长上下文和新型架构

《大语言模型》编写团队：李军毅

长上下文模型

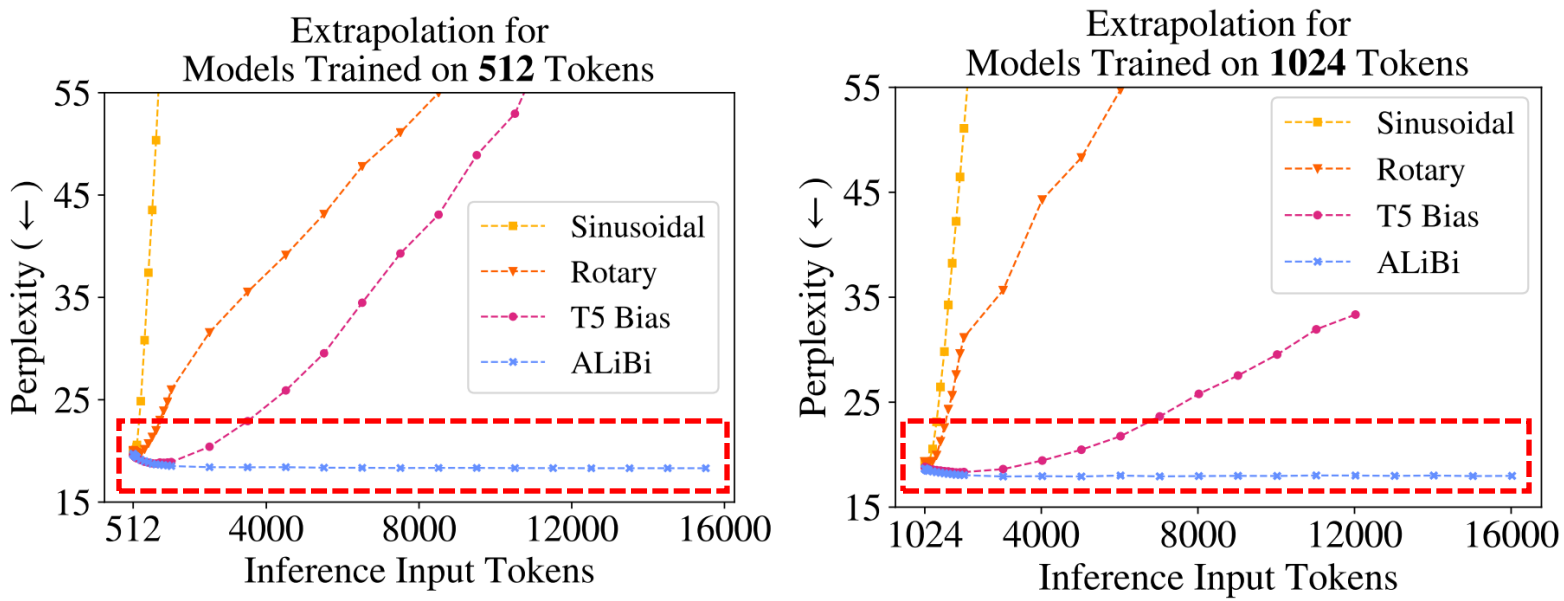
- 长上下文场景
  - 长文档分析
  - 论文阅读
  - 多轮对话
  - 故事创作

大模型	上下文窗口长度
LLaMA	2048
LLaMA-2	4096
LLaMA-3	8192
LLaMA-3.1	128K
Mistral-v3	32K
Qwen-2.5	128K
DeepSeek-R1	160K
Claude-3	200K
Gemini-1.5	2M
GPT-4o	128K
o1	200K

长上下文模型

AI Box

- 长度外推 (Length Extrapolation)
  - 在超出原始上下文窗口的文本上依旧保持与上下文窗口内部相似的性能
  - 代表模型：AliBi、T5 Bias、xPos

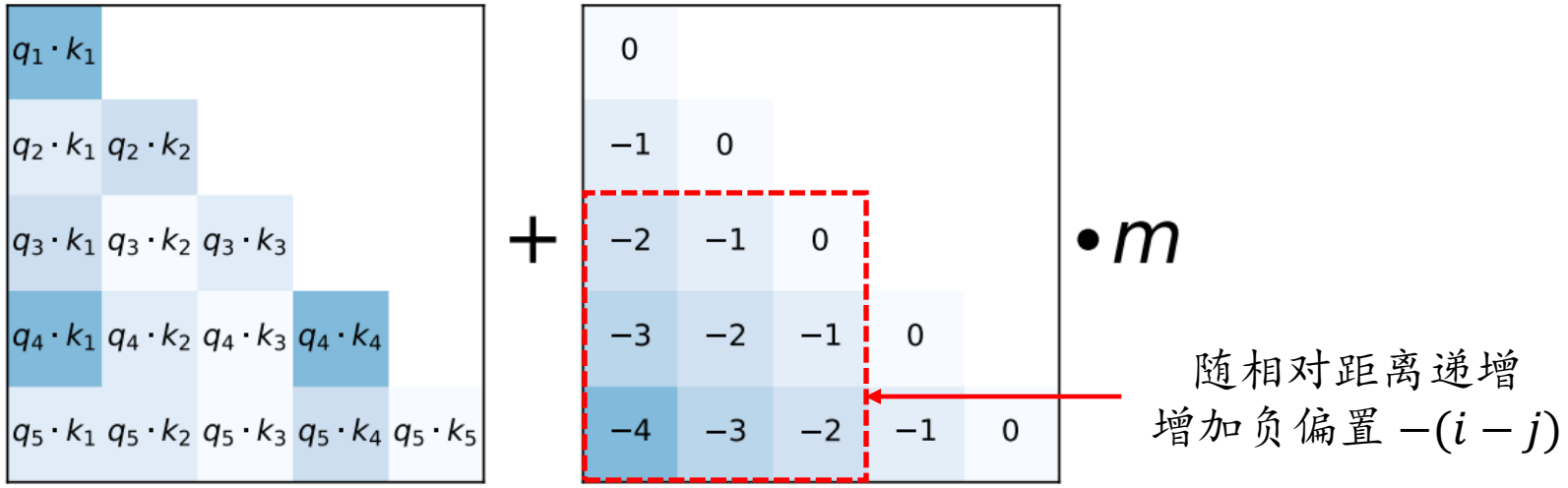


Train Short, Test Long: Attention with Linear Biases Enables Input Length. **ICLR 2022.**

