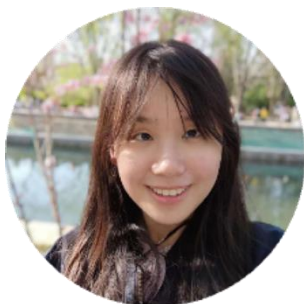


RAT: Bridging RNN Efficiency and Attention Accuracy via Chunk-based Sequence Modeling

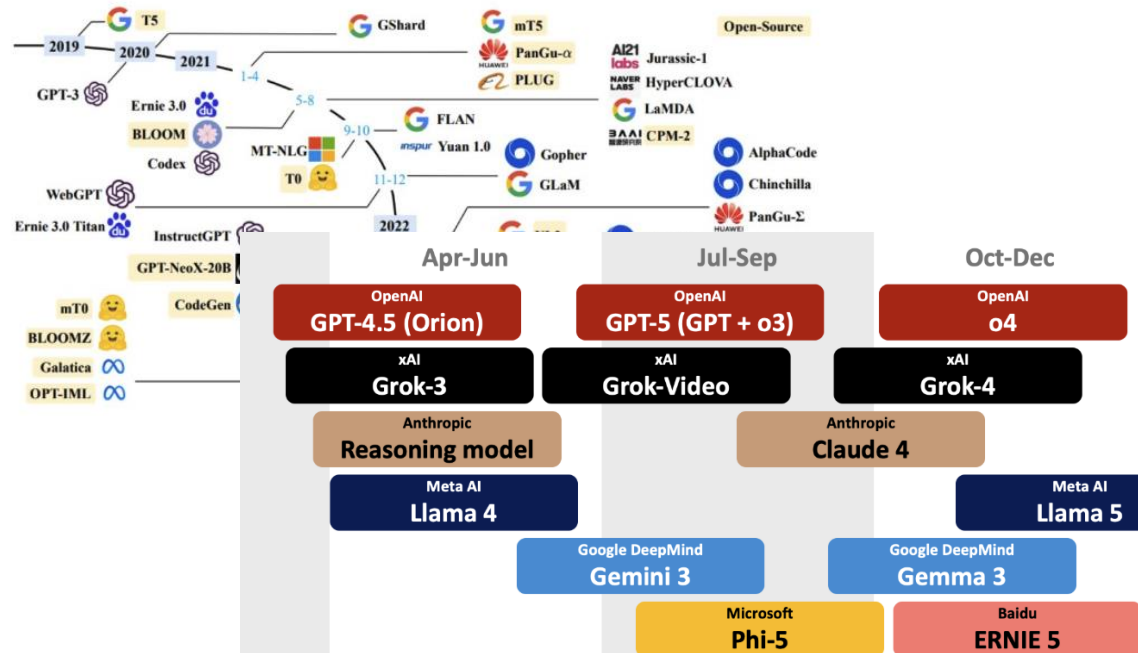


Xiuying Wei, Anunay Yadav, Razvan Pascanu, Caglar Gulcehre

EPFL



Many works on attention and recurrence!



- Large language models emerge very quickly these years, and attention governs their architecture!

There lacks an intermediate design!

Mamba: Linear-Time Sequence Modeling with Selective State Spaces

Albert Gu¹ and Tri Dao²

Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality

Published as a conference paper at ICLR 2025

GATED DELTA NETWORKS:
IMPROVING MAMBA2 WITH DELTA RULE

Songlin Yang^{*}
MIT CSAIL
yangs166@mit.edu

Jan Kautz
NVIDIA
jkautz@nvidia.com

Ali Hatami
NVIDIA
ahatamiz@nvidia.com

ABSTRACT

Linear Transformers have gained attention as efficient alternatives to Transformers, but their performance in retrieval and long-context limited. To address these limitations, recent work has explored two mechanisms: gating for adaptive memory control and the delta update memory modifications. We observe that these mechanisms together—gating enables rapid memory erasure while the delta rule facilitates parallel training algorithm optimized for modern hardware. Our new Gated DeltaNet, consistently surpasses existing models like DeltaNet across multiple benchmarks, including language modeling, sense reasoning, in-context retrieval, length extrapolation, and understanding. We further enhance performance by developing hybrids that combine Gated DeltaNet layers with sliding window attentioners, achieving both improved training efficiency and superior task performance. Code: <https://github.com/NVlabs/GatedDeltaNet>

1 INTRODUCTION

The Transformer architecture has significantly advanced the capabilities of (LLMs), showcasing exceptional performance across a wide range of tasks. This mechanism excels in precise sequence modeling and a processing capabilities of modern GPUs during training. However, the quadratic scaling with sequence length, leading to substantial computational challenges for both training and inference.

