



2024 AI+研发数字峰会

AI+ Development Digital summit

AI驱动研发变革 促进企业降本增效

北京站 08/16-17

SUBLLM新架构： 文本下采样机制革新大语言模型效率

王全东 小米大模型团队

科技生态圈峰会 + 深度研习



—1000+ 技术团队的选择



上海站

K+全球软件研发行业创新峰会

时间: 2024.06.21-22



敦煌站

K+思考周®研习社

时间: 2024.10.17-19



香港站

K+思考周®研习社

时间: 2024.11.10-12



K+峰会详情



上海站

Ai+研发数字峰会

时间: 2024.05.17-18



北京站

Ai+研发数字峰会

时间: 2024.08.16-17



深圳站

Ai+研发数字峰会

时间: 2024.11.08-09



AiDD峰会详情



2024 AI+研发数字峰会

AI+ Development Digital summit

深圳站 11/08-09

AI 驱动研发变革 促进企业降本增效

2024深圳站-议题设置

| | | | |
|--------|--------------------------------|------------------|-------------|
| AI+产品线 | LLM驱动产品创新 | LLM驱动需求与业务分析 | AI驱动设计与用户体验 |
| AI+开发线 | AI 原生应用开发框架与技术 | AI Agents在研发落地实践 | LLM驱动编程与单测 |
| AI+测试线 | LLM驱动测试分析与设计 | 基于LLM生成测试脚本与数据 | LLM和AI应用的评测 |
| AI+工程线 | AI+DevOps 与工具 (LLM 时代的平台工程) | 大模型对齐与安全 | 端侧大模型与云端协同 |
| AI+领域线 | 领域大模型 SFT 与优化 | 知识增强与数据智能 | 大厂专场 |

扫描右侧二维码
查看更多会议详情



早鸟票限时抢购中 (截止到9月30日)

¥3680

早鸟票

¥2800

学生票



王全东

小米大模型团队 大模型高级算法工程师

中国科学院声学研究所博士、美国佐治亚理工访问学者、中科院认证高级工程师，长期从事大语言模型、多模态、语音识别等领域研究，曾获多项顶会竞赛冠亚军奖项，已发表顶会论文十余篇，拥有专利多项。深度参与了小米自研大模型从0到1的研发过程，荣获2024年度CCF计算机应用创新技术一等奖。近期和Daniel Povey等提出SUBLLM新架构，被量子位等科技媒体报道。

目录

CONTENTS

1. 长文本模型的技术挑战
2. SUBLLM架构
3. 主要实验结果
4. 分析与讨论
5. 总结与展望

PART 01

长文本模型的技术挑战

▶▶ 长文本模型的技术挑战

长文本需求旺盛

多人会议摘要

行业报告
新闻摘要

学术论文分析

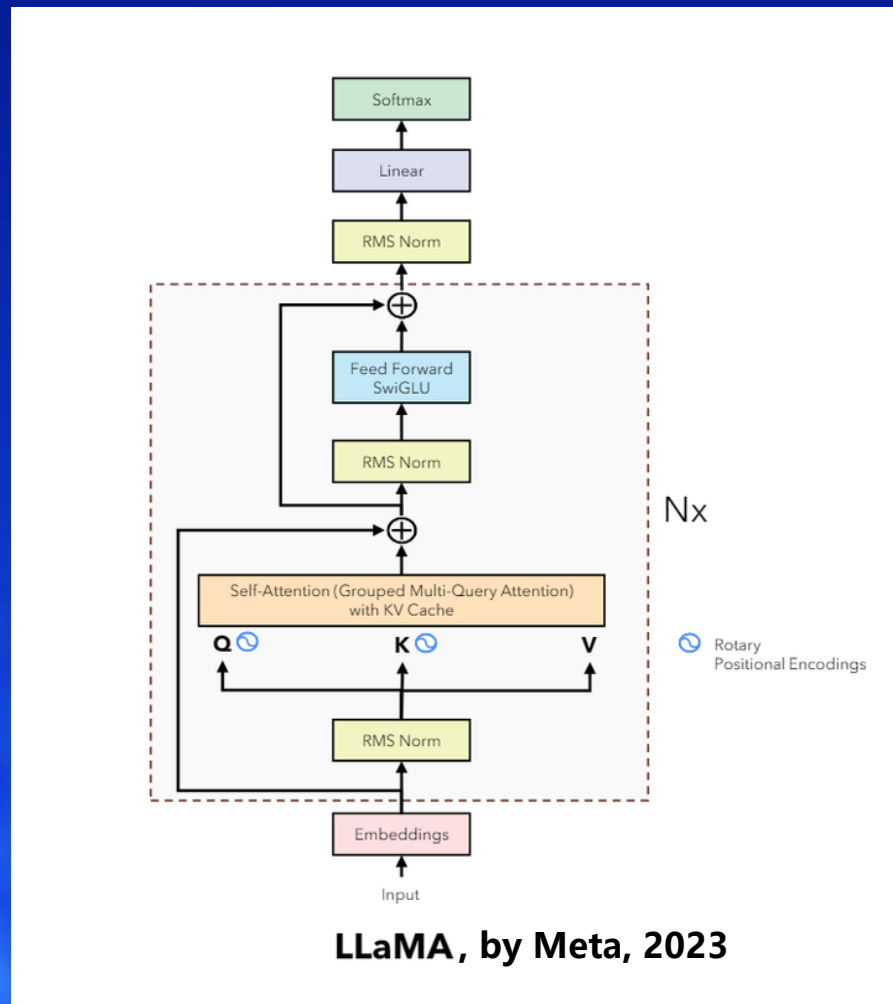
长文写作
长篇翻译

...

