deepseek

# MoE 混合专家
# 深度解读

ZOMI

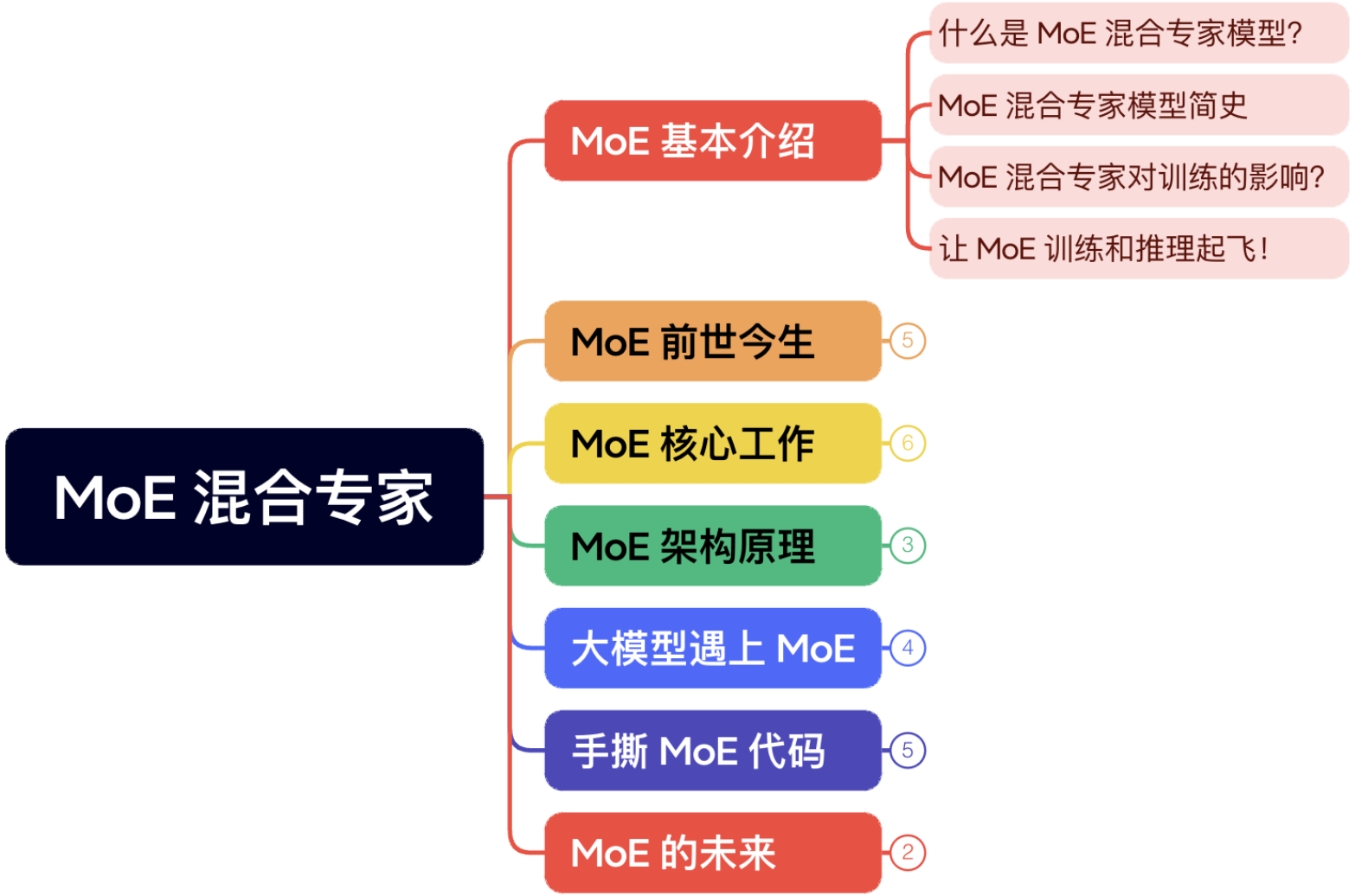
# Question?

1. 为什么幻方 DeepSeek V3 和 R1 模型能够做到这么便宜的 Tokens/pre $?
2. 幻方的 DeepSeek MoE 架构到底有什么主要特性使得算力利用率上去?
3. 幻方的 DeepSeek MoE 架构会不会降低对训练算力和推理算力的需求?

# 视频目录大纲

1. 什么是 MoE 混合专家模型?
2. MoE 混合专家模型简史
3. MoE 混合专家对训练的影响?
4. 让 MoE 训练和推理起飞!
5. 对产业的思考与小结

# 视频目录大纲


MoE 混合专家
MoE 基本介绍
什么是 MoE 混合专家模型?
MoE 混合专家模型简史
MoE 混合专家对训练的影响?
让 MoE 训练和推理起飞!
MoE 前世今生 5
MoE 核心工作 6
MoE 架构原理 3
大模型遇上 MoE 4
手撕 MoE 代码 5
MoE 的未来 2


好好想想

# 视频目录大纲

- MoE 混合专家
  - MoE 基本介绍 4
  - MoE 前世今生
    - 90 年代初期（奠基工作）
    - RNN 时代（架构形成）
    - Transformer 时代（提升效果）
    - GPT 时代（智能涌现）
    - Mixtral 为例 MoE 可视化
  - MoE 核心工作 6
  - MoE 架构原理 3
  - 大模型遇上 MoE 4
  - 手撕 MoE 代码 5
  - MoE 的未来 2

