# When Do Transformers Shine in RL? Decoupling Memory from Credit Assignment




Tianwei Ni, Michel Ma, Benjamin Eysenbach, Pierre-Luc Bacon

Speakers: Tianwei & Michel

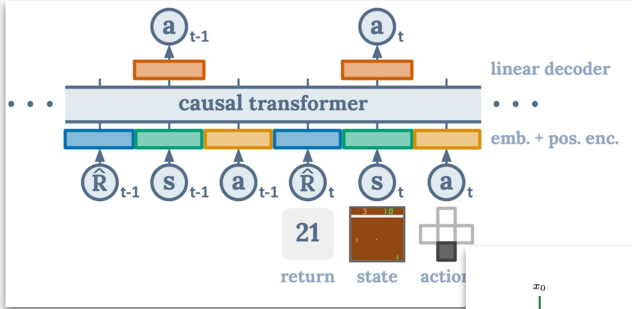
NeurIPS 2023 (oral)

Code

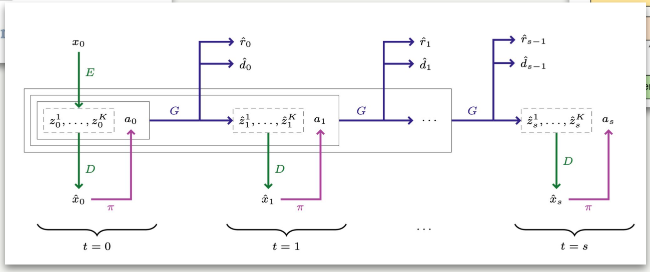
# Mystery in Transformers and RL

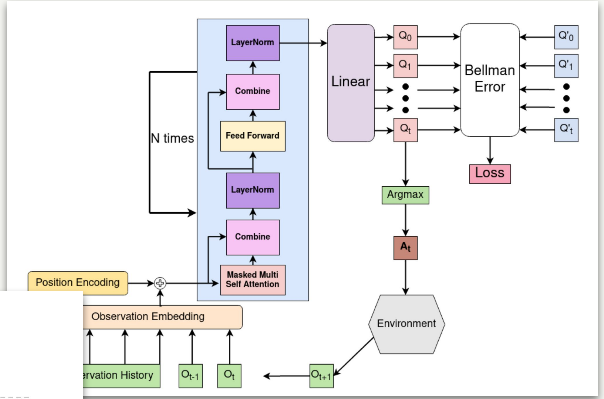
# Transformers have been successful in **RL**

- **Offline RL:** Decision Transformer (Chen et al., 2021)
- **Model-Based RL:** IRIS (Micheli et al., 2022)
- **Model-Free RL:** Deep Transformer DQN (Eslinger et al., 2022)




**Why?**






Mila

# Why do Transformers shine in **SL**?
# Excel at long-term dependencies

**Attention Is All You Need**

**Ashish Vaswani*** Google Brain avaswani@google.com

**Noam Shazeer*** Google Brain noam@google.com

**Niki Parmar*** Google Research nikip@google.com

**Jakob Uszkoreit*** Google Research usz@google.com

**Llion Jones*** Google Research llion@google.com

**Aidan N. Gomez*** † University of Toronto aidan@cs.toronto.edu

**Łukasz Kaiser*** Google Brain lukaszkaiser@google.com

**Illia Polosukhin*** ‡ illia.polosukhin@gmail.com

*“[…] the Transformer, a model architecture […] relying entirely on attention mechanism to draw global dependencies”*