长文本模型的技术挑战

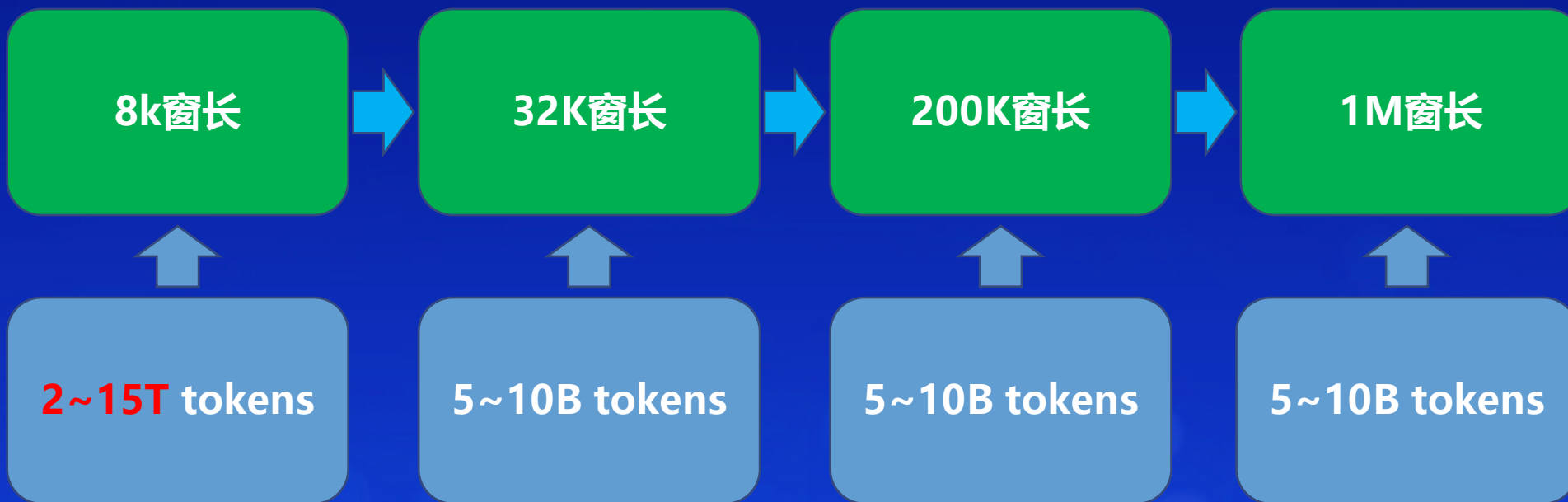
长文本模型结构：

Decoder only Transformer结构：Llama 类似结构，attention的平方复杂度



▶▶ 长文本模型的技术挑战

训练成本高：attention的平方复杂度

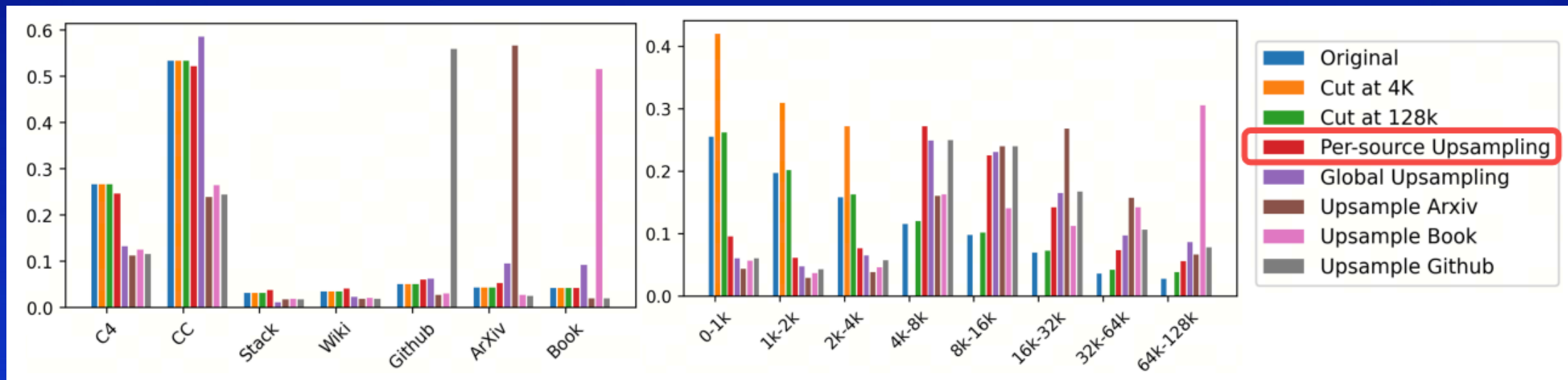


主要优化点

▶▶ 长文本模型的技术挑战

模型窗长扩展方法：数据方向，训练成本不高

1. Data Engineering for Scaling Language Models to 128K Context

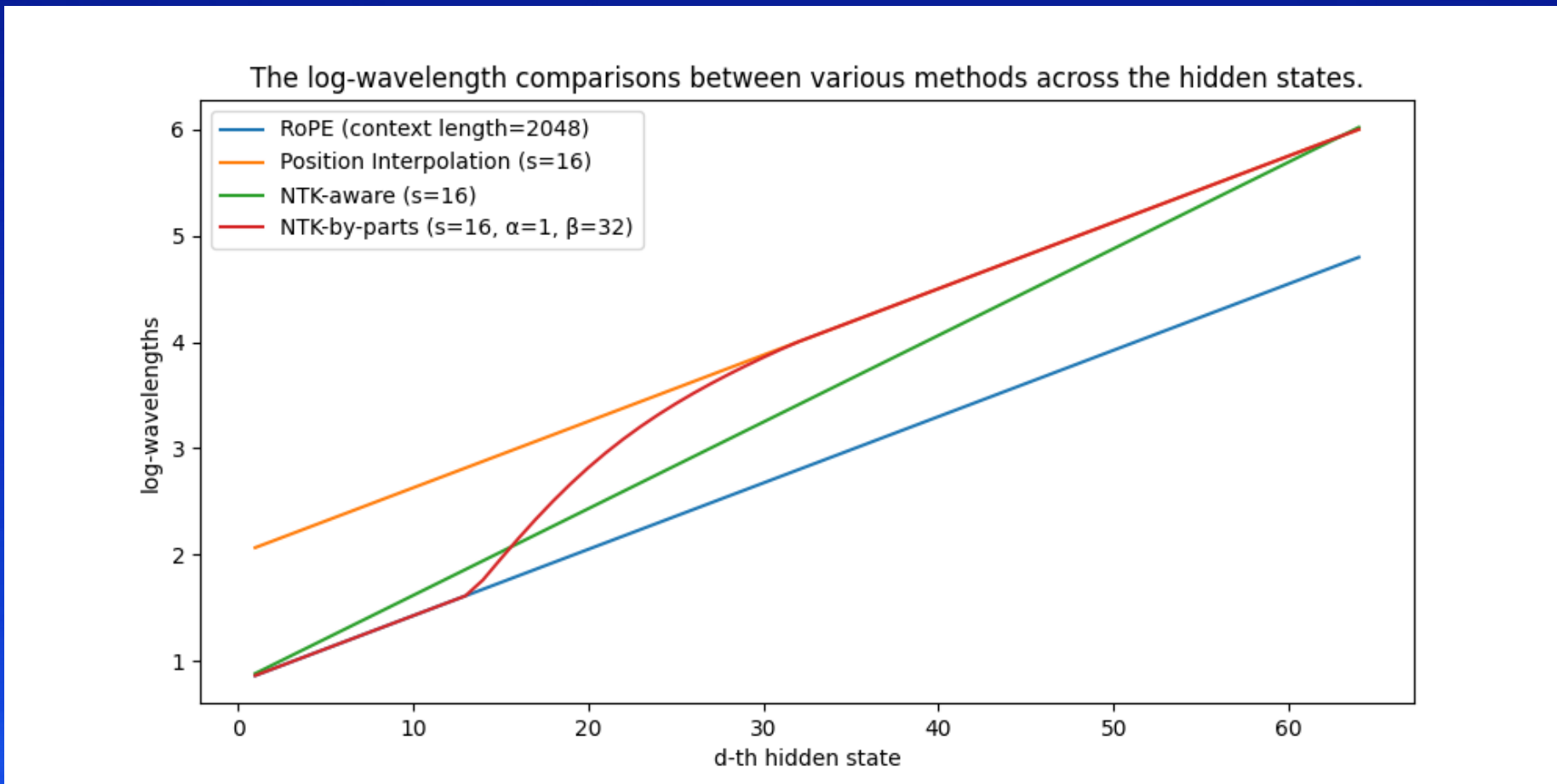


▶ 长文本模型的技术挑战

模型窗长扩展方法：位置编码方向，训练成本不高

2. YaRN: Efficient Context Window Extension of Large Language Models

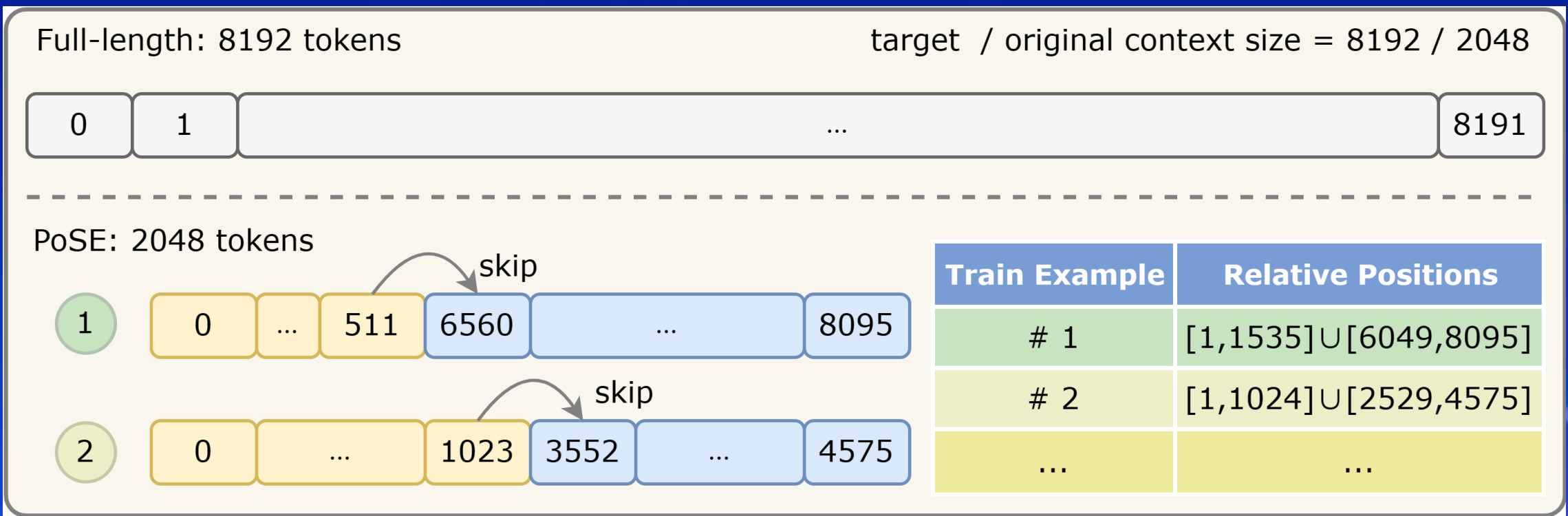
