20th April 2023

TARM: Token Averaging Recurrent Memory Transformer

Paper presented as part of the presentation 1 :

```
@inproceedings{bulatovrecurrent,
  title={Recurrent Memory Transformer},
  author={Bulatov, Aydar and Kuratov, Yuri and Burtsev, Mikhail},
  booktitle={Advances in Neural Information Processing Systems (NeurIPS)},
  year={2022}
}
```

Course Instructor

Prof. C. Krishna Mohan

Presenter

Rahul Vigneswaran K*

CS23MTECH02002

Assigned TA

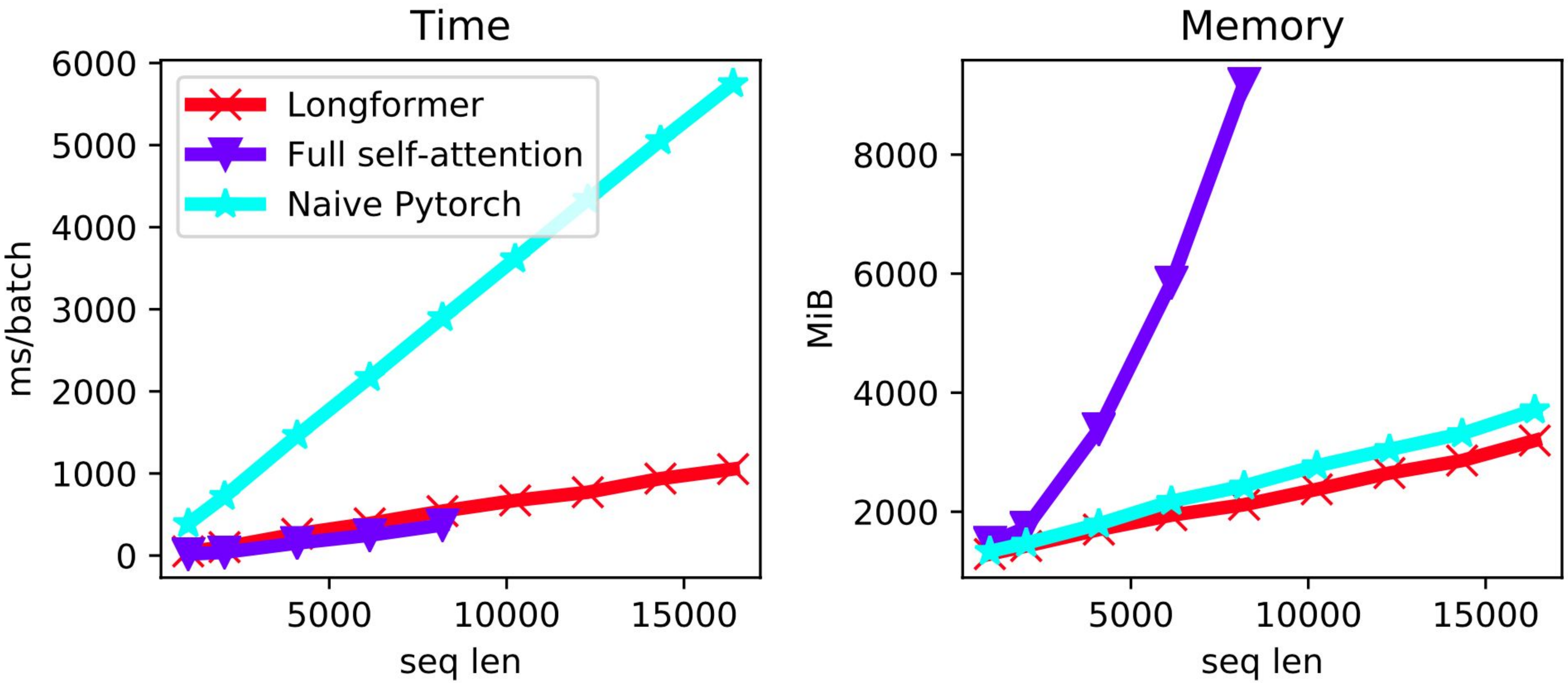
Peketi Divya

*Presented as part of the coursework for Visual Computing (CS6450)

Table of Contents

Introduction: What problem are we trying to solve?

Introduction: What problem are we trying to solve?



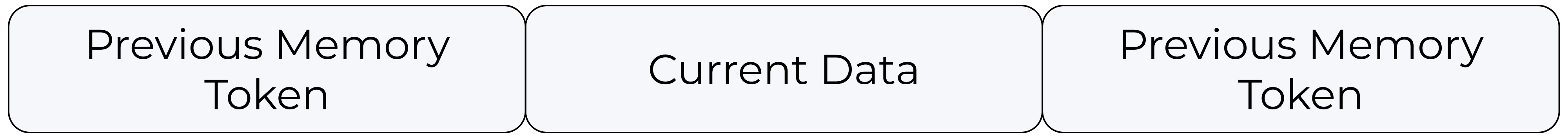
Beltagy, Iz, Matthew E. Peters, and Arman Cohan. "Longformer: The Long-Document Transformer." (Arxiv 2020)

Methodology

Current Data

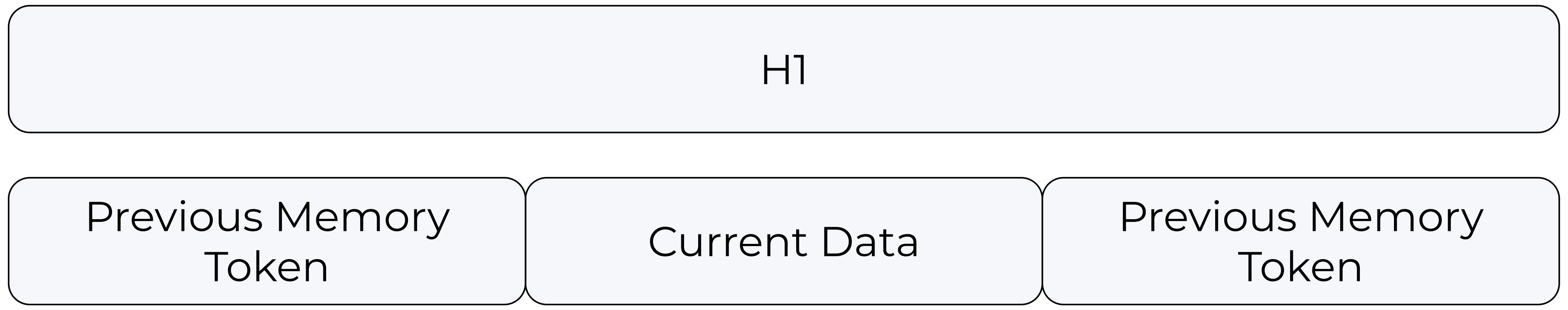
Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology