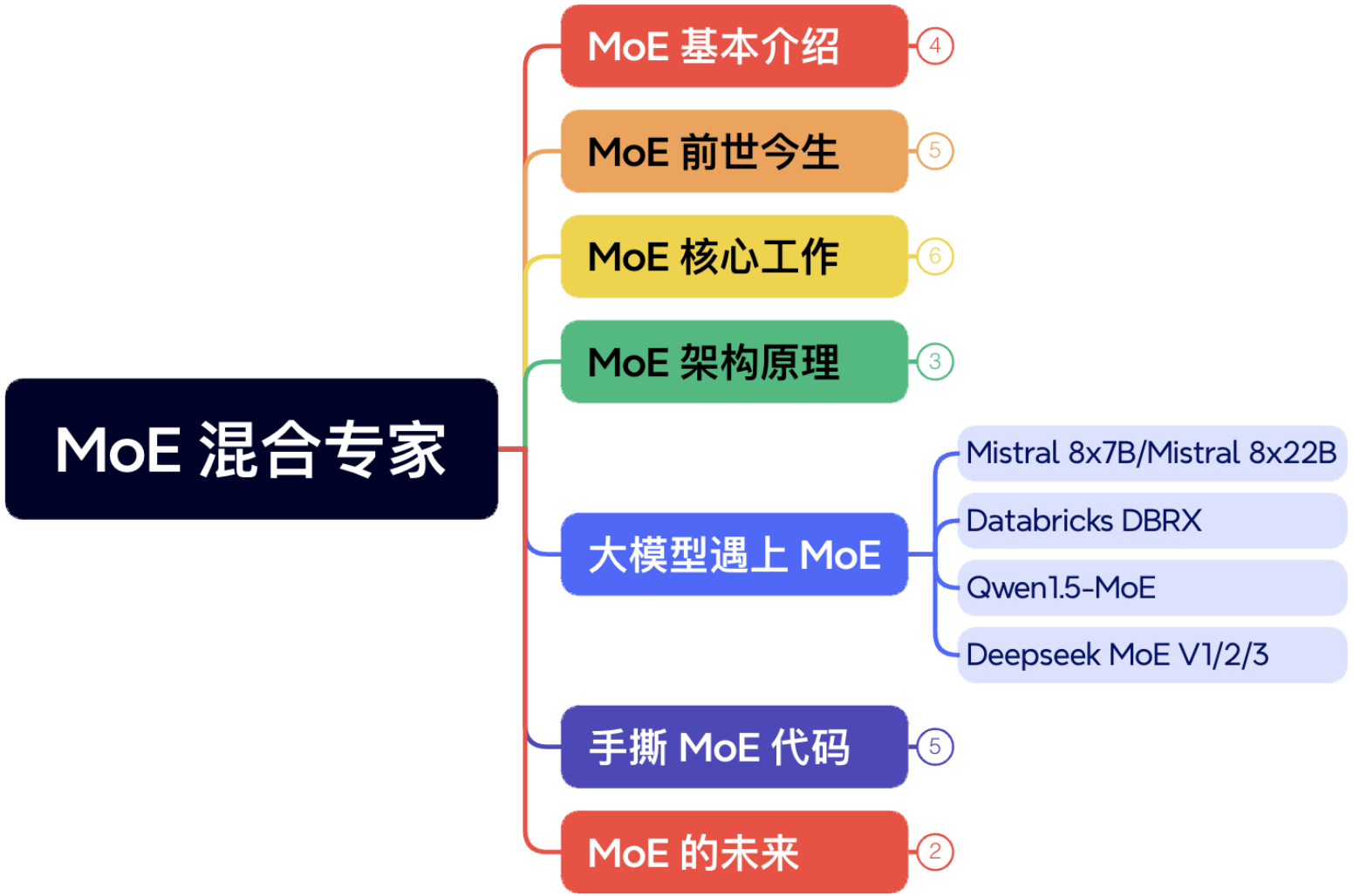
# 视频目录大纲

- MoE 混合专家
  - MoE 基本介绍 4
  - MoE 前世今生 5
  - MoE 核心工作
    - 奠基工作：Adaptive Mixtures of Local Experts
    - LSTM MoE：Outrageously Large Neural Networks ...
    - GShard: Scaling Giant Models with Conditional ...
    - Switch Transformers: Scaling to Trillion Parameter Models ...
    - GLaM: Efficient Scaling of Language Models with Mixture ...
    - ST-MoE: Designing Stable and Transferable Sparse Exp
  - MoE 架构原理 3
  - 大模型遇上 MoE 4
  - 手撕 MoE 代码 5
  - MoE 的未来 2

# 视频目录大纲

- MoE 混合专家
  - MoE 基本介绍 4
  - MoE 前世今生 5
  - MoE 核心工作 6
  - MoE 架构原理
    - 门控网络 Gating Network 到路由 Switch Routing
    - 平衡专家利用率（Balancing Expert Utilization）
    - 高效训练：数据、模型、专家并行
  - 大模型遇上 MoE 4
  - 手撕 MoE 代码 5
  - MoE 的未来 2

# 视频目录大纲


MoE 混合专家
MoE 基本介绍
4
MoE 前世今生
5
MoE 核心工作
6
MoE 架构原理
3
大模型遇上 MoE
Mistral 8x7B/Mistral 8x22B
Databricks DBRX
Qwen1.5-MoE
Deepseek MoE V1/2/3
手撕 MoE 代码
5
MoE 的未来
2


好好想想

# 视频目录大纲

- MoE 混合专家
  - MoE 基本介绍 (4)
  - MoE 前世今生 (5)
  - MoE 核心工作 (6)
  - MoE 架构原理 (3)
  - 大模型遇上 MoE (4)
  - 手撕 MoE 代码
    - 简单版从零实现一个 MOE
    - DeepSpeed 中 MoE 模型（Gshard 架构）
    - MoE 并行（专家并行）
    - 专家并行通信加速（All2All）
    - Megatron MoE 代码解读
  - MoE 的未来 (2)

# 视频目录大纲

- MoE 混合专家
  - MoE 基本介绍 4
  - MoE 前世今生 5
  - MoE 核心工作 6
  - MoE 架构原理 3
  - 大模型遇上 MoE 4
  - 手撕 MoE 代码 5
  - MoE 的未来
    - V-MoE：视觉和多任务 MoE
    - Soft MoE：可微的稀疏混合专家