NTK-by-parts + 温度控制



▶ 长文本模型的技术挑战

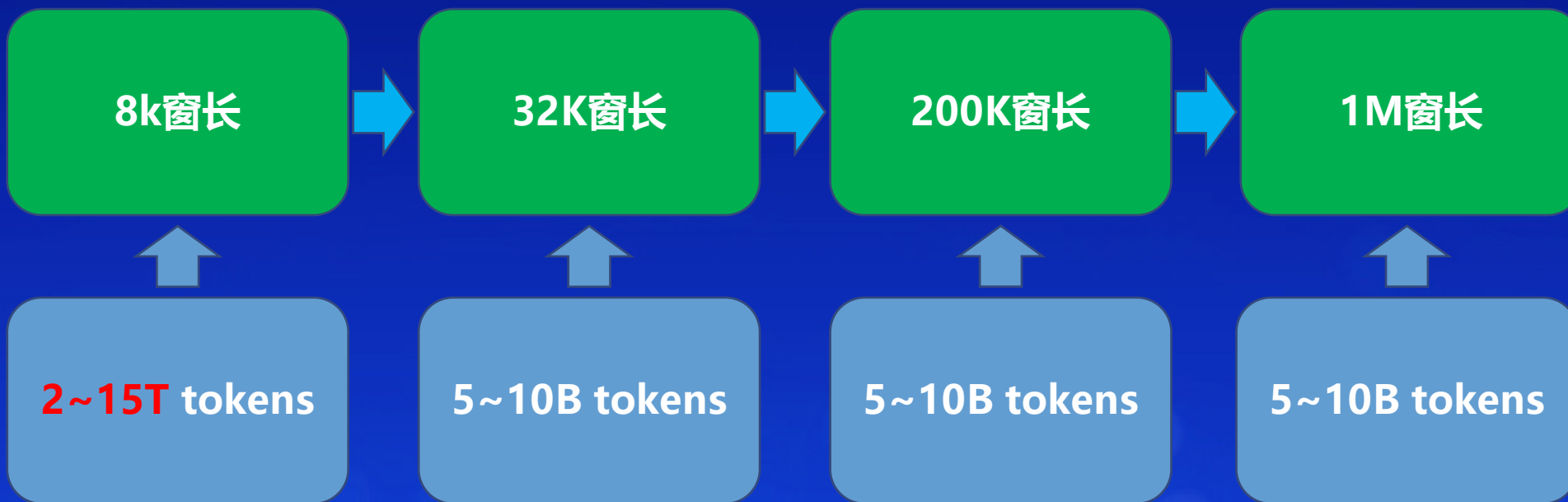
模型窗长扩展方法：位置编码方向，训练成本不高

3. PoSE: Efficient Context Window Extension of LLMs via Positional Skip-wise Training



长文本模型的技术挑战

训练成本主要在8k预训练阶段：attention的平方复杂度



主要优化点

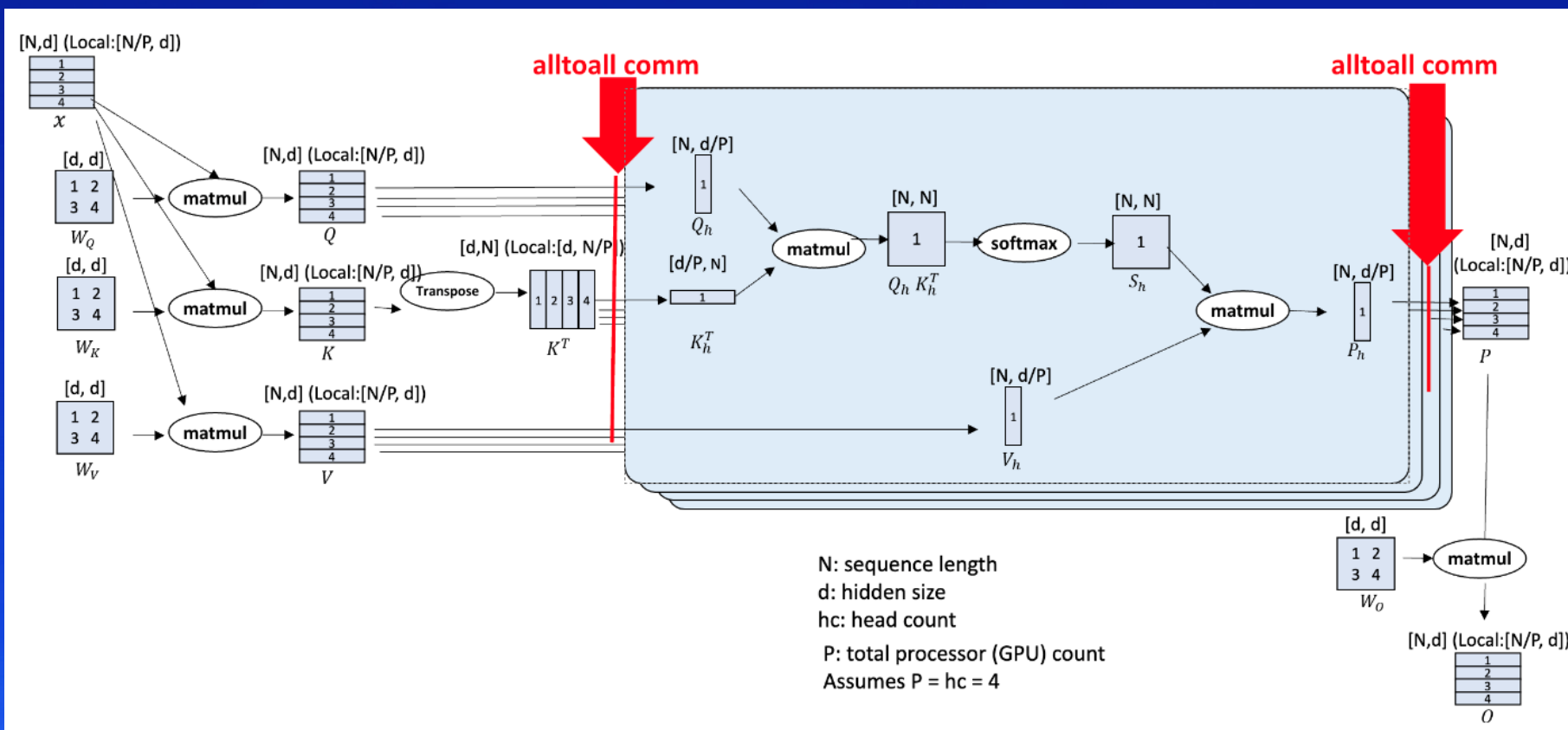
长文本模型的技术挑战

Decoder only Transformer长文本模型训练infra开发

1. DeepSpeed Ulysses:

优势：对Attention 的实现不敏感，适合各种attention方法

劣势：序列并行度不能超过头数



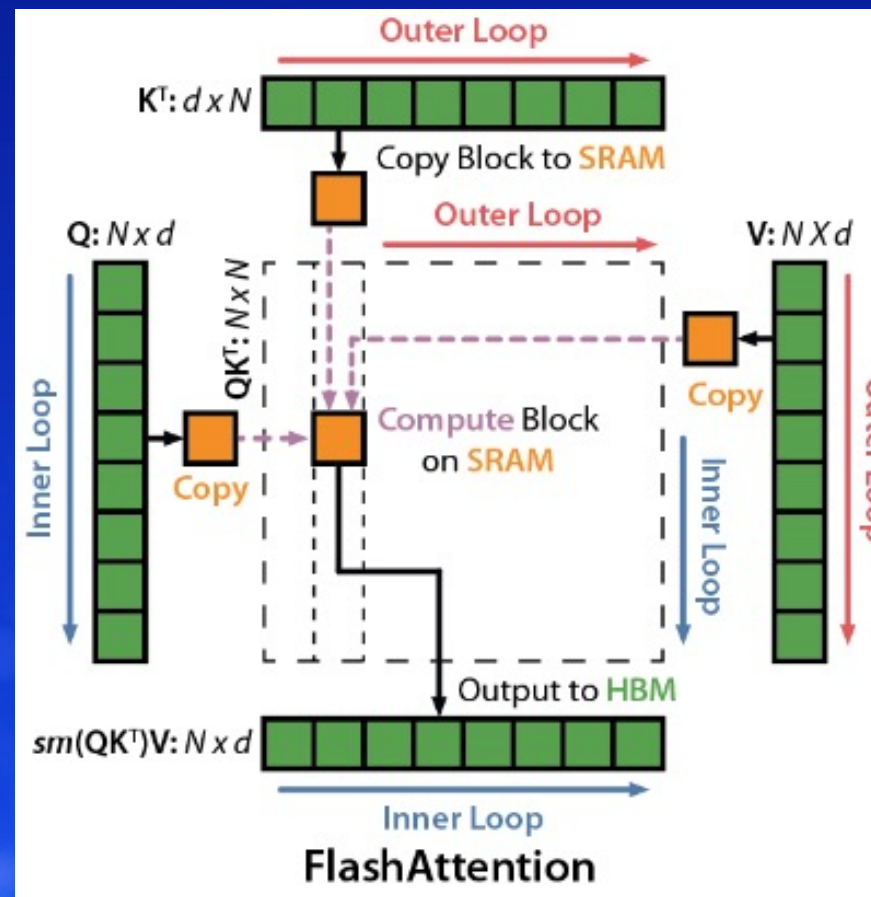
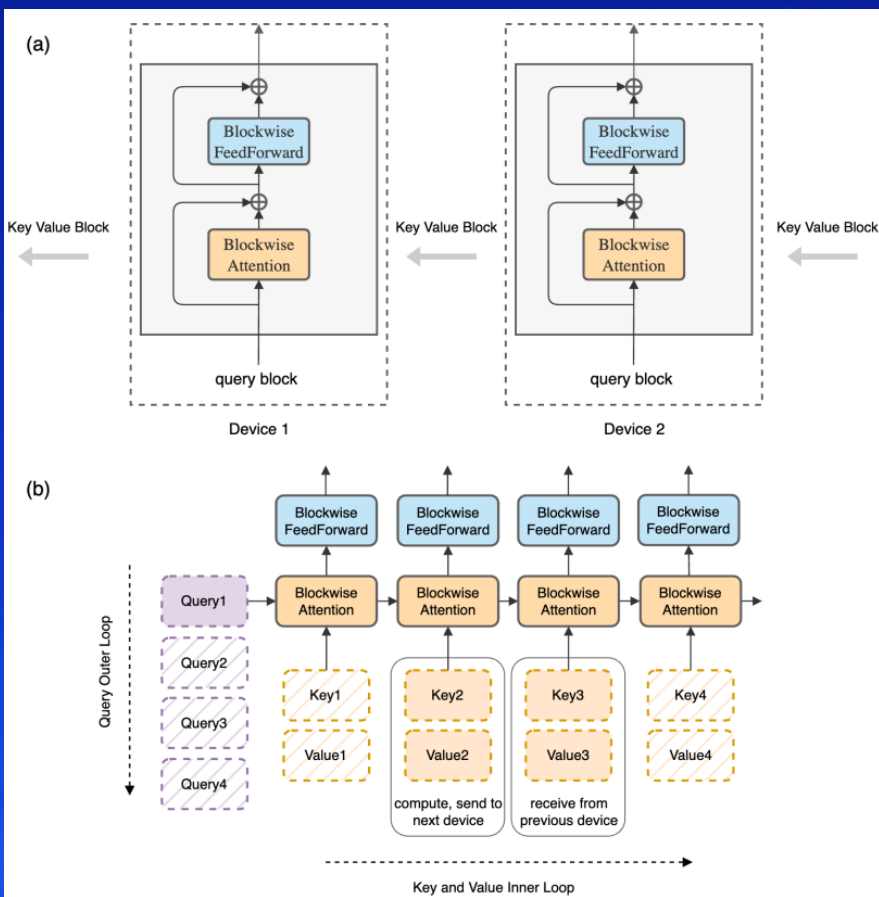
长文本模型的技术挑战