the ability to have both efficient parallelization and precise sequence modeling. This paper provides multiple viewpoints connecting the strengths of SSMs and attention.¹

Google DeepMind

2024-3-1

Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models

Soham De¹, Samuel L. Smith¹, Anushan Fernando¹, Aleksandar Botev¹, George Cristian-Muraru¹, Albert Gu², Ruba Haroun¹, Leonard Berrada¹, Yutian Chen¹, Srivatsan Srinivasan¹, Guillaume Desjardins¹, Arnaud Doucet¹, David Budden¹, Yee Whye Teh¹, Razvan Pascanu¹, Nando De Freitas¹ and Caglar Gulcehre¹

¹Equal contributions, ²Google DeepMind, ³Work done while at Google DeepMind

Recurrent neural networks (RNNs) have fast inference and scale efficiently on long sequences, but they are difficult to train and hard to scale. We propose Hawk, an RNN with gated linear recurrences, and Griffin, a hybrid model that mixes gated linear recurrences with local attention. Hawk exceeds the reported performance of Mamba on downstream tasks, while Griffin matches the performance of Llama-2 despite being trained on over 6 times fewer tokens. We also show that Griffin can extrapolate on sequences significantly longer than those seen during training. Our models match the hardware efficiency of Transformers during training, and during inference they have lower latency and significantly higher throughput. We scale Griffin up to 14B parameters, and explain how to shard our models for efficient distributed training.

1. Introduction

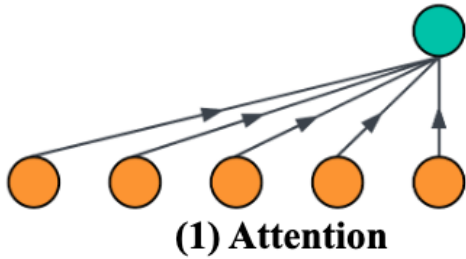
Recurrent neural networks (RNNs) played a central role in the early days of deep learning and NLP research (Elman, 1990; Siegelmann and Sontag, 1991; Hochreiter and Schmidhuber, 1997; Mikolov et al., 2010; Bahdanau et al., 2014; Sutskever et al., 2014), and achieved practical success in many applications, including Google’s first end-to-end machine translation system (Wu et al., 2016). However in recent years, both deep learning and NLP have been dominated by the Transformer architecture (Vaswani et al., 2017), which interleaves multi-layer perceptrons (MLPs) and multi-head attention (MHA). Transformers achieve better performance than RNNs in practice and are also very efficient at utilizing modern hardware (Kaplan et al., 2020). Transformer-based large language models trained on massive datasets collected from the web have achieved remarkable success (Brown et al., 2020; Rae et al., 2021; Hoffmann et al., 2022; Touvron et al., 2023; Achiam et al., 2023; Gemini Team Google, 2023).

Despite their successes, Transformers are difficult to scale efficiently to long sequences due to the quadratic complexity of global attention. Additionally, the linear growth of the Key-Value (KV) cache with the sequence length makes Transformers slow during inference. Although Multi-Query Attention (MQA) (Shazeer, 2019) partially mitigates this issue by reducing the cache size by a constant factor, the cache still grows linearly in sequence length. Recurrent language models present a compelling alternative as they compress the entire sequence into a fixed-sized hidden state which is updated iteratively. However to replace Transformers, new RNN models must demonstrate not only comparable performance at scale but also achieve similar hardware efficiency (Gu et al., 2021a; Mehta et al., 2022; Smith et al., 2022; Oriente et al., 2023b; Dao et al., 2023b; Pali et al., 2023; Gu and Dao, 2023).

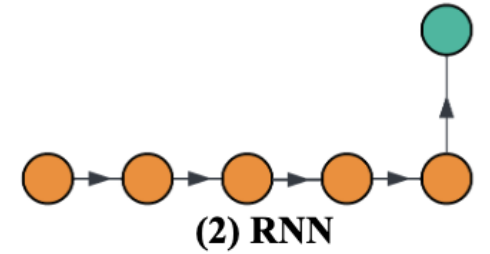
arXiv:2412.06464v3 [cs.CL] 6 Mar 2025

arXiv:2402.19427v1 [cs.LG] 29 Feb 2024

- A wave of recent efforts to revisit recurrent models or propose novel linear recurrent models



Attention vs. Recurrence!



