Improving Recall in Recurrent LMs

Omar Abul-Hassan[*]
Department of Mathematics
Stanford University
Stanford, CA, USA

Abstract

Recurrent large language models (LLMs) offer significant memory and computational efficiencies over Transformer architectures. However, they face challenges in recalling information from long contexts, leading to degraded in-context learning (ICL) performance. This project proposes three enhancements to the Just-Read-Twice (JRT) framework [1]: dynamic prefix-length adjustment in JRT-RNN, exploration of diverse feature map functions within Prefix Linear Attention (PLA), and the implementation of hierarchical JRT-Prompting strategies. Through these modifications, we aim to improve recall accuracy and computational efficiency, bridging the performance gap between recurrent models and Transformers in recall-intensive tasks.

1 Introduction

Recurrent large language models (LLMs), such as Mamba and RWKV, have emerged as efficient alternatives to Transformer-based models. They maintain constant memory usage during inference and exhibit lower computational costs for processing long sequences [16, 8]. These properties are attractive for real-world applications that demand efficient and scalable language modeling over extended contexts. However, a crucial limitation of current recurrent models lies in their struggle to recall information accurately from these long contexts, which can diminish their performance in in-context learning (ICL) tasks.

The Just-Read-Twice (JRT) framework [1] attempts to address this recall bottleneck by allowing the model to re-read context segments, improving the alignment of internal representations and thereby enhancing recall. Yet, even with JRT, certain limitations remain: fixed prefix lengths may not adapt well to diverse context complexities, limited feature representations in Prefix Linear Attention (PLA) may not capture all relevant relationships, and flat prompting strategies may fail to leverage hierarchical structure in the data.

In this work, we propose three enhancements to the JRT framework:

1. **Dynamic Prefix-Length Adjustment:** Adapt prefix lengths in JRT-RNN based on context complexity.
2. **Feature Map Exploration in PLA:** Introduce diverse activation and kernel approximations to improve the quality of attention.
3. **Hierarchical JRT-Prompting:** Incorporate structural cues from hierarchical prompts to emphasize critical context segments.

Our experiments, conducted on recall-intensive benchmarks, demonstrate consistent gains in recall accuracy and confirm that these strategies can bridge some of the performance gap between recurrent

[*] omarah@stanford.edu

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

models and Transformers. While the improvements are modest, they highlight the potential of flexible memory allocation, richer feature maps, and hierarchical prompt structuring in enhancing the recall capabilities of recurrent LLMs.

2 Related Work

The recent literature has shown growing interest in improving long-context processing and recall in LLMs. Transformer-based models [15, 7, 14] have historically demonstrated strong recall capabilities due to their direct attention mechanisms, but their memory and computational costs scale quadratically with sequence length. Various methods—such as sparse attention [4], memory-augmented Transformers [13], and retrieval-based augmentation [10, 2]—have been proposed to mitigate these costs and improve long-range recall.

Recurrent models [12, 8, 16] offer a different route by using fixed-size hidden states that do not blow up with sequence length. However, classical recurrent architectures like LSTMs and GRUs [5] often struggle with long-range dependencies, and even recent large recurrent LMs have yet to match the recall performance of Transformers. The JRT framework [1] addresses part of this gap by re-reading the input, thereby giving recurrent models multiple passes over the data. Nevertheless, JRT-based recurrent LMs still lag behind their Transformer counterparts in handling highly complex, long-range dependencies.

Our work builds on JRT by investigating three complementary directions. First, we dynamically adjust prefix lengths to optimally allocate memory resources, which is reminiscent of adaptive context allocation strategies explored in hierarchical models [11, 3]. Second, we diversify PLA feature maps, drawing inspiration from kernel-based approximations of the softmax function [9] and learned representations in attention [6]. Third, we introduce hierarchical prompting, a concept related to hierarchical encodings found in structured datasets and multi-level attention mechanisms [13]. By combining these insights, we aim to enhance the recall capabilities of recurrent LLMs and approach Transformer-level performance on challenging, long-context tasks.

3 Problem Description

Efficient recurrent LLMs operate under strict memory and computation constraints. They process long sequences in a fixed order and maintain a hidden state that ideally captures essential information. However, these hidden states cannot dynamically expand to store all relevant details, and the model lacks the direct attention mechanisms that Transformers use to selectively recall distant tokens.