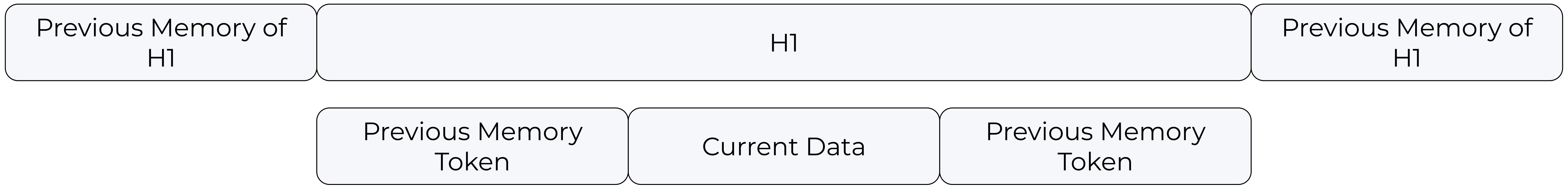


Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology

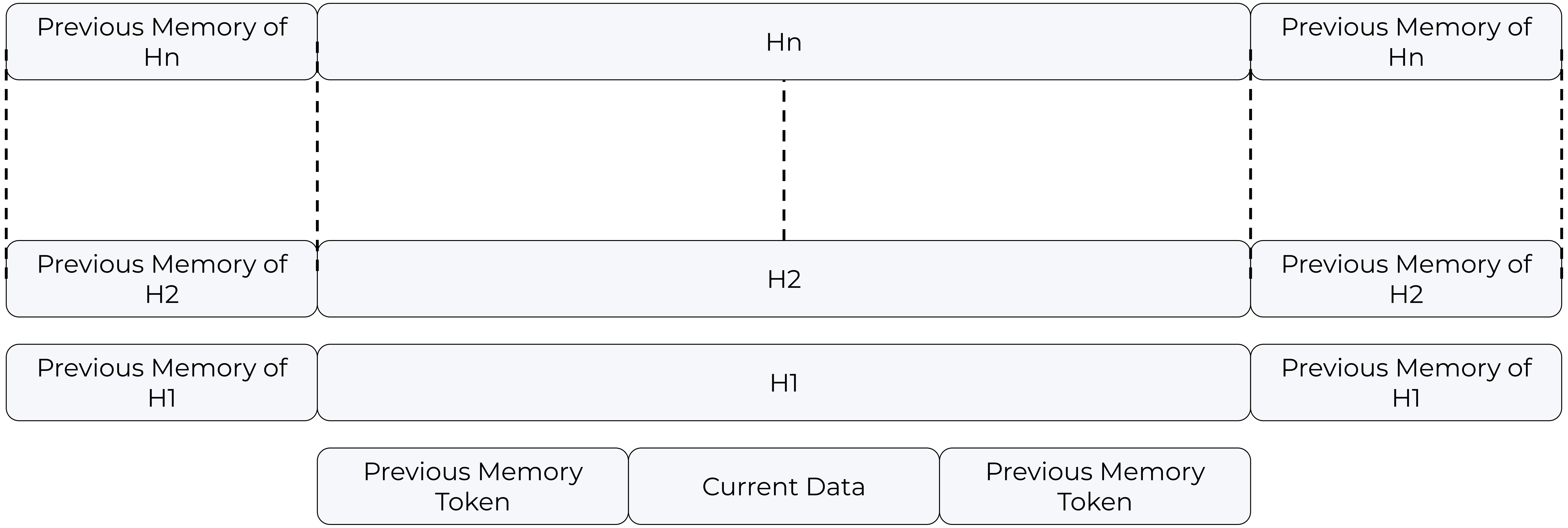


Methodology



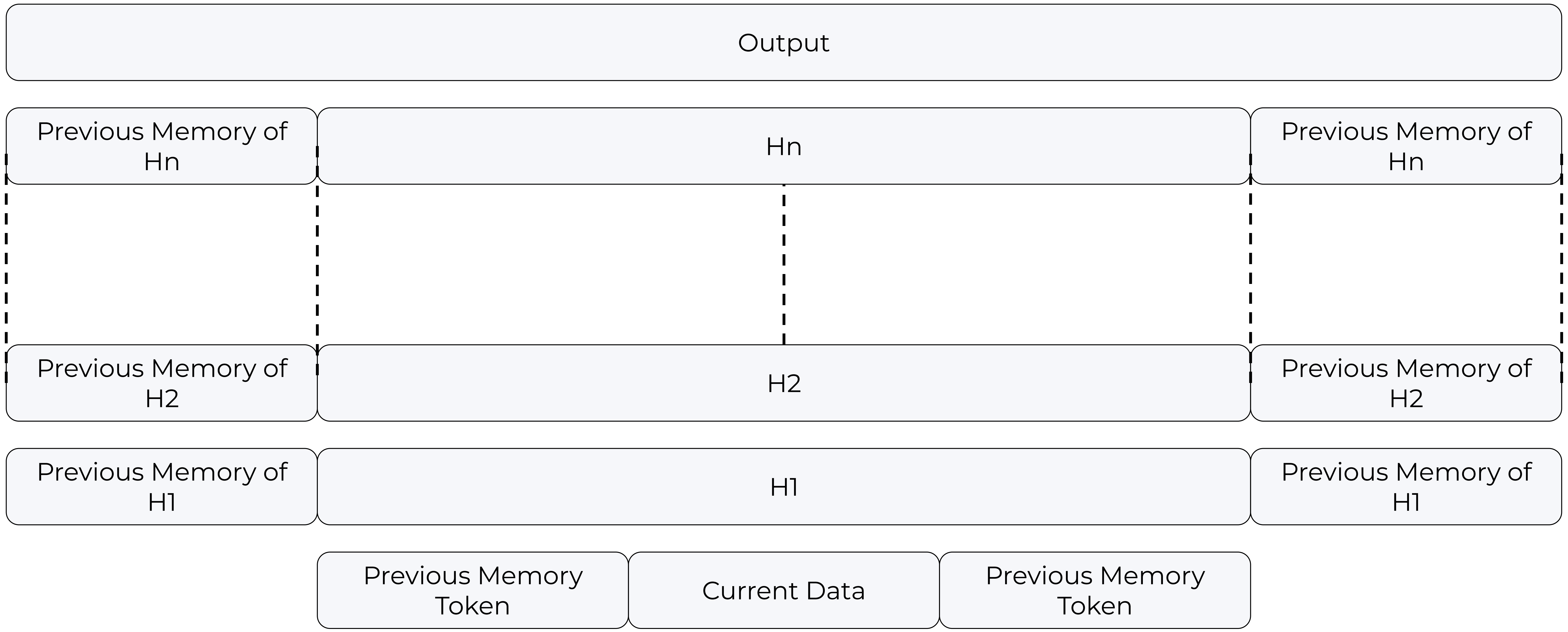
Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology



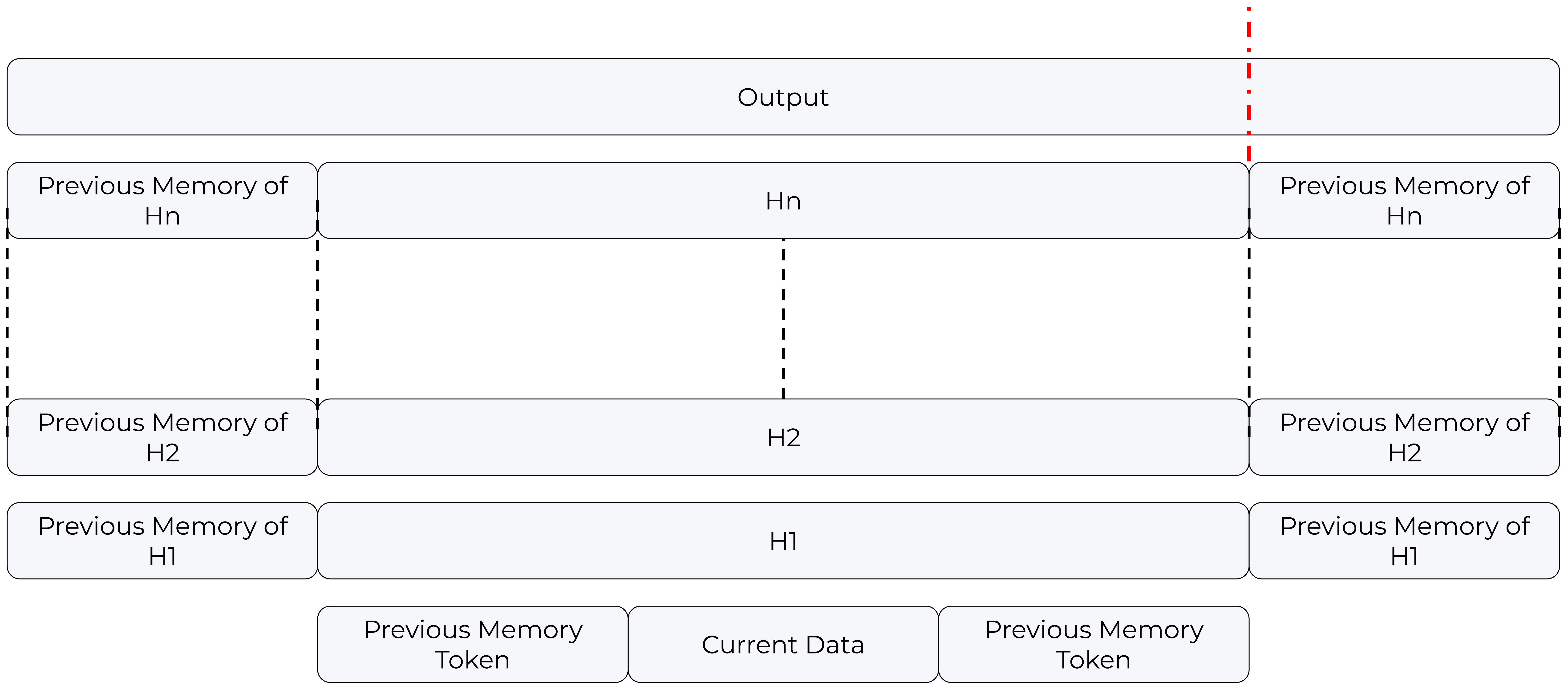
Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology



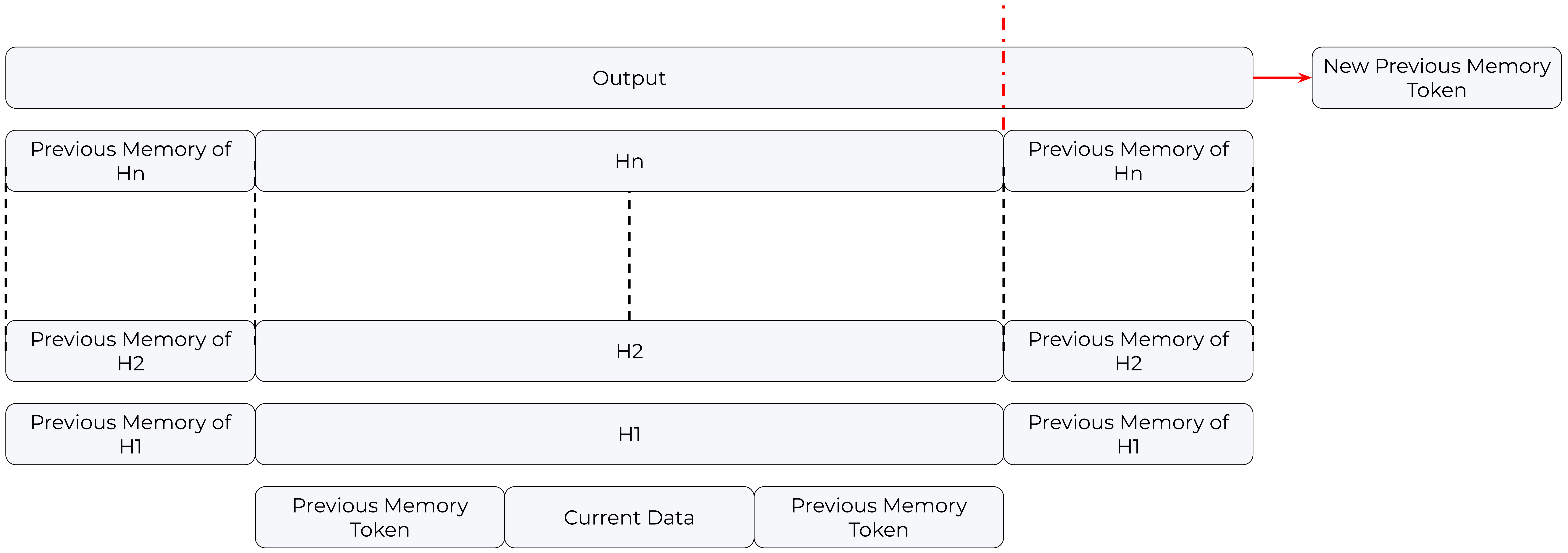
Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology



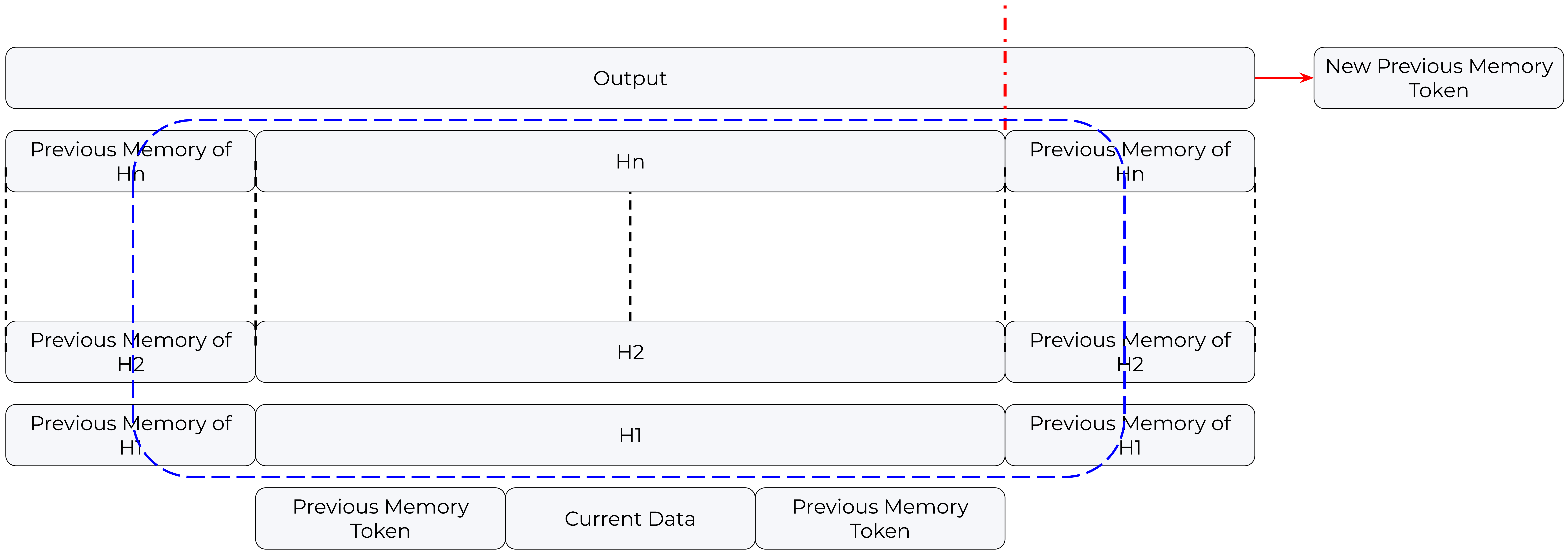
Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology



