Effective Architectures for Long Contexts and Reasoning

Matvey Kairov
kairov.mv@phystech.edu

MIPT Center for Cognitive Modeling

2025 – Scientific Research Seminar

Table of Contents

- Introduction
- RMT Architecture Overview
- Task Dataset
- Motivation
- Problem Formulation
- Proposed Methodology
  - Architectural Modifications
  - Heuristic Modifications
- Initial Experiments & Results
- Conclusion & Future Work

Introduction

- Transformers [1] are the most widely adopted type of architecture for solving tasks on sequences.
- The attention mechanism, integral to Transformers, has quadratic computation complexity and memory consumption.
- On very long sequences (e.g. $> 10^4$ tokens long) and sequences longer than training-time contexts traditional Transformers rapidly degrade in performance.
- Architectures such as the Recurrent Memory Transformer (RMT) [2] were proposed with the goal of reducing complexity of processing long input sequences.

[1] Vaswani et al. "Attention is all you need". 2017

[2] Bulatov et al. "Recurrent Memory Transformer". 2022.

RMT Architecture Overview

- RMT extends the Transformer model by incorporating recurrent memory to process long sequences efficiently.
- For segment τ, given an initial hidden state H_τ^0, memory tokens H_τ^{mem} are appended at the beginning and the end of the hidden state:

$$\tilde{H}_\tau^0 := [H_\tau^{mem} \circ H_\tau^0 \circ H_\tau^{mem}], \quad (1)$$

$$\bar{H}_\tau^N := \text{Transformer}(\tilde{H}_\tau^0), \quad (2)$$

$$\bar{H}_\tau^N =: [H_\tau^{read} \circ H_\tau^0 \circ H_\tau^{write}]. \quad (3)$$

H_τ^{write} is used as the memory state on the next segment.

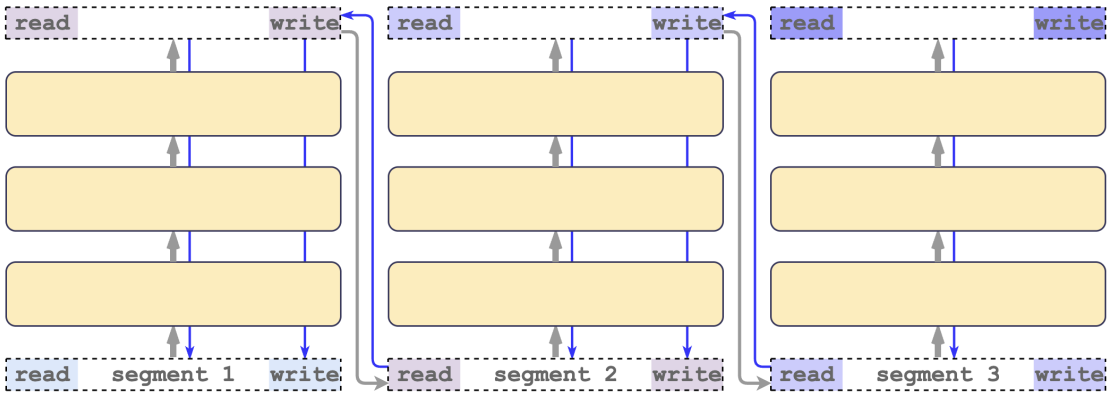


Figure: RMT architecture.

Task Dataset

- Standard text datasets do not account for the needs of long-context memory retrieval.
- We chose **BABILong** [3] as our long-context benchmark.
- It requires the model to retrieve relevant facts from a vast pool of information.
- A BABILong task consists of a large noise text N with inserted facts F.
- Based on those facts, the model is expected to answer a question Q, which is appended at the end of the sequence.