The JRT framework tries to alleviate this by reading the context twice, allowing the model to refine its internal representation after a first pass. Despite notable improvements, three core challenges persist:

- **Fixed Prefix-Length JRT-RNN:** A static prefix length may not optimally balance recall accuracy and efficiency. More complex contexts might require longer prefixes, while simpler ones might do well with shorter prefixes.
- **Single Feature Map in PLA:** PLA currently uses a single feature map function. Relying on one feature map limits the model's ability to align keys and queries under diverse contextual patterns, potentially restricting recall accuracy.
- **Flat JRT-Prompting:** Flat repetition of the entire prompt does not leverage hierarchical structure. Important segments require more emphasis for recall, while others may not need multiple exposures.

Our aim is to address these issues simultaneously. By tailoring prefix lengths dynamically, exploring diverse feature map functions, and leveraging hierarchical prompting, we hope to push recurrent LLMs closer to the performance envelope of Transformer-based models.

4 Methods

4.1 Dynamic Prefix-Length Adjustment

Objective: Adapt prefix lengths to match context complexity for improved recall.

Methodology:

- **Entropy-Based Complexity Assessment:** Measure the entropy of token distributions. High-entropy passages suggest complex information patterns, justifying longer prefixes.
- **Heuristic Adjustment:** For high-entropy contexts, increase the prefix length by a preset ratio. For low-entropy contexts, maintain or reduce the prefix length. This heuristic ensures a balanced allocation of memory and attention resources.
- **Integration with PLA:** Modify the `PrefixLinearAttention` class to handle variable prefix lengths. The model updates internal states seamlessly when prefix-length changes are triggered.

4.2 Feature Map Exploration in PLA

Objective: Enhance PLA by experimenting with multiple feature map functions to better approximate attention patterns.

Methodology:

- **Non-Linear Activations:** Investigate GELU and ReLU-based feature maps to capture non-linear relationships.
- **Kernel Approximations:** Integrate polynomial and Gaussian kernel approximations to better mimic softmax-based attention.
- **Learnable MLP Layers:** Insert small MLPs to learn adaptive feature maps, enabling the model to tailor query-key alignment dynamically.

4.3 Hierarchical JRT-Prompting

Objective: Exploit the hierarchical structure of data for better selective recall.

Methodology:

- **Multi-Level Segmentation:** Divide the input into hierarchical chunks (e.g., paragraphs and sentences).
- **Layered Repetition:** Repeat critical top-level segments (e.g., paragraphs) less frequently than lower-level segments (e.g., sentences), or vice versa, depending on the nature of the data.
- **Overlapping Windows:** Introduce partial overlaps to maintain continuity and ensure crucial context boundaries are emphasized.
- **Seamless Integration:** This hierarchical repetition plugs directly into the existing JRT-Prompt pipeline without altering fundamental architecture components.

5 Experiments

5.1 Evaluation Setup

Datasets: We evaluated on recall-intensive ICL benchmarks, including SWDE, FDA, SQuADv2, Natural Questions, TriviaQA, and Drop. These datasets challenge the model's ability to retrieve and utilize relevant information from long contexts.

Models and Baselines:

- **Recurrent LLMs:** We tested with 2 pretrained recurrent LLMs - Mamba and Based, both at 1B parameters.
- **Transformer Baselines:** We used Llama-based Transformers (1.3B) [14] as strong baselines with known high recall performance.

Metrics:

- **Recall Accuracy:** Percentage of correctly recalled answers.
- **Perplexity:** Measures general language modeling quality.
- **Computational Efficiency:** Inference time and memory usage.
- **Throughput:** Tokens processed per second.

5.2 Results

5.2.1 Dynamic Prefix-Length Adjustment

Table 1 compares a JRT-RNN model with fixed prefix length against one using dynamic prefix-length adjustment. The dynamic approach improves recall accuracy by about 2.1% with a minor increase in inference time.

Table 1: Impact of Dynamic Prefix-Length Adjustment on Recall Accuracy and Efficiency

Model	**Recall Accuracy (%)**	**Inference Time (ms)**
JRT-RNN (Fixed Prefix)	46.1	150
JRT-RNN (Dynamic Prefix)	48.2	158

5.2.2 Feature Map Exploration in PLA

As shown in Table 2, introducing non-linear activations, kernel approximations, and learned mappings improved recall. The learnable MLP-based feature map achieved the highest recall accuracy of about 50.4%.

Table 2: Effect of Different Feature Maps on Recall Accuracy

Feature Map	**Recall Accuracy (%)**	**Perplexity**
Linear	48.3	35.2
GELU	49.7	34.9
ReLU	47.5	35.6
Gaussian Kernel	49.0	35.1
Learnable MLP	50.4	34.7

5.2.3 Hierarchical JRT-Prompting

Table 3 demonstrates that hierarchical prompting confers a modest improvement in recall accuracy (2%) with a small increase in inference time.