# 01

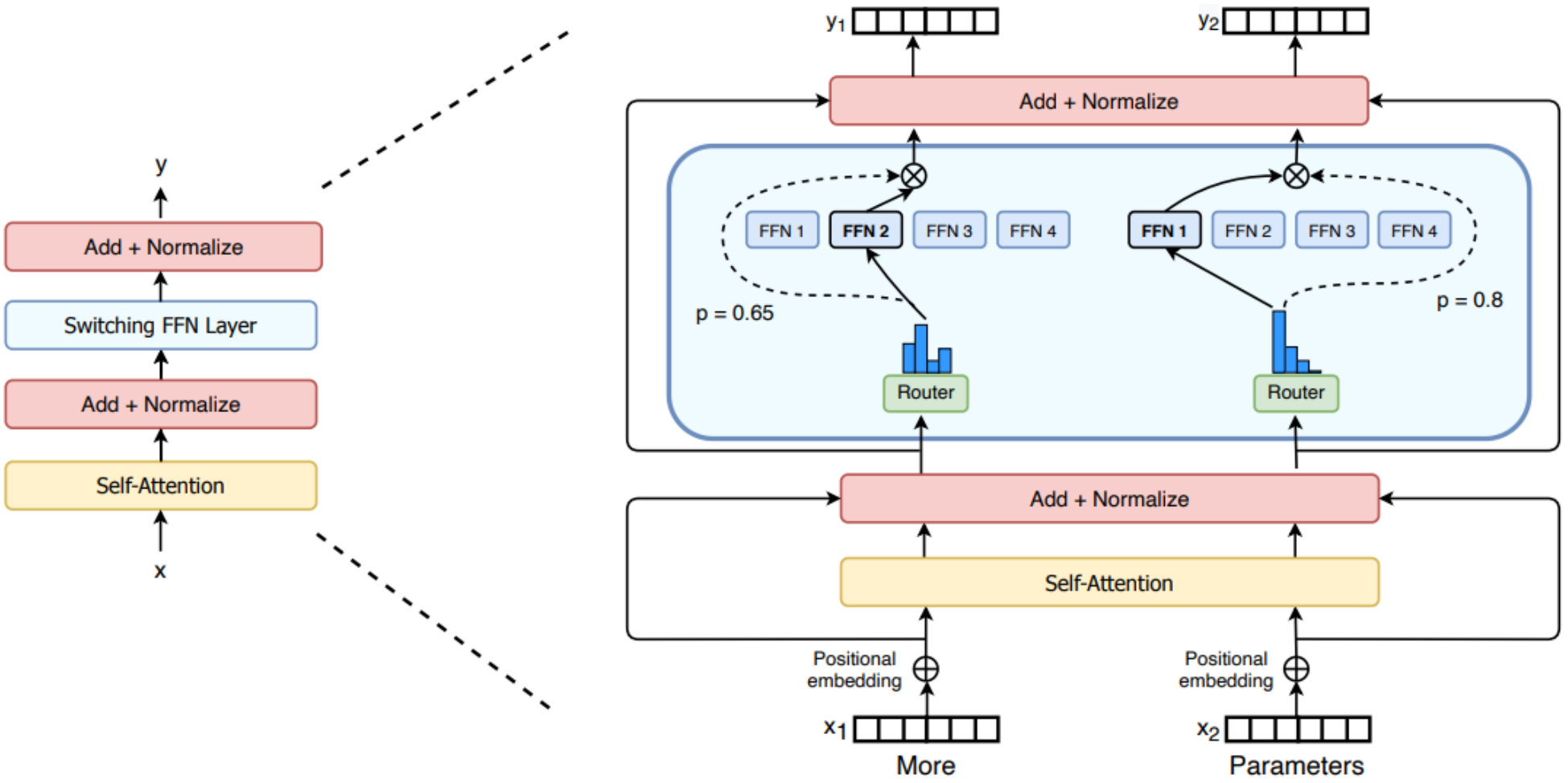
# 什么是 MoE 混合专家模型

# What is a Mixture of Experts?

- MoE 主要由两个关键部分组成:

1. **稀疏 MoE 层：** MoE 层代替传统 Transformer 中 FFN 层。MoE 层包含若干“专家 Expert”，每个专家本身是一个独立的神经网络。
2. **门控网络或路由：** 用于决定哪些 token 发送到哪个专家。例如，More 可能被发送到第二个专家 FFN2，Parameters 被发送到第一个专家 FFN1。有时，一个 token 可以被发送到多个专家 Expert。 token 的路由方式是 MoE 中一个关键点，因为路由器由学习的参数组成，并且与网络的其他部分一同进行预训练。

# What is a Mixture of Experts?



# MoE 的挑战

- MoE 提供更高效的预训练和与稠密模型相比更快的推理速度，但也伴随着一些挑战：
  - **训练挑战：**MoE 能够实现更高效的计算预训练，在微调阶段往往面临泛化能力不足，易于引发过拟合现象，或者预训练难以收敛。
  - **推理挑战：**MoE 模型拥有大参数量，但在推理过程中只激活其中一部分专家参数，这使得推理速度快于具有相同数量参数的稠密模型。然而，MoE 需要将所有参数加载到内存 HBM，因此对内存 HBM 需求高。

# MoE 的挑战

- e,g. Mixtral 8x7B MoE，需要足够 HBM 来容纳 47B 参数。
  - **Why 47B?** 之所以是 47B 而不是 8 x 7B = 56B，MoE 模型中，只有 FFN 层被视为独立专家，模型其他参数共享。
  - **How to share?** 设每个 Token 只使用两个专家，那么推理速度类似使用 12B 模型 （而不是 14B 模型），因为虽然进行 2x7B 矩阵乘法计算，但 MoE 层通过通信来实现参数共享，而非重复计算。

# 02

# MoE 混合专家模型简史

# MoE 简史

- 混合专家模型 MoE 理念起源于 1991 年的论文
- 出自 Geoffrey Hinton 和 Michael I. Jordan 两位大神之手

## Adaptive Mixtures of Local Experts

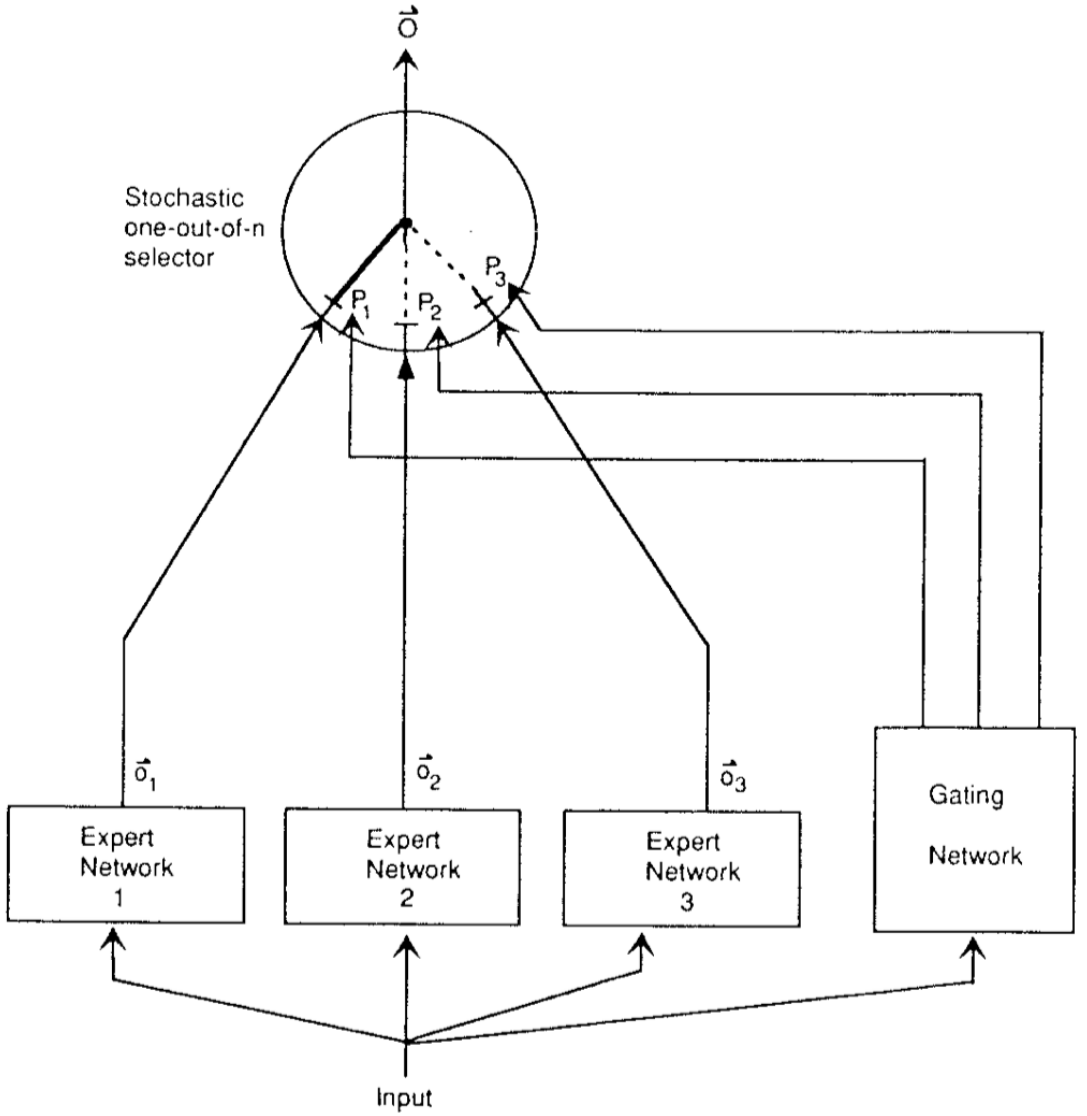
Robert A. Jacobs
Michael I. Jordan
*Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139 USA*

Steven J. Nowlan
Geoffrey E. Hinton
*Department of Computer Science, University of Toronto, Toronto, Canada M5S 1A4*

We present a new supervised learning procedure for systems composed of many separate networks, each of which learns to handle a subset of the complete set of training cases. The new procedure can be viewed either as a modular version of a multilayer supervised network, or as an associative version of competitive learning. It therefore provides a new link between these two apparently different approaches. We demonstrate that the learning procedure divides up a vowel discrimination task into appropriate subtasks, each of which can be solved by a very simple expert network.