# Temporal dependency: *memory*?

# Long-range dependence

文A 1 language

Article Talk Read Edit source View history

From Wikipedia, the free encyclopedia

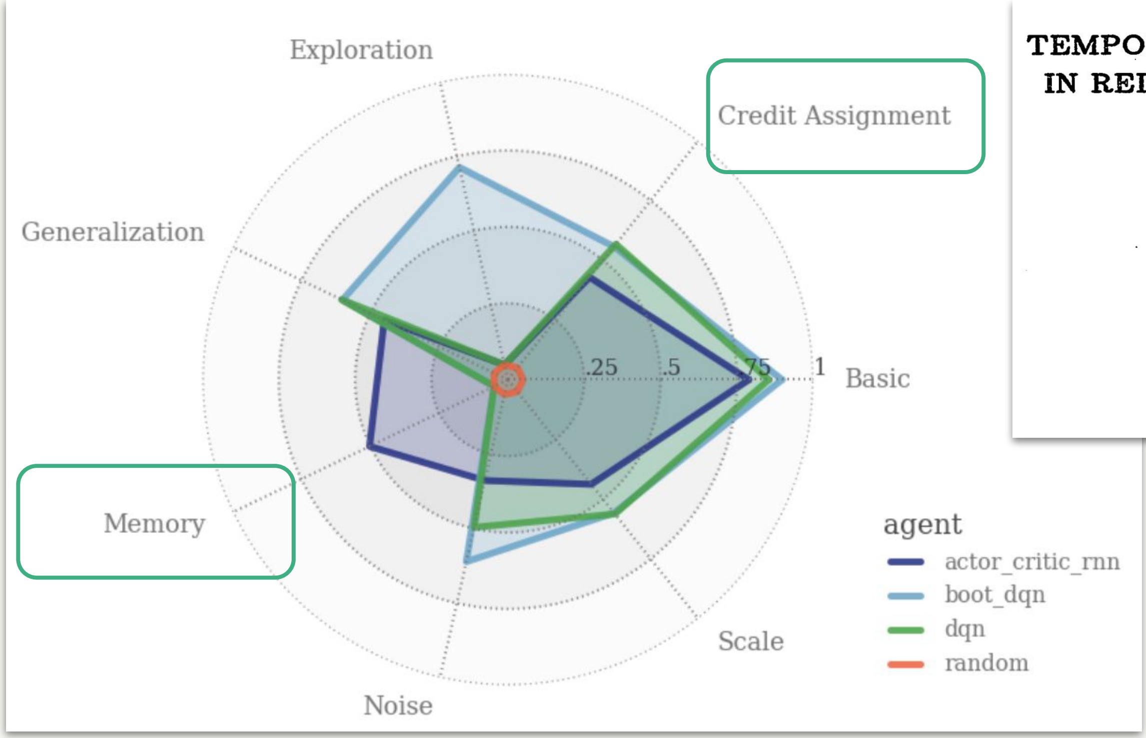
**Long-range dependence** (**LRD**), also called **long memory** or **long-range persistence**, is a phenomenon that may arise in the analysis of spatial or time series data. It relates to the rate of decay of statistical dependence of two points with increasing time interval or spatial distance between the points. A

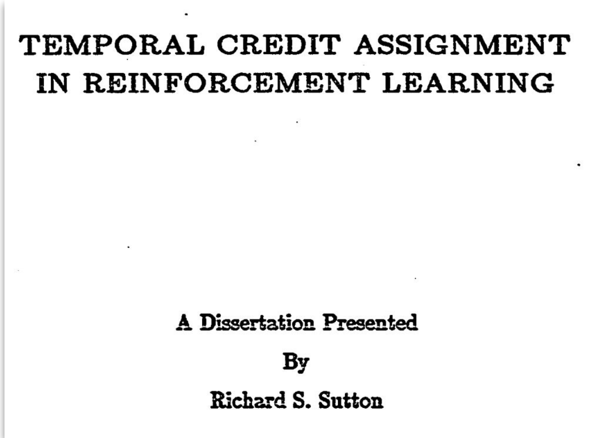
## The Problem of Learning Long-Term Dependencies in Recurrent Networks

Yoshua Bengio†, Paolo Frasconi‡, and Patrice Simard†
†AT&T Bell Laboratories
‡Dip. di Sistemi e Informatica, Universitá di Firenze

***Supervised learning perspective***

# Capabilities in RL: memory and credit assignment





TEMPORAL CREDIT ASSIGNMENT IN REINFORCEMENT LEARNING

A Dissertation Presented

By

Richard S. Sutton

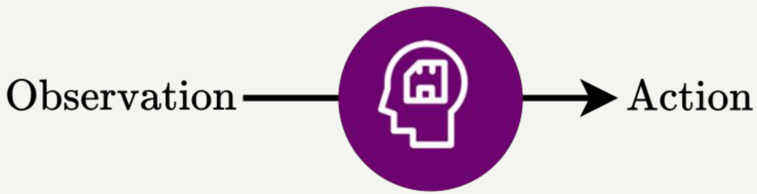


From behavior suite (Osband et al., 2019)