Decoder only Transformer长文本模型训练infra开发

2. Ring-attention: “大号”的flash attention

优势：并行度的扩展性较好

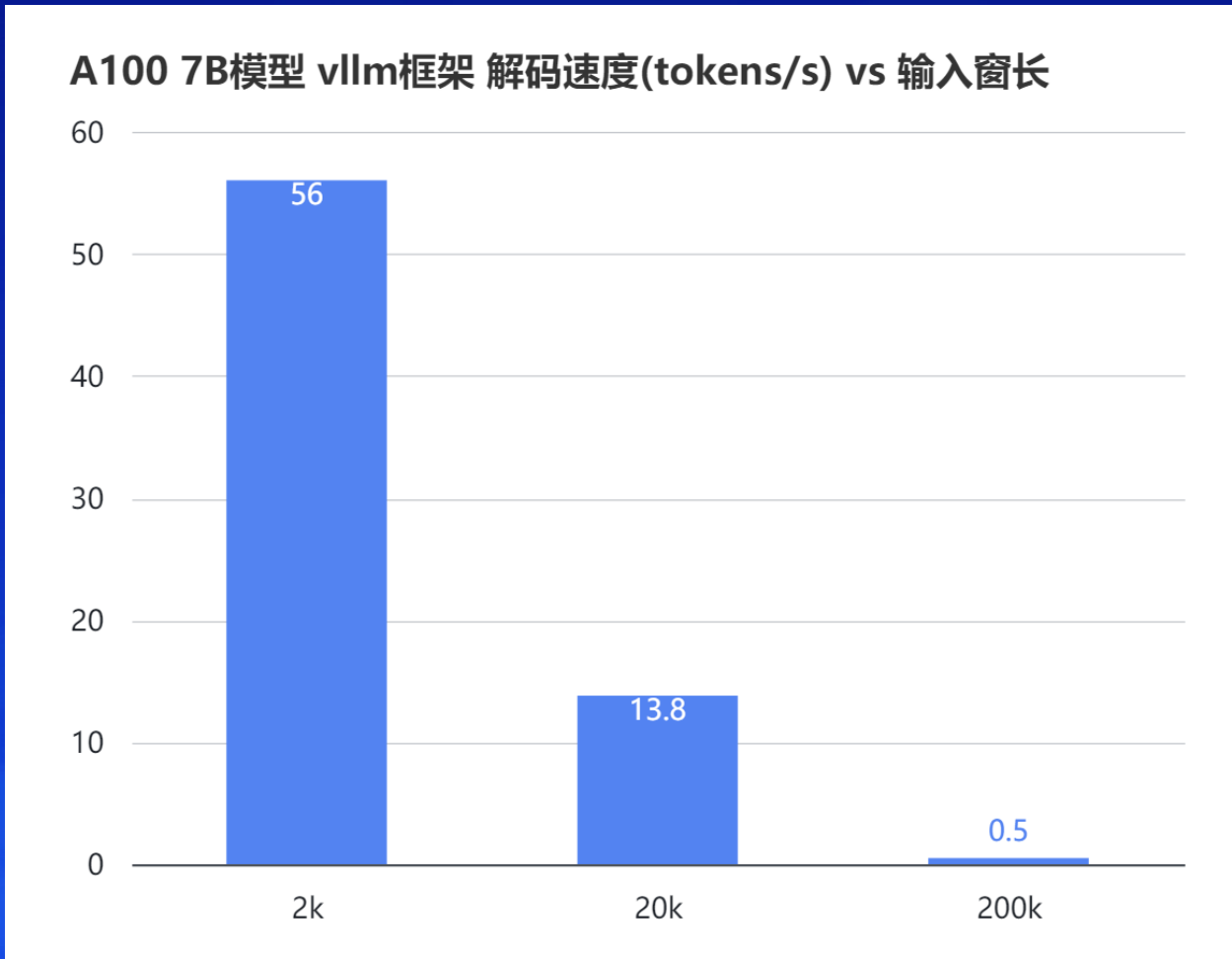
劣势：对Attention变种不友好，eg. Sparse Attention



长文本模型的技术挑战

推理成本高：attention的平方复杂度

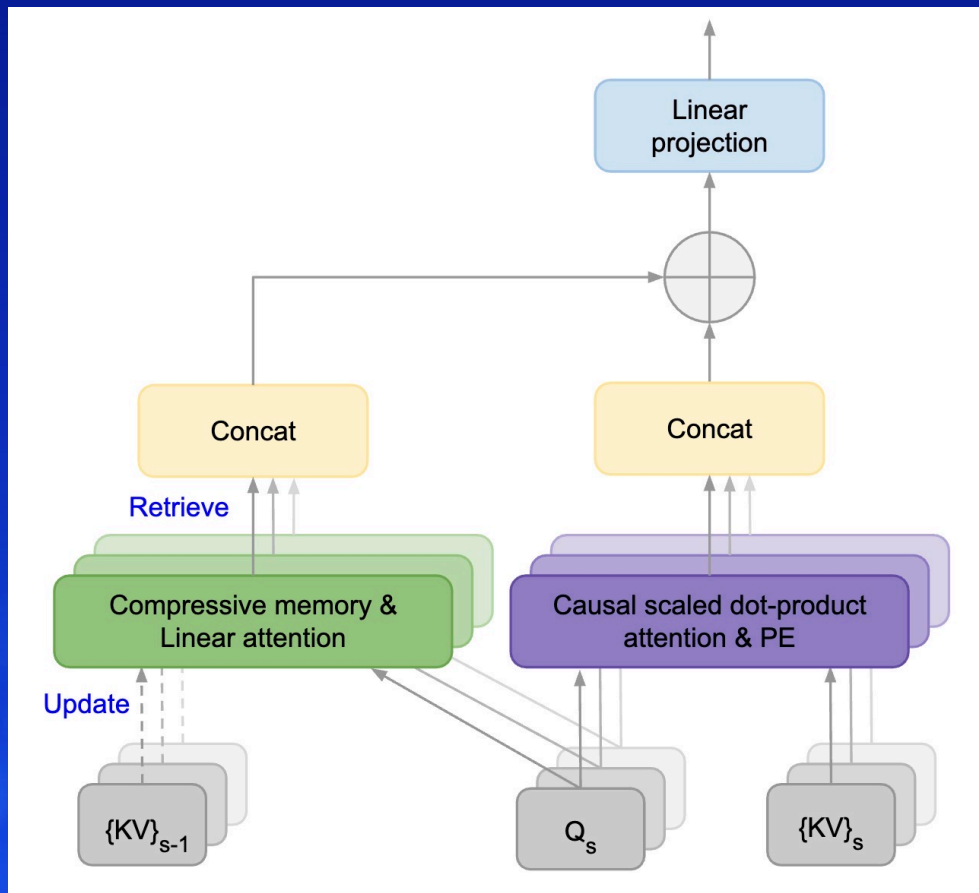
推理速度角度，200k 比 20k 贵28倍，比2k 贵112倍



长文本模型的技术挑战

其他长文本模型结构：

1. Infini-Transformer: 长期压缩记忆和局部因果注意力attention

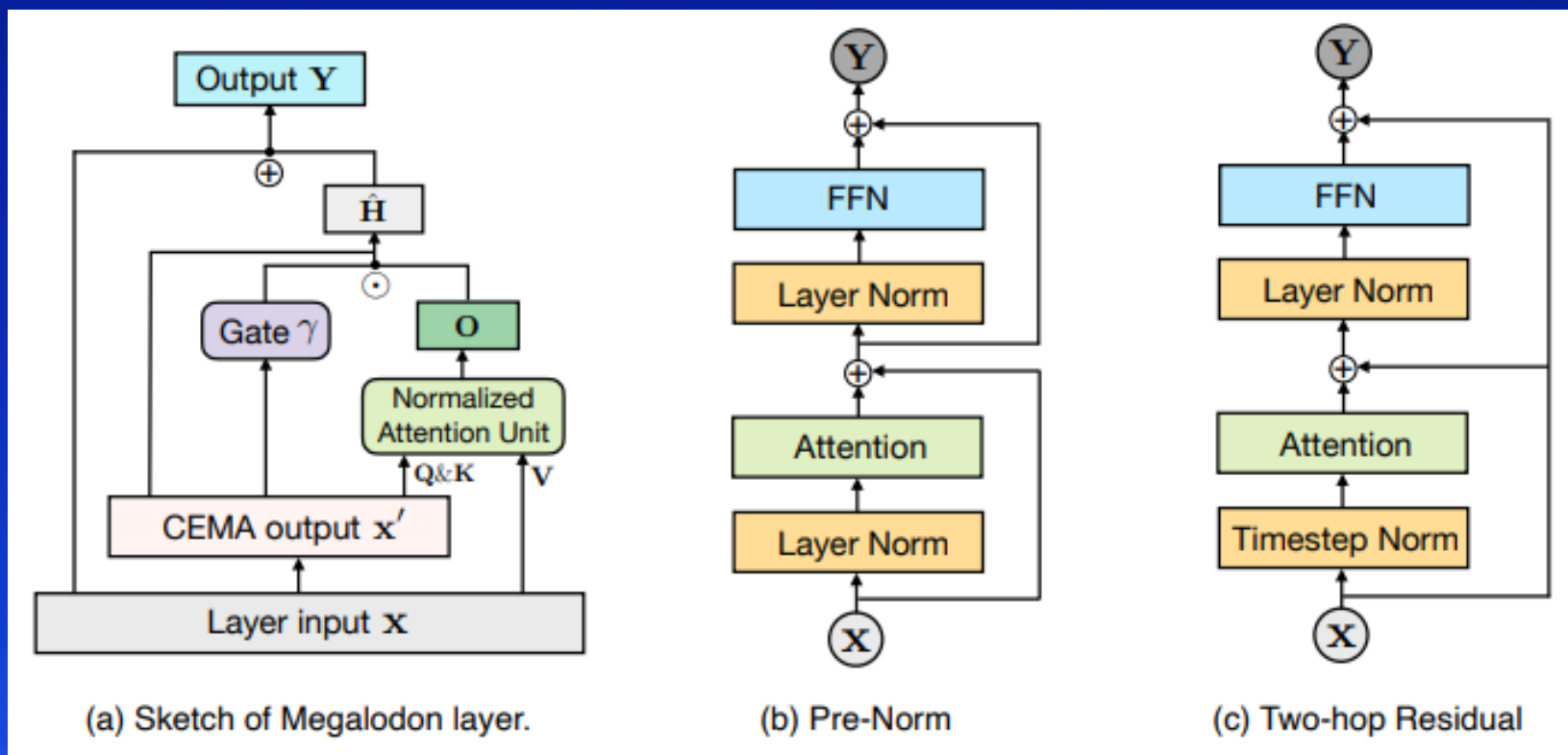


Infini-Transformer 模型结构 by Google, 2024

▶ 长文本模型的技术挑战

其他长文本模型结构：

2. MEGALODON：继承MEGA(带有门控注意力的指数移动平均) 并改进



MEGALODON 模型结构 by Meta, 2024

▶▶ 长文本模型的技术挑战

加速方法很多 备受关注 SUBLLM应运而生

| 方法 | 推理加速 | 训练加速 | Attention based大模型 | 兼容主流attention预训练大模型: eg. Llama,可持续训练 | 减少kv cache | 官方开源 |
|--|------|------|--------------------|--------------------------------------|------------|------|
| 投机解码 | √ | × | √ | √ | × | √ |
| medusa | √ | × | √ | √ | × | √ |
| 动态稀疏attention: eg. Minference | √ | × | √ | √ | × | √ |
| Kv cache压缩/剪裁: eg. StreamingLLM,H2O,DMC,CEPE | √ | × | √ | √ | √ | √ |
| CoLT5 | √ | √ | √ | × | × | × |
| Mamba2,RWKV | √ | √ | × | × | × | √ |
| RecurrentGemma: rnn+local attention | √ | √ | × | × | √ | √ |
| YOCO | √ | √ | √ | × | √ | √ |
| MEGALODON | √ | √ | √ | × | √ | √ |
| Infini-Transformer | √ | √ | √ | √ | √ | × |
| Mixture-of-depths | √ | √ | √ | √ | √ | × |
| SUBLLM | √ | √ | √ | √ | √ | √ |