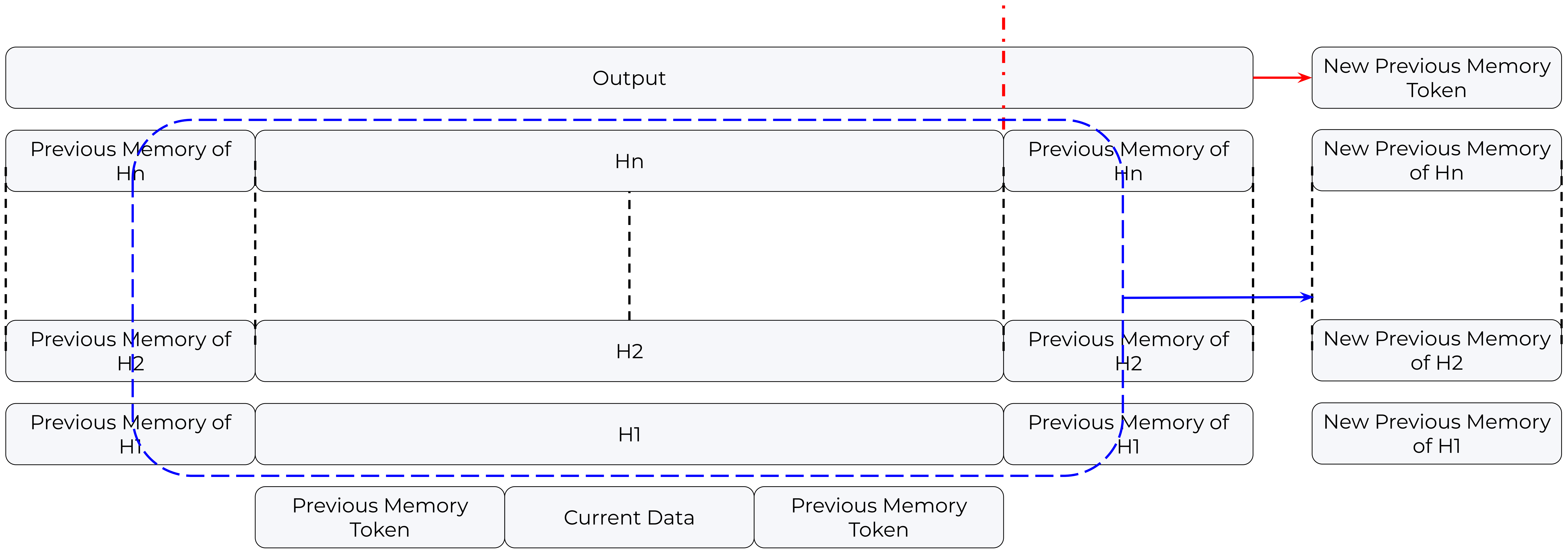
Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology



Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology



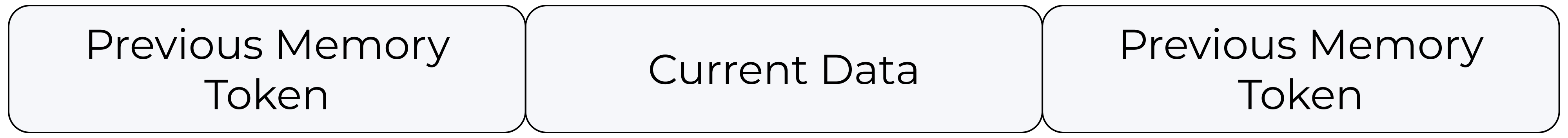
Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology

Current Data

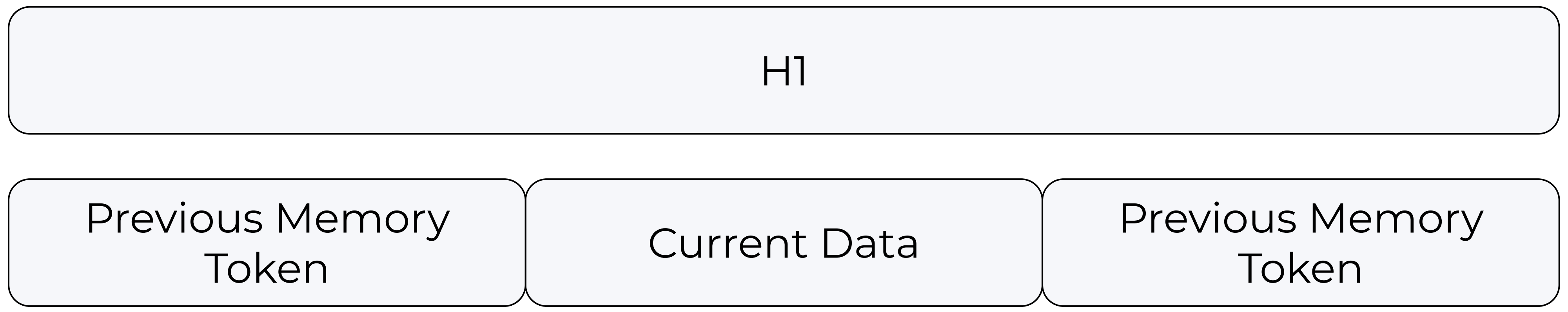
Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology



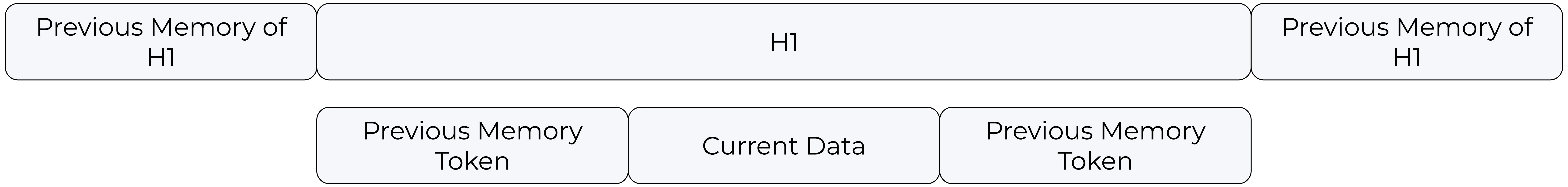
Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology



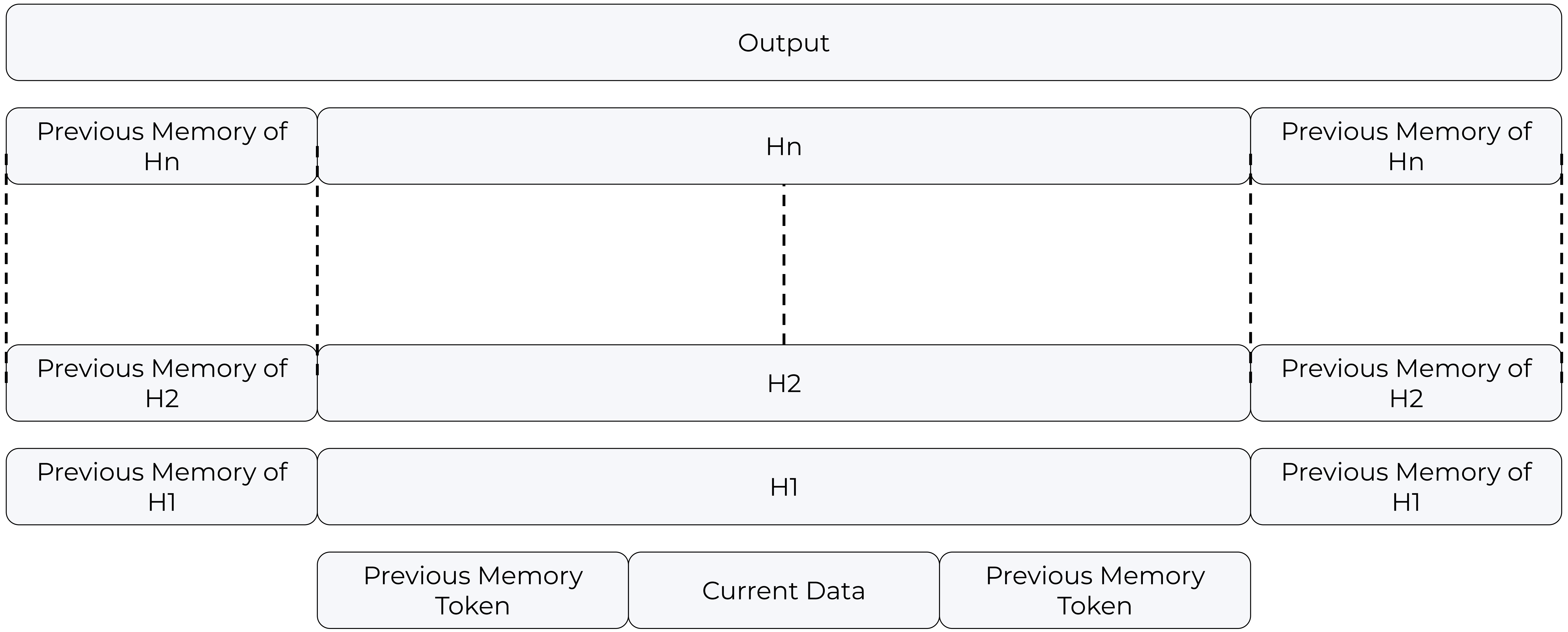
Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology



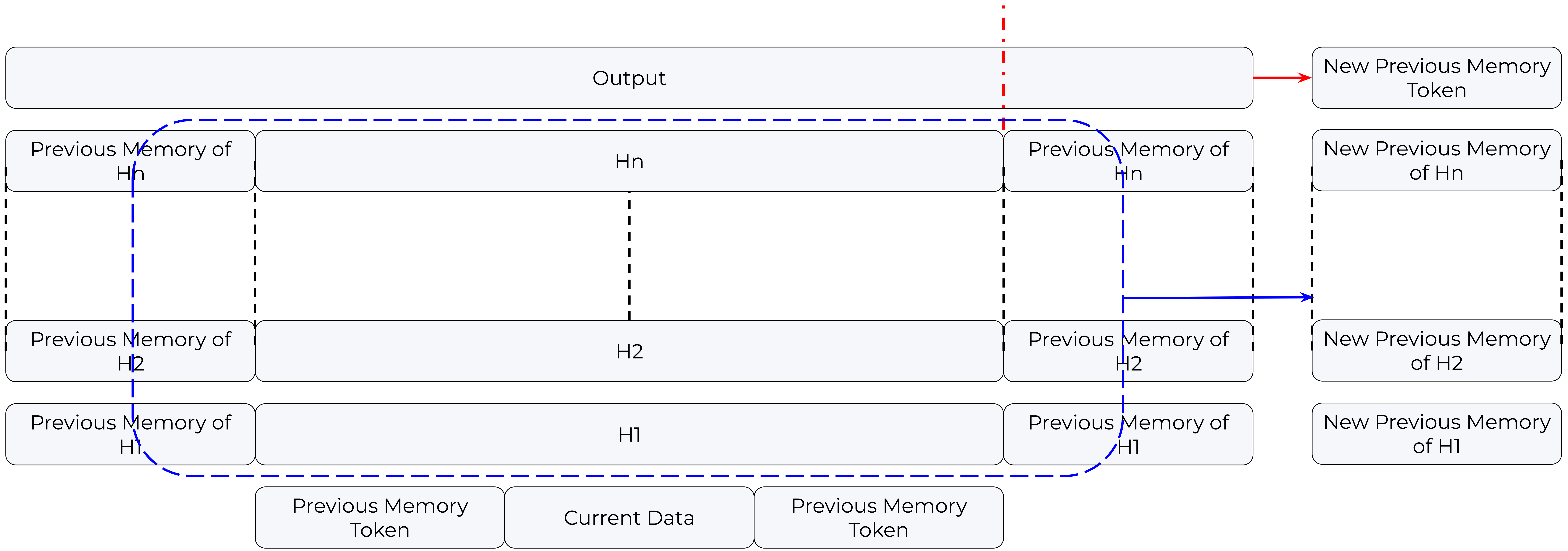
Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology



Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Methodology



Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Implementation details

- Language Modelling Task
  - Model
    - Transformer
      - 16-layer
      - 10 heads
      - 410 in/out neuron count
      - 2100 intermediate neuron count
  - Batch size - 60
  - Segment size - 160
  - Optimizer
    - Adam
      - Learning Rate Schedule
        - Linear schedule learning rate starting from 0.00025 for 200,000 steps

- Quadratic Equations Task
  - Model
    - Transformer
      - 6-layer
      - 6 heads
      - 128 in/out neuron count
      - 256 intermediate in/out neuron count
  - Batch size - 32
  - Segment size - 180
  - Optimizer
    - Adam
      - Learning Rate Schedule
        - Constant learning rate 1e-4 with reduction on plateau with decay factor of 0.5 for 600,000 steps

Due to compute constraints, we were able to run **TARM (Ours)** only for 120,000 steps, so we compare the corresponding best RMT result for the same number of steps.

Bulatov, Aydar, Yuri Kuratov, and Mikhail Burtsev. "Recurrent Memory Transformer." Advances in Neural Information Processing Systems. (NeurIPS 2022)

Reproduced Results

Task	Dataset	Metric	RMT (From Paper)	RMT (Reproduced)
LM*	WT-103	PPL ↓	**23.99**	24.291 (↑0.301)
Step-by-Step	Quadratic Equations	Accuracy ↑	99.8	**99.9** (↑0.1)

Table 1. Comparison of reproduced RMT results on various tasks. The best results are shown in bold, and the second-best results are underlined. The symbol ↑ and ↓ indicate the improvement and degradation of the result compared to the reproduced RMT model. LM* denotes language modeling task. PPL denotes perplexity.