# Memory and Credit Assignment: they are distinct!

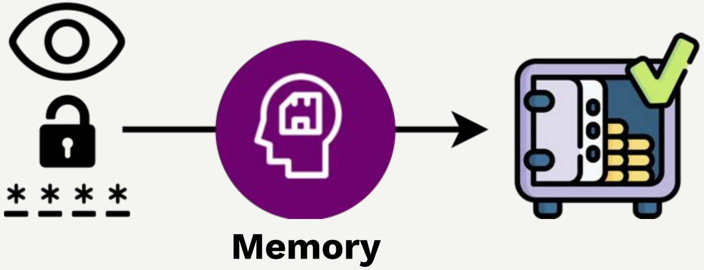
**Memory**
The ability to recall distant past events

**Temporal credit assignment**
The ability to determine *when* the actions that deserved credit occurred (Sutton, 1984)

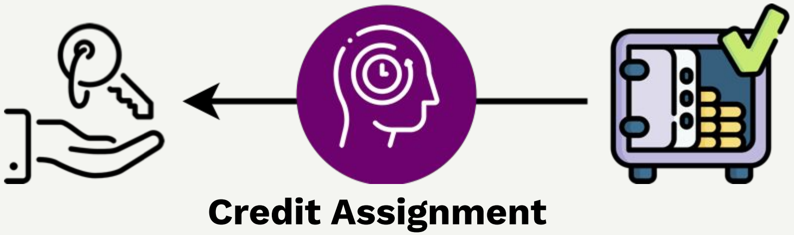



# We have intuition!

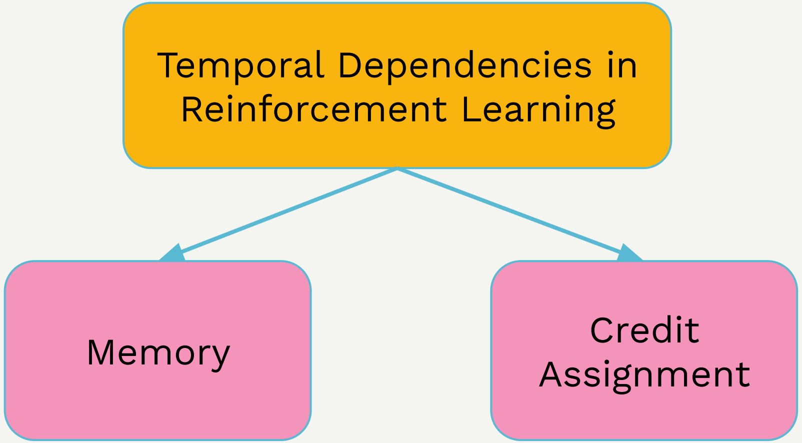
Scenario 1: Alice *remembers her passcode* set a month earlier, and opens a safe full of money.




Scenario 2: Bob *picks up a key* (then he can always see the key), and a month later he opens a safe full of money.



# Why do Transformers shine in RL? Memory or Credit Assignment? Or Both?

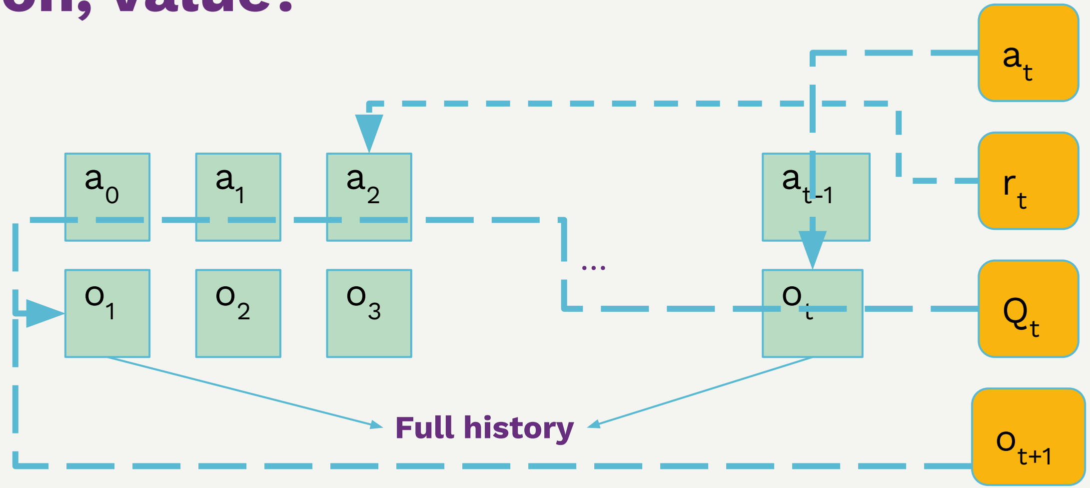



Although we have intuition, we don’t have clear mathematical definitions. This limits our understanding.