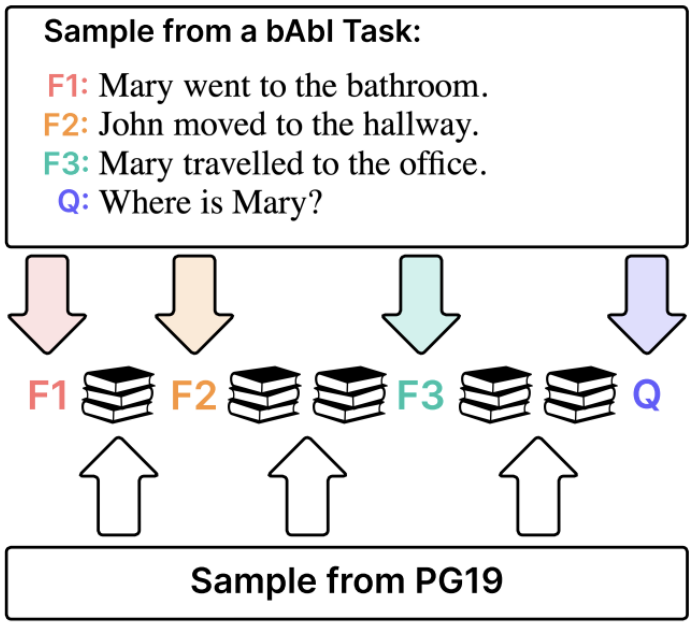


Figure: BABILong task example.

[3] Kuratov et al. "BABILong: Testing the Limits of LLMs with Long Context Reasoning-in-a-Haystack". 2024.

Motivation

- RMT and its variants such as ARMT [4] and RMT-R [5] achieve outstanding performance on long contexts.
- Authors employ a convoluted training curriculum which involves training on sequences of varying length in a strict order.
- This leads to a relatively long training time and difficulties when integrating these models with popular deep learning APIs (e.g. HF Transformers Trainer [6]).

The goal of the current research is to eliminate the need for a complicated training curriculum while maintaining high performance on language modeling tasks.

---

[4] Rodkin et al. "Associative Recurrent Memory Transformer". 2024.

[5] Kuratov et al. "In Search of Needles in a 11M Haystack: Recurrent Memory Finds What LLMs Miss". 2024.

[6] `https://huggingface.co/docs/transformers/main_classes/trainer`

Problem Formulation

The optimization process considers two main objectives:

1. Model Performance — maximizing the likelihood of observed data.
2. Memory Retrieval Efficiency — maximizing the impact of facts and minimizing the impact of noise on model predictions.
3. Training Efficiency — reducing computational overhead during optimization.

Given a set of parameter configurations Θ and a set of training heuristics H, the general objective function can be formulated as:

$$\theta^*, h^* = \arg \min_{\theta \in \Theta, h \in H} [\mathcal{L} + \lambda \mathcal{C} + \gamma \mathcal{I}] \tag{4}$$

Problem Formulation

BABILong tasks consist of facts F, noise N, questions Q and labels Y. Given the number of sequences in the dataset M, the number of classes C and predictions Y', the objective takes the form:

$$\mathcal{L} = -\sum_{i=1}^{M}\sum_{c=1}^{C} \mathbf{1}_{y_i=c} \log p_{\theta,h}\left(y_i' = c | n_i, q_i, y_i\right). \tag{5}$$

The second term regulates penalties for inefficient training heuristics (given a set of heuristics costs c_k), high computational costs, the total number of parameters, and slow convergence.

$$\mathcal{C} = \sum_k c_k \cdot \mathbf{1}_{h_k} + \beta\|\theta\|_0 + \text{FLOPs}\left(\theta\right) + \gamma T_{\text{epochs}}. \tag{6}$$

The last term regulates the amount of information retrieved by the model from facts and noise

$$\mathcal{I} = -\sum_{i=1}^{M} \mathtt{I}\left(\mathrm{y}_i, \mathrm{f}_i | \mathrm{q}_i\right) + \alpha \sum_{j=1}^{M} \mathtt{I}\left(\mathrm{y}_j, \mathrm{n}_j | \mathrm{q}_j\right), \tag{7}$$

where $\mathtt{I}$ is the mutual information function.

Proposed Methodology

The proposed methodology involves modifying both architecture and training heuristics to enable efficient single-pass training.