长上下文模型

- 长度外推 (Length Extrapolation)
  - ALiBi 位置编码
    - 引入了与相对距离成比例关系的惩罚因子来调整注意力分数

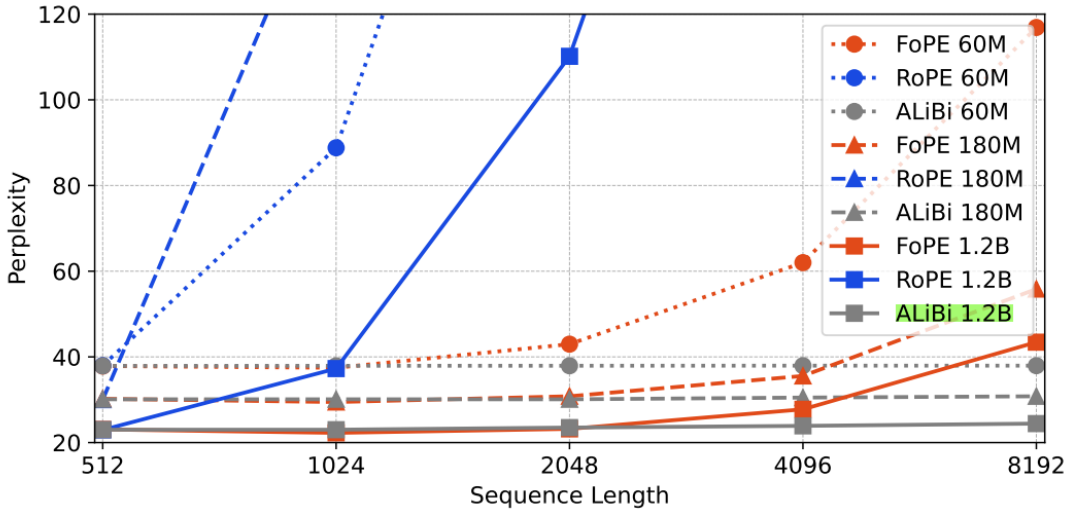
$$A_{ij} = \boldsymbol{x}_i \boldsymbol{W}^Q \boldsymbol{W}^{K\top} \boldsymbol{x}_j^\top - m(i-j),$$



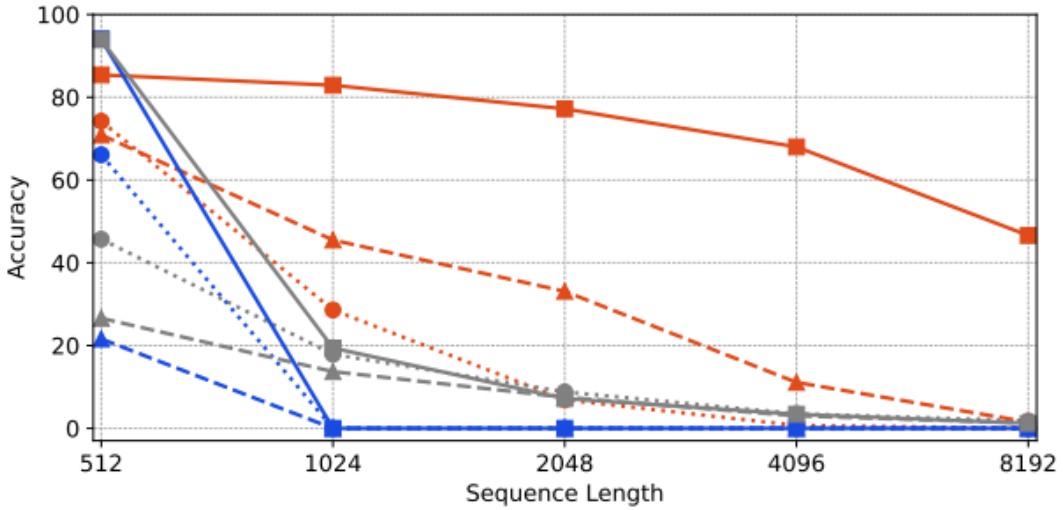
Train Short, Test Long: Attention with Linear Biases Enables Input Length. **ICLR 2022.**

长上下文模型

- 长度外推 (Length Extrapolation)
  - ALiBi 位置编码
    - 仍然无法保证在超出上下文窗口后对文本的理解能力



Perplexity 保持稳定



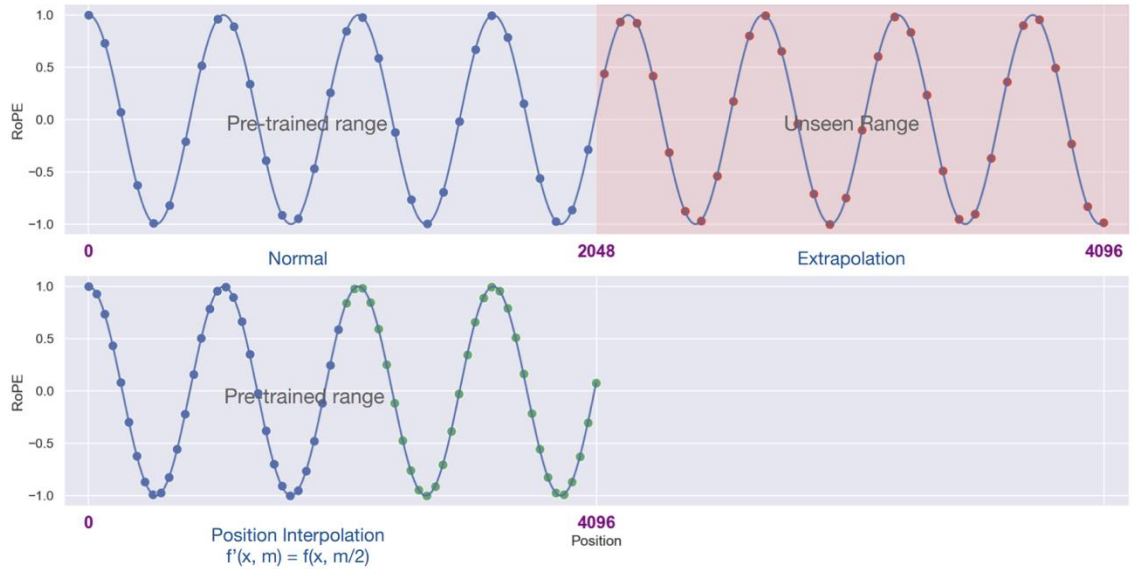
检索准确率下降

Fourier Position Embedding: Enhancing Attention's Periodic Extension for Length Generalization. **arXiv 2024.**

长上下文模型

AI Box

- 改进方法
  - 扩展位置编码
    - 模型在一定长度的数据上训练，超过训练长度的位置编码没有得到充分训练
    - 目标：原始上下文窗口 T_{max} 扩展为目标上下文窗口 T'_{max}



Extending Context Window of Large Language Models via Position Interpolation. **arXiv 2024.**

长上下文模型

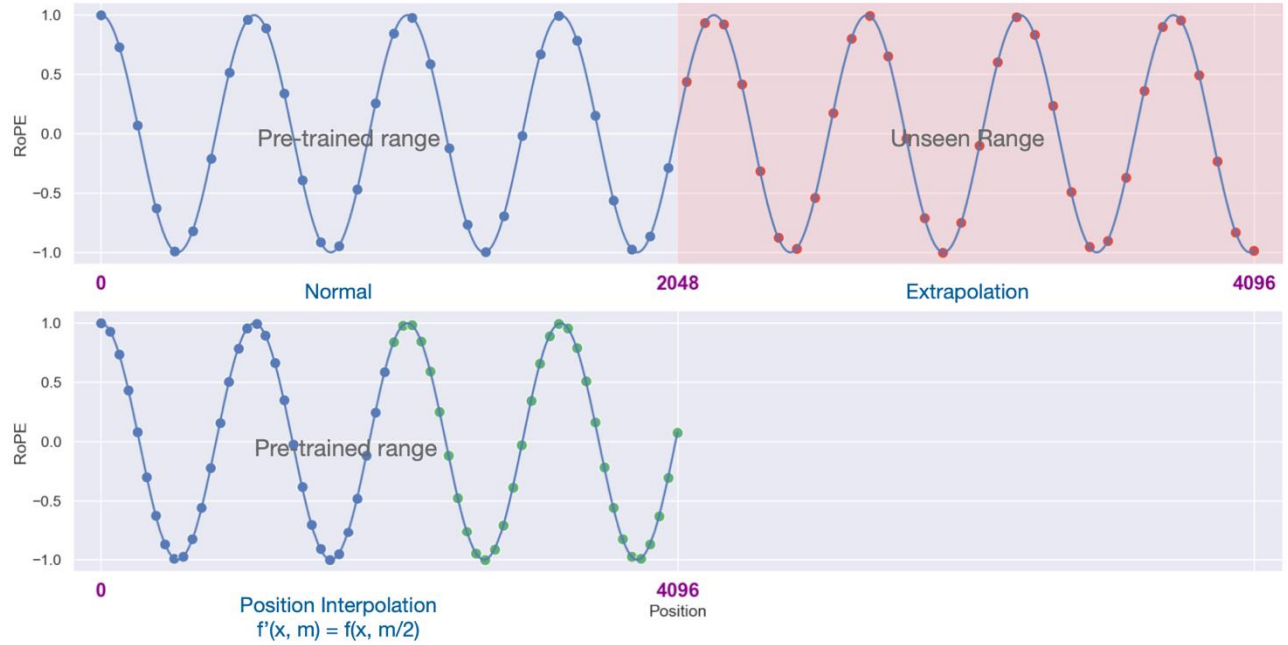
- 扩展位置编码 (以RoPE为例)
  - 在每个子空间 i 上，相对位置 t 的旋转角度为