# Measuring Temporal Dependencies in RL

# Memory lengths (intuition)

- History: all previous observations and actions
- **How long is the minimal history required to predict / generate current reward, observation, action, value?**






Mila

# Memory lengths for each RL component and their relations

**Definitions**

- $m^{\mathcal{M}}_{\text{reward}}$: min $n$ s.t. $\mathbb{E}[r_t \mid h_{1:t}, a_t] = \mathbb{E}[r_t \mid h_{t-n+1:t}, a_t]$.
- $m^{\mathcal{M}}_{\text{transit}}$: min $n$ s.t. $o_{t+1} \perp\!\!\!\perp h_{1:t}, a_t \mid h_{t-n+1:t}, a_t$.
- $l_{\text{mem}}(\pi)$: min $n$ s.t. $a_t \perp\!\!\!\perp h_{t-l_{\text{ctx}}(\pi)+1:t} \mid h_{t-n+1:t}$.
- $l^{\mathcal{M}}_{\text{value}}(\pi)$: min $n$ s.t. $Q^{\pi}_n(h_{t-n+1:t}, a_t) = Q^{\pi}(h_{1:t}, a_t)$.

**Theorem**

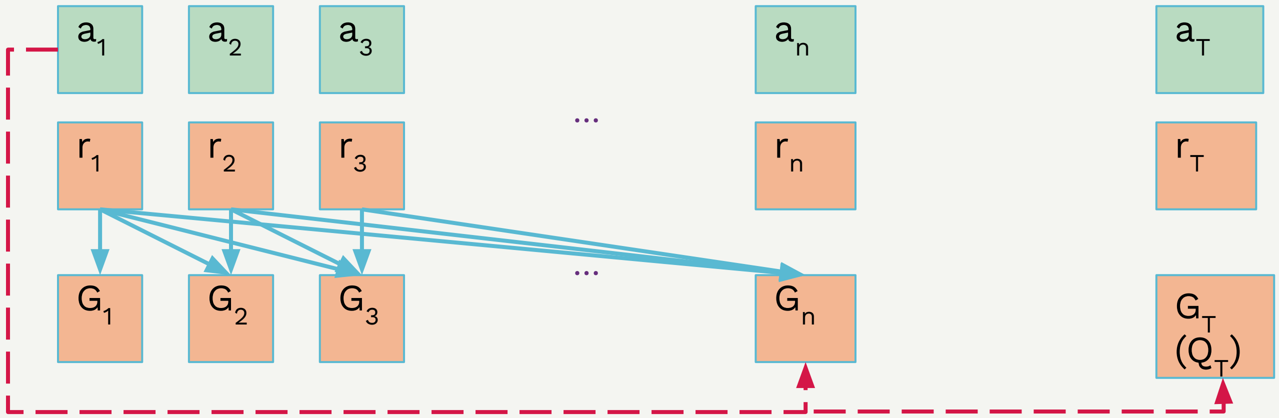
*For any optimal policy $\pi^*$ with minimal policy memory length,*

$$l_{\text{mem}}(\pi^*) \leq l^{\mathcal{M}}_{\text{value}}(\pi^*) \leq \max(m^{\mathcal{M}}_{\text{reward}}, m^{\mathcal{M}}_{\text{transit}}) := m^{\mathcal{M}}$$

**Optimal Policy memory length <= Optimal Value memory length <= max( reward memory length, transition memory length )**

# Credit assignment length (Intuition)

**How long does it take for a greedy action to see its benefits regarding its *n-step rewards* ($G_n$)?**






Mila

# When does an optimal action become best at n-step rewards?

**Informal Definition**

For an optimal action $a_t^* \in argmax_{a_t} Q^*(h_{1:t}, a_t)$, find the minimal $n$ s.t.

$$G_n^*(h_{1:t}, a_t^*) > G_n^*(h_{1:t}, a_t'), \quad \forall a_t' \neq a_t^*$$

where $G_n^*(h_{1:t}, a_t) = \mathbb{E}_{\pi^*}\left[\sum_{k=0}^{n-1} \gamma^k r_{t+k} \mid h_{1:t}, a_t\right]$ is the sum of $n$-step rewards (e.g., immediate reward, $Q$ value).

