

DeepSeek-R1

R1 model and Ideas used in DeepSeek models

Outline

- Overview
- Ideas used in previous DeepSeek models
- Deepseek-R1 model
- Summary

DeepSeek-R1

Overview

- DeepSeek is a Chinese startup founded in 2023 with ~ 200 employees
- First open-weight reasoning model developed by DeepSeek
- Competing with OpenAI o1 reasoning model
- DeepSeek-R1 reveals its thinking, while o1 does not
- Uses many ideas used in DeepSeek's previous models
- DeepSeek team has proposed many efficient methods for training and inference



Models from DeepSeek

Jan 2024

April 2024

DeepSeek LLM
Scaling Open-Source Language Models with Longtermism

DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models

Damai Dai^{*1,2}, Chengqi Deng¹, Chenggang Zhao^{*1,3}, R.X. Xu¹, Huazuo Gao¹, Deli Chen¹, Jiashi Li¹, Wangding Zeng¹, Xingkai Yu^{*1,4}, Y. Wu¹, Zhenda Xie¹, Y.K. Li¹, Panpan Huang¹, Fuli Luo¹, Chong Ruan¹, Zhifang Sui², Wenfeng Liang¹

DeepSeek-Coder: When the Large Language Model Meets Programming - The Rise of Code Intelligence

Daya Guo^{*1}, Qihao Zhu^{*1,2}, Dejian Yang¹, Zhenda Xie¹, Kai Dong¹, Wentao Zhang¹, Guanting Chen¹, Xiao Bi¹, Y. Wu¹, Y.K. Li¹, Fuli Luo¹, Yingfei Xiong², Wenfeng Liang¹

DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models

Zihong Shao^{1,2*}, Peiyi Wang^{1,3*}, Qihao Zhu^{1,3*}, Runxin Xu¹, Junxiao Song¹, Xiao Bi¹, Haowei Zhang¹, Mingchuan Zhang¹, Y.K. Li¹, Y. Wu¹, Daya Guo^{1*}

¹DeepSeek-AI, ²Tsinghua University, ³Peking University

Jun 2024

Dec 2024

Jan 2025

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

DeepSeek-AI

research@deepseek.com

DeepSeek-V3 Technical Report

DeepSeek-AI

research@deepseek.com



DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

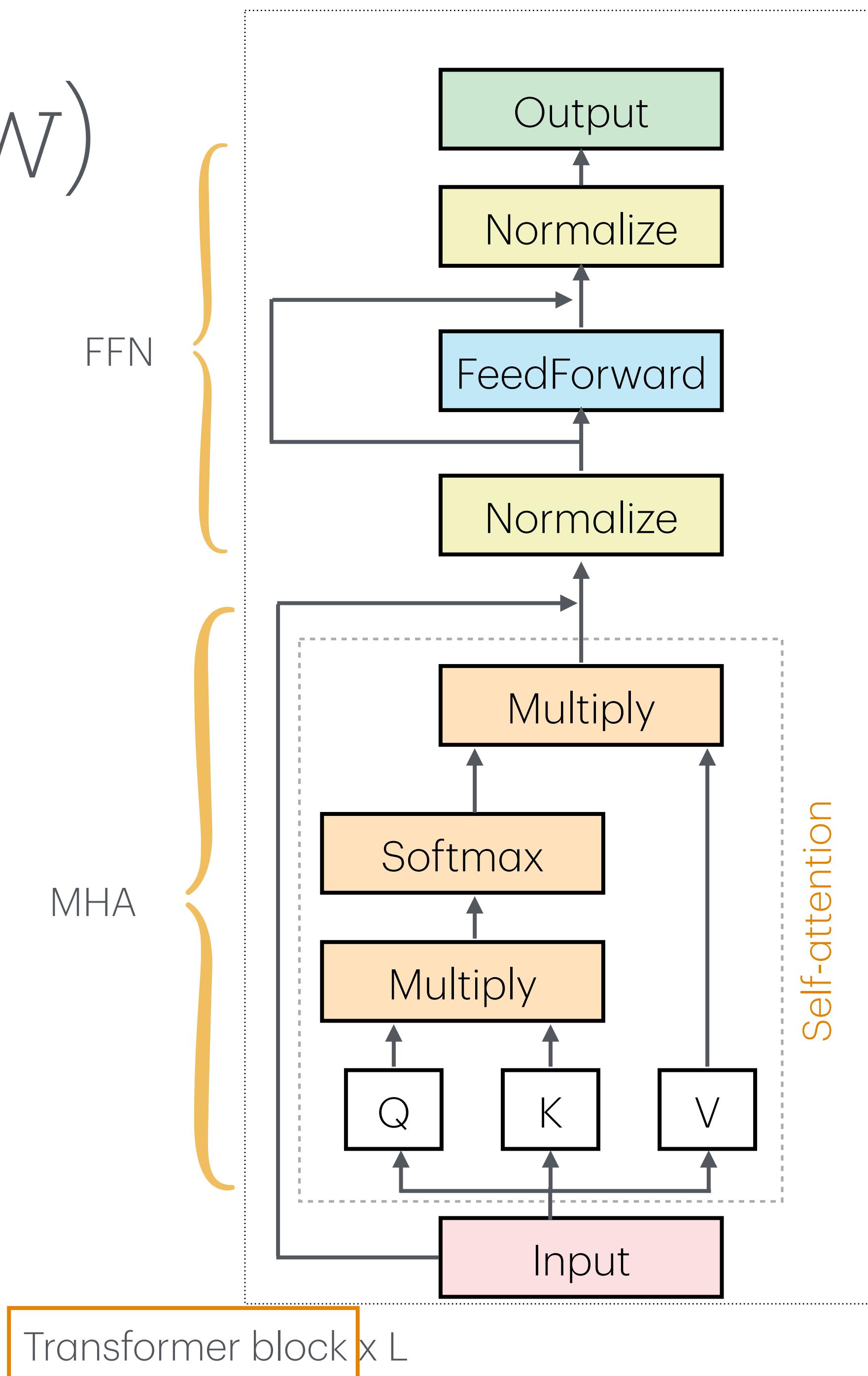
DeepSeek-AI

research@deepseek.com

Ideas used in DeepSeek models

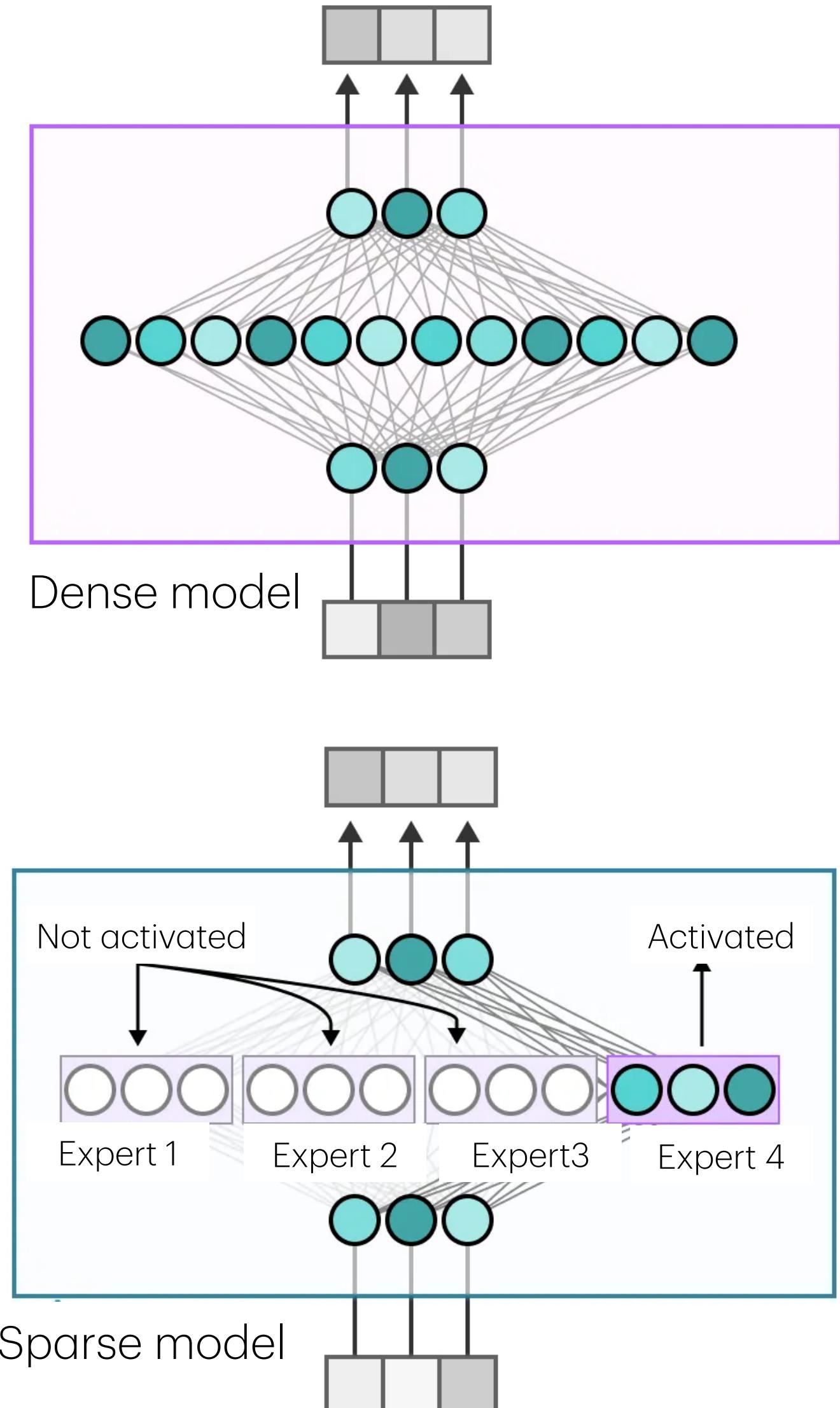
Transformer model (review)

- Transformer architecture/model improves the issues with RNN, such as providing attention paths, enabling parallel computations.
- It achieved great success in NLP
- **Transformer block** is a core block for the Transformer architecture, which includes
 - multi-headed self attention (MHA)
 - feedforward network (FFN)



Mixture of Expert, MoE (Review)

- According to scaling law*, model performance depends on i) # model parameters, N, ii) size of the dataset, D, iii) amount of compute in training, C
- MoE is a way to train larger model with lower training compute budget. Also MoE model requires lower inference budget
- MoE create a sparse model. Only 1 out of 4 experts is used. It is computationally efficient due to sparsity.
- MoE helps to increase parameters while keeping the compute cost manageable.