- **Full-token access**

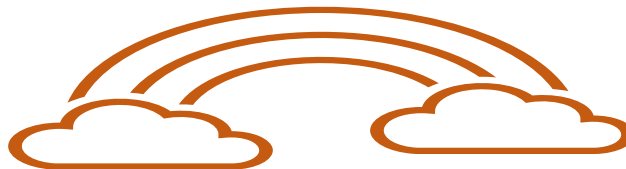
- full-size memory and precise history retrieval
- heavy computation

Strong performance!

- **Full-sequence compression**

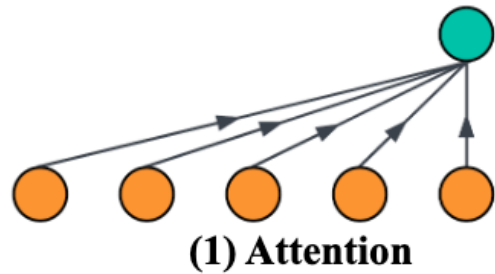
- fixed-size and holistic representation
- degraded memory and limited precise information retrieval
- cheap computation

High Efficiency!

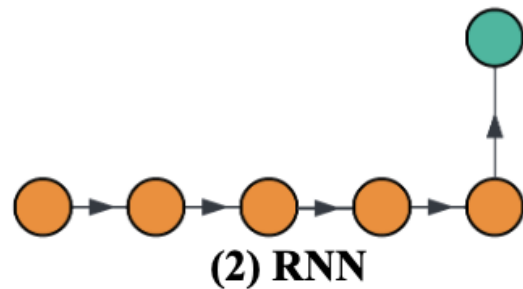


RAT: Chunk-Based Intermediate design

RAT: Chunk-Based Intermediate design

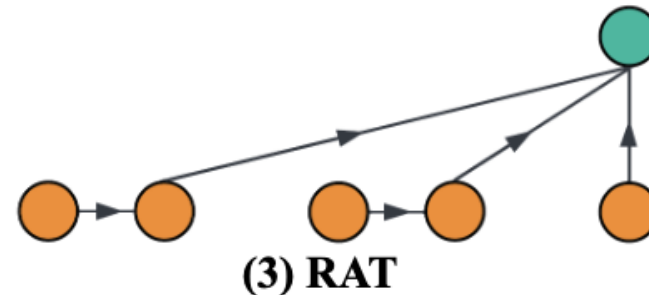


Chunk size as 1



Chunk size as the
sequence length

- Interpret the input as a sequence of shorter chunks
 - Intra-chunk: Recurrence can excel on short sequences.
 - Inter-chunk: Attention has the direct distant access but with reduced computation.
 - Intermediate design by adjusting the chunk size.



Flexible chunk size: mitigate the fixed-size representation
limitation of (2) and the inefficiency of (1)