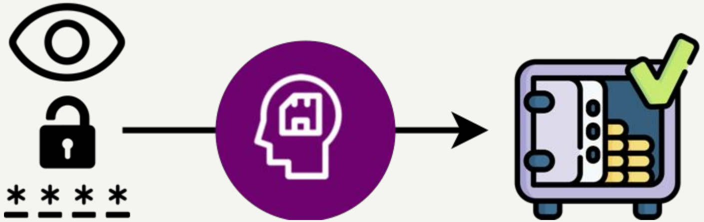
# Examples: decoupling memory and credit assignment

Scenario 1: Alice remembers her passcode set a month earlier, and opens a safe full of money.

- Credit assignment length = 1 day
- (Policy) Memory length = 1 month

Scenario 2: Bob picks up a key (then he can always see the key), and a month later he opens a safe full of money.

- Credit assignment length = 1 month
- (Policy) Memory length = 1 day





Mila

# Characterizing existing RL *benchmarks*

| | Task $\mathcal{M}$ | $T$ | $l_{\text{mem}}(\pi^*)$ | $m^{\mathcal{M}}$ | $c^{\mathcal{M}}$ |
|---|---|---|---|---|---|
| Memory | Reacher-pomdp (Yang and Nguyen, 2021) | 50 | long | long | short |
| | Memory Cards (Esslinger et al., 2022) | 50 | long | long | 1 |
| | TMaze Long (Noise) (Beck et al., 2019) | 100 | $T$ | $T$ | 1 |
| | Memory Length (Osband et al., 2020) | 100 | $T$ | $T$ | 1 |
| | Mortar Mayhem (Pleines et al., 2023) | 135 | long | long | $\leq 25$ |
| | Autoencode (Morad et al., 2023) | 311 | $T$ | $T$ | 1 |
| | Numpad (Parisotto et al., 2020) | 500 | long | long | short |
| | PsychLab (Fortunato et al., 2019) | 600 | $T$ | $T$ | short |
| | Passive Visual Match (Hung et al., 2018) | 600 | $T$ | $T$ | $\leq 18$ |
| | Repeat First (Morad et al., 2023) | 831 | 2 | $T$ | 1 |
| | Ballet (Lampinen et al., 2021) | 1024 | $\geq 464$ | $T$ | short |
| | Passive Visual Match (Sec. 5.1; Our experiment) | 1030 | $T$ | $T$ | $\leq 18$ |
| | $17^2$ MiniGrid-Memory (Chevalier-Boisvert et al., 2018) | 1445 | $\leq 51$ | $T$ | $\leq 51$ |
| | Passive T-Maze (Eg. 1; Ours) | 1500 | $T$ | $T$ | 1 |
| | $15^2$ Memory Maze (Pasukonis et al., 2022) | **4000** | long | long | $\leq 225$ |
| | HeavenHell (Esslinger et al., 2022) | 20 | $T$ | $T$ | $T$ |
| | T-Maze (Bakker, 2001) | 70 | $T$ | $T$ | $T$ |
| | Goal Navigation (Fortunato et al., 2019) | 120 | $T$ | $T$ | $T$ |
| | T-Maze (Lambrechts et al., 2021) | 200 | $T$ | $T$ | $T$ |
| | Spot the Difference (Fortunato et al., 2019) | 240 | $T$ | $T$ | $T$ |
| | PyBullet-P benchmark (Ni et al., 2022) | 1000 | 2 | 2 | short |
| | PyBullet-V benchmark (Ni et al., 2022) | 1000 | short | short | short |
| Credit Assignment | Umbrella Length (Osband et al., 2020) | 100 | 1 | 1 | $T$ |
| | Push-r-bump (Yang and Nguyen, 2021) | 50 | long | long | long |
| | Key-to-Door (Raposo et al., 2021) | 90 | short | $T$ | $T$ |
| | Delayed Catch (Raposo et al., 2021) | 280 | 1 | $T$ | $T$ |
| | Active T-Maze (Eg. 2; Ours) | 500 | $T$ | $T$ | $T$ |
| | Key-to-Door (Sec. 5.2; Our experiment) | 530 | short | $T$ | $T$ |
| | Active Visual Match (Hung et al., 2018) | 600 | $T$ | $T$ | $T$ |
| | Episodic MuJoCo (Ren et al., 2022) | **1000** | 1 | $T$ | $T$ |