*: <https://arxiv.org/abs/2001.08361> Scaling Laws for Neural Language Models

Mixture of Expert, MoE (Review)

- FFN layer of Transformer block is substituted with MoE layer
- Each expert is an identical structure FFN
- Challenges in MoE architectures
 - Knowledge hybridity: expert learns diverse knowledge due to its limited numbers
 - Knowledge redundancy: multiple experts learn common knowledge due to no common expert(s)
 - Routing collapse: selects a few experts so other experts may not trained fully

Output hidden

$$\mathbf{h}_t^l = \sum_{i=1}^N \left(g_{i,t} \text{FFN}_i \left(\mathbf{u}_t^l \right) \right) + \mathbf{u}_t^l,$$

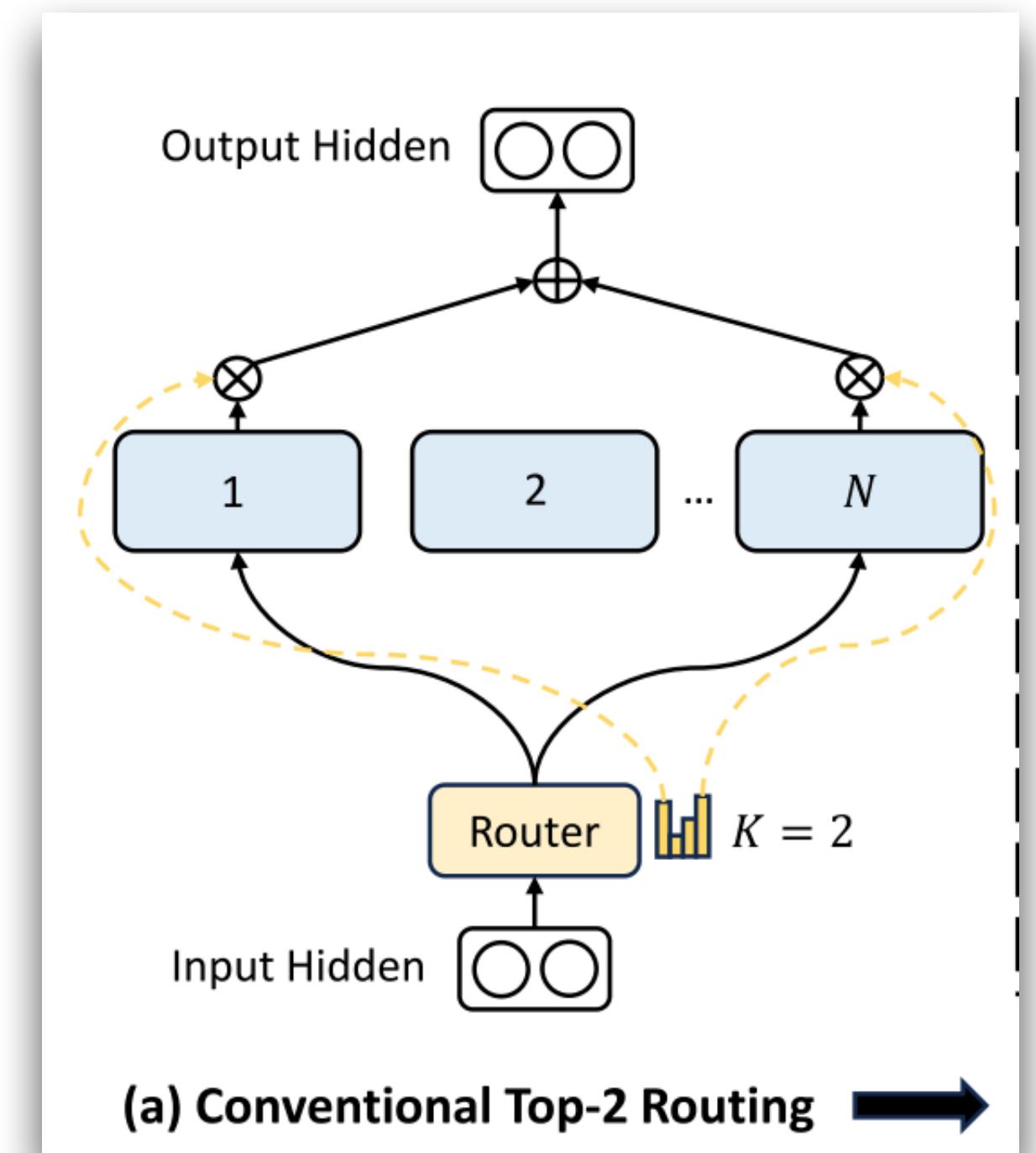
Gate, select expert

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \text{Topk}(\{s_{j,t} | 1 \leq j \leq N\}, K), \\ 0, & \text{otherwise,} \end{cases}$$

Input hidden

$$s_{i,t} = \text{Softmax}_i \left(\mathbf{u}_t^{lT} \mathbf{e}_i^l \right),$$

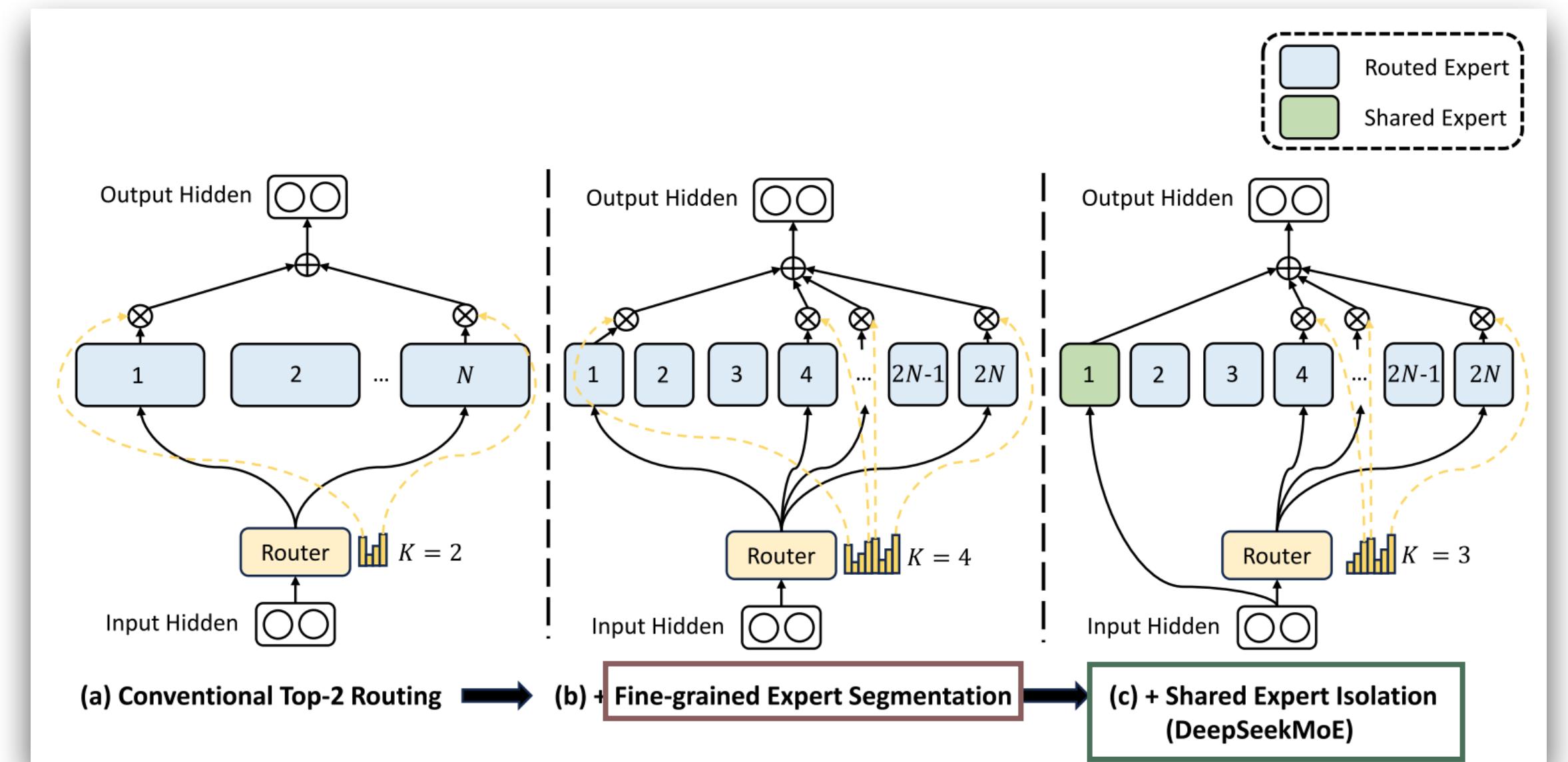
Router



DeepSeekMoE

Design to improve expert specialization

- Fine-grained expert segmentation: splitting experts into finer groups and activating more experts allows them to learn diverse knowledge. Achieving more accurate and targeted knowledge acquisition.
- Shared expert isolation: Dedicate shared experts that are always activated to capture common knowledge across contexts. Prevent other experts from learning redundant knowledge.