$$f(t,i) = \boldsymbol{t} \cdot \boldsymbol{\theta_i}$$

  - 通过调整旋转角度 $f(t,i)$ 达到扩展上下文长度的目标
    - 修改相对位置索引 $\boldsymbol{t}$：$\boldsymbol{g(t)}$
      - 代表方法：ReRoPE 和 LeakyReRoPE
    - 修改旋转基 $\boldsymbol{\theta_i}$：$\boldsymbol{h(i)}$
      - 代表方法：NTK-RoPE 和 Dynamic-NTK-RoPE

长上下文模型

- 修改位置索引
  - 位置内插：将位置索引成比例缩放，保证旋转角度不超过最大值
    - 所有位置索引乘以一个小于1的系数，即 $g(t) = \frac{T_{\max}}{T'_{\max}} \cdot t$



将上下文窗口长度从 2048 扩展到 4096

Extending Context Window of Large Language Models via Position Interpolation. **arXiv 2024.**

长上下文模型

AI Box

- 修改位置索引
  - 位置截断：设置最大距离阈值 w，阈值内保留，阈值外截断或者插值
    - ReRoPE：将超过阈值的位置索引设为固定值
    - LeakyReRoPE：将超过阈值的位置索引线性内插到原始上下文窗口大小

$$g(t)=\begin{cases} t, & t <= w; \\ w, & t > w\text{且使用 ReRoPE;} \\ w+\frac{(T_{\max}-w)(t-w)}{T'_{\max}-w}, & t > w\text{且使用 LeakyReRoPE.} \end{cases}$$

可以直接应用于更长的上下文而无需重新训练，同时保持正常文本的建模能力

但需要对注意力矩阵做二次计算，增加计算开销

长上下文模型

AI Box

- 修改旋转基
  - 关键子空间：波长超过上下文窗口长度的子空间 ($\lambda_i > \mathrm{T_{max}}$)，无法对完整的旋转周期进行训练，需要调整旋转角度
  - 对旋转基 θ_i 进行缩放

$$f(T'_{\max}, i) = T'_{\max} \cdot h(i) \leq T_{\max} \cdot \theta_i$$

    - 修改旋转基的底数 b ($\theta_i = b^{-2(i-1)/H}$)
    - 旋转基截断

Effective Long-Context Scaling of Foundation Models. **NAACL 2024.**

长上下文模型

- 修改旋转基
  - 修改旋转基的底数 b
    - 增大底数以减小旋转基

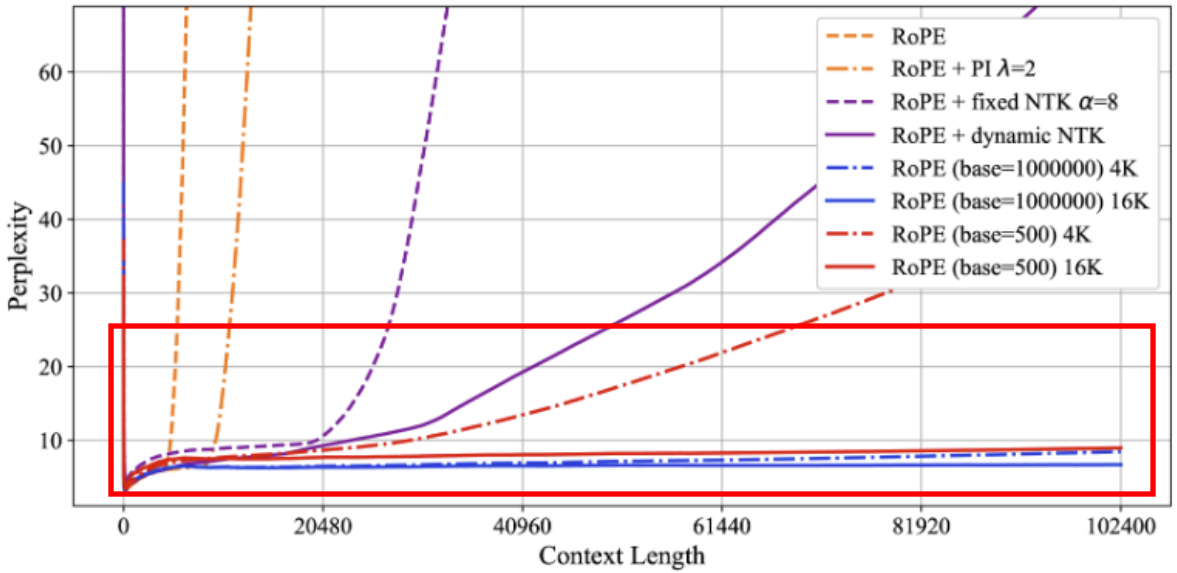
$$h(i) = (\alpha \cdot b)^{-(i-1)/H} \quad (\alpha \geq 1)$$

    - NTK-RoPE

$$\alpha = (T'_{\max}/T_{\max})^{H/H-2}$$

    - Dynamic-NTK-RoPE

$$\alpha = \max(1, T/T_{\max})$$



底数从500增加为1M，可处理上下文长度增加

无需额外训练就能够处理更长的上下文窗口

Scaling Laws of RoPE-based Extrapolation. **ICLR 2024.**

长上下文模型

AI Box

- 修改旋转基
  - 旋转基截断
    - 设置两个阈值 a 和 c，将子空间的基分为三部分

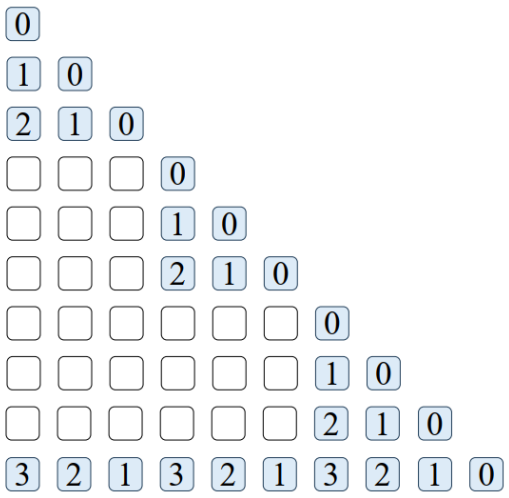
$$h(i) = \begin{cases} \theta_i, & \theta_i \geq c; \quad \text{较小的基保留原先的值} \\ \beta, & c \geq \theta_i \geq a; \quad \text{处于中间的基设置为一个较小的固定值} \\ 0, & \theta_i \leq a. \quad \text{较大的基直接置为 0} \end{cases}$$