PART 02

SUBLLM架构

▶▶ SUBLLM架构

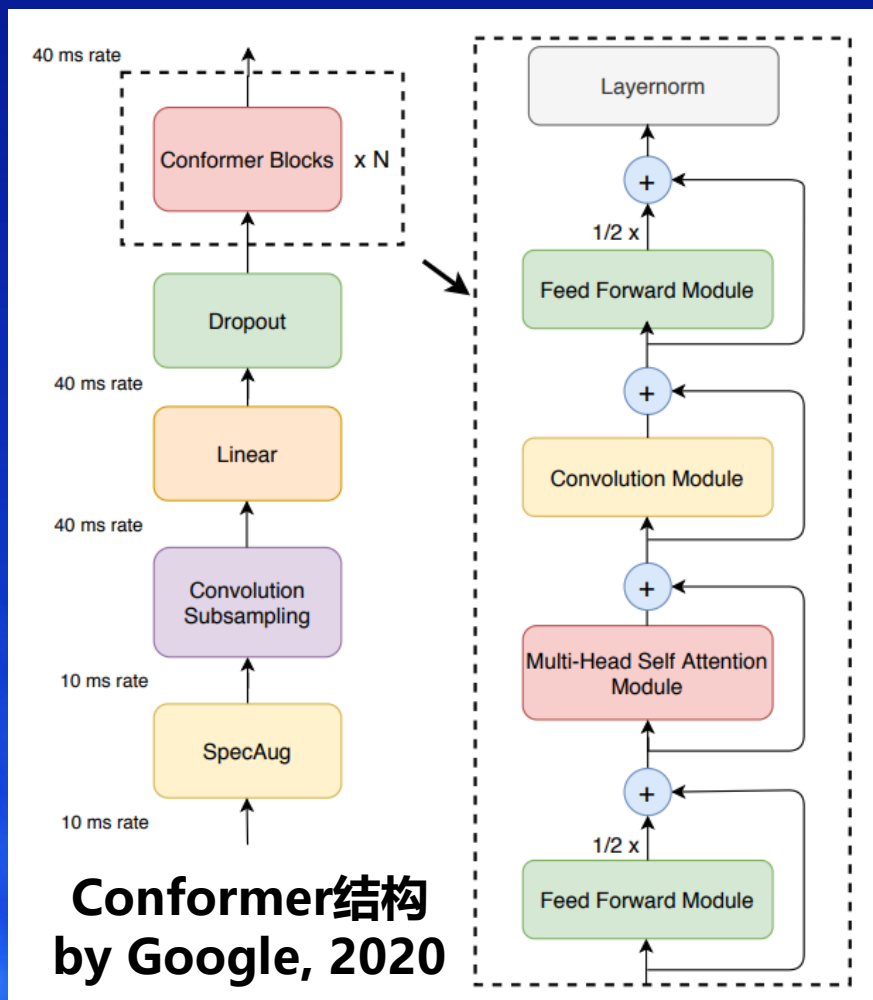
目标:

1. 开发一种优化资源使用的架构，同时保持模型能力不变。
2. 区分重要token和不重要token，重要token占主要算力
3. 兼容现有attention based模型生态，模型广泛应用的关键

► SUBLLM架构

受语音领域启发：语音信号下采样减少冗余 保留必要信息

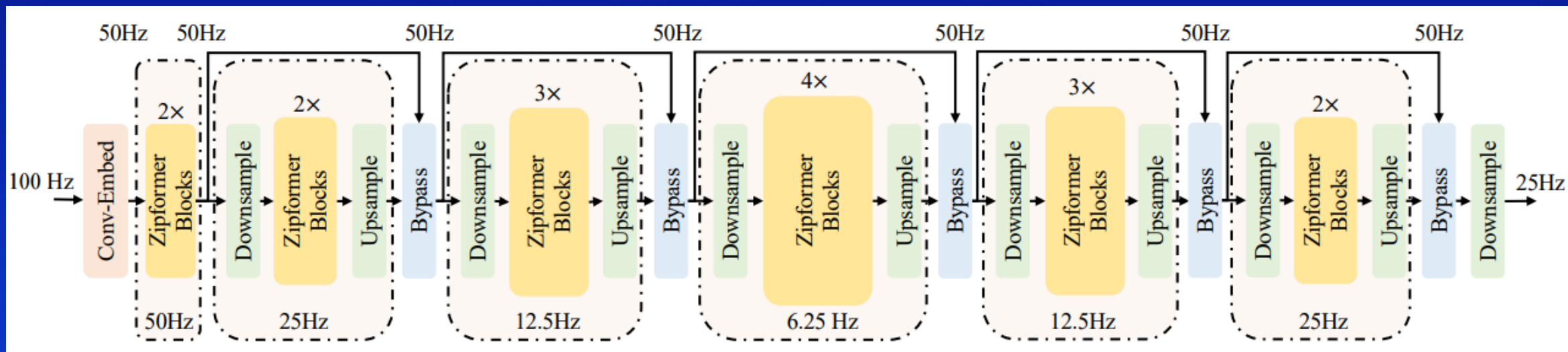
语音识别中语音分帧后降采样4倍



► SUBLLM架构

受语音领域启发：语音信号下采样减少冗余 保留必要信息

语音识别Zipformer，最高降采样16倍

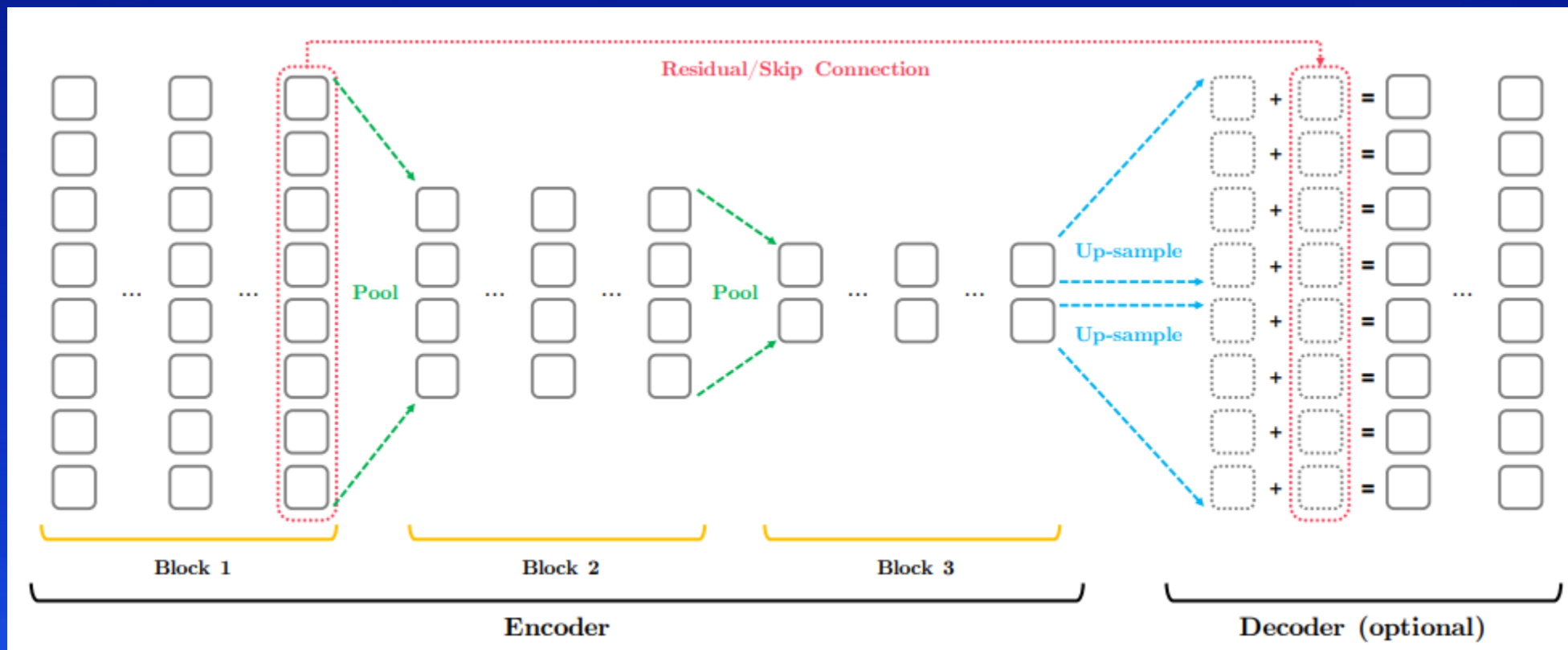


Zipformer 模型结构 by Daniel Povey, 2024

► SUBLLM架构

受语音领域启发：语音信号下采样减少冗余 保留必要信息

文本序列是否存在冗余？ Yes!

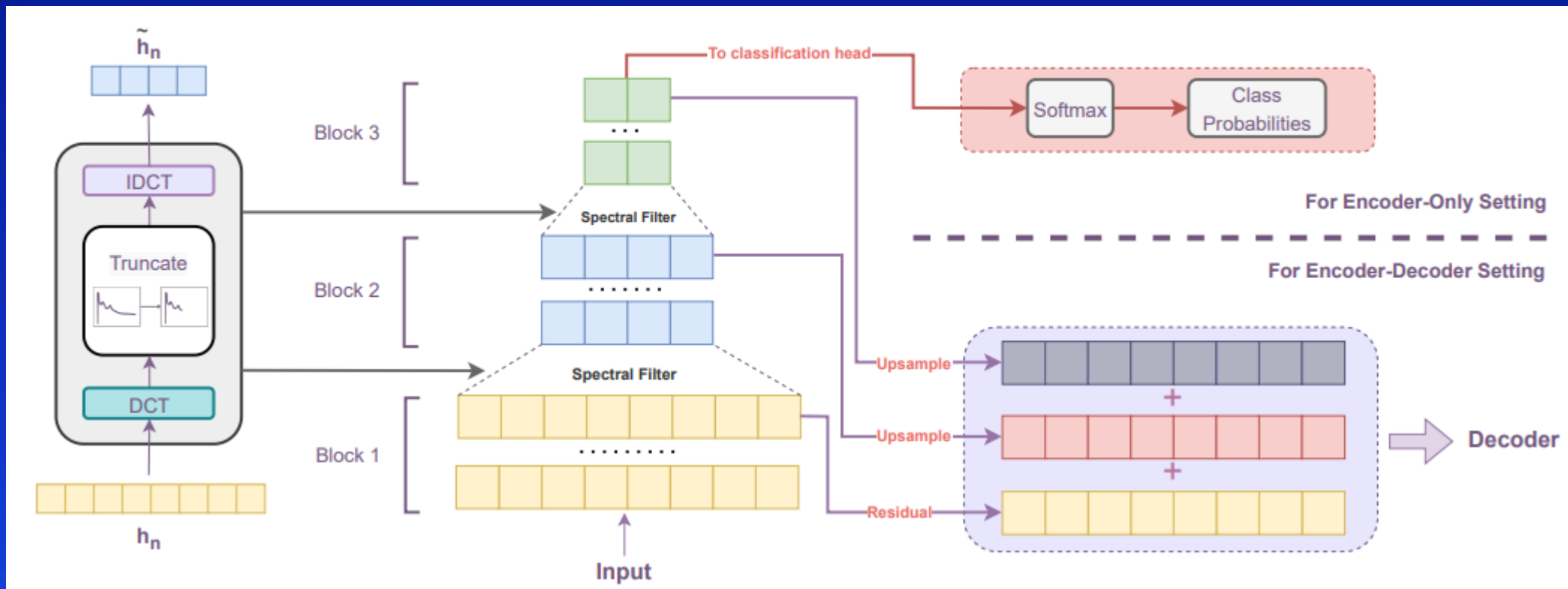


Funnel-Transformer 模型结构 by Google, 2020

► SUBLLM架构

受语音领域启发：语音信号下采样减少冗余 保留必要信息

文本序列是否存在冗余？ Yes!