Table 3: Comparison of Hierarchical vs. Flat JRT-Prompting

Prompting Strategy	**Recall Accuracy (%)**	**Inference Time (ms)**
Flat JRT-Prompting	49.2	155
Hierarchical JRT-Prompting	51.3	161

5.3 Discussion

These enhancements yield incremental yet consistent improvements in recall accuracy and highlight new avenues for optimizing recurrent LLMs. Although Transformers still hold an edge in recall performance, our results suggest that careful tuning of prefix lengths, richer feature maps, and hierarchical prompt structuring can bring recurrent models closer to parity, without sacrificing their inherent efficiency advantages.

6 Conclusion and Discussions

In this work, we introduced dynamic prefix-length adjustment, feature map exploration in PLA, and hierarchical JRT-Prompting to enhance the recall capabilities of recurrent LLMs within the JRT framework. Our experiments show modest gains in recall accuracy and stability, confirming that these strategies can help recurrent models better handle the complexities of long contexts.

Future work includes:

- **Adaptive RL-Based Prefix Lengths:** Train an RL agent to dynamically optimize prefix lengths beyond heuristic rules.
- **More Complex Feature Maps:** Explore multi-head or hybrid feature maps that blend kernel approximations and learned transformations.
- **Refined Hierarchical Prompts:** Experiment with more sophisticated segmentation strategies and learned hierarchies to further improve recall quality.

By continuing to refine these approaches, we aim to further close the gap between recurrent and Transformer-based models in recall-intensive scenarios, ultimately enabling more efficient and powerful long-context language modeling.

References

[1] Simran Arora et al. *Just read twice: closing the recall gap for recurrent language models*. 2024. arXiv: 2407.05483 [cs.CL]. URL: https://arxiv.org/abs/2407.05483.

[2] Sebastian Borgeaud et al. *Improving language models by retrieving from trillions of tokens*. 2022. arXiv: 2112.04426 [cs.CL]. URL: https://arxiv.org/abs/2112.04426.

[3] Devichand Budagam et al. *Hierarchical Prompting Taxonomy: A Universal Evaluation Framework for Large Language Models*. 2024. arXiv: 2406.12644 [cs.CL]. URL: https://arxiv.org/abs/2406.12644.

[4] Rewon Child et al. *Generating Long Sequences with Sparse Transformers*. 2019. arXiv: 1904.10509 [cs.LG]. URL: https://arxiv.org/abs/1904.10509.

[5] Kyunghyun Cho et al. *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. 2014. arXiv: 1409.1259 [cs.CL]. URL: https://arxiv.org/abs/1409.1259.

[6] Krzysztof Choromanski et al. *Rethinking Attention with Performers*. 2022. arXiv: 2009.14794 [cs.LG]. URL: https://arxiv.org/abs/2009.14794.

[7] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[8] Albert Gu and Tri Dao. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. 2024. arXiv: 2312.00752 [cs.LG]. URL: https://arxiv.org/abs/2312.00752.

[9] Angelos Katharopoulos et al. *Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention*. 2020. arXiv: 2006.16236 [cs.LG]. URL: https://arxiv.org/abs/2006.16236.

[10] Urvashi Khandelwal et al. *Generalization through Memorization: Nearest Neighbor Language Models*. 2020. arXiv: 1911.00172 [cs.CL]. URL: https://arxiv.org/abs/1911.00172.

[11] Yang Liu and Mirella Lapata. *Hierarchical Transformers for Multi-Document Summarization*. 2019. arXiv: 1905.13164 [cs.CL]. URL: https://arxiv.org/abs/1905.13164.

[12] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. *Regularizing and Optimizing LSTM Language Models*. 2017. arXiv: 1708.02182 [cs.CL]. URL: https://arxiv.org/abs/1708.02182.

[13] Jack W. Rae et al. *Compressive Transformers for Long-Range Sequence Modelling*. 2019. arXiv: 1911.05507 [cs.LG]. URL: https://arxiv.org/abs/1911.05507.

[14] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[15] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[16] Hanwei Zhang et al. *A Survey on Visual Mamba*. 2024. arXiv: 2404.15956 [cs.CV]. URL: https://arxiv.org/abs/2404.15956.

Appendix

Additional Experiments and Observations

Extended Hyperparameter Ablations: Beyond the experiments reported in the main text, we explored varying prefix-length increments (5%, 20%) under different entropy thresholds. Larger increments did not yield significant gains in recall-related accuracy/metrics.

Hierarchical Prompting Sensitivity: We performed a sensitivity analysis by varying the overlap in hierarchical segments. Overlaps of 10% to 30% showed that a moderate overlap (around 20%) generally struck the best balance between continuity and redundancy.