- **Architectural Modifications:**
  - Introduce structural adjustments to improve memory efficiency.
  - Optimize parameterization to maintain inference-time constraints.
- **Training Heuristics:**
  - Explore alternative scheduling strategies to replace progressive segment training.

Proposed Methodology

Center for Cognitive Modeling

We proposed the following architectural and heuristic modifications to the original RMT model:

- **Single-Pass RMT Training**: we train RMT in a single pass, without curriculum learning.
- **RMT-R**: we introduce memory retrieval mechanisms for improved handling of long-term dependencies.
- **RMT-B**: we implement memory updates on each transformer layer.
- **RMT-BR**: we combine RMT-B with the memory retrieval mechanism.
- **RMT-MS**: we pass memory tokens both between memory segments and neighboring Transformer layers.

Methodology: RMT-R

- RMT-R preserves tokens from previous segments and attends to them in each segment.
- After each segment, memory tokens are combined with past tokens using a transformer-like block.
- We use Rotary Positional Embeddings (RoPE) [7] to retain memory order during the Attention operation. We performed an ablation study to prove the effectiveness of RoPE in this architecture.

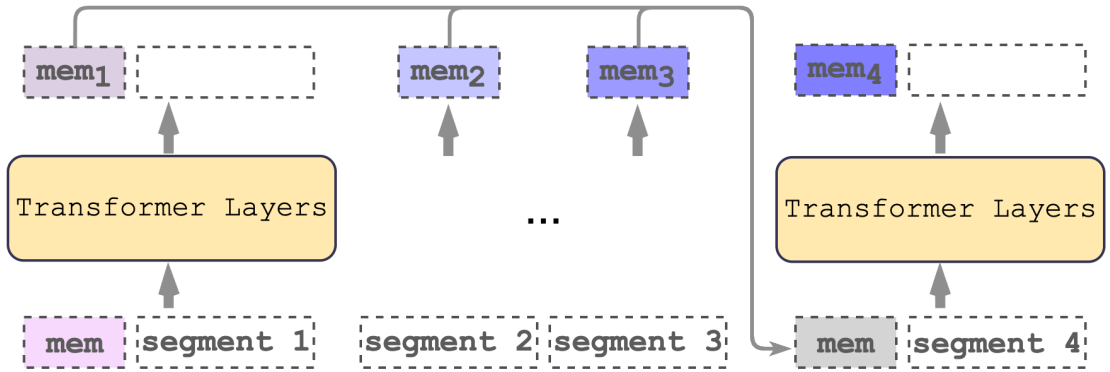


Figure: RMT-R architecture.

[7] Su et al. "RoFormer: Enhanced Transformer with Rotary Position Embedding". 2021.

Methodology: RMT-B & RMT-BR

- RMT-B applies the idea behind RMT on each individual Transformer layer.
- Each layer has its own set of memory tokens and performs memory updates.
- RMT-BR inherits the memory aggregation module from RMT-R with the same working principle, but on a lower level.