Output hidden

$$\mathbf{h}_t^l = \sum_{i=1}^{K_s} \text{FFN}_i(\mathbf{u}_t^l) + \sum_{i=K_s+1}^{mN} (g_{i,t} \text{FFN}_i(\mathbf{u}_t^l)) + \mathbf{u}_t^l,$$

Gate, select
expert

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \text{Topk}(\{s_{j,t} | K_s + 1 \leq j \leq mN\}, mK - K_s), \\ 0, & \text{otherwise,} \end{cases}$$

$$s_{i,t} = \text{Softmax}_i(\mathbf{u}_t^{lT} \mathbf{e}_i^l)$$

Finer
grained
experts
Shared experts

Router

DeepSeekMoE

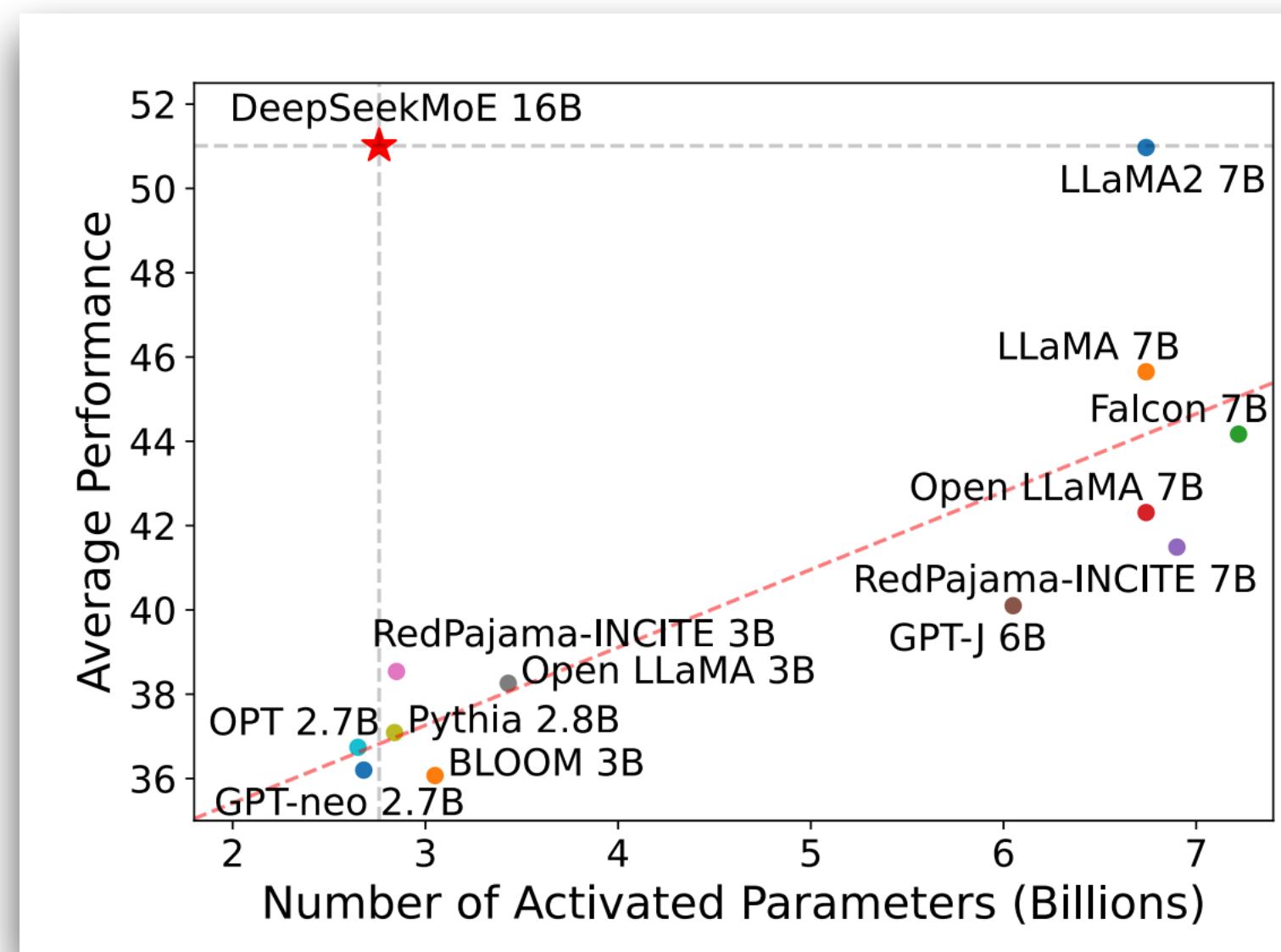
- To validate their idea, they used a small model with 2B parameters, DeepSeekMoE 2B
- DeepSeekMoE 2B outperform GShard 2.9B, which is large and x1.5 more activated experts.
- DeepSeekMoE 2B approaches close to DeepSeek 2B (dense model), which is a upper bound of MoE model

Metric	# Shot	Dense	Hash Layer	Switch	GShard	DeepSeekMoE
# Total Params	N/A	0.2B	2.0B	2.0B	2.0B	2.0B
# Activated Params	N/A	0.2B	0.2B	0.2B	0.3B	0.3B
FLOPs per 2K Tokens	N/A	2.9T	2.9T	2.9T	4.3T	4.3T
# Training Tokens	N/A	100B	100B	100B	100B	100B
Pile (Loss)	N/A	2.060	1.932	1.881	1.867	1.808
HellaSwag (Acc.)	0-shot	38.8	46.2	49.1	50.5	54.8
PIQA (Acc.)	0-shot	66.8	68.4	70.5	70.6	72.3
ARC-easy (Acc.)	0-shot	41.0	45.3	45.9	43.9	49.4
ARC-challenge (Acc.)	0-shot	26.0	28.2	30.2	31.6	34.3
RACE-middle (Acc.)	5-shot	38.8	38.8	43.6	42.1	44.0
RACE-high (Acc.)	5-shot	29.0	30.0	30.9	30.4	31.7
HumanEval (Pass@1)	0-shot	0.0	1.2	2.4	3.7	4.9
MBPP (Pass@1)	3-shot	0.2	0.6	0.4	0.2	2.2
TriviaQA (EM)	5-shot	4.9	6.5	8.9	10.2	16.6
NaturalQuestions (EM)	5-shot	1.4	1.4	2.5	3.2	5.7

Table 1 | Evaluation results for validation experiments. **Bold** font indicates the best. Compared with other MoE architectures, DeepSeekMoE exhibits a substantial performance advantage.

DeepSeekMoE

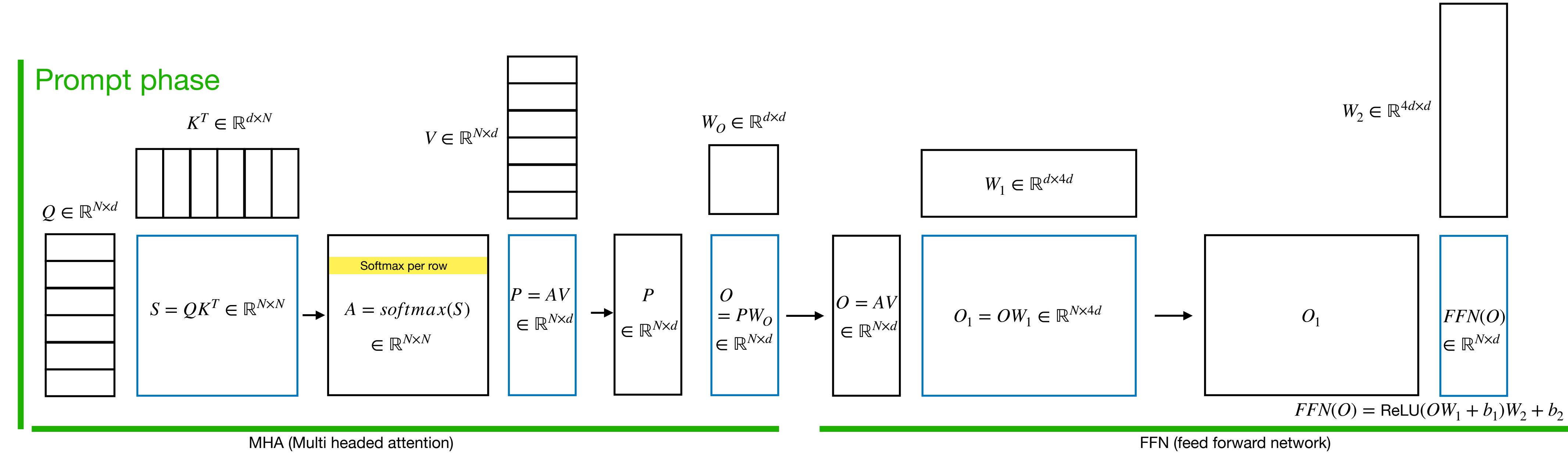
- DeepSeekMoE 16B achieves similar performance with DeepSeek 7B with only 40.5% computation!



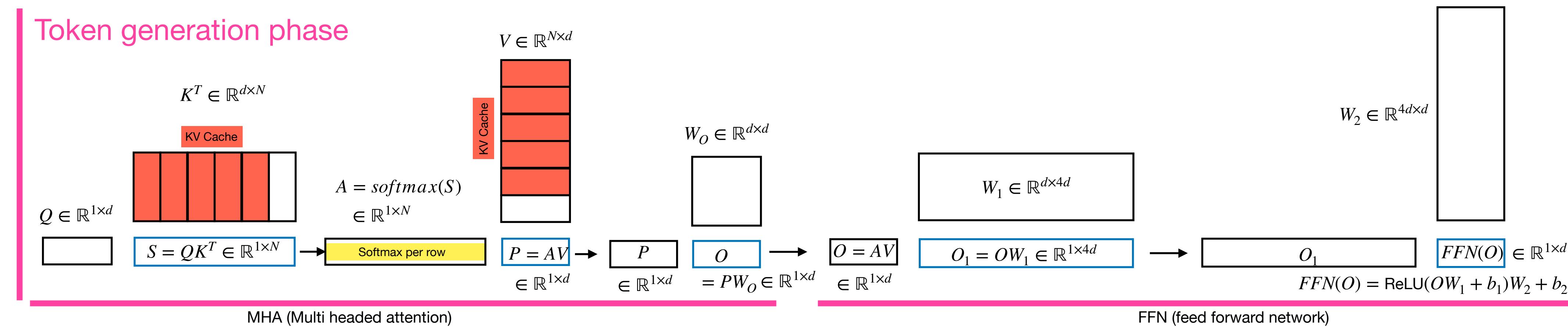
Metric	# Shot	DeepSeek 7B (Dense)	DeepSeekMoE 16B
# Total Params	N/A	6.9B	16.4B
# Activated Params	N/A	6.9B	2.8B
FLOPs per 4K Tokens	N/A	183.5T	74.4T
# Training Tokens	N/A	2T	2T
Pile (BPB)	N/A	0.75	0.74
HellaSwag (Acc.)	0-shot	75.4	77.1
PIQA (Acc.)	0-shot	79.2	80.2
ARC-easy (Acc.)	0-shot	67.9	68.1
ARC-challenge (Acc.)	0-shot	48.1	49.8
RACE-middle (Acc.)	5-shot	63.2	61.9
RACE-high (Acc.)	5-shot	46.5	46.4
DROP (EM)	1-shot	34.9	32.9
GSM8K (EM)	8-shot	17.4	18.8
MATH (EM)	4-shot	3.3	4.3
HumanEval (Pass@1)	0-shot	26.2	26.8
MBPP (Pass@1)	3-shot	39.0	39.2
TriviaQA (EM)	5-shot	59.7	64.8
NaturalQuestions (EM)	5-shot	22.2	25.5
MMLU (Acc.)	5-shot	48.2	45.0
WinoGrande (Acc.)	0-shot	70.5	70.2
CLUEWSC (EM)	5-shot	73.1	72.1
CEval (Acc.)	5-shot	45.0	40.6
CMMU (Acc.)	5-shot	47.2	42.5
CHID (Acc.)	0-shot	89.3	89.4