有效防止位置索引较大时超出预期分布的旋转角度，提升长度外推能力

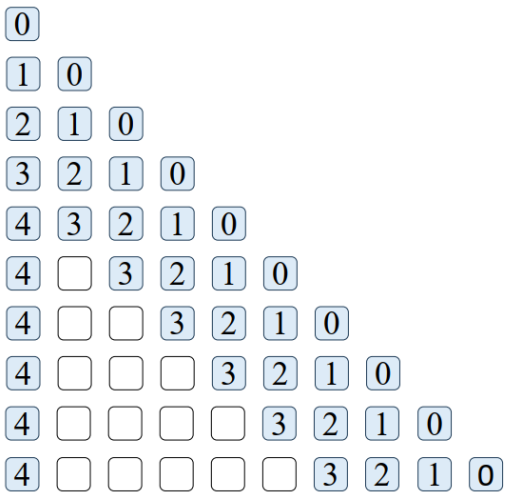
削弱了子空间对不同位置索引的分区能力

长上下文模型

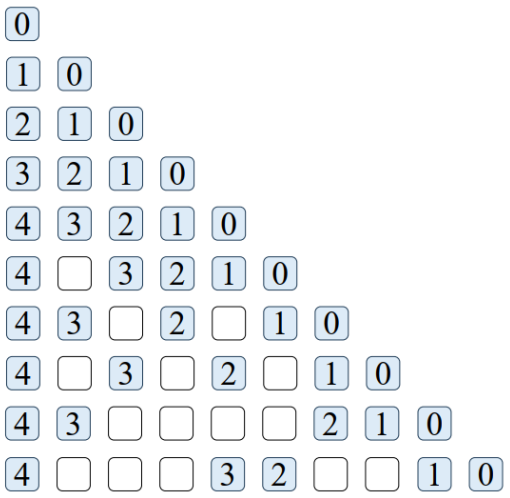
- 调整上下文窗口
  - 采用受限注意力机制实现对长文本的建模
  - 例如并行上下文窗口、Λ型上下文窗口、词元选择



(a) 并行上下文窗口



(b) Λ形上下文窗口

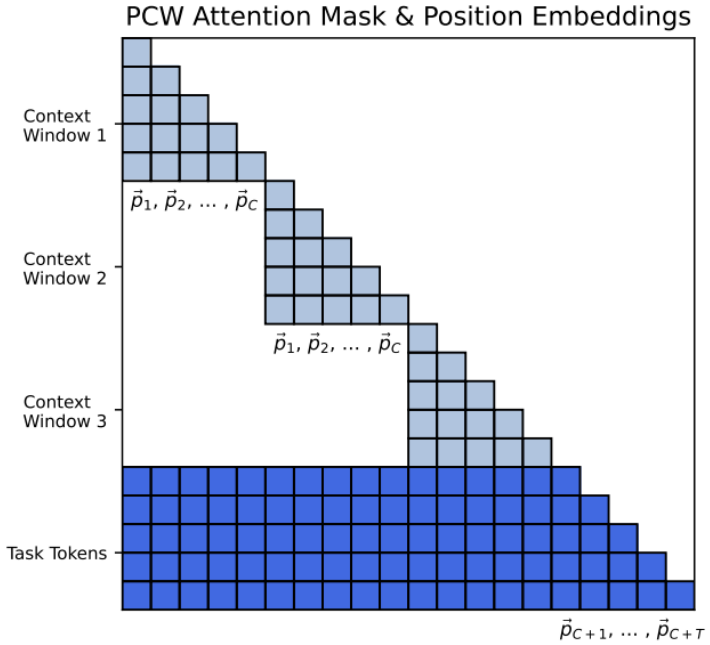


(c) 词元选择

长上下文模型

AI Box

- 并行上下文窗口
  - 将文本分成若干片段，每个片段单独编码，生成时关注所有前序词元
  - 代表方法：PCW

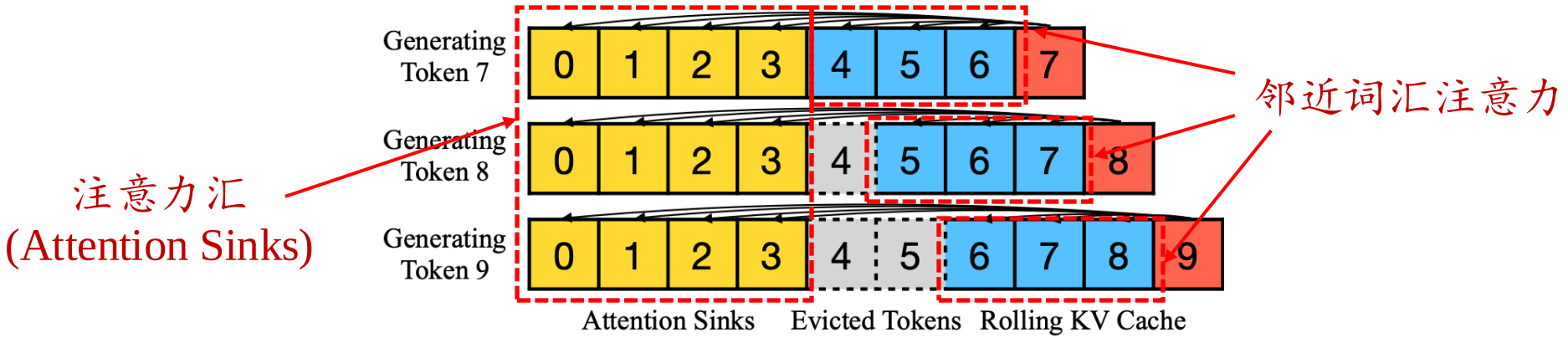


- 窗口之间共享位置编码
- 该方法无法有效区分不同片段的顺序关系
- 在某些特定任务性能受限

Parallel Context Windows for Large Language Models. **ACL 2023.**

长上下文模型

- Λ型上下文窗口
  - 每个词元仅关注序列最开始和邻近的词元
  - 代表方法：StreamingLLM，LM-infinite



在有限资源下实现长度外推，但忽略的词元信息影响上下文能力

Efficient Streaming Language Models with Attention Sinks. **ICLR 2024.**

长上下文模型

AI Box

- 词元选择
  - 基于查询与词元相似度的选择
    - 将词元按照距离分为近距离词元和远距离词元(存储于外部向量库)
    - 可通过检索获得最相关的远距离词元，补充远程语义信息
    - 代表方法：Focused Transformer
  - 基于查询与分块相似度的选择
    - 对文本分块并压缩为向量表示，使用k近邻方法选择最相关的分块并重新排序词元
    - 代表方法： LongHeads，InfLLM

Focused Transformer: Contrastive Training for Context Scaling. **NeurIPS 2023.**

长上下文模型

AI Box

上下文窗口	思想	代表模型	优缺点
并行上下文窗口	将文本分成若干片段，每个片段单独编码，生成时关注所有前序词元	PCW	该方法无法有效区分不同片段的顺序关系
Λ型上下文窗口	将文本分成若干片段，每个片段单独编码，生成时关注所有前序词元	StreamingLLM LM-infinite	在有限资源下实现长度外推 忽略词元信息影响上下文能力
词元选择	基于查询与词元相似度(按距离划分词元，选择最相关远距离词元)	Focused Transformer	需要构建额外的向量库 存储远距离词元的向量表示
	基于查询与分块相似度（选择最相关的文本块）	LongHeads InfLLM	通常能够在保证性能的同时 降低计算复杂度和内存需求