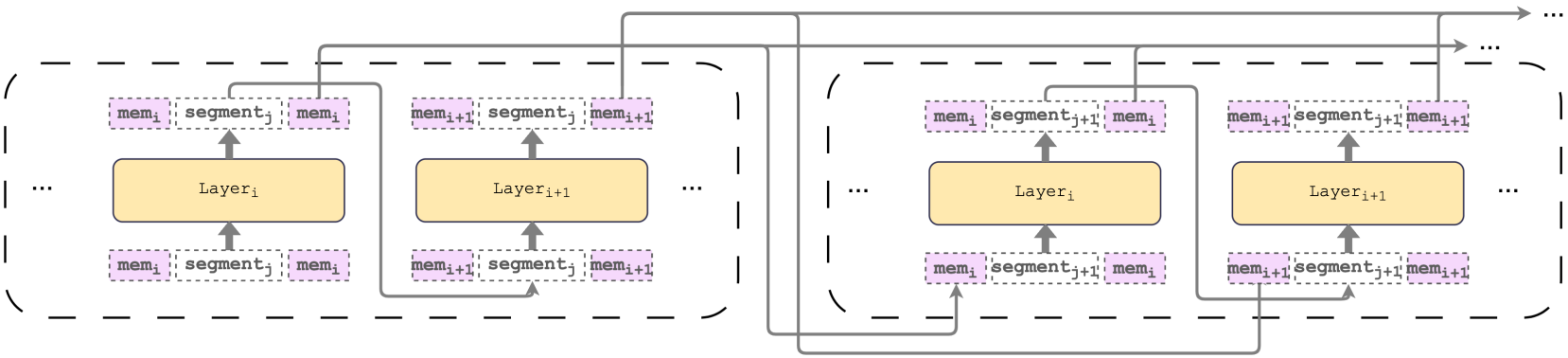


Figure: RMT-BR architecture.

Methodology: RMT-MS

- RMT-B and RMT-BR cause an information bottleneck at the hidden states.
- To prevent this, we replace writing memory of each Transformer layer with the writing memory of the previous layer.