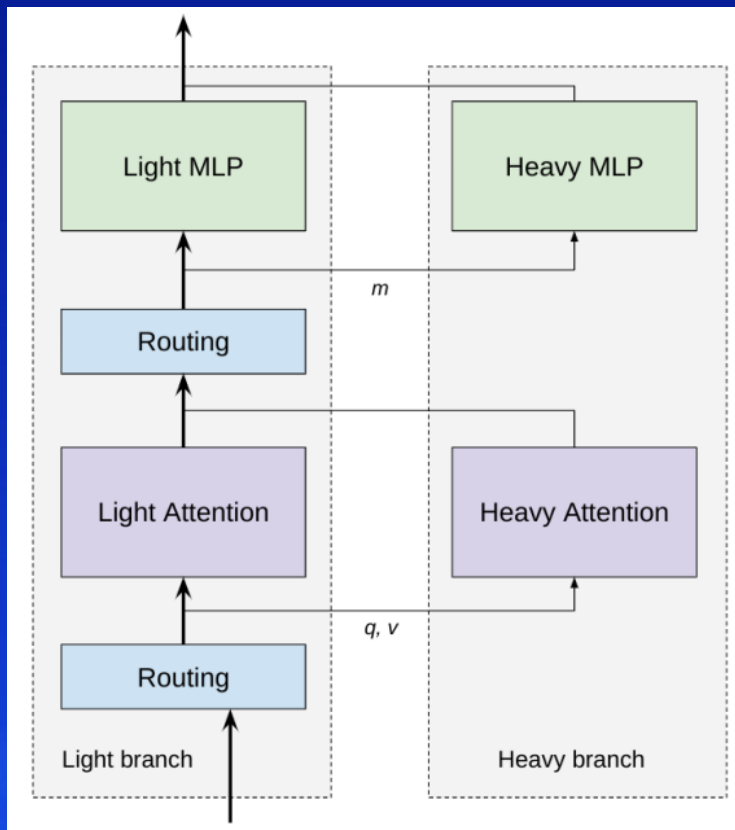


Fourier Transformer 模型结构, 2023

► SUBLLM架构

受语音领域启发：语音信号下采样减少冗余 保留必要信息

文本序列中的token是否同等重要？ No!

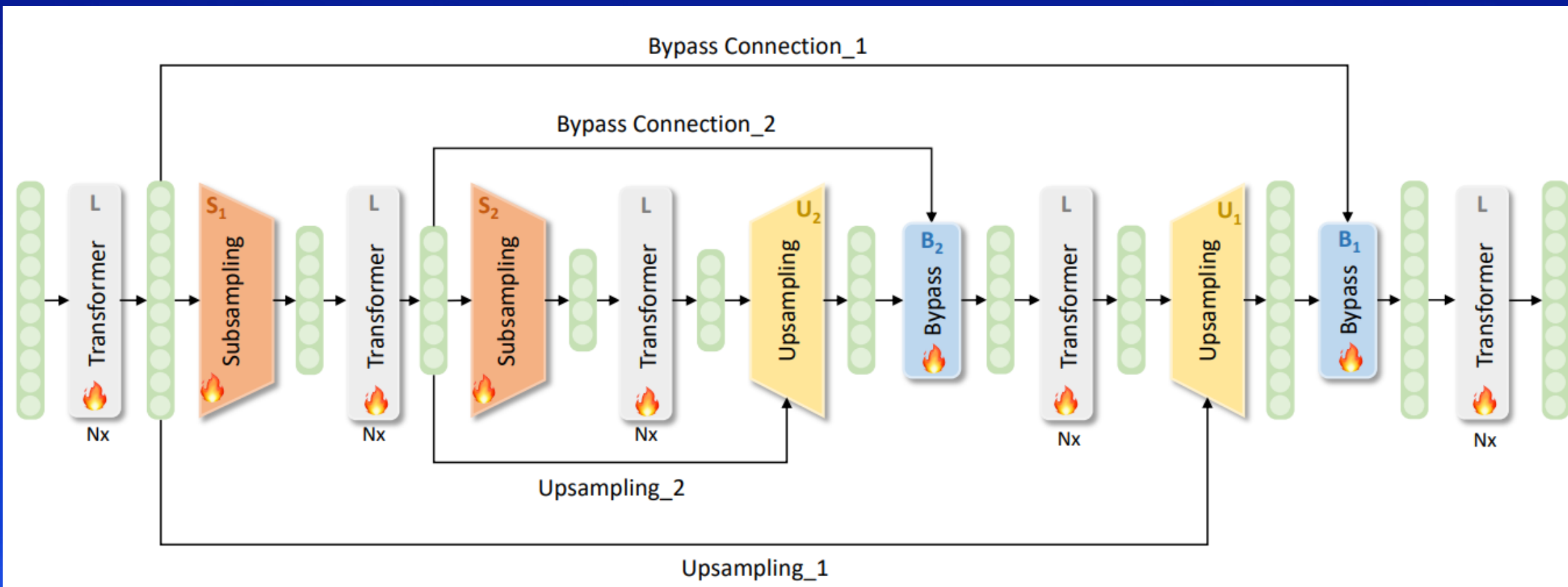


CoLT5 模型结构 by Google, 2023

► SUBLLM架构

受语音领域启发：语音信号下采样减少冗余 保留必要信息

SUBLLM：采样(Sub-sampling)-上采样(Up-sampling)-旁路(Bypass) LLM



SUBLLM 模型结构 by 小米AI实验室

<https://arxiv.org/pdf/2406.06571>, accepted by ECAI 2024

► SUBLLM架构

受语音领域启发：语音信号下采样减少冗余 保留必要信息

SUBLLM：采样(Sub-sampling)-上采样(Up-sampling)-旁路(Bypass) LLM

| Blocks | S/U Num | Model representation |
|--------|---------|--|
| 15 | 1 | $5L_S_1_5L_U_1_B_1_5L$ |
| 15 | 2 | $3L_S_1_3L_S_2_3L_U_2_B_2_3L_U_1_B_1_3L$ |
| 24 | 2 | $5L_S_1_5L_S_2_4L_U_2_B_2_5L_U_1_B_1_5L$ |

新模块插入位置示例，嵌套结构，逐渐下采样+逐渐上采样

Learnable Subsampling Module

1. 通过去除不太重要的tokens来缩短序列长度
2. 使用Score层来衡量token的重要性

序列采样, index选择
预设采样保留比例d

$$\begin{aligned} \mathbf{x}' &= \text{INDEXSELECT}(\mathbf{x}, \mathcal{I}) \\ &= x_{\mathcal{I}_1}, \dots, x_{\mathcal{I}_{N'}} \end{aligned}$$

$$N' = \lceil N * d \rceil$$

Score层给每个token打分
再得到weight

$$\begin{aligned} \mathbf{s} &= s_1, \dots, s_N \\ &= \mathcal{S}(\mathbf{x}_1), \dots, \mathcal{S}(\mathbf{x}_N) \\ w_n &= \text{CLAMP}(\text{BALANCER}(s_n, [0, 1])) \end{aligned}$$

$$\mathbf{w} = w_1, \dots, w_N$$

保留index, 舍弃index

$$\begin{aligned} \mathcal{I} &= \{i | w_i \in \text{TOPK}(\mathbf{w}, N')\} \\ \hat{\mathcal{I}} &= \{i | w_i \notin \text{TOPK}(\mathbf{w}, N')\} \end{aligned}$$

► SUBLLM架构

Learnable Subsampling Module

妙处1：解决训练和推理的不一致性

位置编码下采样

$$RelPos(\mathbf{x}'_i, \mathbf{x}'_j) = \text{ROPE}(\mathbf{x}'_i, \mathbf{x}'_j, (\mathcal{I}_i - \mathcal{I}_j))$$

Inference Mode: 推理时看不到未来token, 设定阈值为0, 正数分值的token保留

$$\mathcal{I}_{infer} = \{i | w_i \geq v\}$$

Balancer: 通过对梯度进行惩罚来限制score的正数比例=预设下采样保留比例
只保留正数分值的token

Upsampling Module

通过将采样后的tokens与原始序列合并来恢复序列长度

妙处2：用减法使得token的完全选择可导，使score层有token判别能力

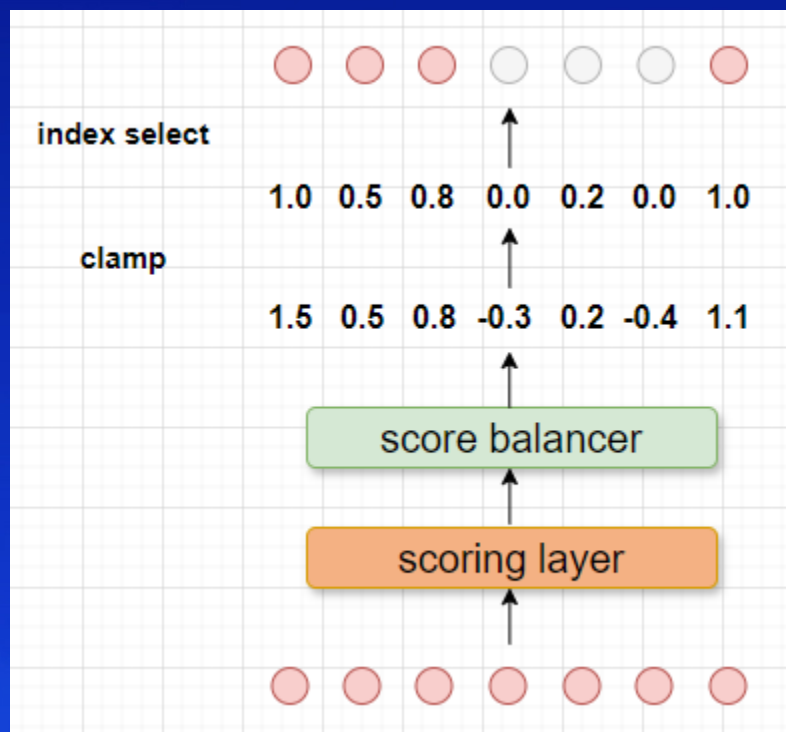
$$\begin{aligned} \mathbf{w}_{kept} &= \text{INDEXSELECT}(\mathbf{w}, \mathcal{I}) \\ \mathbf{w}_{discarded} &= \text{INDEXSELECT}(\mathbf{w}, \hat{\mathcal{I}}) \\ \mathbf{w}_{sample_i} &\sim \text{UNIFORM}(\mathcal{E}), \text{ for } i = 1, 2, \dots, N' \\ \mathbf{w}_{scaling} &= \mathbf{w}_{kept} - \mathbf{w}_{sample} \end{aligned}$$