Reproduced Results

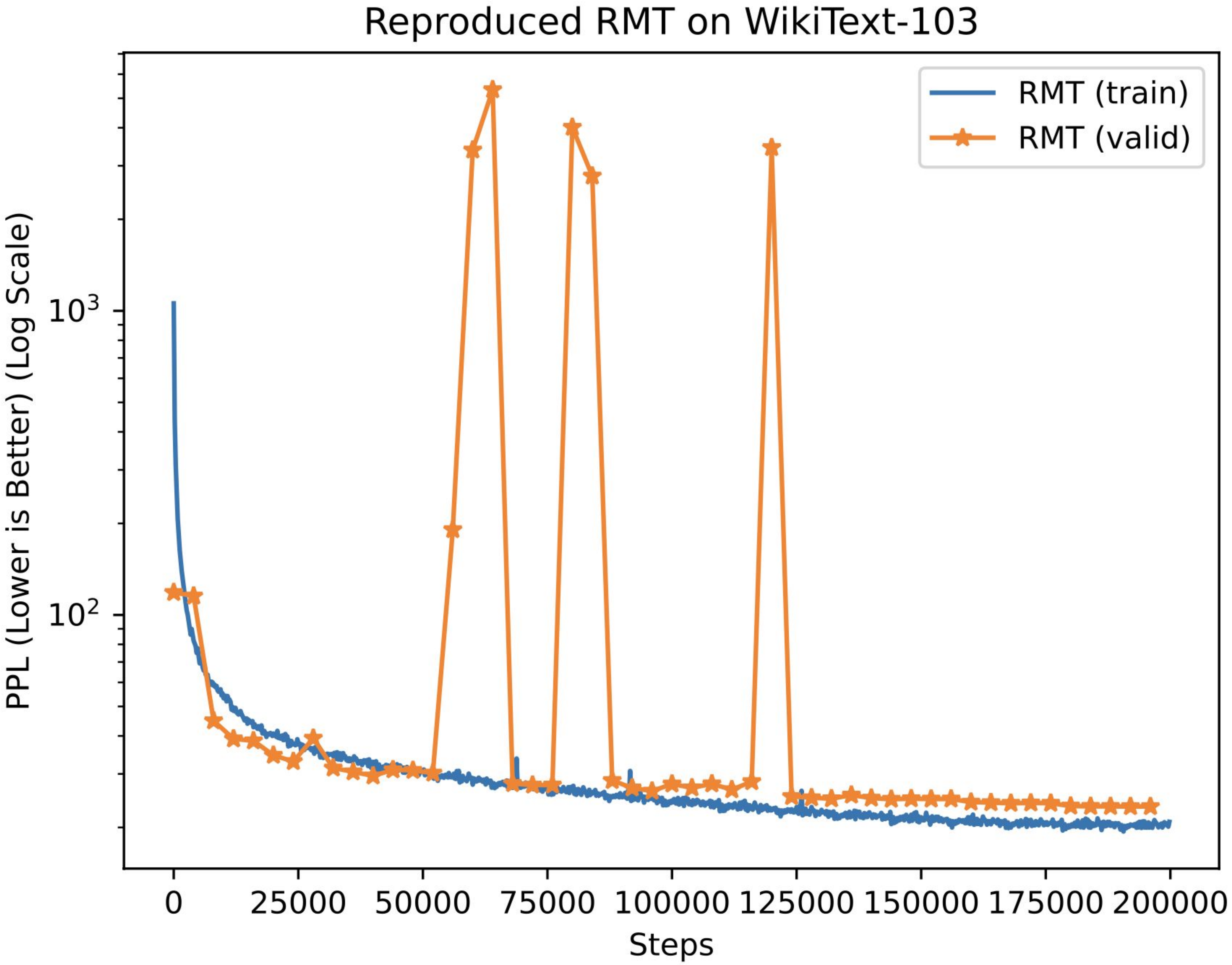


Figure 1. Reproduced step-wise perplexity score based on both training and validation sets. We use a log-scale for y-axis for better visibility of the lower scores which are obstructed by the outliers.

Our Contributions

Our Contributions

01 We introduce a novel token based memory method called **TARM** that increases the capacity of the memory token without actually increasing the memory token size.

Our Contributions

01 We introduce a novel token based memory method called **TARM** that increases the capacity of the memory token without actually increasing the memory token size.

02 We show how **TARM** increases the stability of the training which enables faster convergence.

Our Contributions

01 We introduce a novel token based memory method called **TARM** that increases the capacity of the memory token without actually increasing the memory token size.

02 We show how **TARM** increases the stability of the training which enables faster convergence.

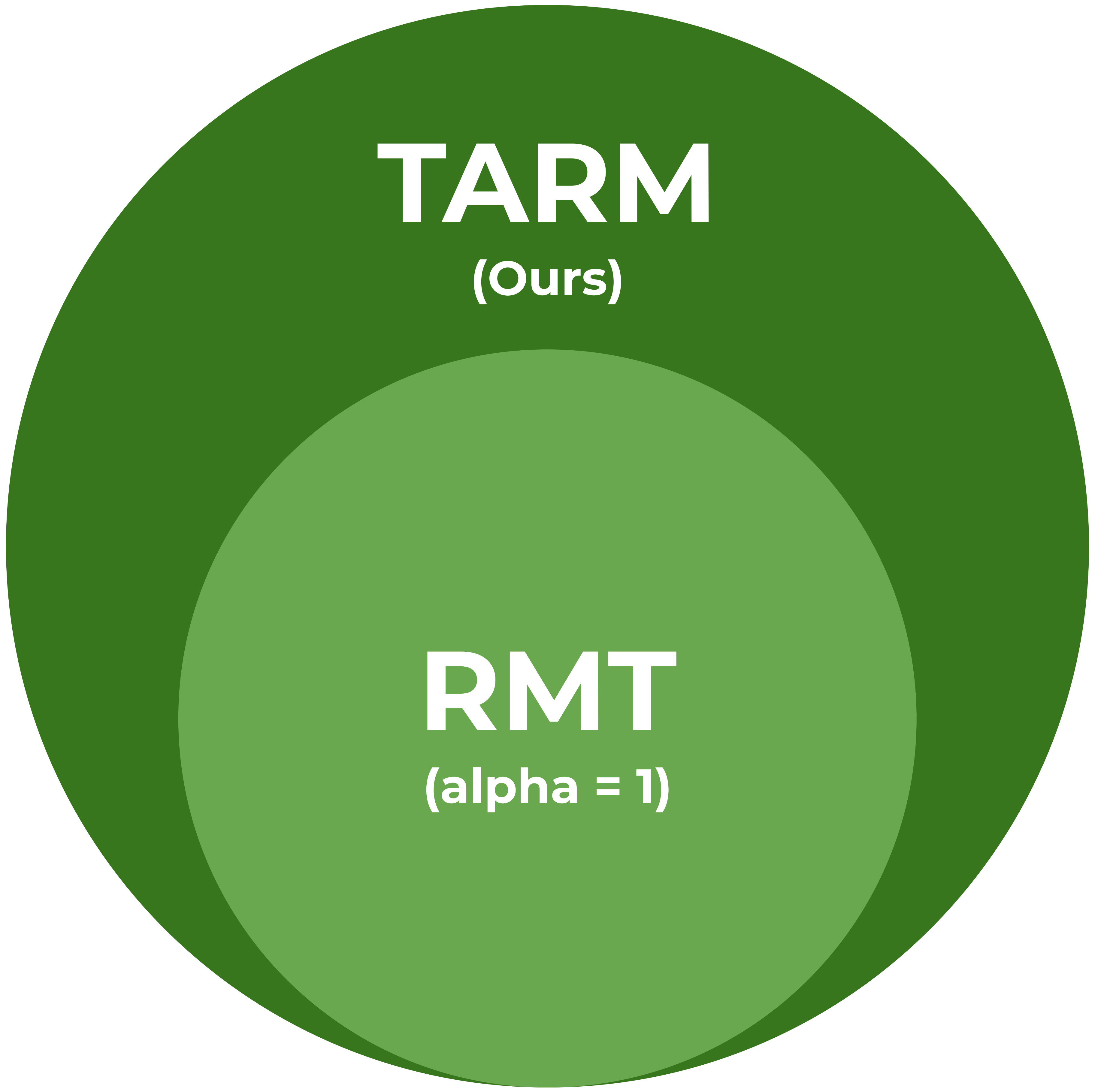
03 We evaluate the effectiveness of **TARM** approach on language modelling tasks and demonstrate that it outperforms the original RMT model on similar settings.