Many tasks **entangle memory and credit assignment.**

**Most memory tasks evaluate memory length <= 1000**

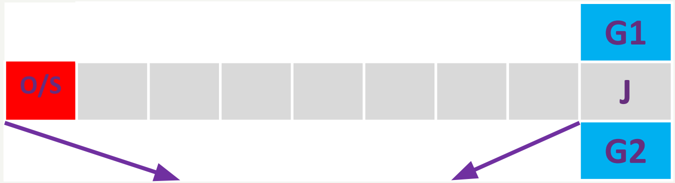


Mila

# Evaluating Transformer-based RL

# Proposing configurable toy tasks: Passive and Active T-Mazes

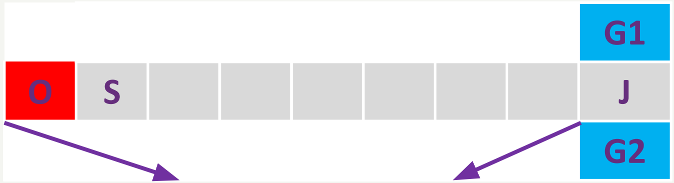
**Passive T-Maze**




Corridor Length T

**Mem len: T**
**CA len: 1**

**Active T-Maze**




Corridor Length T

**Mem len: T**
**CA len: T**

O: Oracle (to get the info of G, randomized in each episode)
S: Start; J: Junction
G1, G2: Goal candidates (to get bonus)

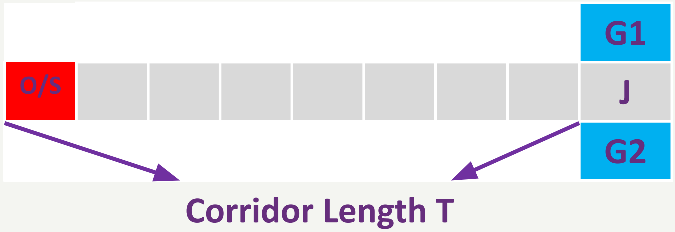
*Going to O is only credited when the agent reaches G.*



Mila

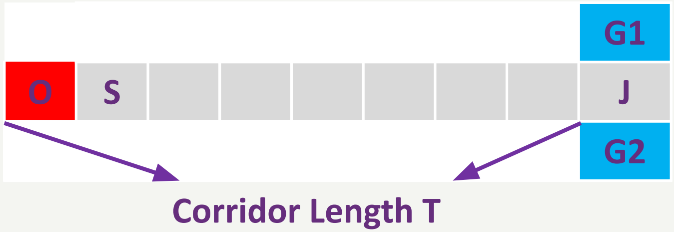
# Proposing configurable toy tasks: Passive and Active T-Mazes

**Passive T-Maze**




**Mem len: T**
**CA len: 1**

**Active T-Maze**




**Mem len: T**
**CA len: T**

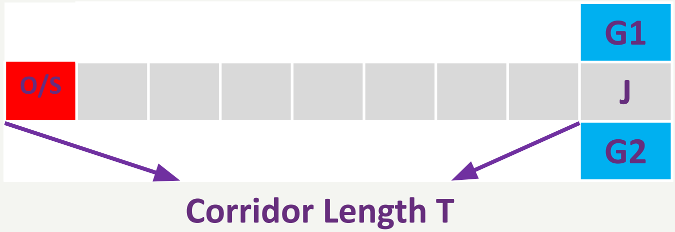
- Agent starts at state **S**
- Goal state is randomized between **G1** and **G2**
- State **O** tells the agent where the goal state is.
- At state **J**, it must make a choice