Two phases of LLM inference (review)

- Prompt (prefill) phase: all prompt tokens are processed in parallel, high compute intensity. Prefill phase runs only once



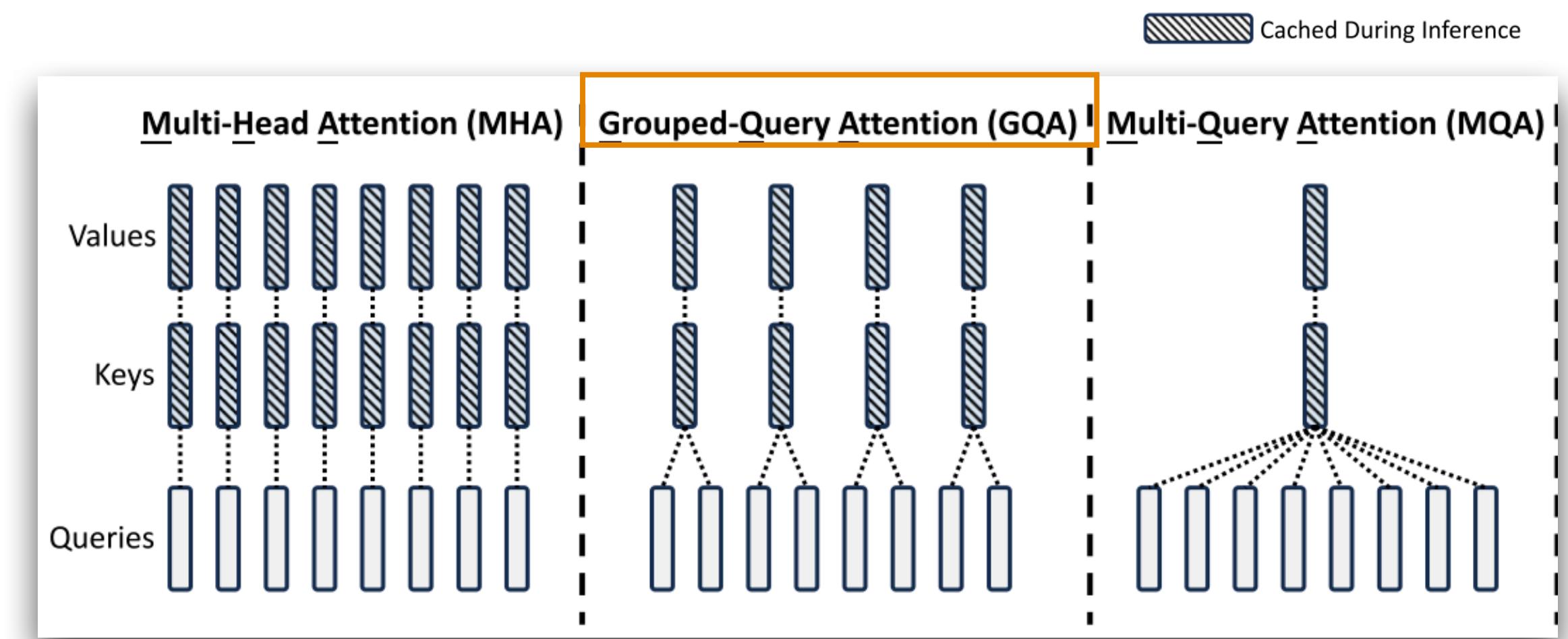
- Token generation phase: generating only one token at a time. To save compute, key and value are cached (KV-cache)



MHA and KV-Cache (review)

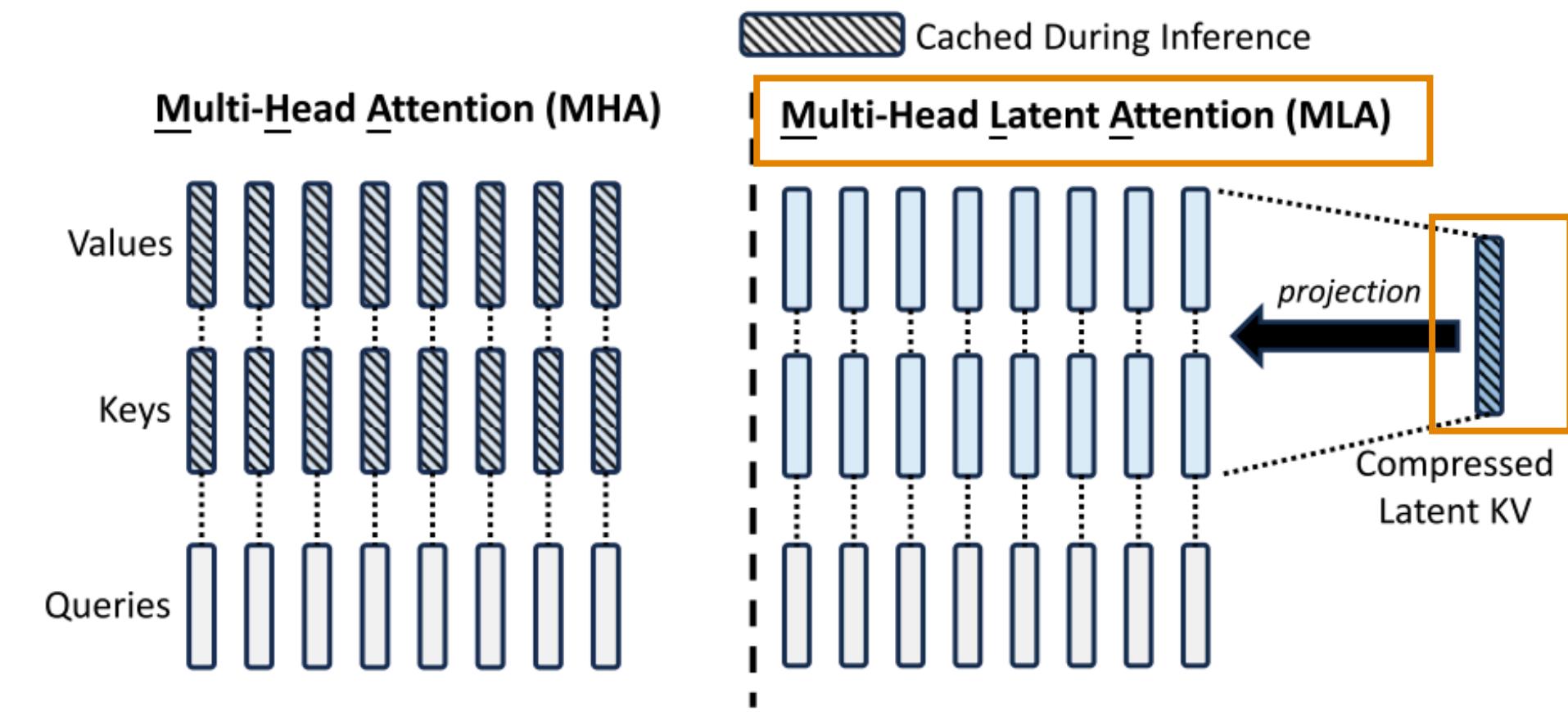
MHA (multi-head attention), KV (Key and Value)

- To reduce inference time, key and value are cached during inference.
- It poses challenges to keep them in the limited GPU memory
 - Size of KV-cache: $\#batch * \#layers * \#KV\ heads * emb * seq\ len * 2 (K\&V) * 2 (\text{bytes})$
 - large KV-size limits large batch size and sequence length
- Approaches to reduce KV-Cache size: GQA (group query), MQA (multi query)
 - Issues: compromise performance by reducing KV-cache



MLA (Multi-head Latent Attention)

- Use low-rank key-value joint compression: results in much smaller amount of KV-cache
- Better performance drop compared to MHA
 - Size of KV-cache: #batch * #layers * 1 * emb_c * seq len * 2 (K&V) * 2 (bytes)



$$\mathbf{c}_t^{KV} = W^{DKV} \mathbf{h}_t, \\ \mathbf{k}_t^C = W^{UK} \mathbf{c}_t^{KV}, \\ \mathbf{v}_t^C = W^{UV} \mathbf{c}_t^{KV},$$

$$\mathbf{c}_t^Q = W^{DQ} \mathbf{h}_t, \\ \mathbf{q}_t^C = W^{UQ} \mathbf{c}_t^Q,$$

Not for KV-cache, but for reducing activation memory

MLA (Multi-head Latent Attention)

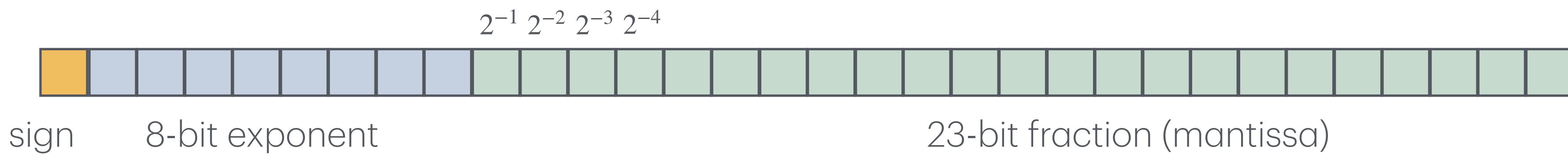
Decoupled with Rotary positional encoding (RoPE)

- Rotary positional encoding (RoPE): widely used positional encoding
 - Rotate embeddings in 2D space
 - Applied to query and key.
- With RoPE applied, lower the MLA's efficiency
- Use additional multi-head queries and a shared key to carry RoPE.

$$\begin{aligned} [\mathbf{q}_{t,1}^R; \mathbf{q}_{t,2}^R; \dots; \mathbf{q}_{t,n_h}^R] &= \boxed{\mathbf{q}_t^R} = \text{RoPE}(W^{QR}\mathbf{c}_t^Q), \\ \mathbf{k}_t^R &= \text{RoPE}(W^{KR}\mathbf{h}_t), \\ \mathbf{q}_{t,i} &= [\mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R], \\ \mathbf{k}_{t,i} &= [\mathbf{k}_{t,i}^C; \mathbf{k}_t^R], \\ \mathbf{o}_{t,i} &= \sum_{j=1}^t \text{Softmax}_j\left(\frac{\mathbf{q}_{t,i}^T \mathbf{k}_{j,i}}{\sqrt{d_h + d_h^R}}\right) \mathbf{v}_{j,i}^C, \\ \mathbf{u}_t &= W^O [\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; \dots; \mathbf{o}_{t,n_h}], \end{aligned}$$