长上下文模型

- 基于长文本数据的继续预训练
  - 数据量：少量长文本数据可以实现有效的上下文窗口扩展
    - 1B tokens 可将 7B和13B LLaMA上下文长度有效拓展到100K

使用1B tokens训练的性能

Data Engineering for Scaling Language Models to 128K Context. **ICML 2024.**

长上下文模型

AI Box

- 基于长文本数据的继续预训练
  - 数据混合
    - 领域多样化，并与预训练数据分布比例尽量相似
    - 去除杂乱文本，保留连贯长文本，并对多篇相关文本聚合以及上采样

	C4	CC	Stack	Arxiv	Wiki	Book	Github
0 - 4K Context Length							
Original	2.038	1.760	1.519	1.660	1.424	2.085	0.907
v.s. Per-source	+ .002	+ .008	– .001	– .008	– .040	– .065	– .008
v.s. Global	+ .008	+ .010	+ .015	– .020	– .020	– .140	+ .015
v.s. Code↑	+ .010	+ .016	+ .010	+ .006	– .026	+ .030	– .023
v.s. Book↑	+ .010	+ .016	+ .021	+ .000	– .010	– .175	+ .029
v.s. Arxiv↑	+ .006	+ .016	+ .013	– .060	– .030	+ .040	+ .025
4K - 128K Context Length							
Original	1.560	1.650	0.786	1.075	1.313	1.852	0.447
v.s. Per-source	– .010	– .010	– .006	– .011	– .044	– .014	+ .002
v.s. Global	– .010	– .006	– .001	– .016	– .040	– .018	– .007
v.s. Code↑	– .008	– .002	– .003	– .007	– .042	– .010	– .029
v.s. Book↑	– .010	– .006	+ .001	– .007	– .037	– 0.30	+ .000
v.s. Arxiv↑	– .008	– .002	+ .002	– .036	– .039	– .010	– .004

与原始分布一致

	EN	Δ	ZH	Δ	Text	Δ
LLaMA2-7B-4K	28.55		13.62		21.41	
HOL. + AGG. + CHA.	32.86	+15.11%	17.18	+26.20%	24.30	+13.46%
HOL.	33.17	+16.20%	18.44	+35.44%	24.63	+15.02%
HOL. + AGG.	33.66	+17.91%	17.14	+25.88%	24.99	+16.70%
HOL. + Upsampling AGG.	**33.92**	+18.80%	**18.94**	+39.09%	**25.15**	+17.45%
InternLM2-7B	51.61		34.07		40.91	
HOL. + AGG. + CHA.	55.03	+6.63%	36.63	+7.52%	44.49	+8.74%
HOL.	55.12	+6.81%	36.97	+8.51%	44.61	+9.04%
HOL. + AGG.	55.54	+7.62%	37.36	+9.67%	44.79	+9.46%
HOL. + Upsampling AGG.	**56.64**	+9.76%	**39.31**	+15.38%	**46.26**	+13.07%

Data Engineering for Scaling Language Models to 128K Context. **ICML 2024.**
LongWanjuan: Towards Systematic Measurement for Long Text Quality. **arXiv 2024.**

长上下文模型

AI Box

- 以 Qwen2.5 和 Kimi K1.5 为例
  - Qwen2.5
    - 修改 RoPE 旋转基的底数 (10,000 调整为1,000,000)，上下文窗口大小从 4096 扩展至 32768
  - Qwen2.5-Turbo
    - 采用渐进式的四阶段训练策略，设置窗口大小分别为 32768、65536、131072 和 262144，将底数从 1,000,000 逐渐调整为 10,000,000
    - 每阶段的训练数据：40% 当前最大长度的序列 + 60% 短序列
  - Kimi K1.5
    - 预训练阶段逐渐扩展上下文窗口从 4K 到 32K，再到 128K

新型模型架构

AI Box

- 参数化状态空间模型 (State Space Model, SSM)
  - RNN和CNN的结合体，利用卷积计算并行编码
  - 仅依赖于前一个状态循环推理
  - 相比于 Transformer 长文本建模效率得到极大改进

模型	可并行性	解码复杂度	训练复杂度
Transformer	✓	$O(TH + H^2)$	$O(T^2H + TH^2)$
标准 SSM	✓	$O(N^2H + H^2)$	$O(TH \log T + THN^2 + TH^2)$
Mamba	×	$O(N^2H + H^2)$	$O(TN^2H + TH^2)$
RWKV	×	$O(H^2)$	$O(TH^2)$
RetNet	✓	$O(H^2)$	$O(TH^2)$
Hyena	✓	$O(TMH + MH^2)$	$O(TMH \log T + TMH^2)$

