ELMUR: External Layer Memory with Update/Rewrite for Long-Horizon RL

Egor Cherepanov, Alexey K. Kovalev, Aleksandr I. Panov

Problem

1. Many real-world decision-making problems are **partially observable**, requiring agents to act under incomplete information.

2. However, most existing RL methods are designed for **fully observable MDPs**, making them poorly suited for these settings.

3. To address partial observability, we must equip policies with **memory mechanisms** that can retain and update past information. Yet, current recurrent and transformer architectures suffer from major limitations:

    1. **Gradient vanishing** in long sequences,
    2. **Instantaneous forgetting** outside the context window,
    3. **No explicit update mechanisms** for stored information.

4. These challenges motivate the design of **ELMUR** (**E**xternal **L**ayer **M**emory with **U**pdate/**R**ewrite), a new transformer architecture that introduces external layer memory with explicit update/rewrite operations to enable robust decision-making over long horizons.

Method

ELMUR enhances transformers with an **external layer memory** that explicitly retains and updates information across long horizons. It integrates three key components:

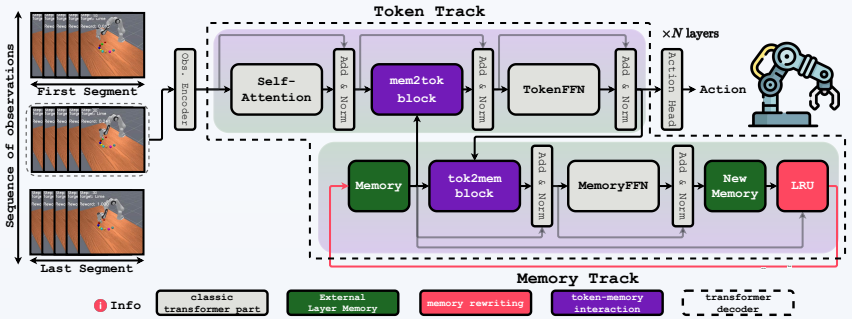
1. **Layer-local persistent memory**
   Each transformer layer is augmented with memory embeddings that persist across input segments, enabling the model to store long-term information beyond the context window, while the entire model is trained recurrently to process arbitrarily long trajectories

2. **Bidirectional token-memory interaction**
   Tokens can both **read from** and **write to** memory embeddings via cross-attention:
    1. **mem2tok:** memory enriches token representations
    2. **tok2mem:** tokens update memory with new information

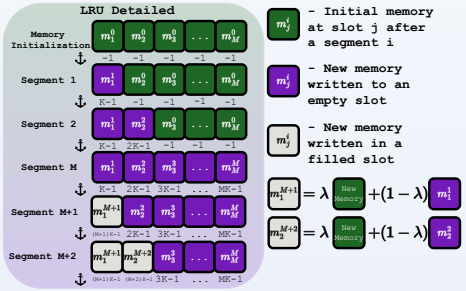
3. **LRU-based update policy**
   A Least Recently Used (LRU) module manages memory updates through either **replacement** or **convex blending**, ensuring old but relevant information is retained while new evidence is efficiently integrated.

Together, these mechanisms allow ELMUR to **extend horizons far beyond the attention window** (up to x100,000 on the synthetic T-Maze task).

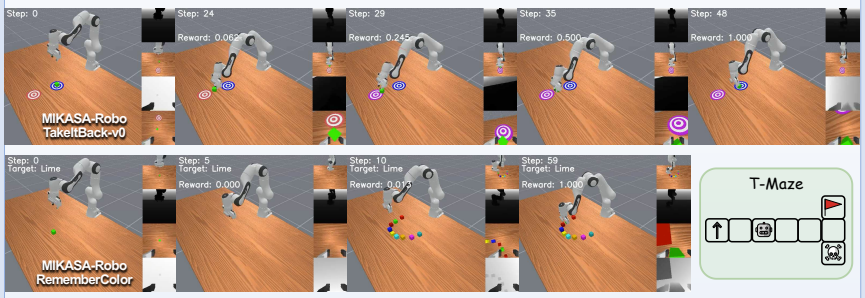
ELMUR Scheme



LRU Module

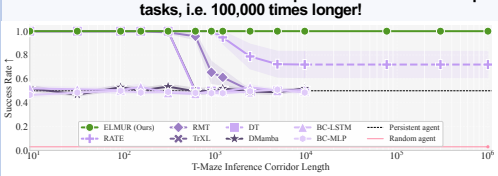


Memory Tasks



Results

ELMUR with context window size 10 steps can solve 1 million steps tasks, i.e. 100,000 times longer!



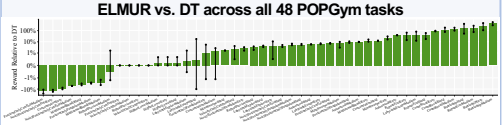
Success rates on MIKASA-Robo tasks

Task	RATE	DT	BC-MLP	CQL-MLP	DP	ELMUR (ours)
RememberColor3-v0	0.65±0.04	0.01±0.01	0.27±0.03	0.29±0.01	0.32±0.01	**0.89±0.07**
RememberColor5-v0	0.13±0.03	0.07±0.05	0.12±0.01	0.15±0.02	0.10±0.02	**0.19±0.03**
RememberColor9-v0	0.09±0.02	0.01±0.01	0.12±0.02	0.15±0.01	0.17±0.01	**0.23±0.02**
TakeItBack-v0	0.42±0.24	0.08±0.04	0.33±0.10	0.04±0.01	0.05±0.02	**0.78±0.03**

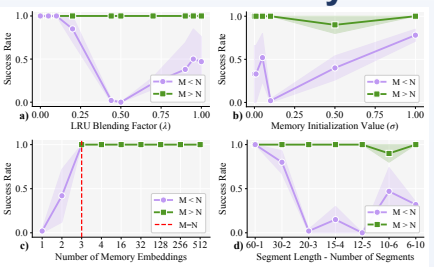
Aggregated returns on 48 POPGym tasks

	RATE	DT	Rand.	BC-MLP	BC-LSTM	ELMUR
All (48)	9.5	5.8	-12.2	-6.8	9.0	**10.4**
Puzzle (33)	0.45	-3.5	-14.6	-11.9	-0.2	**1.2**
Reactive (15)	**9.1**	**9.3**	2.3	5.1	**9.1**	**9.2**

ELMUR vs. DT across all 48 POPGym tasks



Ablation Study



Source & Contacts

Egor Cherepanov
email: cherepanovegor2018@gmail.com
twitter: https://x.com/hirasava_ui

OpenReview