Attention Mechanism	KV Cache per Token (# Element)	Capability
Multi-Head Attention (MHA)	$2n_h d_h l$	Strong
Grouped-Query Attention (GQA)	$2n_g d_h l$	Moderate
Multi-Query Attention (MQA)	$2d_h l$	Weak
MLA (Ours)	$(d_c + d_h^R)l \approx \frac{9}{2}d_h l$	Stronger

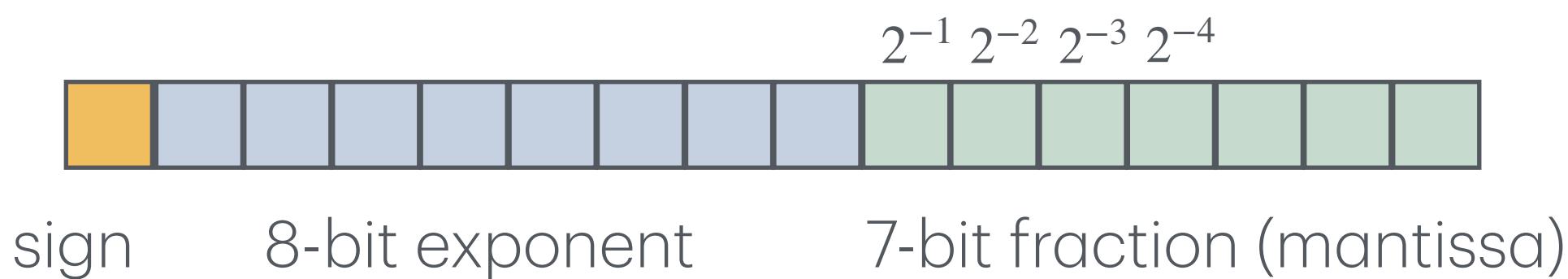
Floating point number



FP32

Represented number = $(-1)^{\text{sign}} \times (1 + \text{fraction}) \times 2^{\text{exponent-base}}$

$$\text{Base} = 127 = 2^{8-1} - 1$$



Brain float16 (BF16): dynamic range is now the same as FP32

- This can be considered as a truncated FP32
- ML applications are more tolerant to quantization error than they are to overflow (dynamic range)

Floating point number

FP8 (E4M3, E5M2). H100 supports FP8



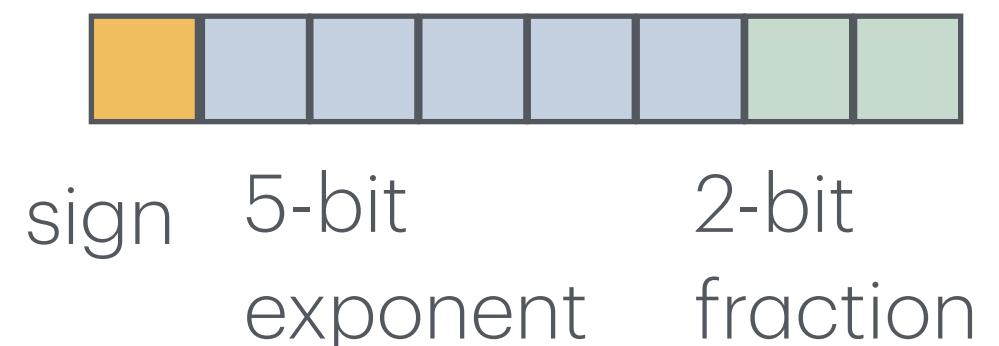
Exponent width determines dynamic range.
Fraction width determines precision. They need to be trade-off.

FP8 (E4M3)

FP8 E4M3 does not have INF, and S.1111.1112 is used for NaN.

Largest FP8 E4M3 normal value is S.1111.1102 =448.

For training, dynamic range is more important.
For inference, precision is more important

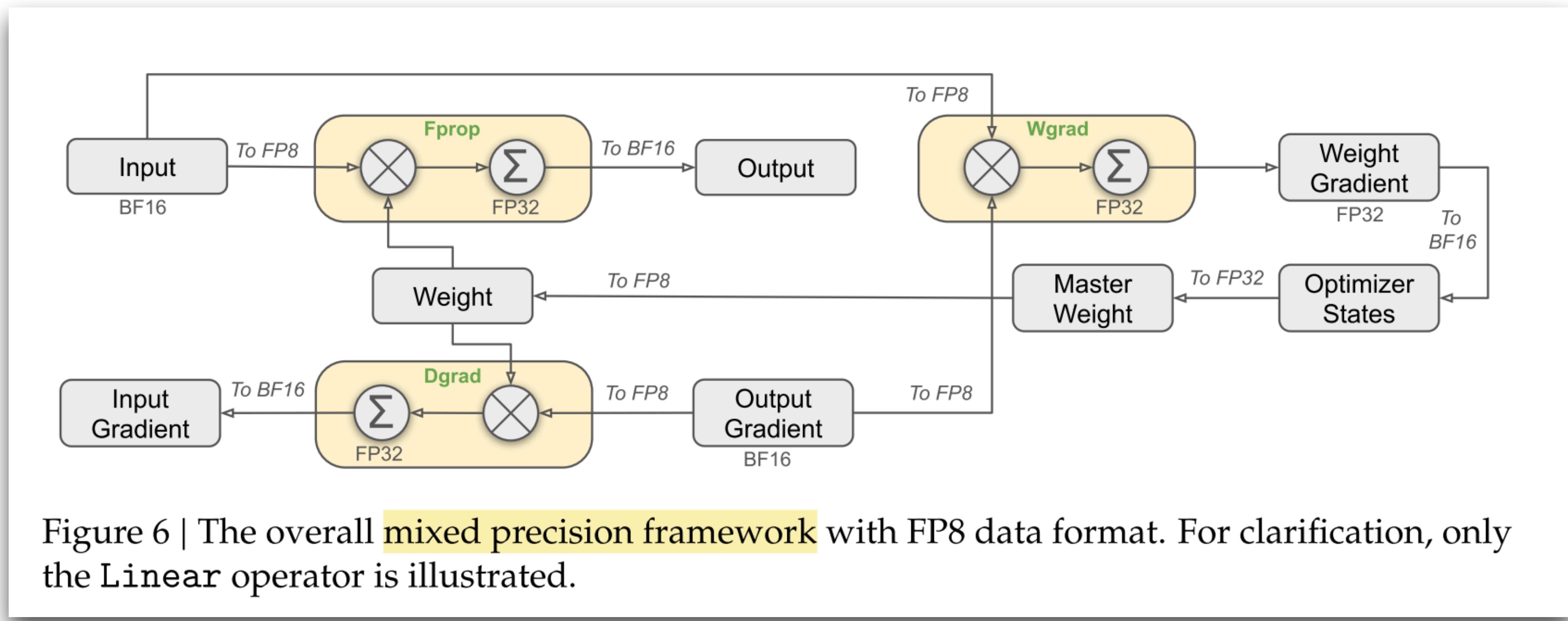


FP8 (E5M2) for gradient computation in backward propagation

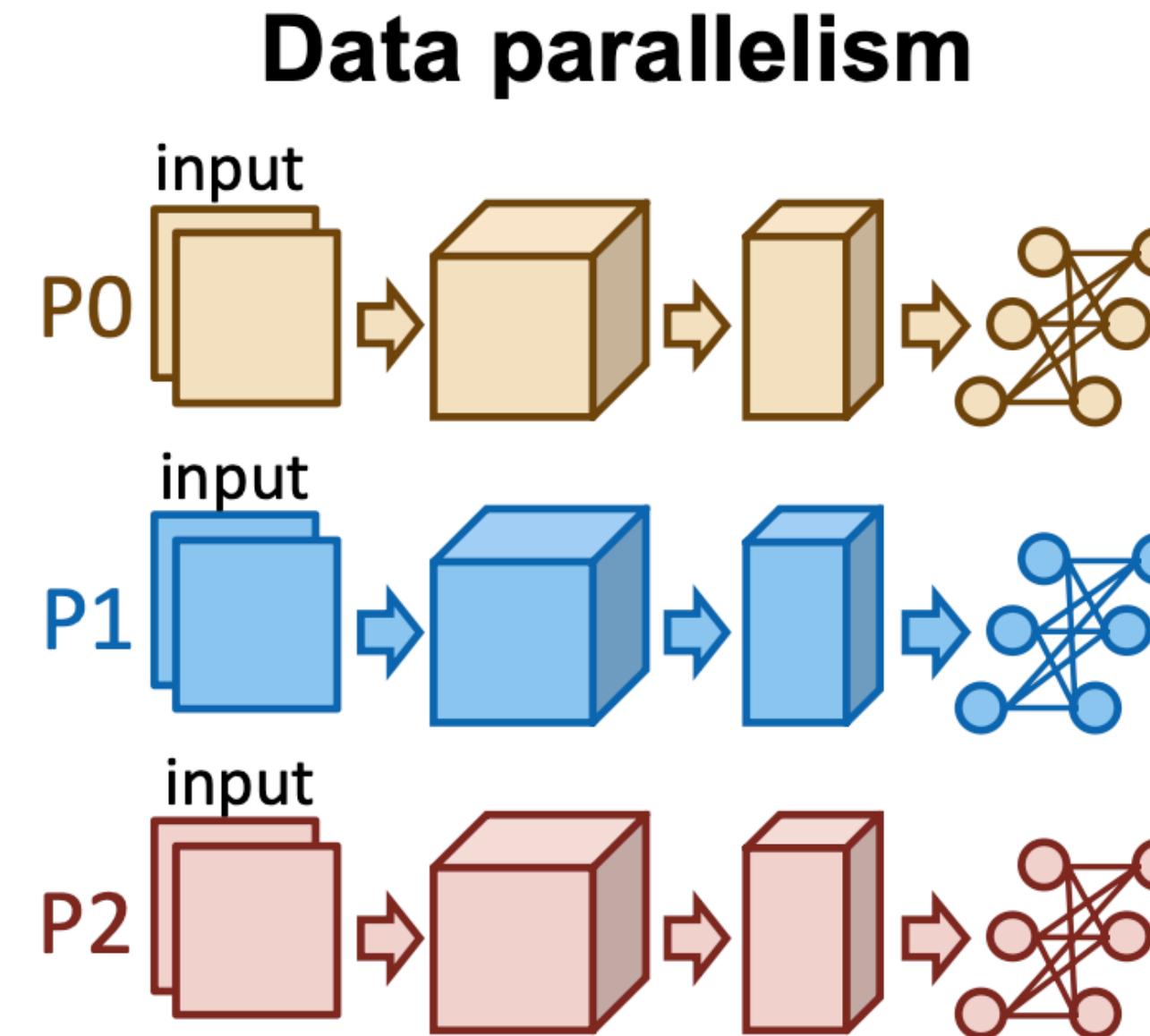
FP8 E5M2 have INF (S.11111.002) and NaN (S.11111.XX2).