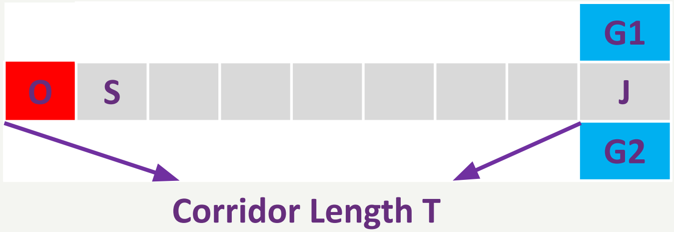
Mila

# Proposing configurable toy tasks: Passive and Active T-Mazes

**Passive T-Maze**




**Active T-Maze**



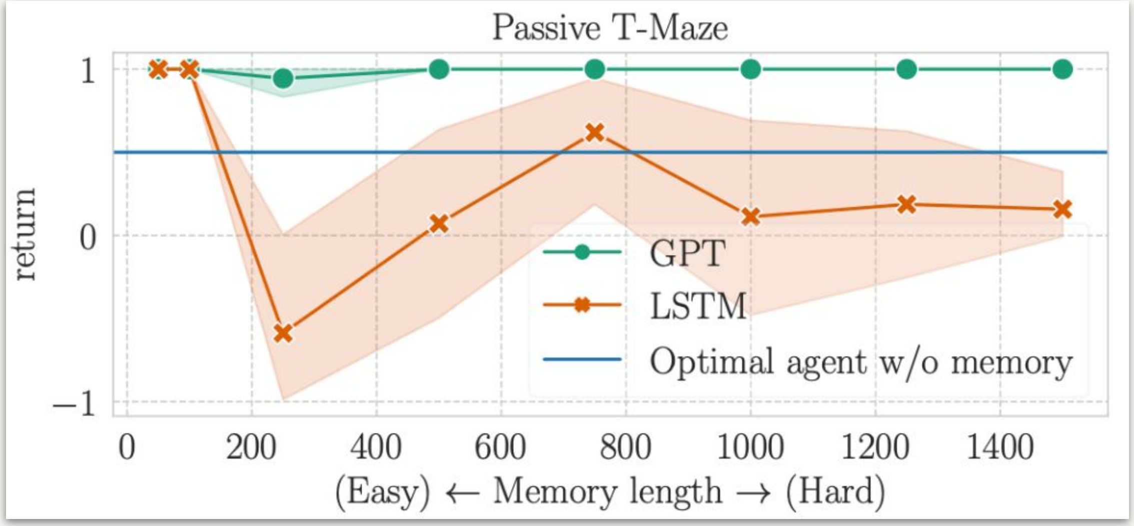

**Mem len: T**
**CA len: 1**

Exploration is easy: a small reward is given for advancing towards **J**.

**Mem len: T**
**CA len: T**

# Transformer-based RL excel at long-term memory.

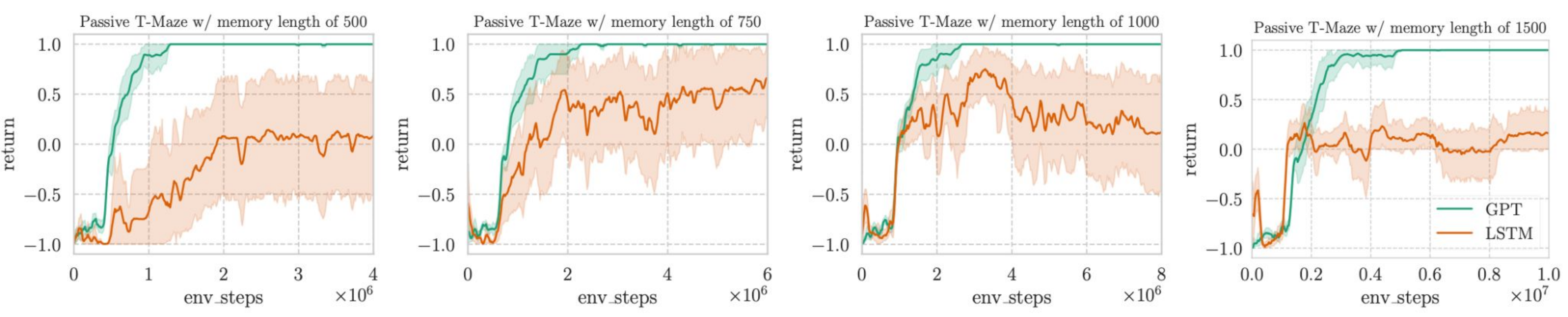
RL algorithm: DDQN w/ eps greedy
Sequence model: LSTM or Decoder-only Transformer (GPT-2)
Context length: full episode length