# MoE 简史

- 与集成学习方法相似，为多个单独网络组成的系统建立一个监管机制。
- 模型架构中为每个网络（专家）处理训练样本的不同子集，专注于输入空间的特定区域。
- 选择哪个专家来处理特定输入？门控网络决定分配给每个专家的权重。训练过程中，专家和门控网络同时接受训练，以优化的性能和决策能力。



# 历史对 MoE 模型架构重要文献

| 文章名称 | 发布时间 | 主要贡献 |
| --- | --- | --- |
| Adaptive Mixtures of Local Experts | 1991年 | 提出了局部专家的概念，通过门控机制选择不同的专家子模型处理不同输入区域，为MoE架构奠定了基础。 |
| Hierarchical Mixtures of Experts and the EM Algorithm | 1994年 | 提出了混合专家模型的基本框架，结合了概率模型和神经网络的思想，使用期望最大化（EM）算法进行训练。这是MoE架构的奠基性工作 |
| Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer | 2017年 | 提出了稀疏门控混合专家层，实现了大规模模型的高效推理。 |
| GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding | 2020年 | 首次将MOE技术引入Transformer架构，提供了高效的分布式并行计算架构。 |
| GLaM: Efficient Scaling of Language Models with Mixture-of-Experts | 2021年 | 利用MoE架构在语言模型中实现了高效的扩展，展示了其在自然语言处理任务中的潜力。 |
| Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity | 2021年 | 提出了稀疏激活的MOE架构，显著提升了模型的预训练速度和推理效率。 |
| PaLM: Scaling Language Modeling with Pathways | 2022年 | 提出了Pathways架构下的PaLM模型，结合了MoE技术，实现了高效的大规模语言模型训练。 |
| Llama-MoE: Scaling Mixture-of-Experts Models for Open Pretraining | 2023年 | 将MoE架构引入开源的Llama系列模型中，展示了MoE在开放预训练中的可行性 |
| DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models | 2024年 | 引入了“细粒度/垂类专家”和“共享专家”的概念，提升了专家的专业性和模型效率。 |

# 近期发布的 MoE 大模型

| 模型 | 发布时间 | 备注 |
| --- | --- | --- |
| GPT4 | 2023年3月 | 23年6月George Hotz爆料GPT4是8×220B模型 |
| Mistral-8×7B | 2023年12月 | Mistral AI，开源 |
| LLAMA-MoE | 2023年12月 | Mate，开源 |
| DeepSeek-MoE | 2024年1月 | 幻方量化(深度求索)，国内首个开源 MoE 模型，有技术报告 |
| Step-2 | 2024年3月 | 阶跃星辰，无开源，无细节发布 |
| MM1 | 2024年3月 | 苹果，多模态MoE，无开源，有技术报告 |
| Grok-1 | 2024年3月 | XAI，开源 |
| Qwen1.5-MoE-A2.7B | 2024年3月 | 阿里巴巴，开源 |
| DBRX | 2024年3月 | Databricks，开源 |
| Mistral-8×22B | 2024年4月 | Mistral AI，开源 |
| WizardLM-2-8×22B | 2024年4月 | 微软，开源 |
| Arctic | 2024年4月 | Snowflake，480B，Dense-MoE Hybrid，开源 |
| Grok-2 | 2024年8月 | XAI，开源 |
| DeepSeek-V3 | 2025 年 1 月 | 幻方量化(深度求索)，国内首个开源 MoE 模型，有技术报告 |
| MiniMax-01 | 2025 年 1 月 | MiniMax 发布的MoE架构大模型，参数规模达4560亿，支持长达400万tokens的输入 |
| Qwen2.5-Max | 2025 年 1 月 | 采用超大规模MOE架构，预训练数据量超过20万亿tokens，支持高达100万token的上下文窗口 |

# 03

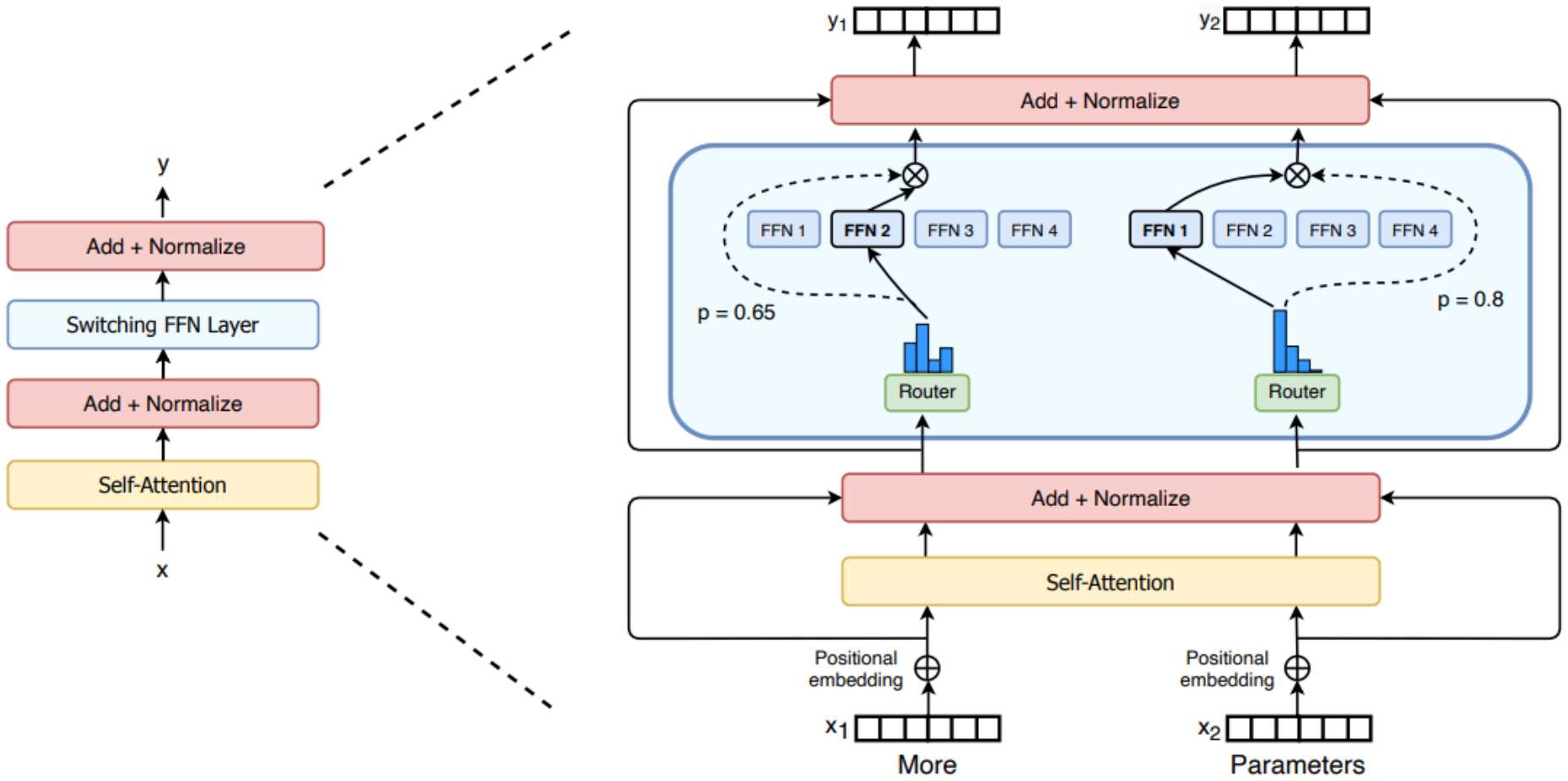
# MoE 混合专家对训练的影响?