Largest FP8 E5M2 normal value is S.11110.112 =57344.

FP8 mixed precision training

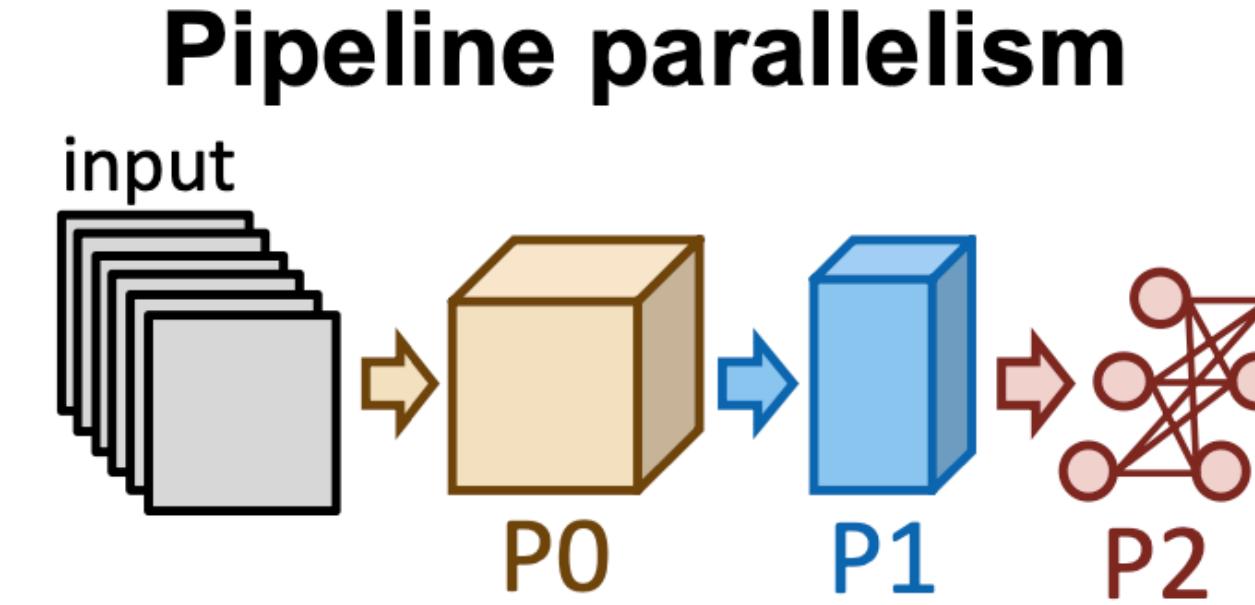


Large scale training (review)



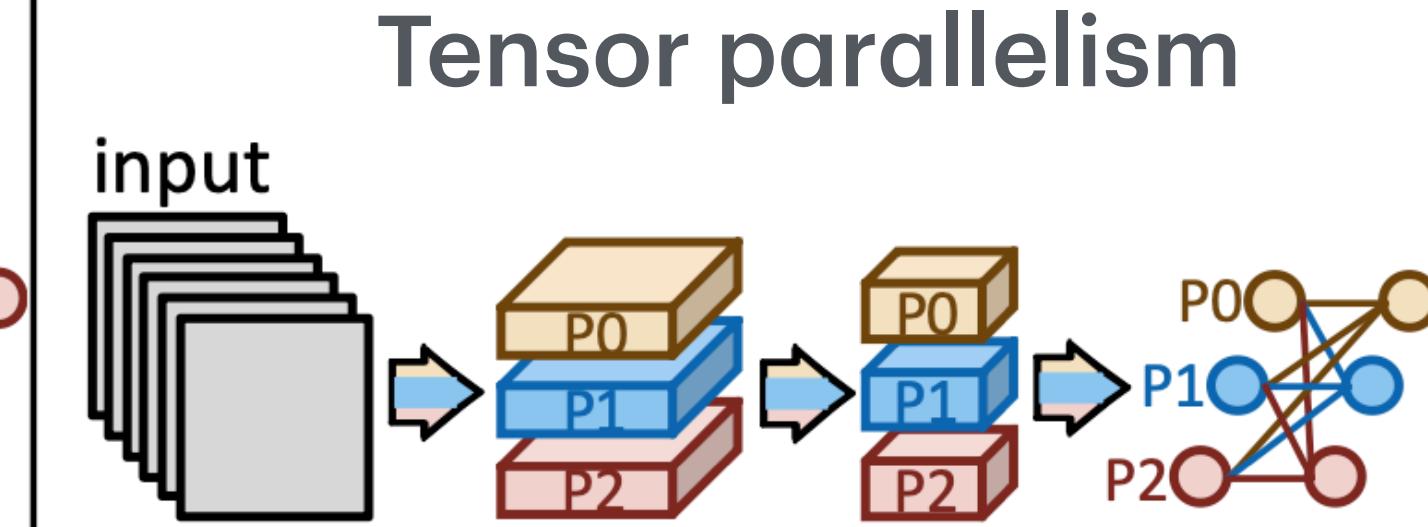
Pros:
a. Easy to realize

Cons:
a. Not work for large models
b. High allreduce overhead



Pros:
a. Make large model training feasible
b. No collective, only P2P

Cons:
a. Bubbles in pipeline
b. Removing bubbles leads to stale weights



Pros:
a. Make large model training feasible

Cons:
b. Communication for each operator (or each layer)

DualPipe & computation/comm overlap

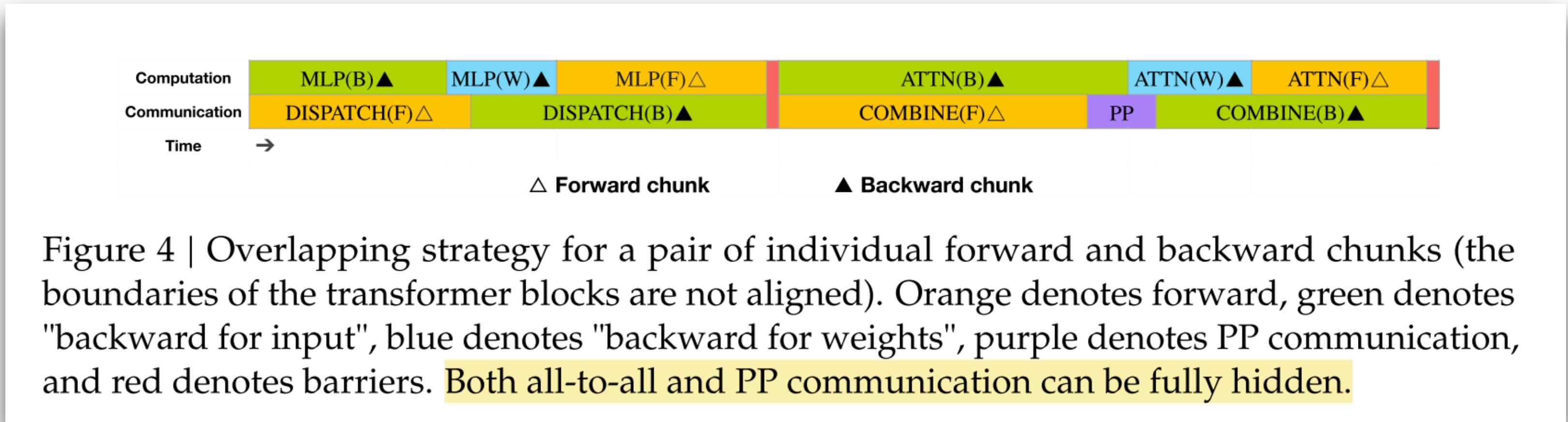


Figure 4 | Overlapping strategy for a pair of individual forward and backward chunks (the boundaries of the transformer blocks are not aligned). Orange denotes forward, green denotes "backward for input", blue denotes "backward for weights", purple denotes PP communication, and red denotes barriers. Both all-to-all and PP communication can be fully hidden.

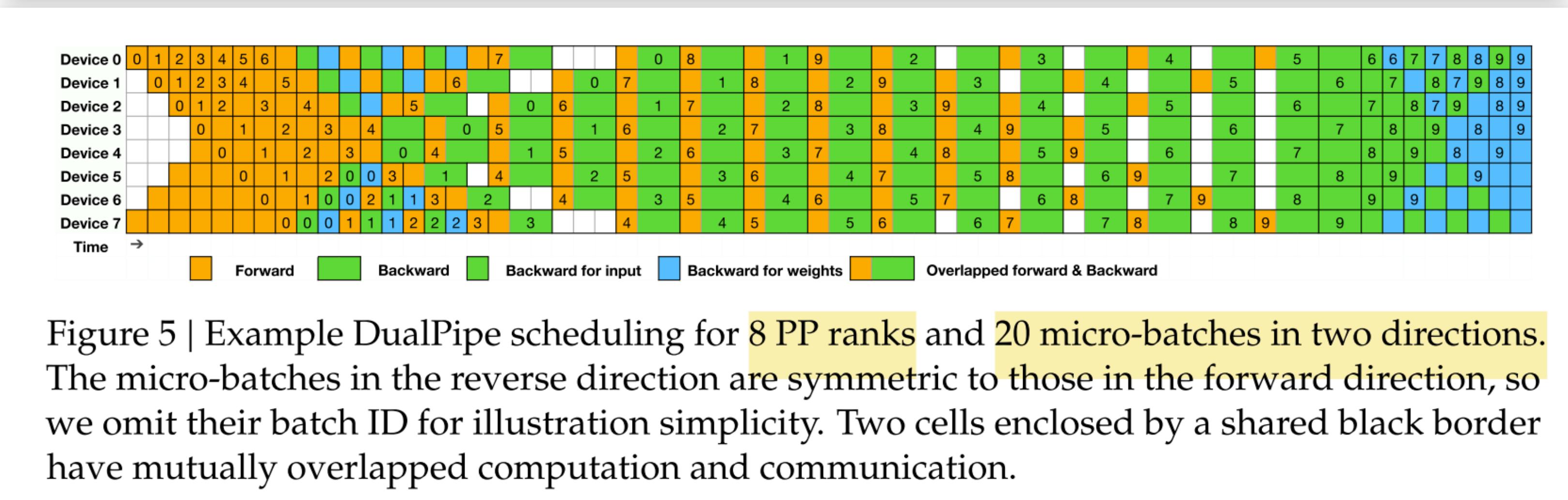


Figure 5 | Example DualPipe scheduling for 8 PP ranks and 20 micro-batches in two directions. The micro-batches in the reverse direction are symmetric to those in the forward direction, so we omit their batch ID for illustration simplicity. Two cells enclosed by a shared black border have mutually overlapped computation and communication.