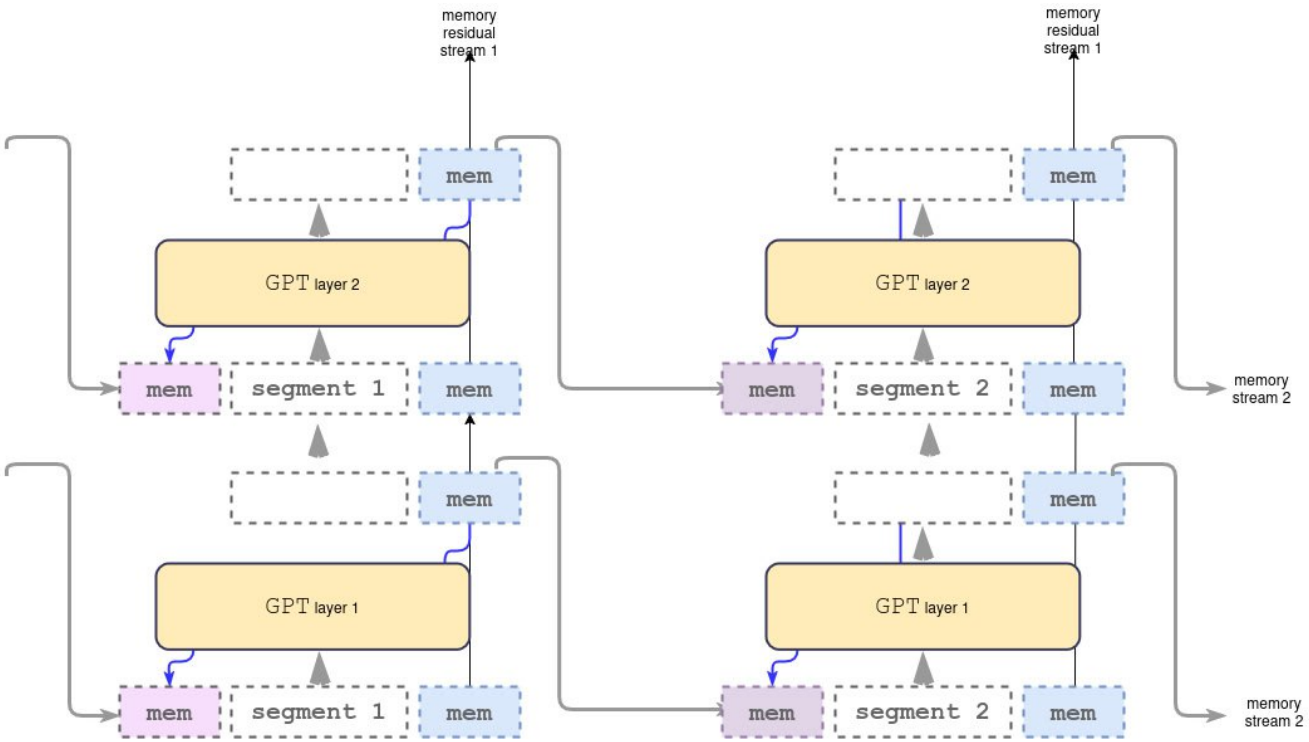
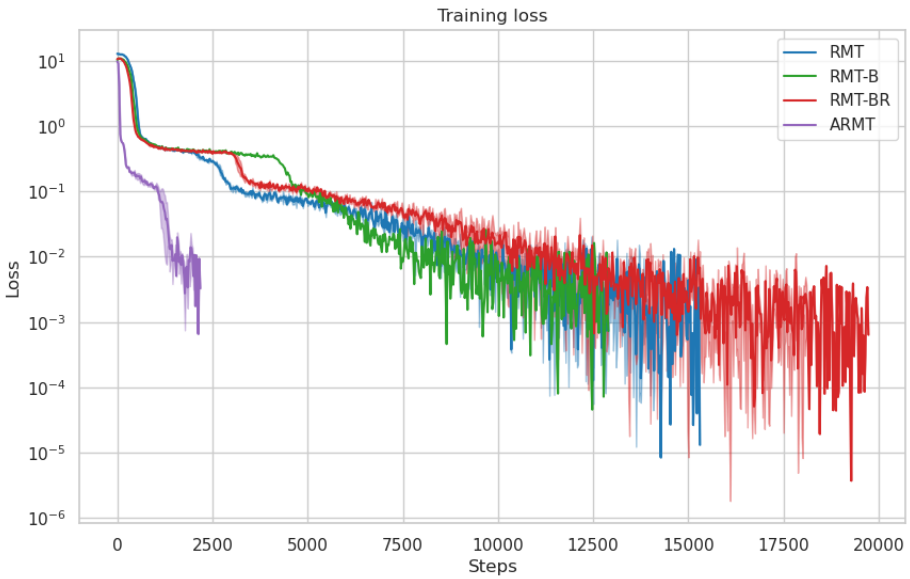
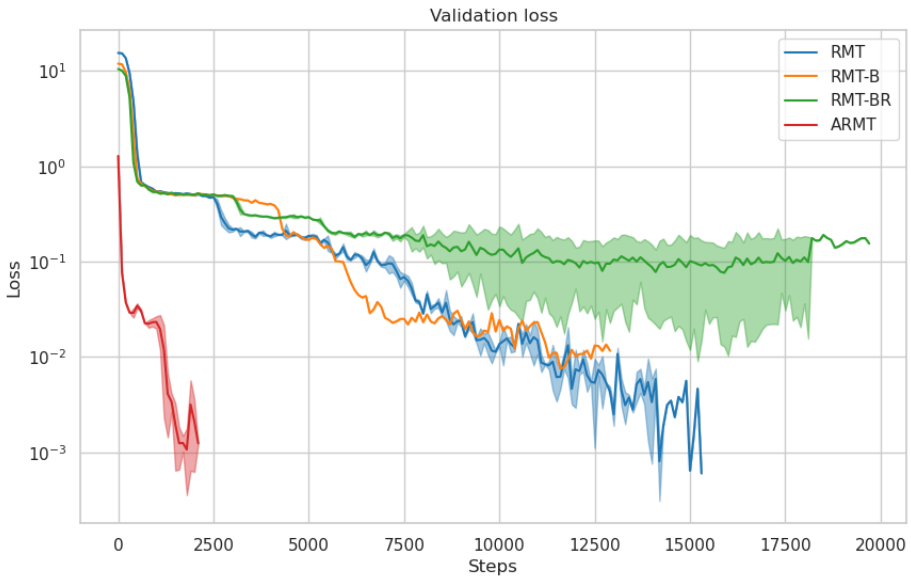


Figure: RMT-MS architecture.