T：序列长度

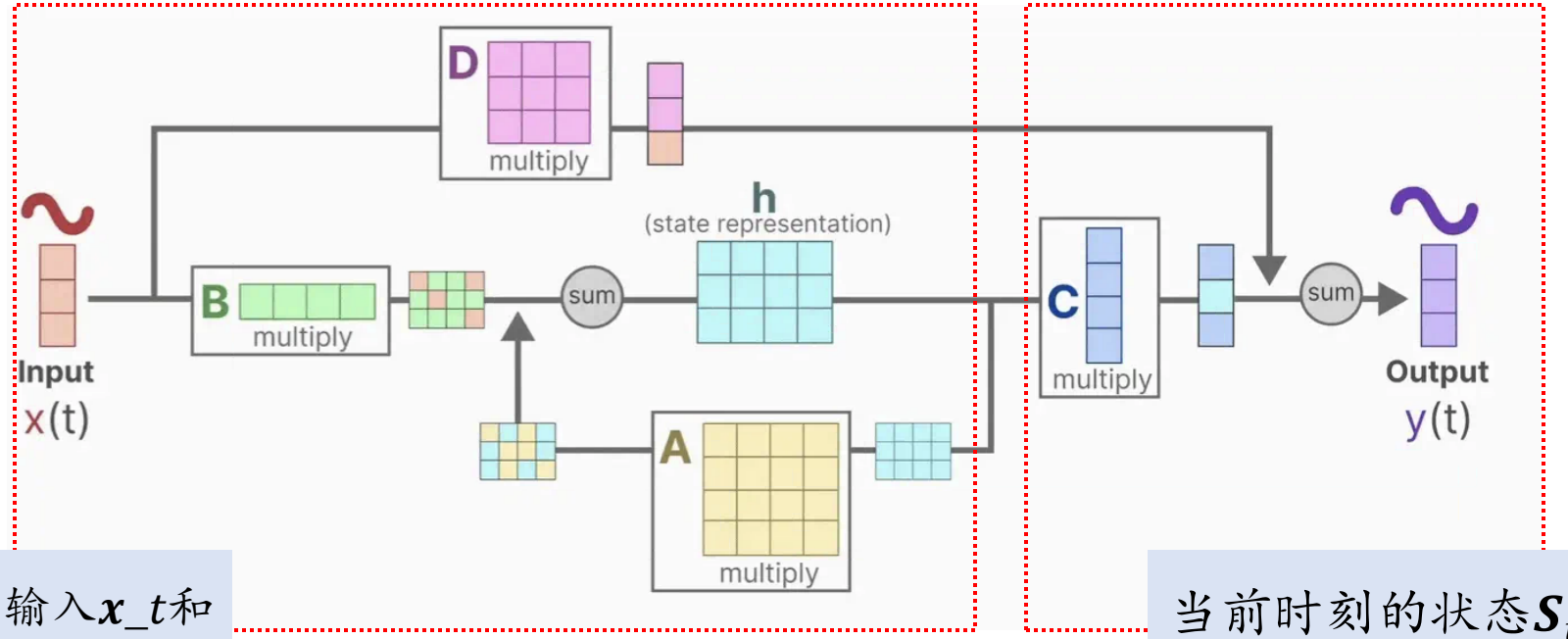
H：输入表示维度

N：状态空间模型压缩后的维度

新型模型架构

AI Box

➢ 参数化状态空间模型



基于当前时刻的输入x_t和前一时刻的状态$S_(t-1)$计算当前状态S_t

$$S_t = A \otimes S_{t-1} + B \otimes x_t$$

当前时刻的状态S_t映射为输出y_t

$$y_t = C \otimes S_t$$

新型模型架构

AI Box

- 参数化状态空间模型
  - 递归分解当前时刻的输出

$$y_t = C \otimes S_t = C \otimes A \otimes (A \otimes S_{t-2} + B \otimes x_{t-1}) + C \otimes B \otimes x_t$$

$$= C \otimes A^{t-1} \otimes Bx_1 + \cdots + C \otimes B \otimes x_t = \sum_{i=1}^{t} C \otimes A^{t-i} \otimes Bx_i$$

对每一时刻输入的卷积

卷积核

$$K = (C \otimes B, C \otimes A \otimes B, \ldots, C \otimes A^{t-1} \otimes B, \ldots)$$

$$y = x * K,$$

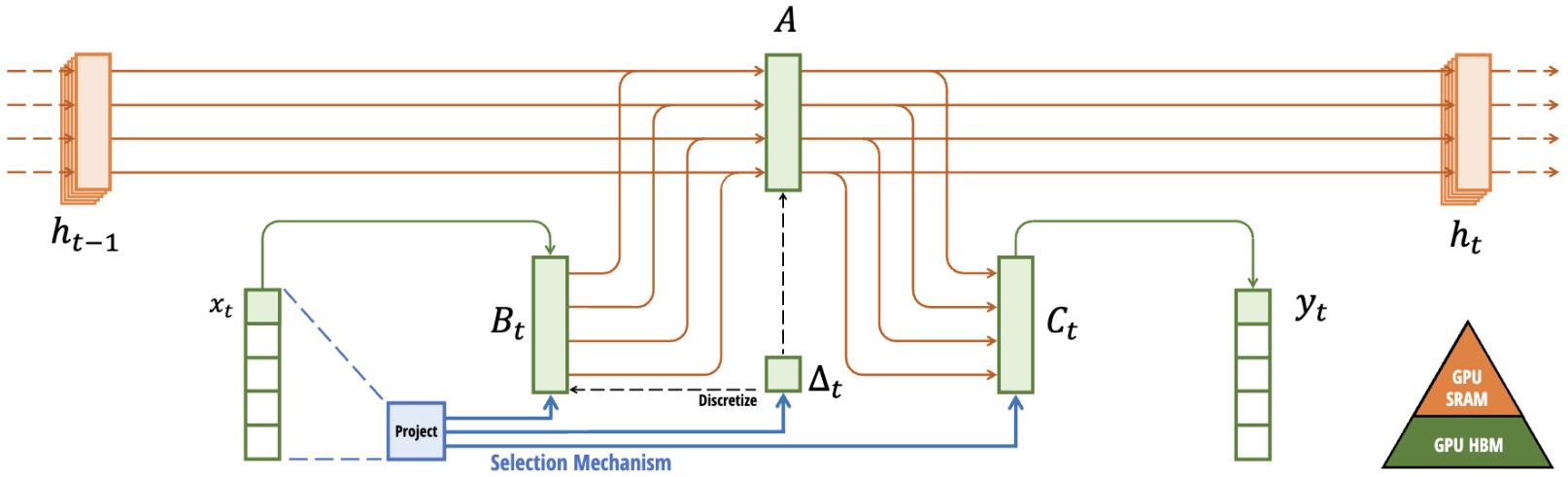
可以使用傅里叶变换实现高效卷积计算

Combining Recurrent, Convolutional, and Continuous-time Models with Linear State-Space Layers. **NeurIPS 2021.**

状态空间模型变种

AI Box

- Mamba
  - 引入基于当前输入的信息选择机制
    - 矩阵 $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ 表示成基于输入 $\boldsymbol{x}_t$ 的非线性函数，对历史信息进行选择过滤



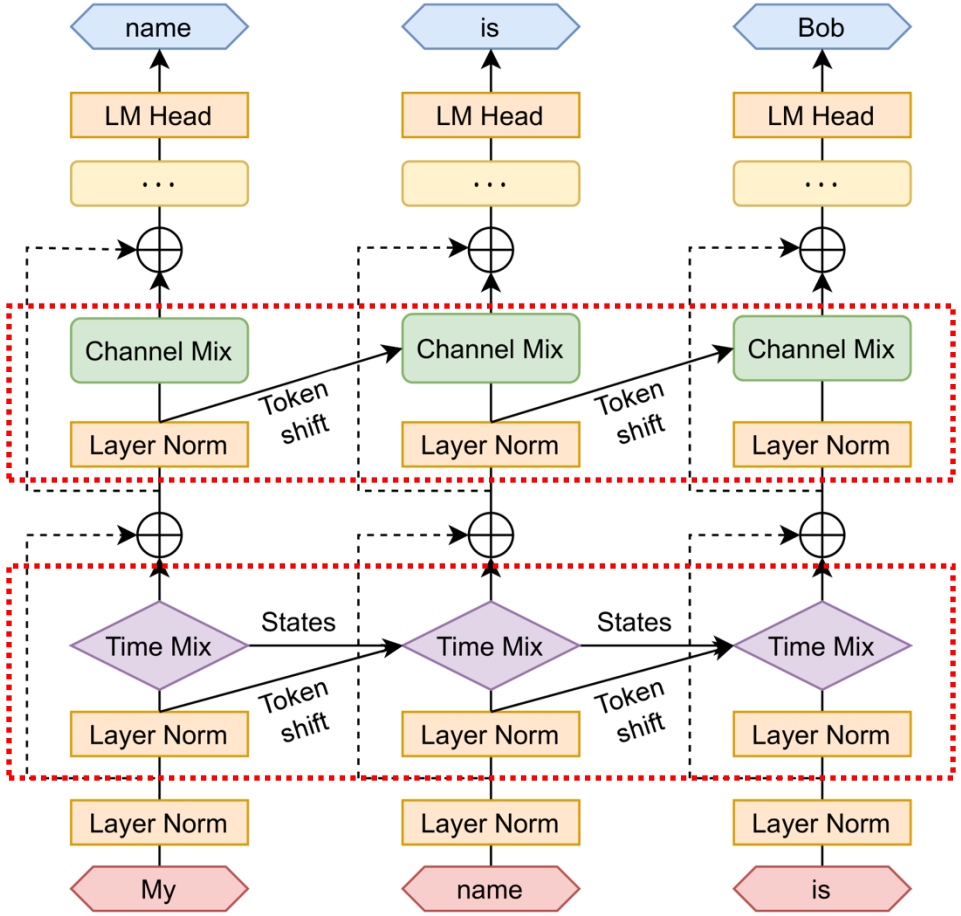
由于在状态计算过程中引入非线性变换，无法利用快速傅里叶变换实现高效卷积计算

Mamba: Linear-Time Sequence Modeling with Selective State Spaces. **COLM 2024.**

状态空间模型变种

- RWKV
  - 词元偏移 (Token Shift)
    - 将当前词元和前一个词元线性插值

      代替当前词元作为输入
  - 时间混合模块 (Time-Mixing)
  - 频道混合模块 (Channel-Mixing)