# 什么是稀疏性？

- 稠密大模型，模型所有参数 w 都会对所有输入数据 x 进行处理计算。
- 稀疏性允许针对模型某些特定部分（Expert）执行计算。
- MoE 架构非所有参数都会在处理每个输入时被激活或使用，根据输入特征，选择部分参数计算。

# 什么是稀疏性？

- 批量大小的不均匀分配和资源利用效率不高：

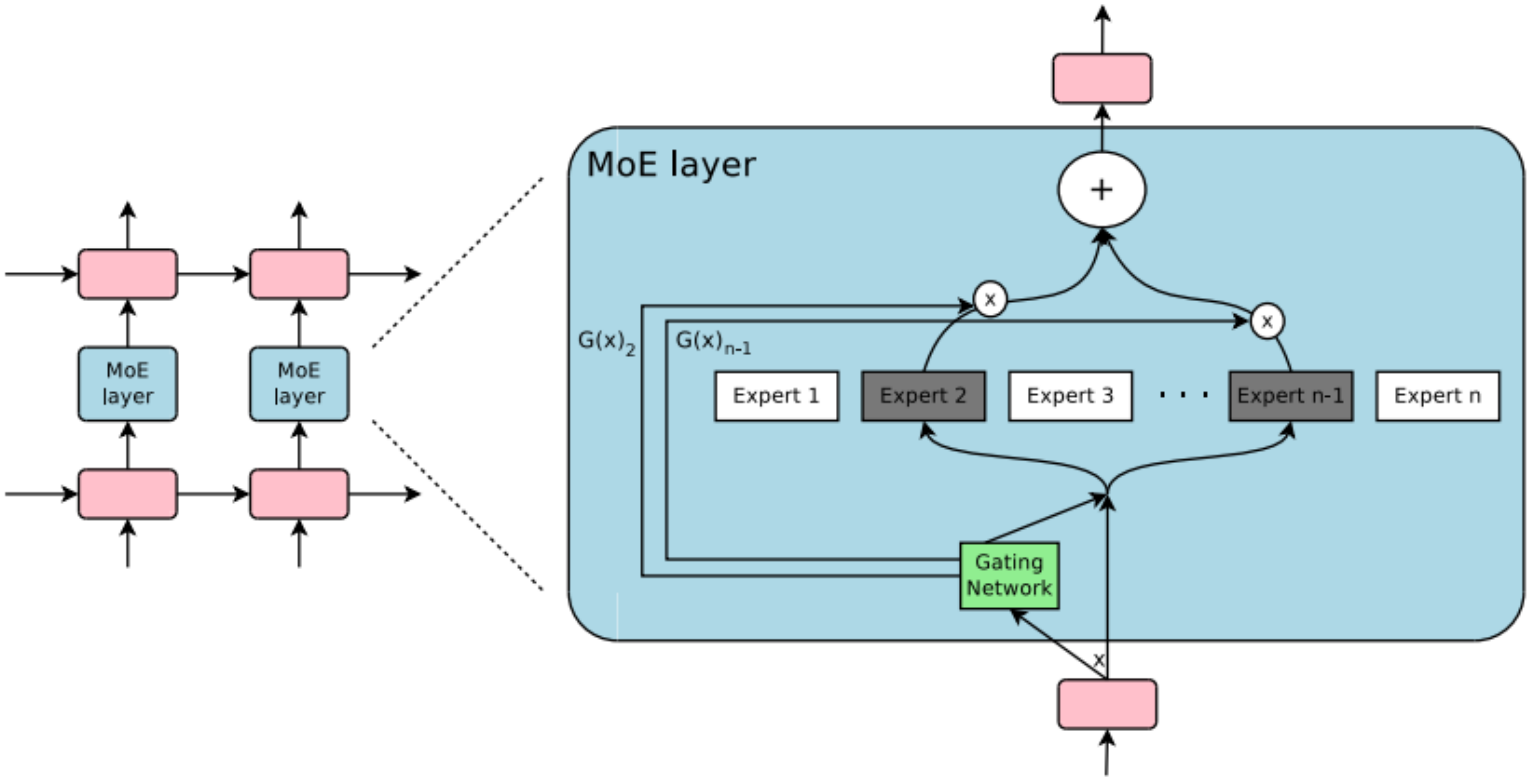


# 什么是稀疏性？

可学习的门控网络 + 专家间负载均衡

# 解决稀疏性计算

- Google Shazeer 对 MoE 在翻译应用中，引入条件计算，在每个样本的基础上激活网络的不同部分。
- 不增加额外计算情况下扩展 MoE 规模，每个 MoE 层中实现更多 Expert，提升专家利用率。



# Token 负载均衡

- 门控网络往往倾向于主要激活相同的几个专家。受欢迎的专家训练得更快，因此更容易被选择。
- 引入了一个辅助损失 Aux Loss，鼓励所有专家相同的重要性，平衡计算量。
- Aux Loss 确保所有专家接收到大致相等数量的训练样本，从而平衡专家间选择。

# 专家如何学习?

- ST-MoE 重表示 Encoder 不同专家倾向于专注于特定类型的 Token 或浅层概念。
- 对 MOE 进行多语言训练中，预期每个专家处理一种特定语言。
- But，由于 Token 路由和负载均衡，没有任何专家被特定用于处理特定语言。

| Expert specialization | Expert position | Routed tokens |
|---|---|---|
| **Sentinel tokens** | Layer 1 | been <extra_id_4><extra_id_7>floral to<br><extra_id_10><extra_id_12><extra_id_15><br><extra_id_17><extra_id_18><extra_id_19>... |
| | Layer 4 | <extra_id_0><extra_id_1><extra_id_2><br><extra_id_4><extra_id_6><extra_id_7><br><extra_id_12><extra_id_13><extra_id_14>... |
| | Layer 6 | <extra_id_0><extra_id_4><extra_id_5><br><extra_id_6><extra_id_7><extra_id_14><br><extra_id_16><extra_id_17><extra_id_18>... |
| **Punctuation** | Layer 2 | , , , , , , , , , - , , , , , ). ) |
| | Layer 6 | , , , , , : . : , & , & & ? & - , , ? , , , . <extra_id_27> |
| **Conjunctions and articles** | Layer 3 | The the the the the the the the the The the the<br>the the the The the the the |
| | Layer 6 | a and and and and and and and or and a and .<br>the the if ? a designed does been is not |
| **Verbs** | Layer 1 | died falling identified fell closed left posted lost felt<br>left said read miss place struggling falling signed died<br>falling designed based disagree submitted develop |
| **Visual descriptions**<br>*color, spatial position* | Layer 0 | her over her know dark upper dark outer<br>center upper blue inner yellow raw mama<br>bright bright over open your dark blue |
| **Proper names** | Layer 1 | A Mart Gr Mart Kent Med Cor Tri Ca Mart<br>R Mart Lorraine Colin Ken Sam Ken Gr Angel A<br>Dou Now Ga GT Q Ga C Ko C Ko Ga G |
| **Counting and numbers**<br>*written and numerical forms* | Layer 1 | after 37 19. 6. 27 I I Seven 25 4, 54 I two dead we<br>Some 2012 who we few lower each |