\mathbf{w}_{sample} 实际基本为0，不影响weight大小，但是有梯度

$$\mathbf{x}_{new,i} = \begin{cases} \mathbf{w}_{scaling,i} * \mathcal{G}(\mathbf{x}_i) + (1 - \mathbf{w}_{scaling,i}) * \mathbf{x}_i, & \text{if } i \in \mathcal{I} \\ \mathbf{x}_i, & \text{otherwise} \end{cases}$$

▶ SUBLLM架构

Subsampling and Upsampling Module



Bypass Module:

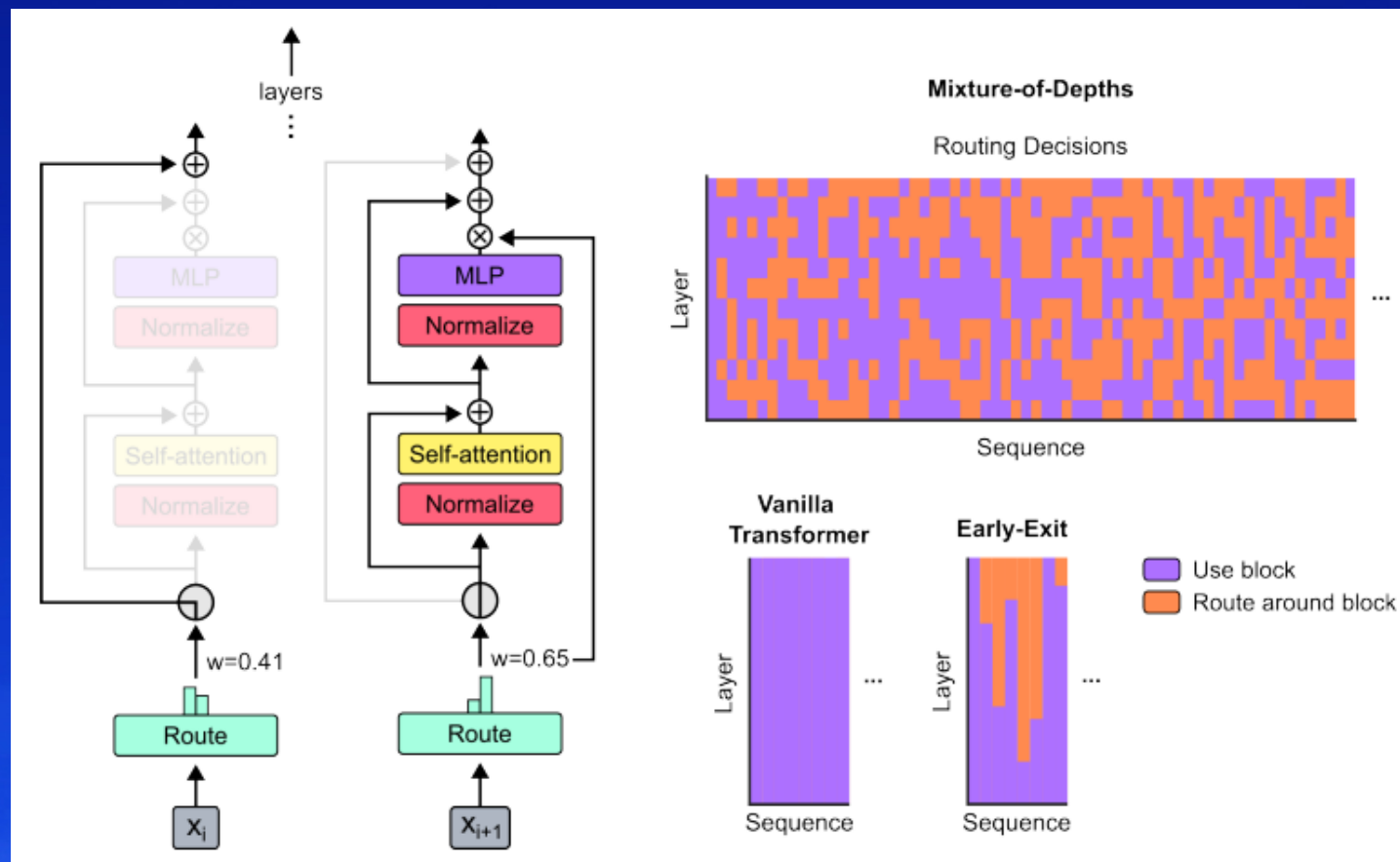
1. 下采样前和上采样后的序列加权求和, per channel
2. 增强训练的收敛性和稳定性。

$$\mathbf{y} = (1 - \mathbf{c}) \odot \mathbf{x} + \mathbf{c} \odot \mathbf{y}$$

| Variant | Valid Loss↓ |
|---------------------------------------|-------------|
| SUBLLM | 3.66 |
| - Bypass Module + Residual Connection | 3.72 |
| - Bypass Module | 3.73 |
| LLaMA | 3.69 |

对比Google: Mixture-of-Depths (MoD) :

1. 给重要token分配更多算力
2. MoD 每间隔一个block做一次topk的token筛选
3. 加速50%



Mixture-of-Depths 模型结构 by Google, 2024

PART 03

主要实验结果

▶▶ 主要实验结果

Pre-Training Corpora

SlimPajama, 100倍模型大小tokens

Pre-Training Details

SUBLLM 1.3B vs Llama 1.3B, SUBLLM仅仅多了8192个参数

SUBLLM 2次下采样, 最短的地方总保留比例仅40%

BF16, Flash Attention2

训练窗长2k 4k 8k

优化器为ScaledAdam

主要实验结果

| Efficiency | Pre-Training | | | Inference | | | | |
|------------|---------------|--------------|--------------|---------------|--------------|--------------|--------------|--------------|
| | Speed-Up | | Mem (GB)↓ | Max Speed-Up | | Speed↑ | Ratio↑ | Mem (GB)↓ |
| TGS↑ | Ratio↑ | TGS↑ | | Ratio↑ | | | | |
| LLaMA | 16,976 | ×1.00 | 65.99 | 18,856 | ×1.00 | 17.83 | × 1.00 | 18.49 |
| SUBLLM | 21,341 | ×1.26 | 55.81 | 24,773 | ×1.31 | 24.43 | ×1.37 | 17.29 |

| Performance | Pre-Training | | Few-Shot Learning | | | | | |
|-------------|--------------|--------------|-------------------|--------------|--------------|--------------|--------------|--------------|
| | Valid Loss↓ | SST2↑ | Amazon↑ | DBpedia↑ | AGNews↑ | Yelp↑ | Hate↑ | Avg.↑ |
| LLaMA | 3.11 | 81.01 | 86.54 | 45.70 | 64.77 | 87.59 | 45.18 | 68.47 |
| SUBLLM | 3.10 | 91.95 | 94.57 | 42.97 | 66.05 | 94.24 | 32.23 | 70.34 |

加速比 26%
最大加速比 31%
推理加速比 37%
训练/推理显存减少 10, 1GB/GPU
预训练valid loss 持平
Few-shot 分数 持平

PART 04

分析与讨论

预训练加速

| Model Size | Context Window | Model | Speed-Up | | | | Max Speed-Up | |
|------------|----------------|--------|----------|--------|-----------|--------|--------------|--------|
| | | | TGS↑ | Ratio↑ | Mem (GB)↓ | Δ Mem | TGS↑ | Ratio↑ |
| 0.25B | 2k | LLaMA | 85,925 | ×1.00 | 60.16 | - | 85,462 | ×1.00 |
| | | SUBLLM | 107,260 | ×1.25 | 53.76 | -6.41 | 107,859 | ×1.26 |
| | 4k | LLaMA | 77,590 | ×1.00 | 77.66 | - | 77,423 | ×1.00 |
| | | SUBLLM | 99,209 | ×1.28 | 69.03 | -8.63 | 100,425 | ×1.30 |
| | 8k | LLaMA | 64,959 | ×1.00 | 74.15 | - | 64,741 | ×1.00 |
| | | SUBLLM | 86,227 | ×1.33 | 66.03 | -8.11 | 87,261 | ×1.35 |
| 1.3B | 2k | LLaMA | 18,405 | ×1.00 | 72.99 | - | 20,284 | ×1.00 |
| | | SUBLLM | 22,831 | ×1.24 | 61.80 | -11.19 | 26,219 | ×1.29 |
| | 4k | LLaMA | 16,976 | ×1.00 | 65.99 | - | 18,856 | ×1.00 |
| | | SUBLLM | 21,341 | ×1.26 | 55.81 | -10.18 | 24,773 | ×1.31 |
| | 8k | LLaMA | 15,080 | ×1.00 | 65.89 | - | 16,587 | ×1.00 |
| | | SUBLLM | 19,390 | ×1.29 | 56.19 | -9.70 | 22,264 | ×1.34 |

推理加速

窗长越大，推理加速越多
8k窗长加速50%

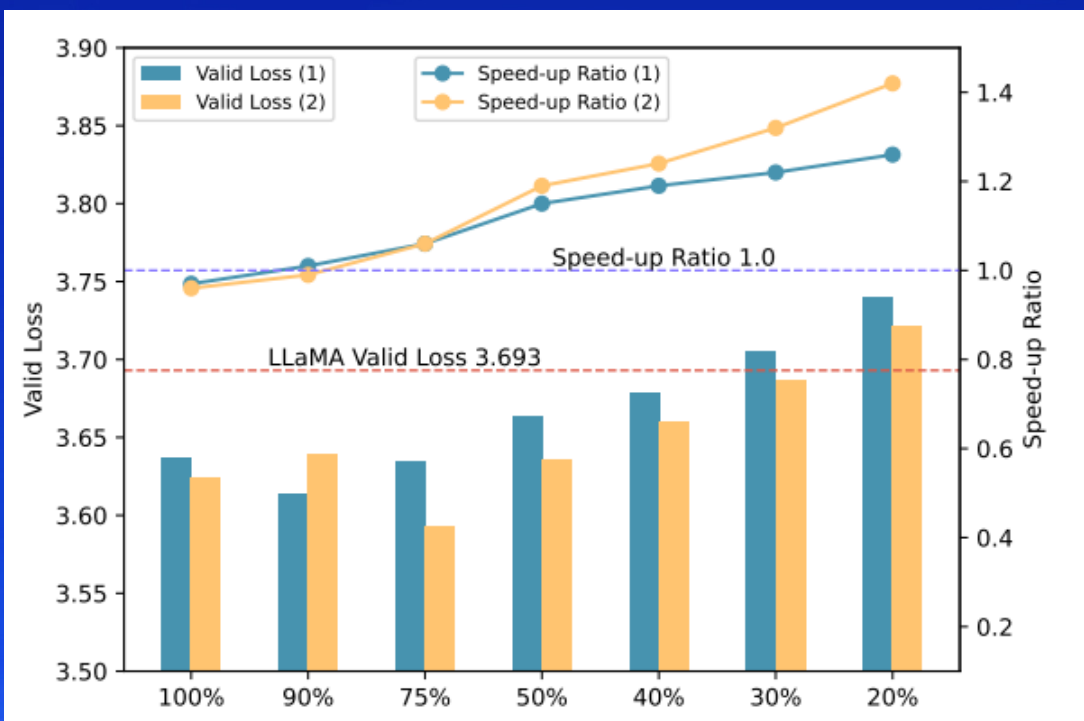
| Context Window | Model | Actual Retention | First Token | | Non-First Tokens | | Memory | |
|----------------|--------|------------------|---------------|--------|------------------|--------|-----------|-------|
| | | | Latency (ms)↓ | Ratio↑ | Speed ↑ | Ratio↑ | Mem (GB)↓ | Δ Mem |
| 2k | LLaMA | - | 695.16 | ×1.00 | 20.82 | ×1.00 | 6.98 | - |
| | SUBLLM | 43% | 496.66 | ×1.40 | 26.71 | ×1.28 | 5.63 | -1.35 |
| 4k | LLaMA | - | 2,051.59 | ×1.00 | 17.83 | ×1.00 | 18.49 | - |
| | SUBLLM | 44% | 1,410.94 | ×1.45 | 24.43 | ×1.37 | 17.29 | -1.20 |
| 8k | LLaMA | - | 16,249.11 | ×1.00 | 12.38 | ×1.00 | 61.05 | - |
| | SUBLLM | 44% | 9,758.40 | ×1.67 | 18.80 | ×1.52 | 58.61 | -2.44 |