Experiments: Simple Task



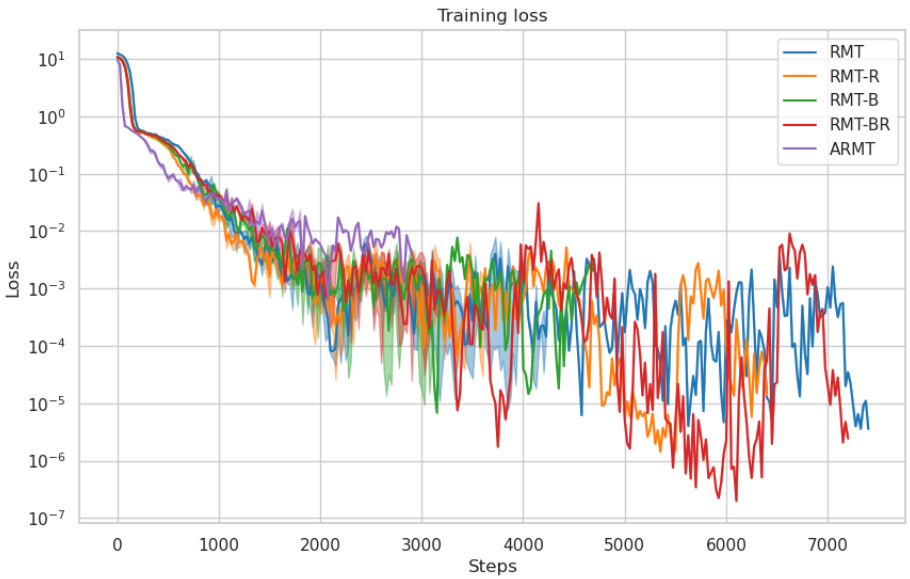
(a) Training loss.



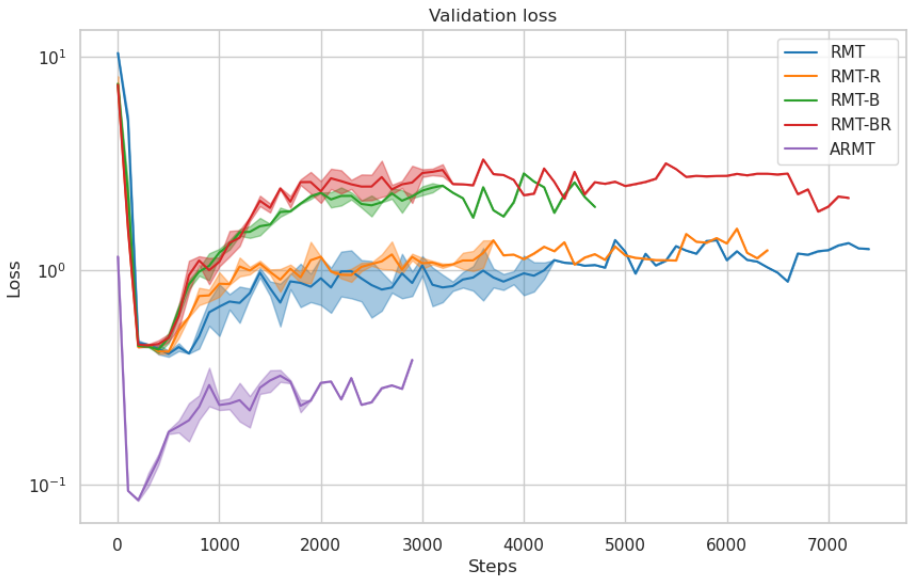
(b) Validation loss

Figure: BABILong QA1: 16 memory tokens, 3 segments $\times$ 128 tokens per segment.