RAT architecture

- **Attention**

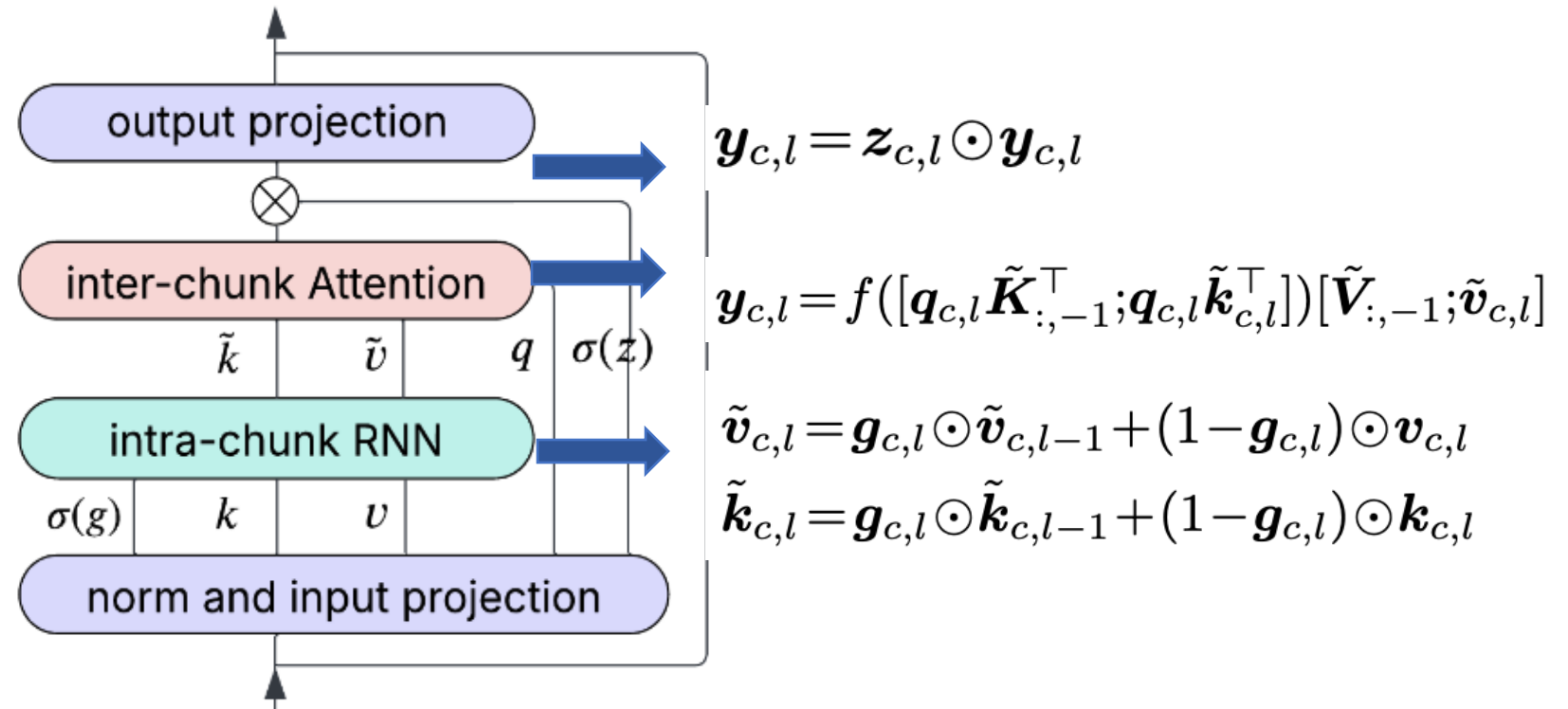
$$\mathbf{y}_t = f(\mathbf{q}_t \mathbf{K}_{:,t}^\top) \mathbf{V}_{:,t}$$

- **Recurrence:** a simple linear recurrence [1, 2] but not limited to this

$$\tilde{\mathbf{v}}_t = \mathbf{g}_t \odot \tilde{\mathbf{v}}_{t-1} + (1 - \mathbf{g}_t) \odot \mathbf{v}_t$$

$$\mathbf{y}_t = \mathbf{z}_t \odot \tilde{\mathbf{v}}_t,$$

- **RAT:** interpret a token t as chunk index and position within a chunk (c, l) .



[1]. Parallelizing linear recurrent neural nets over sequence length.

[2]. Were rnns all we needed?

RAT: scalable and efficient modeling

- **Design details**

- Parameter allocations: decrease from $6D^2$ to $4D^2$.
- Positional encoding: inter-chunk positions and better length generalization.
- Hybrid design with local attention: long-range dependency and local region highlight!

- **Efficiency**

- Reduced FLOPs: $\mathcal{O}(C \cdot D)$ of RAT, $\mathcal{O}(D)$ of recurrence, and $\mathcal{O}(T \cdot D)$ of attention.
- Causal masking in training: online softmax
- Easy impl. without customized kernels: flex attention and parallel scan in training, normal single step update and flash attention in inference
- Compatible with parallelisms

D : model dimension
 T : sequence length
 C : number of chunks
 L : chunk length

Efficiency and accuracy results!

Efficiency

Figure 2: **Latency of the temporal mixing block** (including linear projections) with a model dimension of 2048. (a): full-sequence latency with 200K tokens; (b): generation of 512 tokens at specified positions. We adopt *flash attention* for Attention.