# 专家的数量对预训练有何影响?

- 增加专家可以提升处理样本效率和加速模型运算速度。
- 随着专家数量增加而递减，当专家数量达到 256 或 512 后效果递减。
- 增加专家，在推理过程中需要更多显存来加载整个 MoE 模型。

# 稀疏 VS 稠密，如何选择?

1. 在固定的预训练计算资源下，稀疏模型往往能够实现更优的效果。
2. 在显存较少且吞吐量要求不高的场景，稠密模型则是更合适的选择。

- 直接比较稀疏模型和稠密模型的参数数量不恰当，这两类模型概念和参数量计算方法完全不同。

# 04

# 让 MoE 训练和推理起飞！

# 让 MoE 起飞

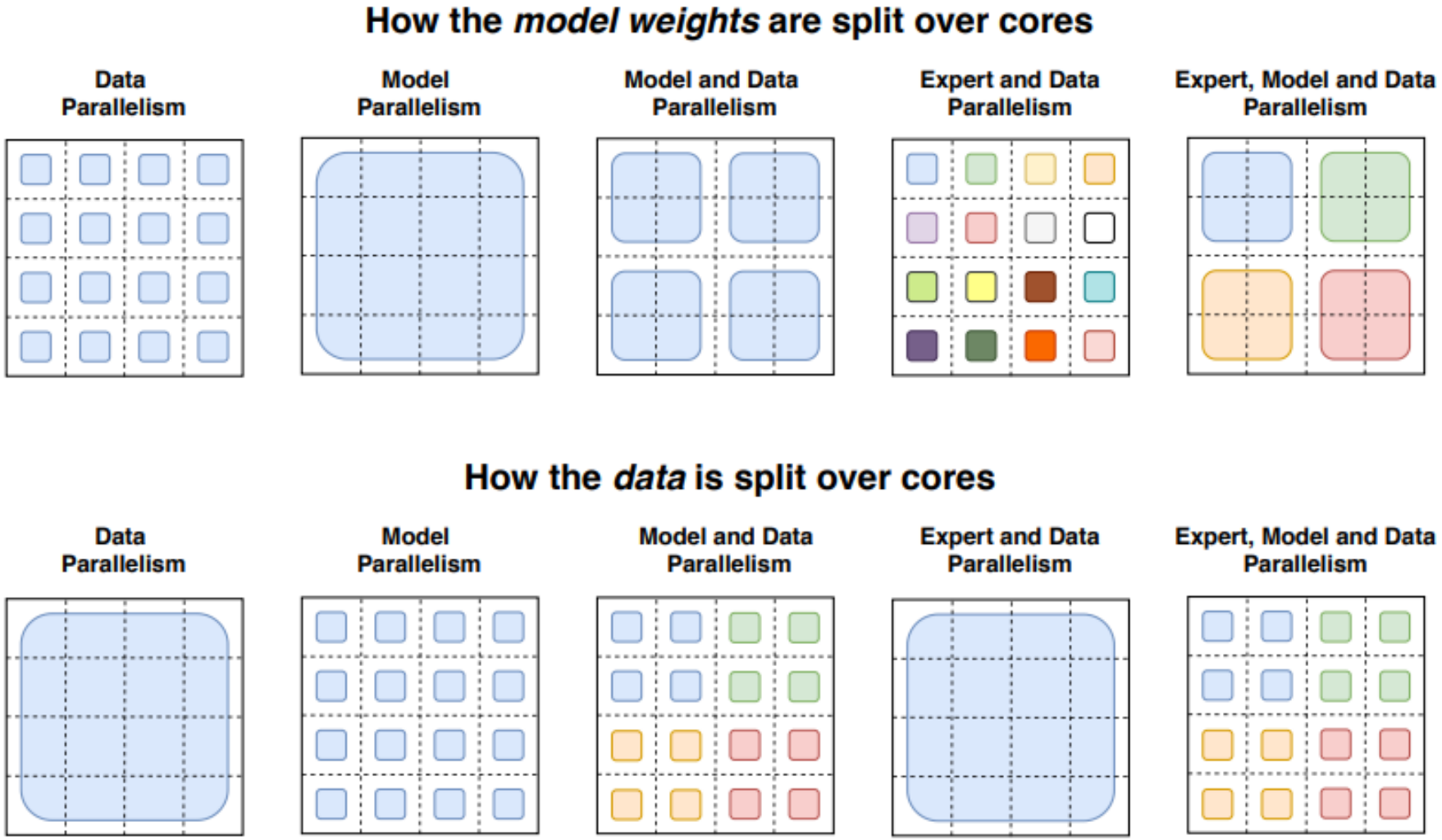
- 早期 MoE （Before 2017 ）采用了分支结构，导致计算效率低下。
- GPU 不是为处理分支结构而设计 + Device 间需要传递数据，网络带宽成为性能瓶颈。

- 专家并行、通信优化、蒸馏可以让 MOE 在预训练和推理阶段更加高效和实用。接下来优化 MoE 模型，让 MoE 起飞!

# 并行计算

- **数据并行：**
  - 相同的权重在所有 NPU 上复制，数据在 NPU 间分割。
- **模型并行：**
  - 模型在 NPU 之间分割，相同的数据在所有 NPU 上复制。
- **模型和数据并行：**
  - NPU 间同时分割模型和数据，不同 NPU 处理不同批次的数据。
- **专家并行：**
  - 专家放置在不同 NPU，与数据并行结合，每个 NPU 不同专家，数据在所有 NPU 间分割。

# 并行计算


How the *model weights* are split over cores
Data Parallelism
Model Parallelism
Model and Data Parallelism
Expert and Data Parallelism
Expert, Model and Data Parallelism
How the *data* is split over cores
Data Parallelism
Model Parallelism
Model and Data Parallelism
Expert and Data Parallelism
Expert, Model and Data Parallelism

# 专家并行

- 专家放置不同 NPU，每个 NPU 处理不同 batch sample；
- 非 MoE 层，专家并行与数据并行相同；
- MoE 层，序列中的 Token 被发送到拥有所需专家 NPU。

# 容量因子和显存带宽

- 提高容量因子（Capacity Factor, CF）可以增强 MOE 性能，带来更高通信成本和对保存激活值的显存需求。
- 设备通信带宽有限，可以选择较小容量因子。评估性能时，根据需要调整容量因子，在 NPU 间通信成本和计算成本之间找到一个平衡点。