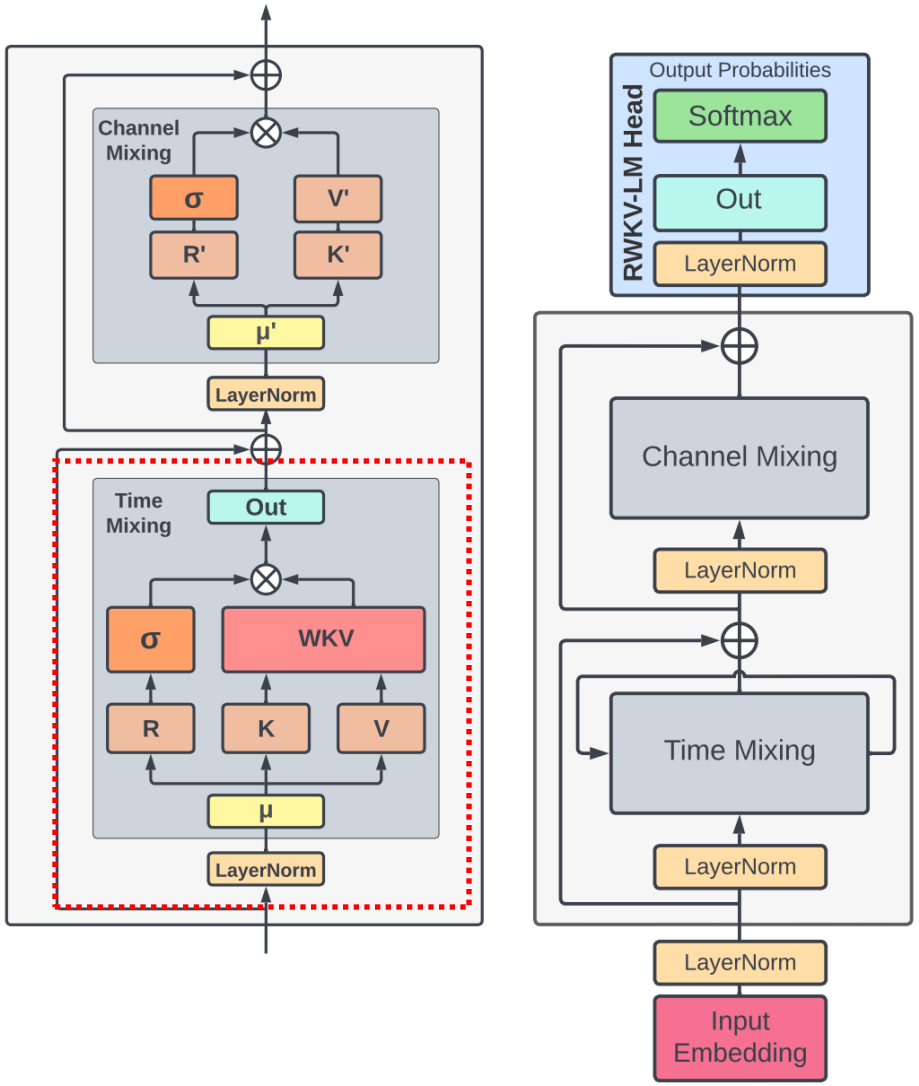


RWKV: Reinventing RNNs for the Transformer Era. **arXiv 2024.**

状态空间模型变种

AI Box

- RWKV
  - 时间混合模块 (Time-Mixing)
    - 代替Transformer中的注意力层
    - 门控RNN，对词元偏移进行更新

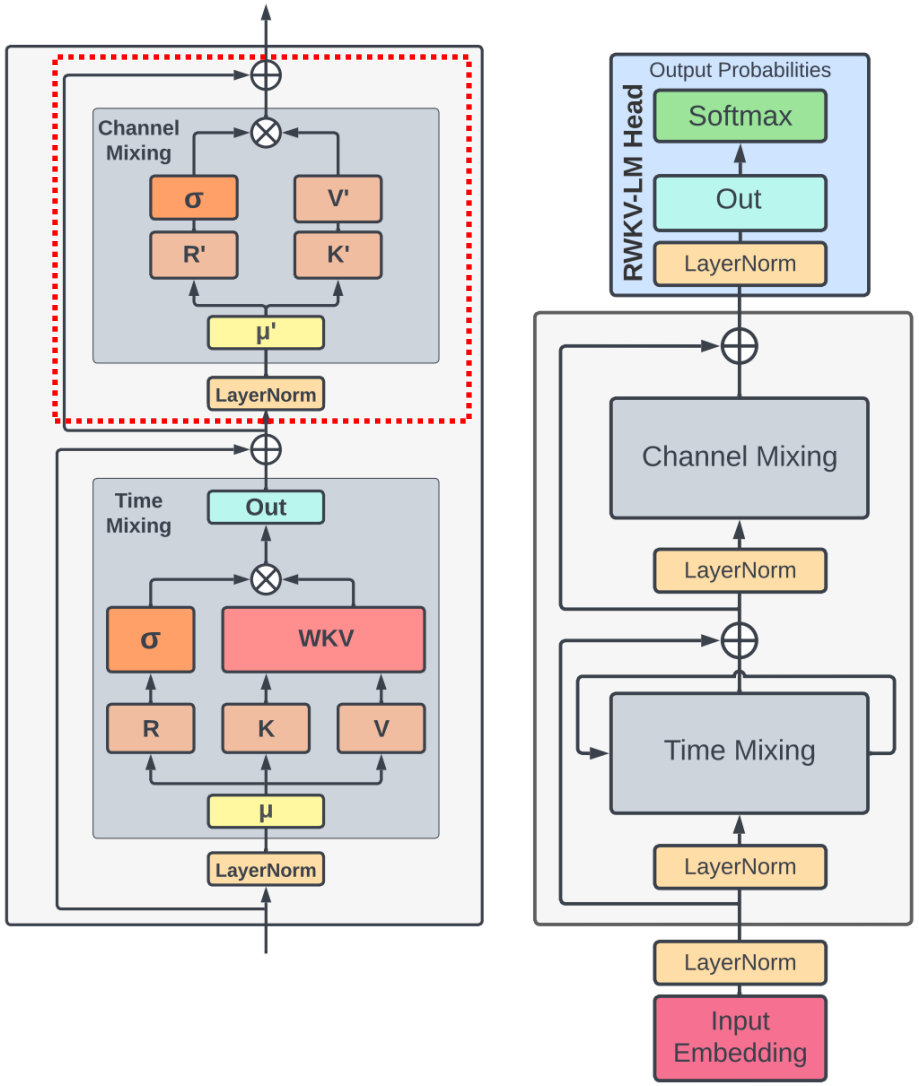


RWKV: Reinventing RNNs for the Transformer Era. **arXiv 2024.**

状态空间模型变种

AI Box

- RWKV
  - 频道混合模块 (Channel-Mixing)
    - 代替Transformer中的前馈网络层
    - 对词元偏移进行映射



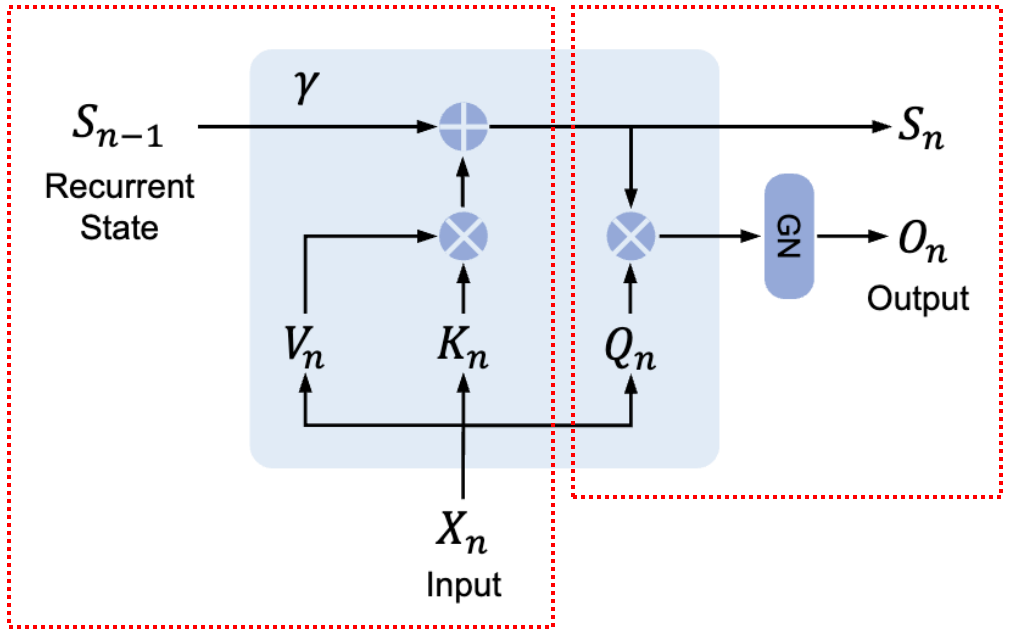
RWKV: Reinventing RNNs for the Transformer Era. **arXiv 2024.**

状态空间模型变种

- RetNet
  - 使用多尺度保留模块 (Multi-scale Retention, MSR) 替换多头注意力

输入词元被映射为 $\boldsymbol{k}, \boldsymbol{q}, \boldsymbol{v}$

$$\boldsymbol{S}_t = \boldsymbol{A}\boldsymbol{S}_{t-1} + \boldsymbol{k}_t^{\mathsf{T}}\boldsymbol{v}_t$$