Training framework

- FP8 mixed precision training: accelerated training and reduce GPU memory usage
- DualPipe algorithm: efficient pipeline parallelism. Fewer pipeline bubble and hides communication during training by communication/computation overlap
- Fully utilize InfiniBand (IB) and NVLink bandwidths with efficient cross-node all-to-all communication overhead
- They can avoid using tensor parallelism, which requires high network throughput, by reducing memory usage
- Highly-efficient AI training framework: efficient communication and distributed file system (3FS)

Training Costs	Pre-Training	Context Extension	Post-Training	Total
in H800 GPU Hours	2664K	119K	5K	2788K
in USD	\$5.328M	\$0.238M	\$0.01M	\$5.576M

Table 1 | Training costs of DeepSeek-V3, assuming the rental price of H800 is \$2 per GPU hour.

DeepSeek-V3

- Base mode for DeepSeek R1 model
- MoE model with 671B total parameters with 37B activated
- 128K context length
- High performance model
- Efficient training platform

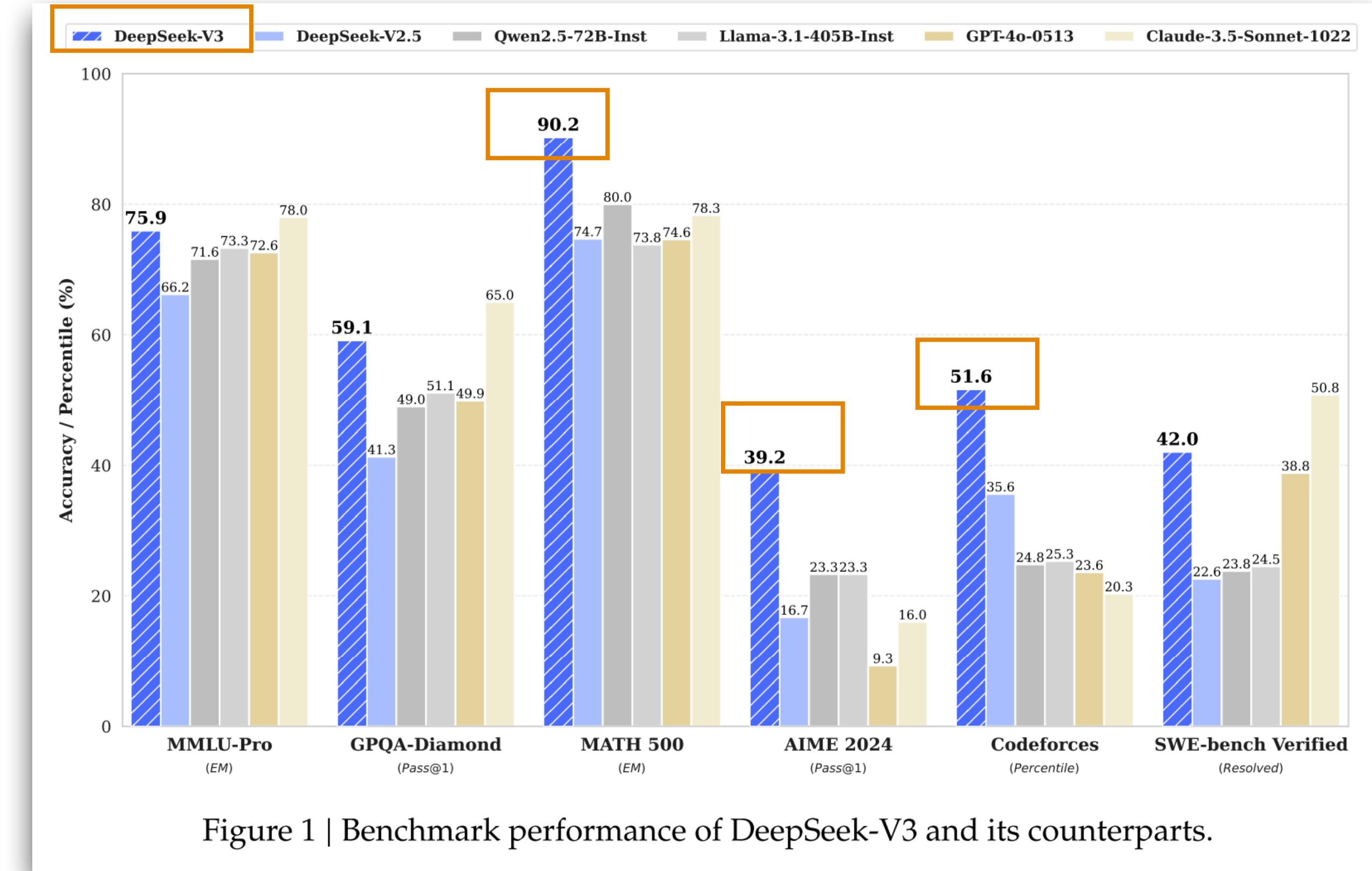


Figure 1 | Benchmark performance of DeepSeek-V3 and its counterparts.

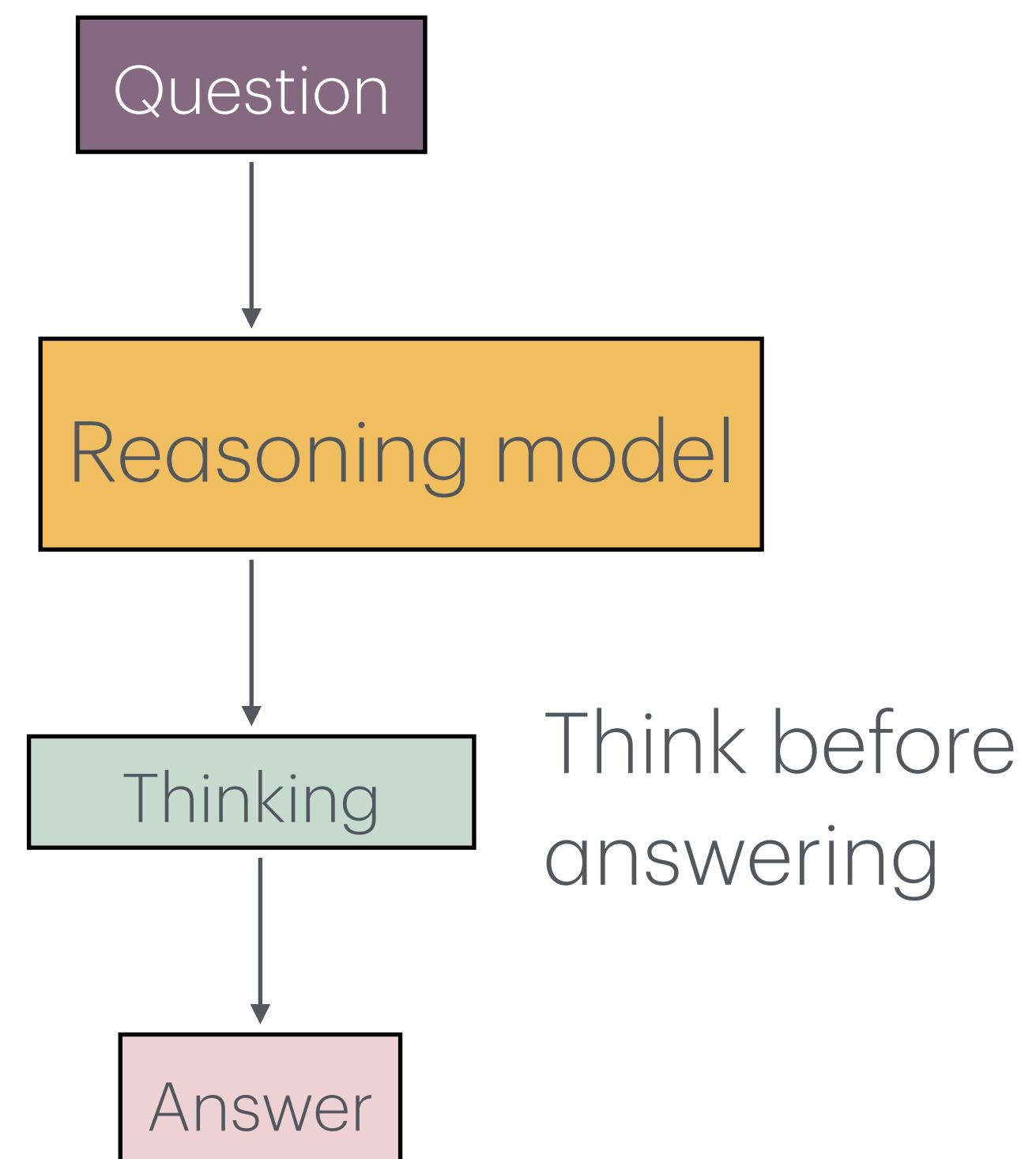
Training Costs	Pre-Training	Context Extension	Post-Training	Total
in H800 GPU Hours	2664K	119K	5K	2788K
in USD	\$5.328M	\$0.238M	\$0.01M	\$5.576M

Table 1 | Training costs of DeepSeek-V3, assuming the rental price of H800 is \$2 per GPU hour.