# 模型蒸馏

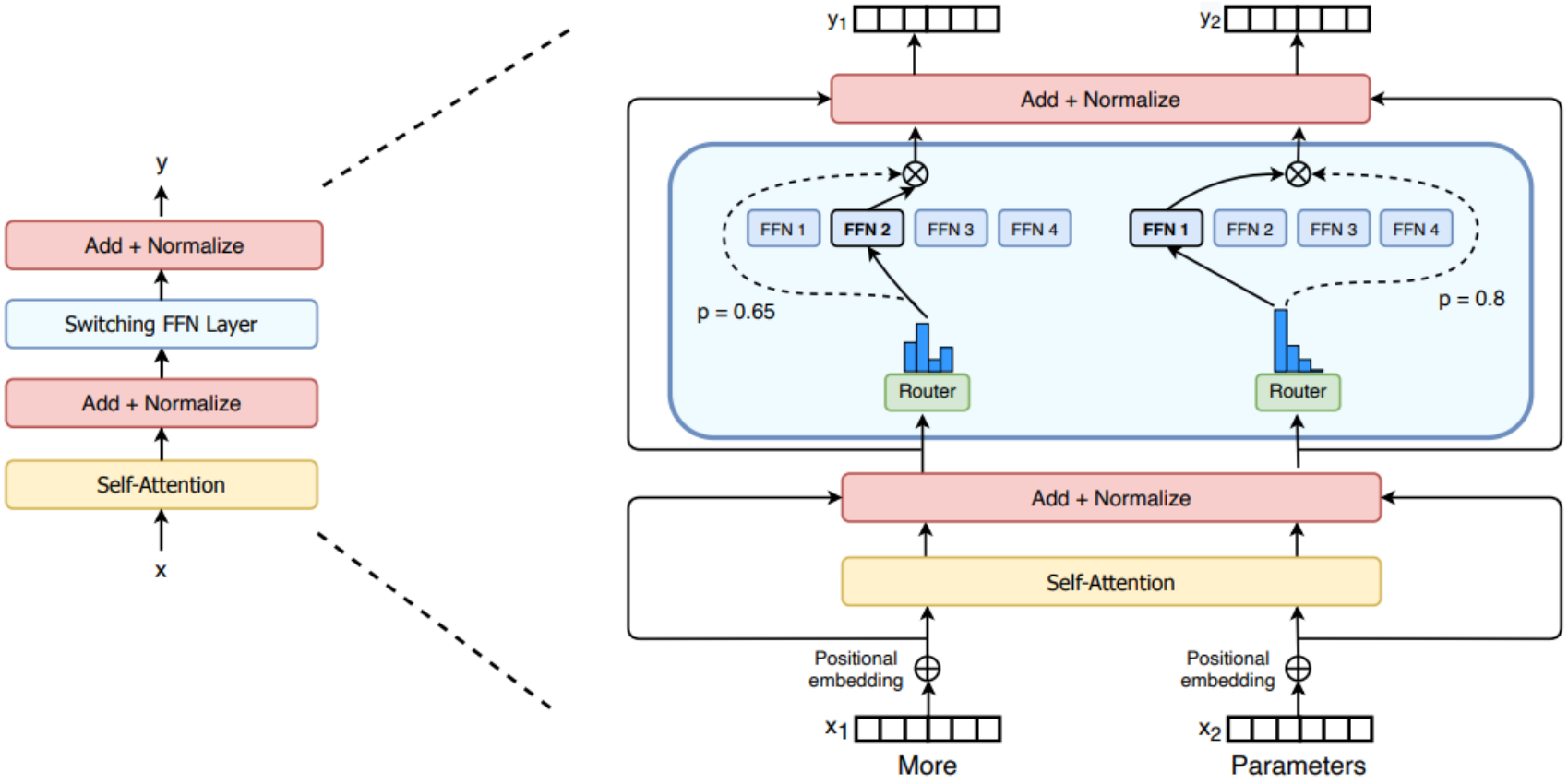
- 蒸馏实验：通过将 MoE 模型蒸馏回其对应稠密模型，保留 30-40% 由稀疏性带来性能提升。预先蒸馏不仅加快预训练速度，还使得在推理中使用更小型模型。

| 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 |

Table 5 | Comparison of DeepSeek-R1 distilled models and other comparable models on reasoning-related benchmarks.

