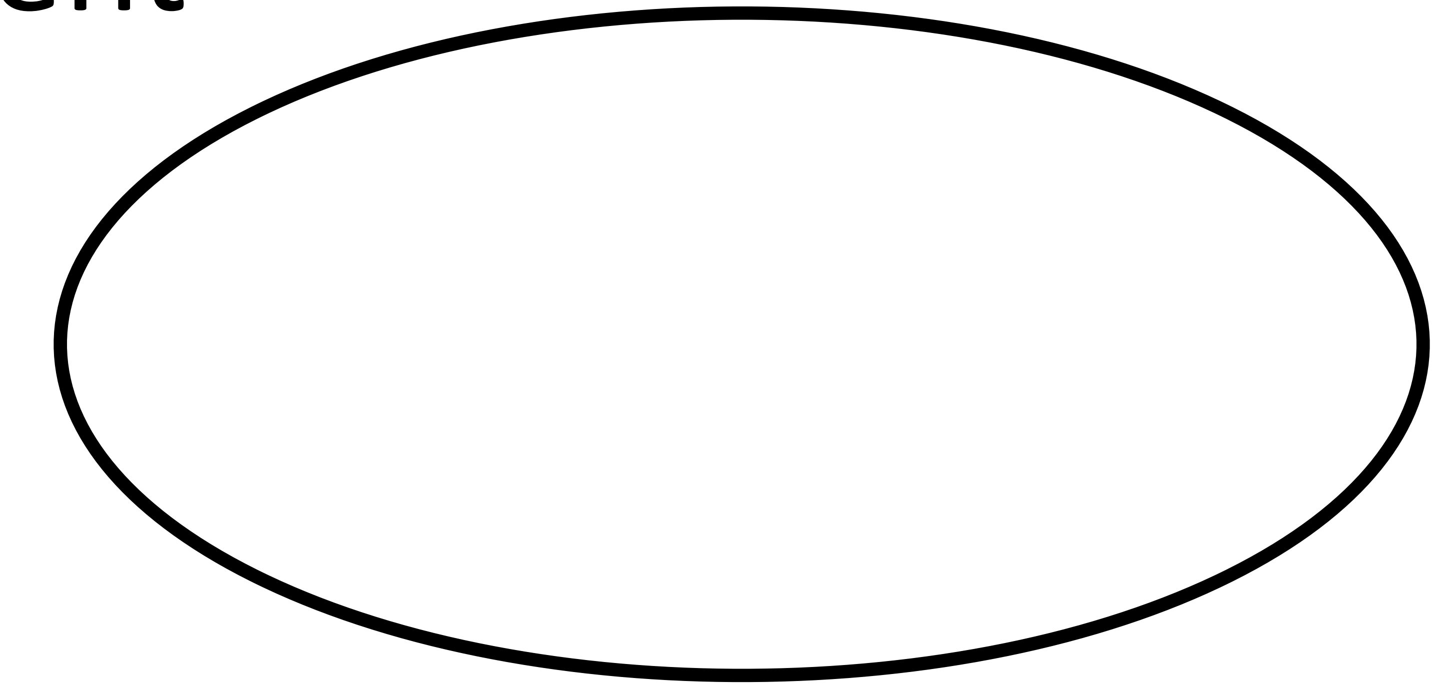
Review — Applications

Reasoning



Assignment 3, 4

Agent



Review — Applications

Reasoning

chain-of-thought

RLVR

Deep reasoning

Assignment 3, 4

Agent

Agent loop

Review — Applications

Reasoning

chain-of-thought

RLVR

Deep reasoning

Assignment 3, 4

Agent

Agent loop

Tool call

Review — Applications

Reasoning

chain-of-thought

RLVR

Deep reasoning

Assignment 3, 4

Agent

Agent loop

Tool call

Environment

Review — Applications

Reasoning

chain-of-thought

RLVR

Deep reasoning

Assignment 3, 4

Agent

Agent loop

Tool call

Environment

Scaffold

Review — Applications

Reasoning

chain-of-thought

RLVR

Deep reasoning

Assignment 3, 4

Agent

Agent loop

Tool call

Environment

Scaffold

Group project + Assignment 4

Review — Applications

Reasoning

chain-of-thought

RLVR

Deep reasoning

Assignment 3, 4

Agent

Agent loop

Tool call

Environment

Scaffold

Group project + Assignment 4

Ethics/Safety

Review — Applications

Reasoning

chain-of-thought

RLVR

Deep reasoning

Assignment 3, 4

Agent

Agent loop

Tool call

Environment

Scaffold

Group project + Assignment 4

Ethics/Safety

Bias

Review — Applications

Reasoning

chain-of-thought

RLVR

Deep reasoning

Assignment 3, 4

Agent

Agent loop

Tool call

Environment

Scaffold

Group project + Assignment 4

Ethics/Safety

Bias

Harmfulness

Review — Applications

Reasoning

- chain-of-thought
- RLVR
- Deep reasoning

Assignment 3, 4

Agent

- Agent loop
- Tool call
- Environment
- Scaffold

Group project + Assignment 4

Ethics/Safety

- Bias
- Harmfulness
- Attack

Review — Applications

Reasoning

- chain-of-thought
- RLVR
- Deep reasoning

Assignment 3, 4

Agent

- Agent loop
- Tool call
- Environment
- Scaffold

Group project + Assignment 4

Ethics/Safety

- Bias
- Harmfulness
- Defense
- Attack

Review — Applications

Reasoning

- chain-of-thought
- RLVR
- Deep reasoning

Assignment 3, 4

Agent

- Agent loop
- Tool call
- Environment
- Scaffold

Group project + Assignment 4

Ethics/Safety

- Hallucination
- Bias
- Harmfulness
- Defense
- Attack

Review — Applications

Reasoning

chain-of-thought
RLVR
Deep reasoning

Assignment 3, 4

Agent

Agent loop
Tool call
Environment
Scaffold

Group project + Assignment 4

Ethics/Safety

RAG
Hallucination
Bias
Harmfulness
Defense
Attack

Review — Applications

Reasoning

chain-of-thought
RLVR
Deep reasoning

Assignment 3, 4

Agent

Agent loop
Tool call
Environment
Scaffold

Group project + Assignment 4

Ethics/Safety

RAG
Hallucination
Bias
Harmfulness
Defense
Attack

Group project

About Large Language Models

About Large Language Models

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

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

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

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!