Experiments: Difficult Task



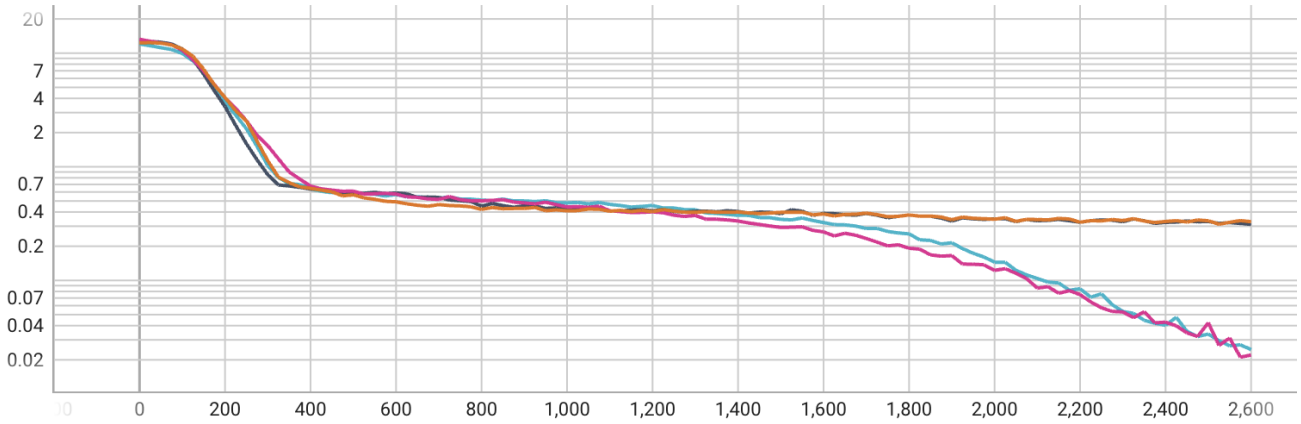
(a) Training loss.



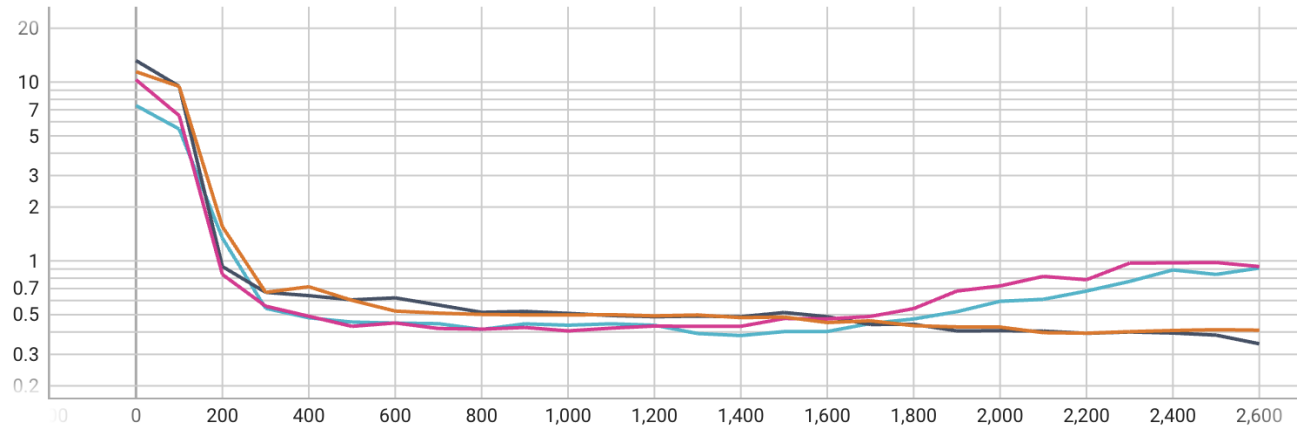
(b) Validation loss

Figure: BABILong QA3: 16 memory tokens, 3 segments $\times$ 128 tokens per segment.

Experiments: RMT-MS



(a) Training loss.



(b) Validation loss.

Figure: BABILong QA1 & QA3: 16 memory tokens, 3 segments $\times$ 128 tokens per segment.