DeepSeek-R1

Reasoning model (Review)

- Reasoning models: OpenAI o1, o3-mini, Google Gemini 2.0 Flash Thinking, DeepSeek-R1
- Thinking “longer” before answering a given question
- Test time (inference) scale: answer can be better if think more
- More thinking == more tokens == more compute == better output



DeepSeek-R1-Zero

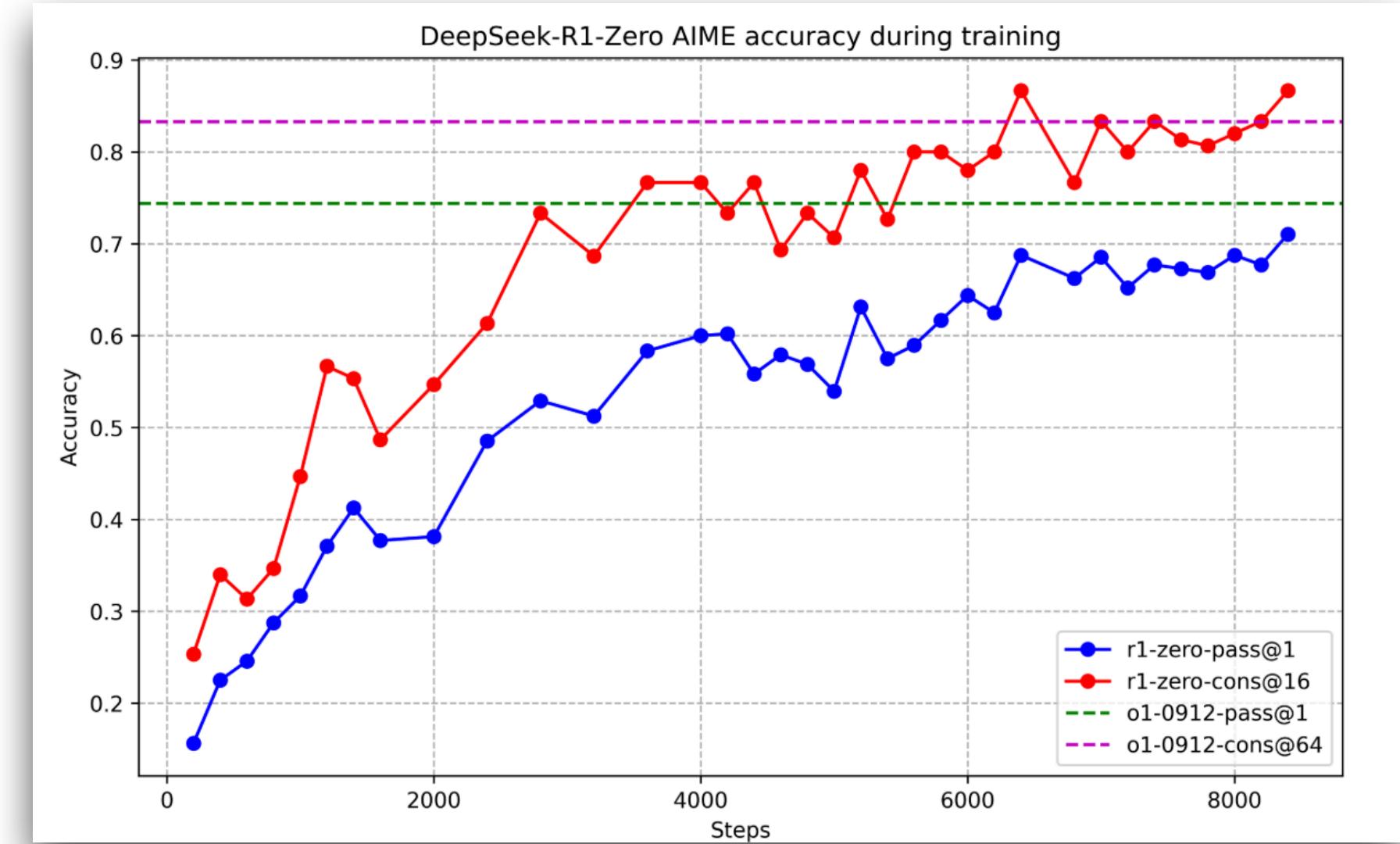
- Directly trained a base model (DeepSeek-V3-Base) with RL only without SFT (supervised fine-tuning)
- After large scale RL training, it shows good performance on reasoning benchmarks
- Group Relative Policy Optimization (GRPO):
 - Similar to PPO but without Critic model and estimate the baseline from group score
 - Some issues in readability, language mixing

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: **prompt**. Assistant:

Table 1 | Template for DeepSeek-R1-Zero. **prompt** will be replaced with the specific reasoning question during training.

DeepSeek-R1-Zero

- Reward model, used rule-based reward model, verifiable reward
- Accuracy rewards: If the final result for math problem is correct. If a code passes the test cases for coding problem
- Format rewards: if the thinking is in between <think> </think>



DeepSeek-R1-Zero

- As test-compute increase, model revisits and reevaluate its own step, and explore alternative approaches. These are not explicitly programmed.
- Aha moment occurred

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a+x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a+x}} = x$, let's start by squaring both ...

$$(\sqrt{a - \sqrt{a+x}})^2 = x^2 \implies a - \sqrt{a+x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

...

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be ...

We started with the equation:

$$\sqrt{a - \sqrt{a+x}} = x$$

First, let's square both sides:

$$a - \sqrt{a+x} = x^2 \implies \sqrt{a+x} = a - x^2$$

Next, I could square both sides again, treating the equation: ...

...

DeepSeek-R1

RL with cold start

1. Cold start

- SFT train to base model with long CoT data
- Base model: DeepSeek-V3-Base
- Long CoT data from DeepSeek-R1-Zero

2. Reasoning oriented RL

- RL with math, coding tasks
- Additional reward for language consistency

3. Rejection sampling and SFT

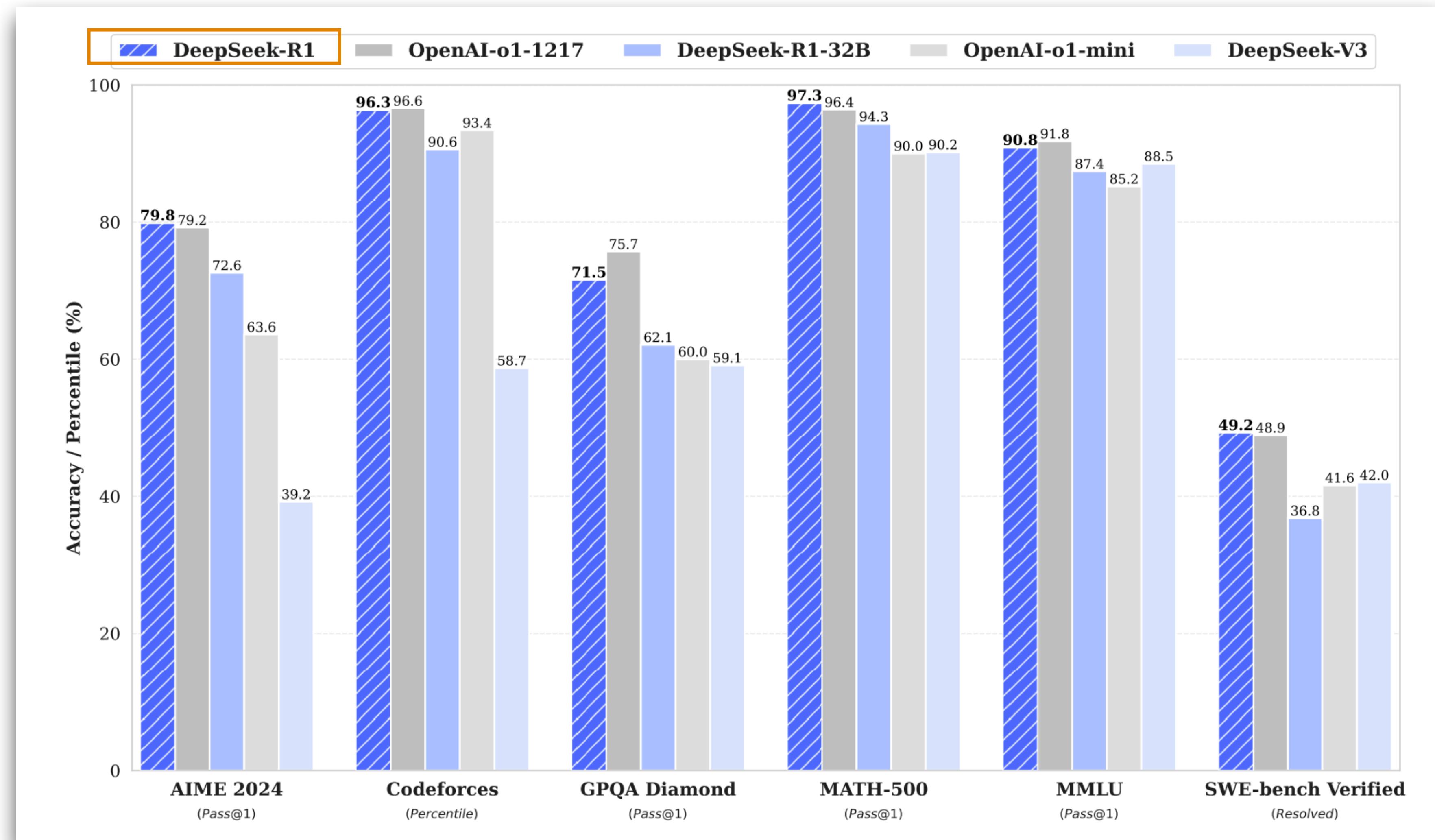
- Reasoning data (600K): Collect data from checkpoint from step 2, and select correct data for SFT
- Non-reasoning data (200k): adapt and reuse dataset of DeepSeek-V3

4. RL for all scenarios

- To improve helpfulness, harmlessness
- Adding human preferences

DeepSeek-R1

Performance



Distilled models

- Directly fine-tuned OSS models (QWen, Llama) using the samples from DeepSeek-R1
- Dataset (800K): from DeepSeek-R1

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	1820
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9	1316
DeepSeek-R1-Distill-Qwen-1.5B	28.9	52.7	83.9	33.8	16.9	954
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481
DeepSeek-R1-Distill-Qwen-32B	72.6	83.3	94.3	62.1	57.2	1691
DeepSeek-R1-Distill-Llama-8B	50.4	80.0	89.1	49.0	39.6	1205
DeepSeek-R1-Distill-Llama-70B	70.0	86.7	94.5	65.2	57.5	1633

Summary

- DeepSeek-R1 is a powerful open-weight reasoning model
- Efficient algorithms for model training helps to reduce training time as well as inference latency
- Detail technical documentation and released code helps community and practitioners

References

1. A Visual Guide to Reasoning LLMs, <https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-reasoning-langs>
2. [MIT 6.5940: TinyML and Efficient Deep Learning Computing](#)