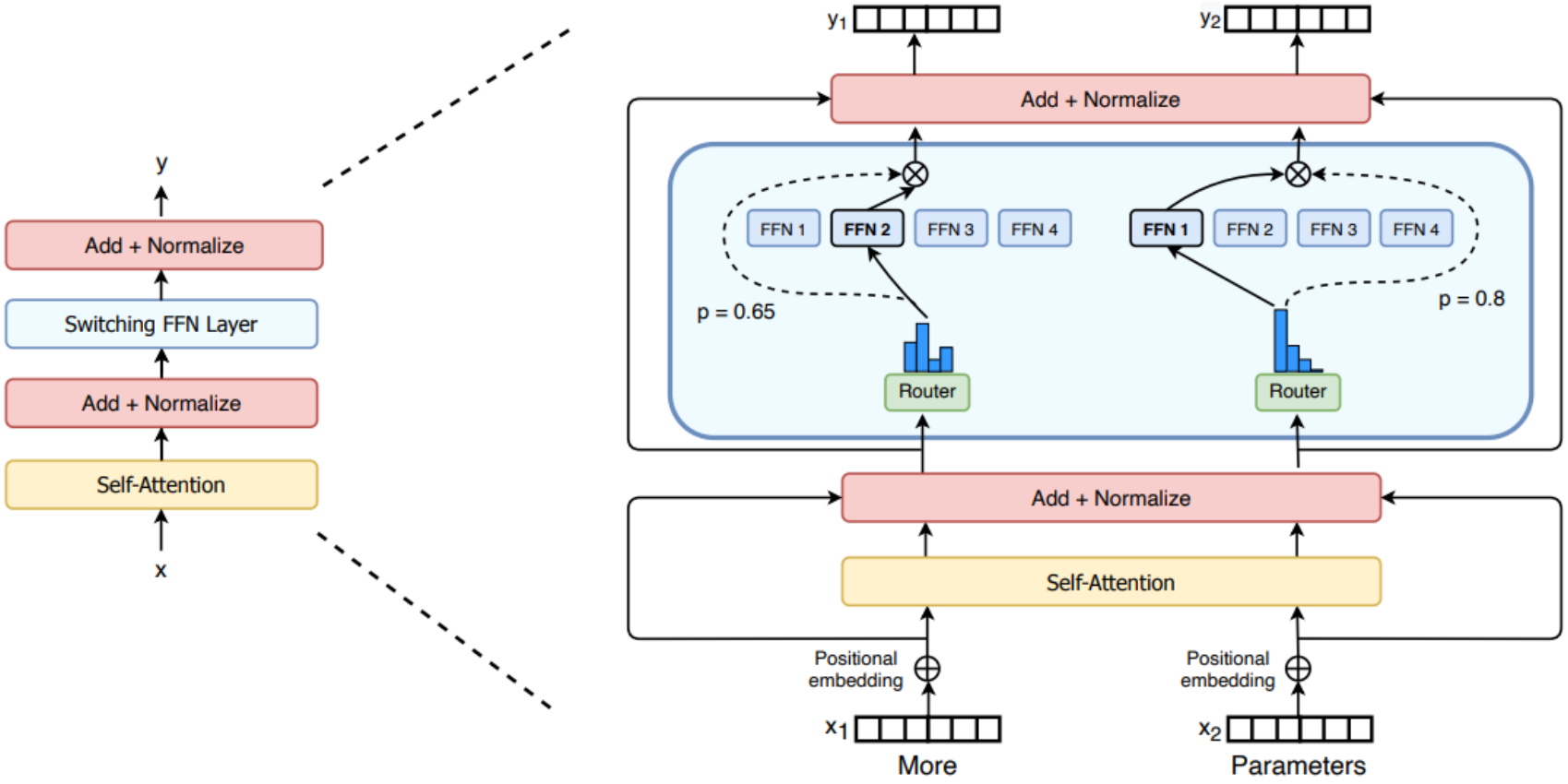
# 任务级别路由

- **任务级别路由：**路由将整个句子或任务直接路由到一个确定性专家。提取出一个用于服务的子模型，有助于简化模型结构。



# 专家网络聚合

- Expert 聚合：合并各个专家权重，推理时减少所需参数数量。在不显著牺牲性能的情况下降低模型稀疏复杂度。

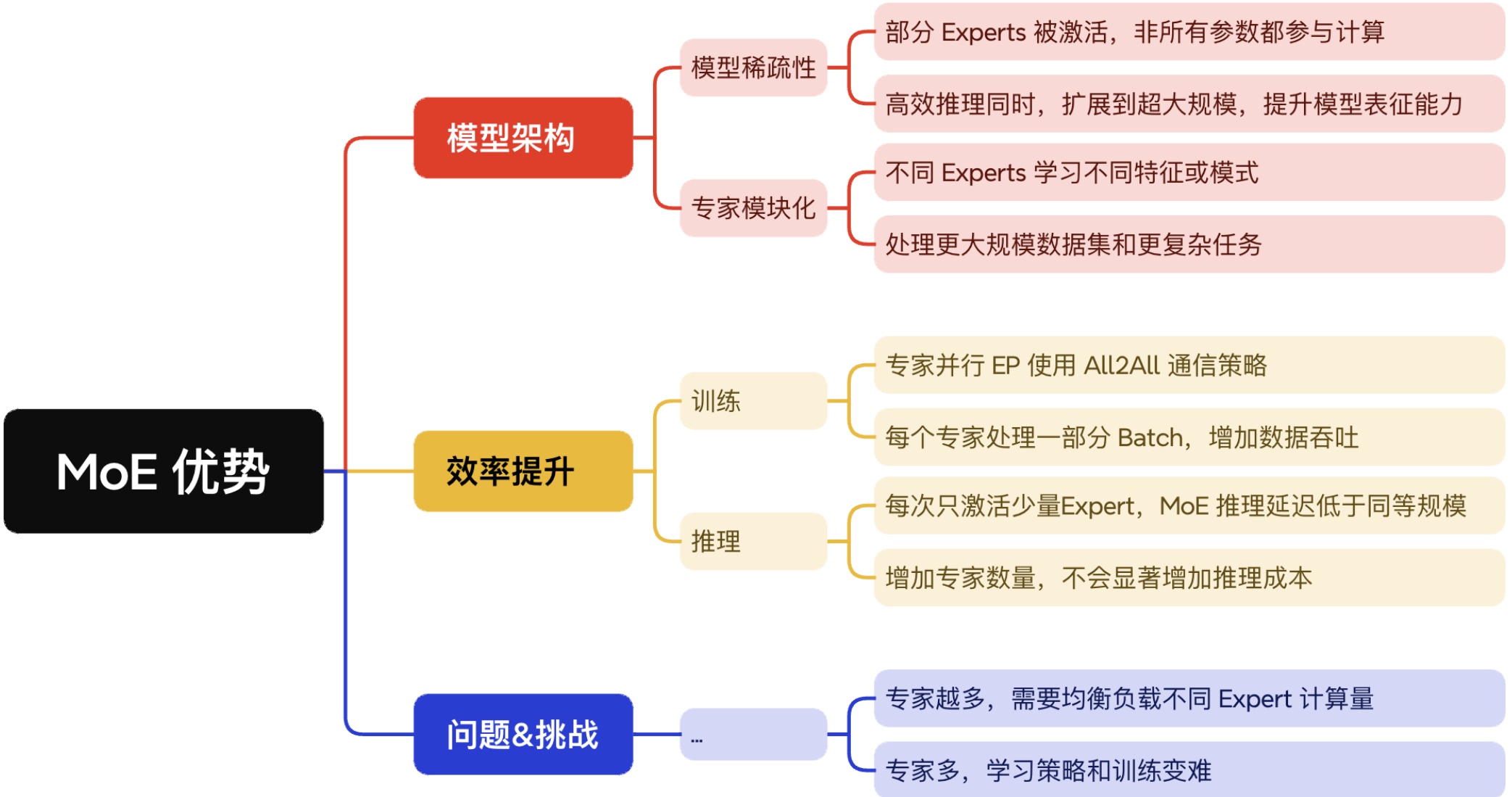


# 思考与小结

# 总结与思考

1. 为什么幻方 DeepSeek V3 和 R1 模型能够做到这么便宜的 Tokens/pre $?
2. 幻方的 DeepSeek MoE 架构到底有什么主要特性使得算力利用率上去?
3. 幻方的 DeepSeek MoE 架构会不会降低对训练算力和推理算力的需求?

- 相同参数下，MOE 架构的天然优势，推理时候只执行部分参数


MoE 优势
模型架构
模型稀疏性
部分 Experts 被激活，非所有参数都参与计算
高效推理同时，扩展到超大规模，提升模型表征能力
专家模块化
不同 Experts 学习不同特征或模式
处理更大规模数据集和更复杂任务
效率提升
训练
专家并行 EP 使用 All2All 通信策略
每个专家处理一部分 Batch，增加数据吞吐
推理
每次只激活少量Expert，MoE 推理延迟低于同等规模
增加专家数量，不会显著增加推理成本
问题&挑战
...
专家越多，需要均衡负载不同 Expert 计算量
专家多，学习策略和训练变难

# Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

**Copyright © 2024 XXX Technologies Co., Ltd.**
**All Rights Reserved.**

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. XXX may change the information at any time without notice.

**ZOMI**

GitHub https://github.com/chenzomi12/AIFoundation

# 引用与参考


- https://mp.weixin.qq.com/s/6kzCMsJuavkZPG0YCKgeig
- https://www.zhihu.com/tardis/zm/art/677638939?source_id=1003
- https://huggingface.co/blog/zh/moe
- https://mp.weixin.qq.com/s/mOrAYo3qEACjSwcRPG7fWw
- https://mp.weixin.qq.com/s/x39hqf8xn1cUlnxEIM0_ww
- https://mp.weixin.qq.com/s/ZXjwnO103e-wXJGmmKi-Pw
- https://mp.weixin.qq.com/s/8Y281VYaLu5jHoAvQVvVJg
- https://blog.csdn.net/weixin_43013480/article/details/139301000
- https://developer.nvidia.com/zh-cn/blog/applying-mixture-of-experts-in-llm-architectures/
- https://www.zair.top/post/mixture-of-experts/
- https://my.oschina.net/IDP/blog/16513157


- PPT 开源在：
- https://github.com/chenzomi12/AIInfra