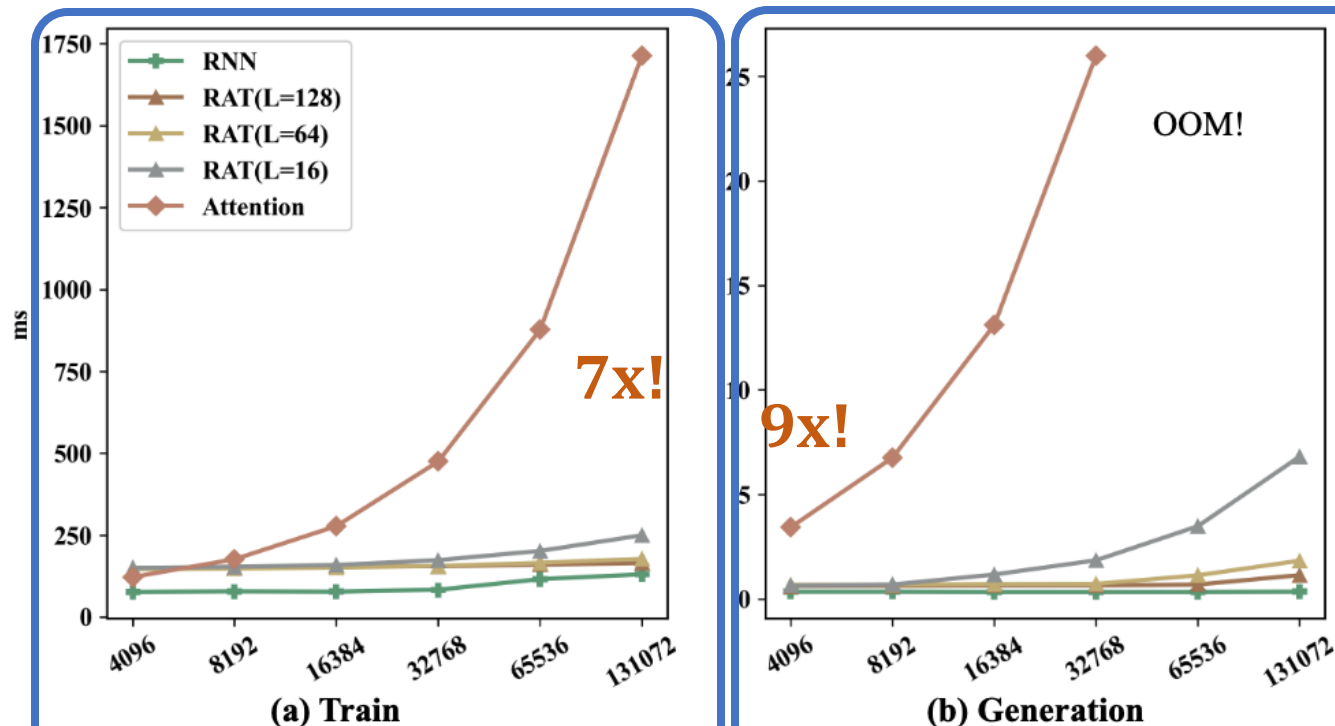


Table 2: **Maximum throughput of full models** (tokens/sec), measured by generating 1024 tokens from a 3072-token prompt. By reducing the KV cache memory and boosting speed, we achieve 10× maximum throughput compared to *flash attention*, and even more on 13B models, as attention suffers from poor GPU utilization at larger scale.

Model	1.3B	7B	13B
RAT(L=16)	31170	10103	5749
Attention	3152	983	534
Ratio	10.2×	10.3×	10.8×

Accuracy

Table 1: Representative results for 1.3B models across pretraining, direct evaluation, and SFT. -SWA denotes interleaving with sliding-window attention (SWA) (window size 1024). Maximum throughput is measured by generating 1024 tokens given a prompt of 3072 tokens on a H100 GPU in GH200 system. See Sec. 4 for details.

Model	Throughput token/sec	Pretrain	Direct Evaluation				SFT	
		Val. PPL	CSR Avg. acc	SQA Avg. F1	Summ Avg. Rouge-L	Code Avg. EditSum	NQA ¹ F1	QMSum Rouge-L
Attention	3052	7.61	56.9	18.2	<u>19.5</u>	<u>23.9</u>	61.3	<u>23.4</u>
RAT(L=16)	31170	7.67	56.7	19.6	20.2	17.4	60.8	23.3
Attention-SWA	4605	<u>7.61</u>	<u>57.1</u>	17.4	19.4	21.7	63.3	<u>23.4</u>
RAT(L=16) -SWA	13582	7.57	58.0	<u>18.8</u>	<u>19.5</u>	28.2	<u>63.2</u>	24.6

- 1.3B model
- 100B web token pretrain
- Commonsense reasoning: short context and general understanding
- Longbench: long context and instruction
- Supervised-finetuning

Takeaways

- Intermediate architecture between recurrence and attention by adjusting the chunk size
 - a single-layer design: trade-off between them.
 - hybrid modelling: greater flexibility with different chunk sizes.
- Memory capacity scales with sequence length with a fixed FLOPs reduction ratio
 - Either classic or advanced recurrence (state space or linear attention models) rely on fixed-size and holistic representations.
 - Partial compression with direct access to prior chunks allows precise retrieval.
- Work with local attention well
 - local attention highlights local computation while RAT focuses more on the long-range dependencies!

Thanks