$$\boldsymbol{o}_t = \boldsymbol{q}_t\boldsymbol{S}_t$$

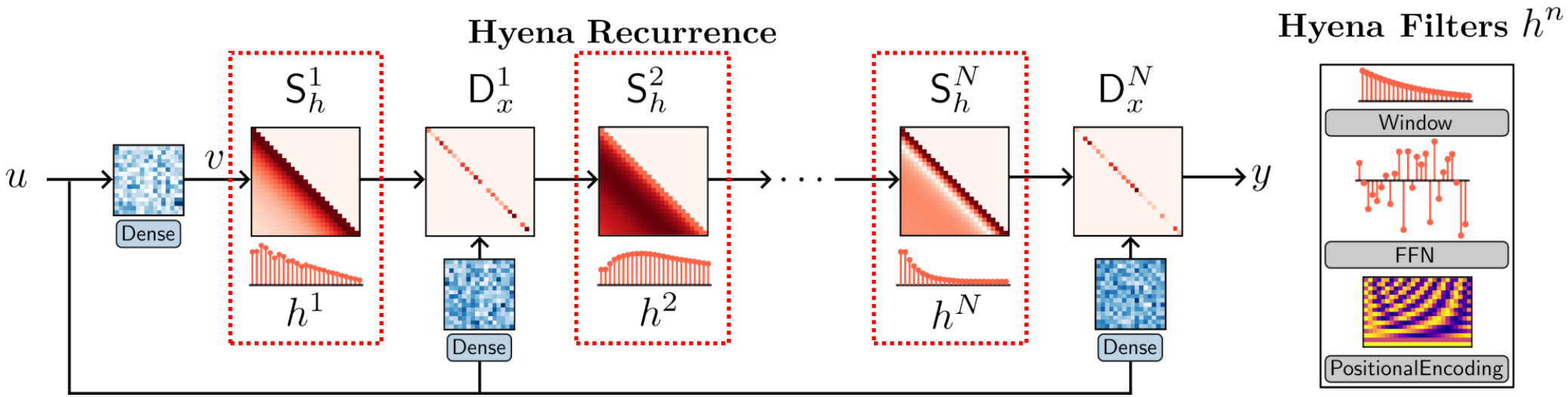
Retentive network: A successor to transformer for large language models. **arXiv 2024.**

状态空间模型变种

AI Box

- Hyena
  - 长卷积模块 (Long Convolution) 替换多头注意力
    - 每层包含 N 个滤波器，每个相对位置索引设置对应的滤波器组成卷积核

$$\boldsymbol{K} = (\boldsymbol{h}(1), \ldots, \boldsymbol{h}(T))$$

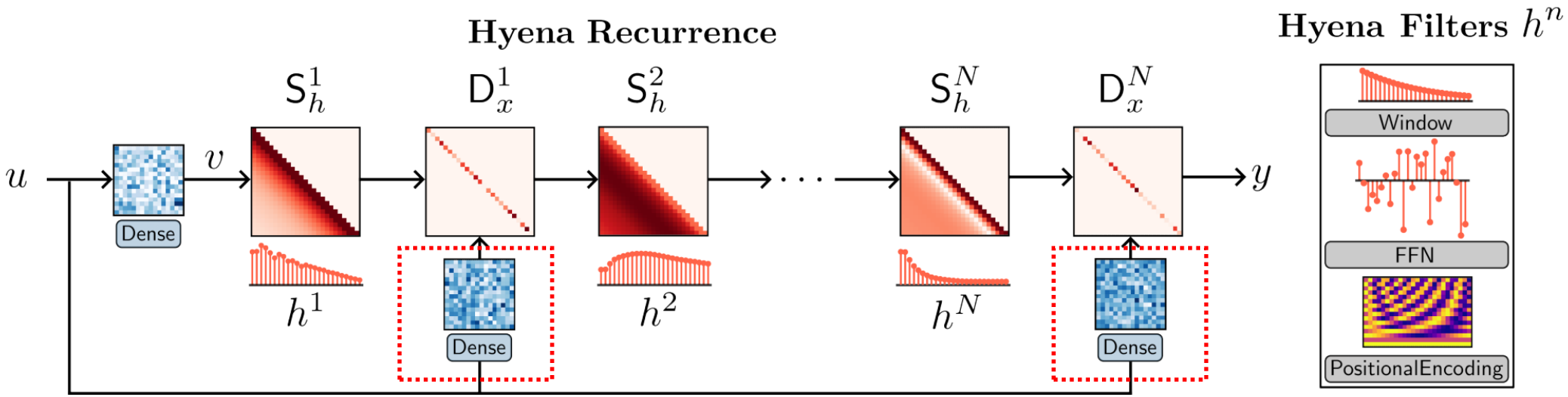


Hyena hierarchy: Towards larger convolutional language models. **ICML 2023.**

状态空间模型变种

AI Box

- Hyena
  - 长卷积模块 (Long Convolution) 替换多头注意力
    - 对于每个卷积核的输出的中间表示 z_t，使用门控函数 $g(t)$进行加权



Hyena hierarchy: Towards larger convolutional language models. **ICML 2023.**

谢谢