Methodology: TARM

RMT

(alpha = 1)

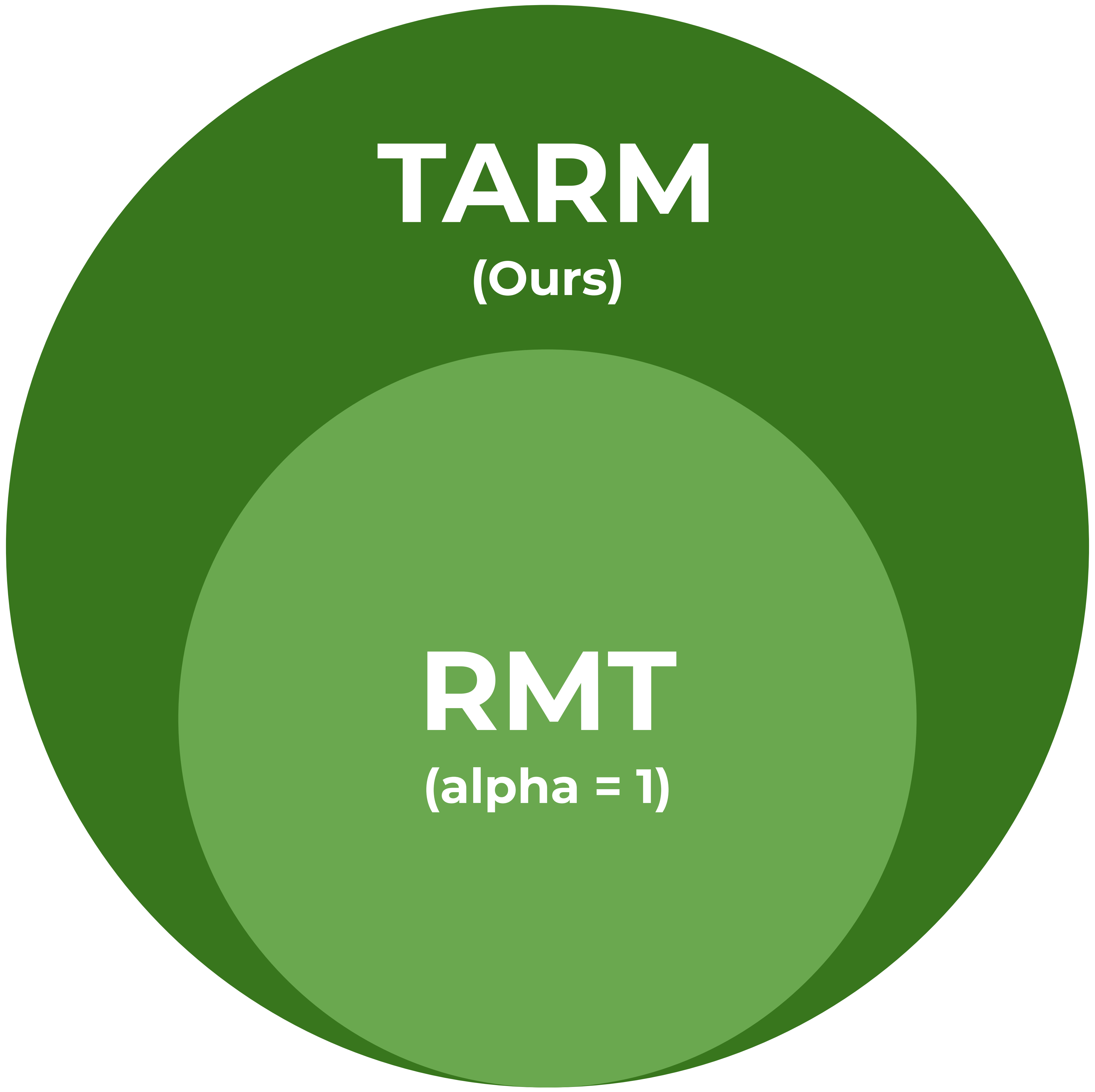
Methodology: TARM



Methodology: TARM

$$E_t = \alpha \times M_t + (1 - \alpha) \times E_{t-1}$$

where:
E_t = weighted memory tokens at time step t
α = weight used in the moving average (damping factor)
M_t = original memory tokens at time step t



TARM increases stability

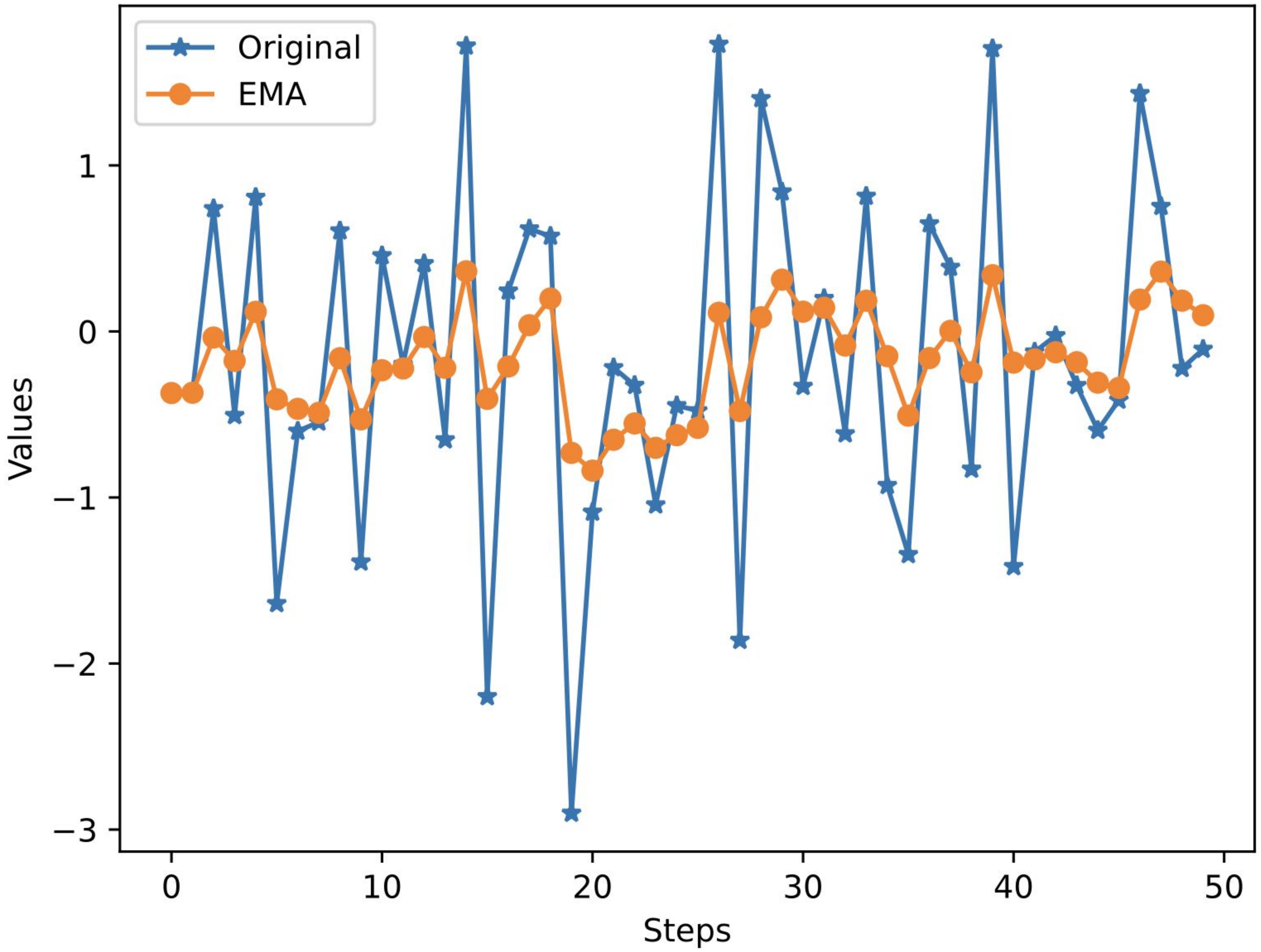


Figure 2. Demonstration of how the Exponential Moving Average (EMA) component used in TARM helps in stabilizing the erratic change in values.