分析与讨论

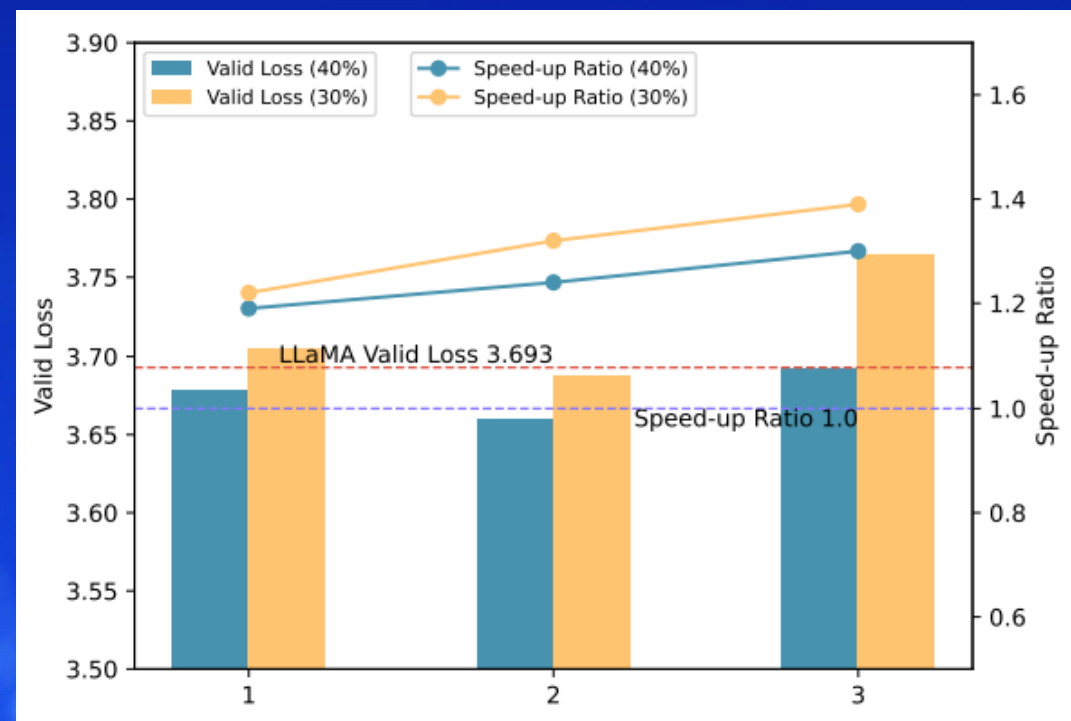
模型结构适用于不同的优化器

| | Adam | | ScaledAdam | |
|---------------|-------------|-------|-------------|-------|
| | Valid Loss↓ | Ratio | Valid Loss↓ | Ratio |
| LLaMA | 3.725 | - | 3.693 | - |
| SUBLLM | 3.743 | ×1.33 | 3.687 | ×1.32 |

寻找最优下采样次数和保留比例：
比例75% valid loss最低

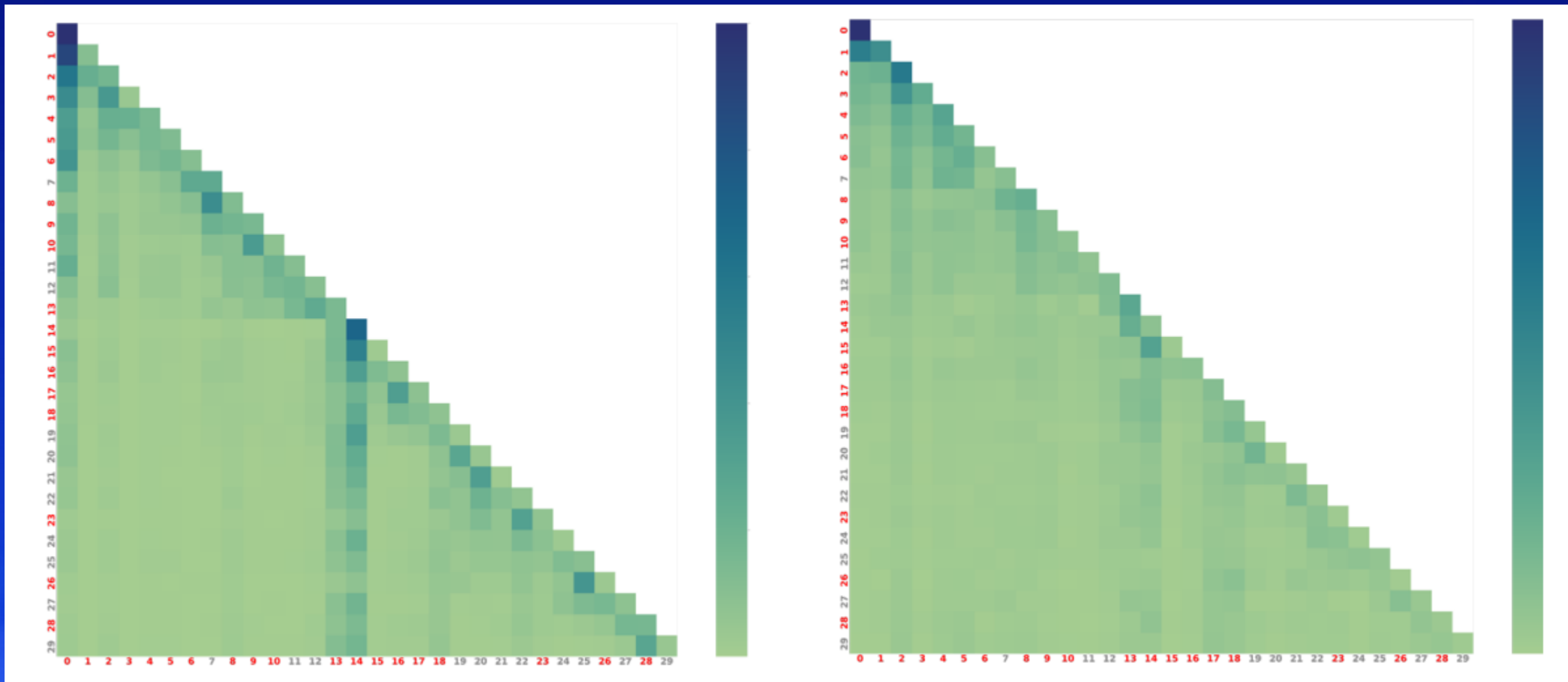


下采样次数2次最优



分析与讨论

下采样的有效性分析



下采样前的attention和保留index分布接近

Llama同一层的attention和保留index分布接近

PART 05

总结与展望

SUBLLM贡献

- 1. 提出SUBLLM新架构**
它结合了下采样、上采样和旁路模块
动态地将资源分配给重要的token
- 2. 提出了一种 token 序列子采样方法**
可以有效地测量 token 重要性分数并按预期控制分数值的分布
在推理过程中实现所需的下采样保留率
- 3. 与 Llama 模型相比**
SUBLLM 在训练和推理方面分别实现了加速，训练加速34%，推理50%
同时显著降低了内存成本，保持了模型能力

未来研究方向

1. 不同的tokenizer有不同的压缩率
如何确保SUBLLM结构的通用性并确定有效的保留比例
2. 应用在长文本场景，比如200K窗长上，可以最低保留多少比例
值得期待一下
3. 多模态模型中的视觉标记具有冗余性
在处理高分辨率图像时，大大降低了VLM的效率
SUBLLM应用在这个领域也将具有落地前景

科技生态圈峰会 + 深度研习



—1000+ 技术团队的选择



上海站

K+全球软件研发行业创新峰会

时间: 2024.06.21-22



敦煌站

K+思考周®研习社

时间: 2024.10.17-19



香港站

K+思考周®研习社

时间: 2024.11.10-12



K+峰会详情



上海站

Ai+研发数字峰会

时间: 2024.05.17-18



北京站

Ai+研发数字峰会

时间: 2024.08.16-17



深圳站

Ai+研发数字峰会

时间: 2024.11.08-09



AiDD峰会详情



2024 AI+研发数字峰会

AI+ Development Digital summit

深圳站 11/08-09

AI 驱动研发变革 促进企业降本增效

2024深圳站-议题设置

| | | | |
|--------|--------------------------------|------------------|-------------|
| AI+产品线 | LLM驱动产品创新 | LLM驱动需求与业务分析 | AI驱动设计与用户体验 |
| AI+开发线 | AI 原生应用开发框架与技术 | AI Agents在研发落地实践 | LLM驱动编程与单测 |
| AI+测试线 | LLM驱动测试分析与设计 | 基于LLM生成测试脚本与数据 | LLM和AI应用的评测 |
| AI+工程线 | AI+DevOps 与工具 (LLM 时代的平台工程) | 大模型对齐与安全 | 端侧大模型与云端协同 |
| AI+领域线 | 领域大模型 SFT 与优化 | 知识增强与数据智能 | 大厂专场 |

扫描右侧二维码
查看更多会议详情



早鸟票限时抢购中 (截止到9月30日)

¥ 3680

早鸟票

¥ 2800

学生票

THANKS