Conclusion

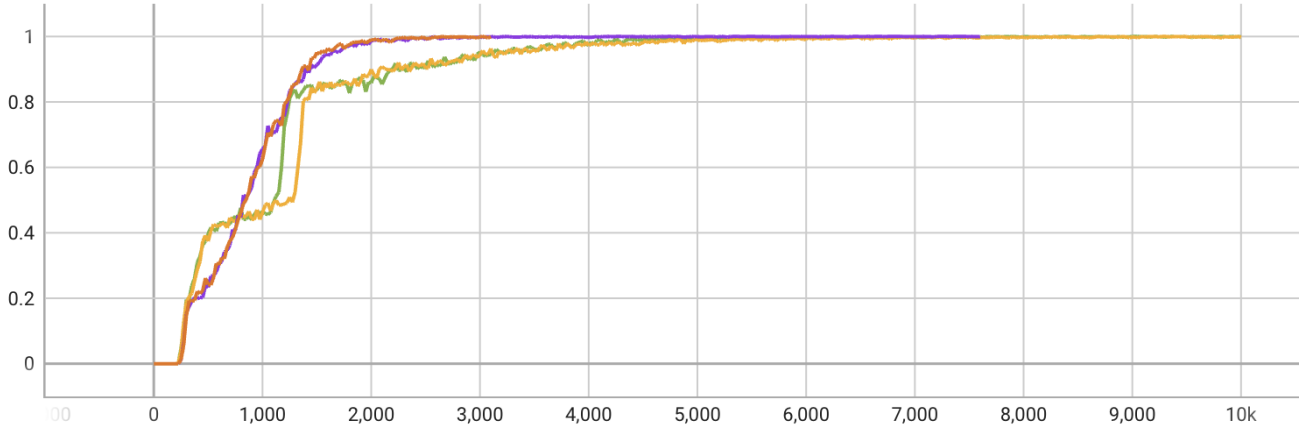
Center for Cognitive Modeling

- We attempted training **RMT** without a curriculum.
- We modified **RMT-R** and proposed **RMT-B**, **RMT-BR** and **RMT-MS**, with the main goal of shortening gradient paths for memory tokens.
- Experiments show that the models tend to overfit before converging on the harder task, while having acceptable performance on the simpler task.
- Future plans:
  - Exploration of model behavior on different stages during training.
  - Addition of in-segment context reconstruction loss.
  - Architecture comparison on curriculum training.

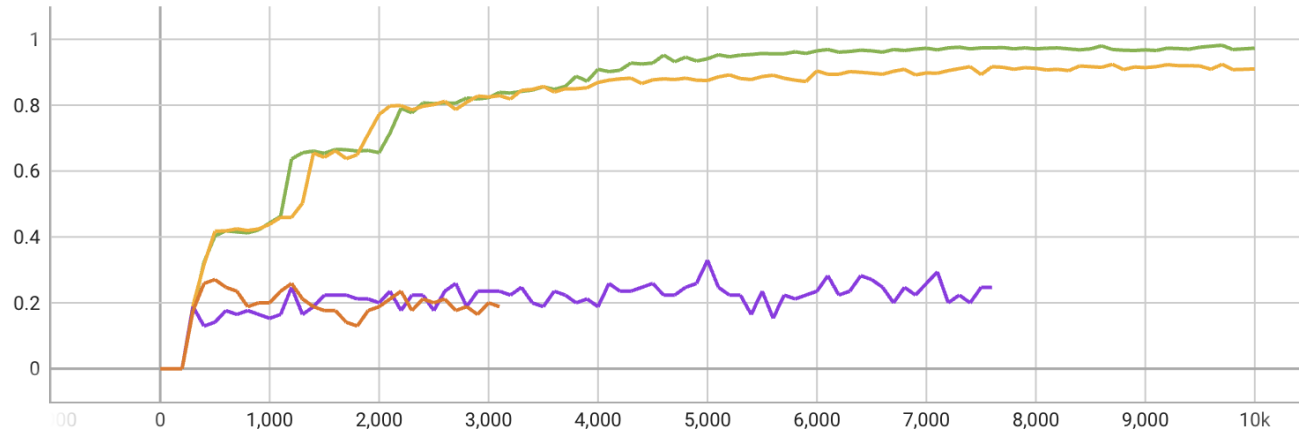
Thank you!

kairov.mv@phystech.edu

RoPE Ablation Study



(a) Training accuracy.



(b) Validation accuracy.

Figure: BABILong QA1 & QA3: 16 memory tokens, 3 segments $\times$ 128 tokens per segment.