TARM increases stability

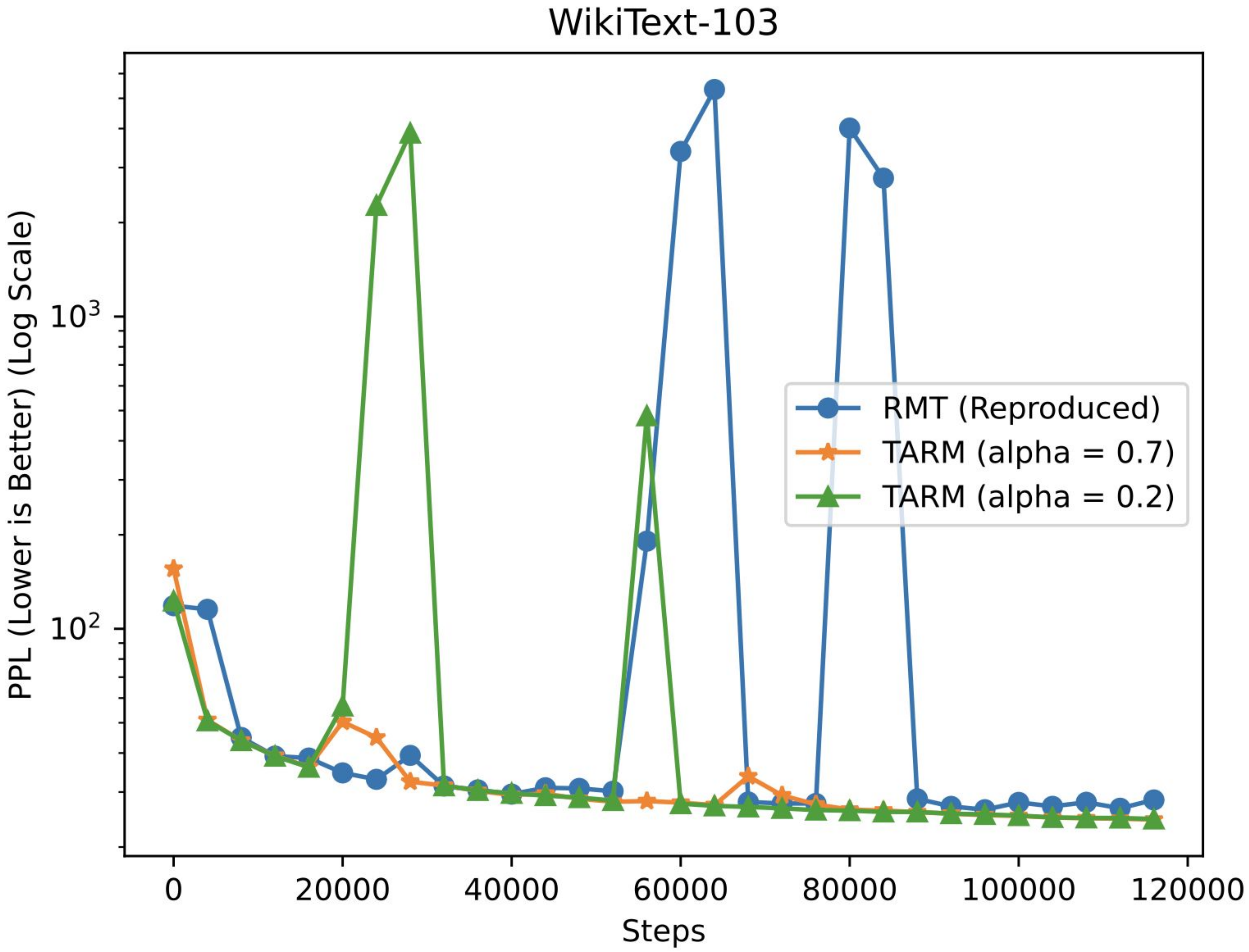


Figure 3. Figure show how TARM stabilizes the validation accuracy over steps and also leads to faster/better convergence when compared to RMT.

Comparison with RMT

Task	**Dataset**	**Metric**	**RMT (Reproduced)**	**TARM (Ours)**
LM*	WT-103	PPL ↓	28.301	**24.493** (↓3.808)

Table 2. Results comparison of RMT and TARM on Language Modelling task. The best results are shown in bold, and the second-best results are underlined. The symbol ↑ and ↓ indicate the improvement and degradation of the result compared to the reproduced RMT model. LM* denotes language modeling task. PPL denotes perplexity.

Ablations

Task	**Dataset**	**Metric**	α	PPL
LM*	WT-103	PPL $\downarrow$	0.2	**24.493**
LM*	WT-103	PPL $\downarrow$	0.7	<u>24.603</u>
LM*	WT-103	PPL $\downarrow$	1.0	28.301

Table 3. Ablation of the hyperparamter α. The best results are shown in bold, and the second-best results are underlined. The symbol $\uparrow$ and $\downarrow$ indicate the improvement and degradation of the result compared to the reproduced RMT model. LM* denotes language modeling task. PPL denotes perplexity.

References

[1] Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 35:11079–11091, 2022. 1, 2

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[3] Matt Mahoney. Large text compression benchmark, 2011. 1

[4] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. 1, 2

TARM: Token Averaging Recurrent Memory Transformers

Rahul Vigneswaran K **CS23MTECH02002** Peketi Divya **Assigned TA** Prof. C. Krishna Mohan **Course Instructor**

Done as part of the coursework on Visual Computing (CS6450).

Abstract

Recurrent Memory Transformers (RMT) are well know for their ability to learn very long-term dependencies and general-purpose memory processing in various applications. However, their capacity too is limited by the size of the memory tokens. In this paper, we propose a novel approach, TARM to increase the memory capacity of RMTs by implementing an exponential moving average on the memory tokens. Our approach allows for improved capture of longer-term dependencies without increasing the size of the memory tokens. We evaluate the effectiveness of this approach on language modelling tasks and demonstrate that it outperforms the original RMT model on similar settings. We also show TARM makes the training much more stable.

1. Introduction

Recurrent Memory Transformers [1] have shown great potential in capturing long-term dependencies and general-purpose memory processing. However, the size of the memory tokens limits the memory capacity of these models, and increasing the size of these tokens can result in higher computational complexity. In this work, we propose a novel approach to address this limitation by implementing an exponential moving average on the memory tokens. The following are the key contributions:

1. We introduce a novel token based memory method called TARM that increases the capacity of the memory token without actually increasing the memory token size.
2. We show how TARM increases the stability of the training which enables faster convergence.
3. We evaluate the effectiveness of this approach on language modelling tasks and demonstrate that it outperforms the original RMT model on similar settings.

2. Methodology

Our proposed method involves using an exponentially weighted moving average which would stablize and smooth out the memory tokens over time. Incidently this would enable the model to access more information from previous segments without actually increasing the size of the memory tokens.

Let the memory tokens at time t be denoted by M_t, and the exponentially weighted memory tokens be denoted by E_t. The formula for updating the exponentially weighted memory tokens at time t is given by:

$$E_t = \alpha \times M_t + (1 - \alpha) \times E_{t-1} \quad (1)$$

where α is the weight used in the moving average, and E_{t-1} is the exponentially weighted memory tokens from the previous time step.

To initialize the exponentially weighted memory tokens, we can set E_0 to be equal to the initial memory tokens M_0. During training, the model uses the exponentially weighted memory tokens E_t instead of the original memory tokens M_t for processing and passing information between segments of the long sequence using recurrence.

3. Experimental setup

The experiments are designed in such a away to evalute the ability of the approach to preserve long-term dependencies across multiple segments.

Our implementation is based on [1] for the ease of reproducibility. WikiText-103 [3] experiments use 16-layer Transformers (10 heads, 410 hidden size, 2100 intermediate FF). We used Adam [2] optimizer with linear schedule learning rate starting from 0.00025 for 200,000 steps for WikiText-103. Vocabulary contains 267735 words for Wikitext-103 tokenizers. For Wikitext-103 an input context length was set to 160 tokens.

Experiments for Quadratic Equations task use 6 layer Transformers (6 heads, 128 hidden size, 256 intermediate FF). We used Adam [2] optimizer with constant learning rate 1e-4 with reduction on plateau with decay factor of 0.5.

1