Perfectly solved by GPT-2


Mila

# Learning curves (10 seeds)

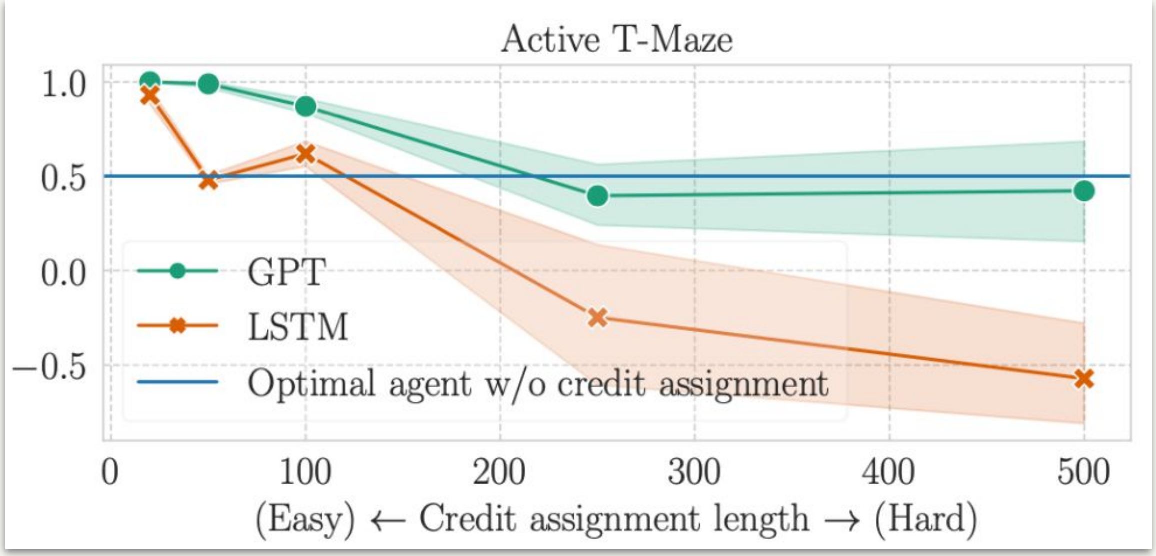



LSTMs can reach the Junction (positive rewards), but not memorize correctly.



Mila

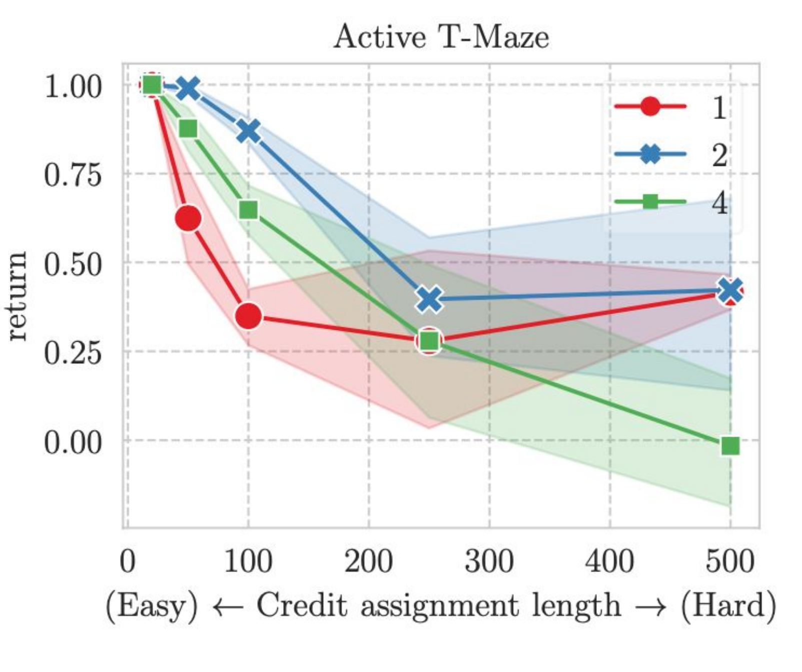
# Transformers do not help long-term credit assignment in RL.




In Active T-Maze, Transformers can reach the Junction.
Transformers helps, **but degrades severely when the CA length >= 250.**


Mila

# Increasing the number of layers in GPT-2 does not help much

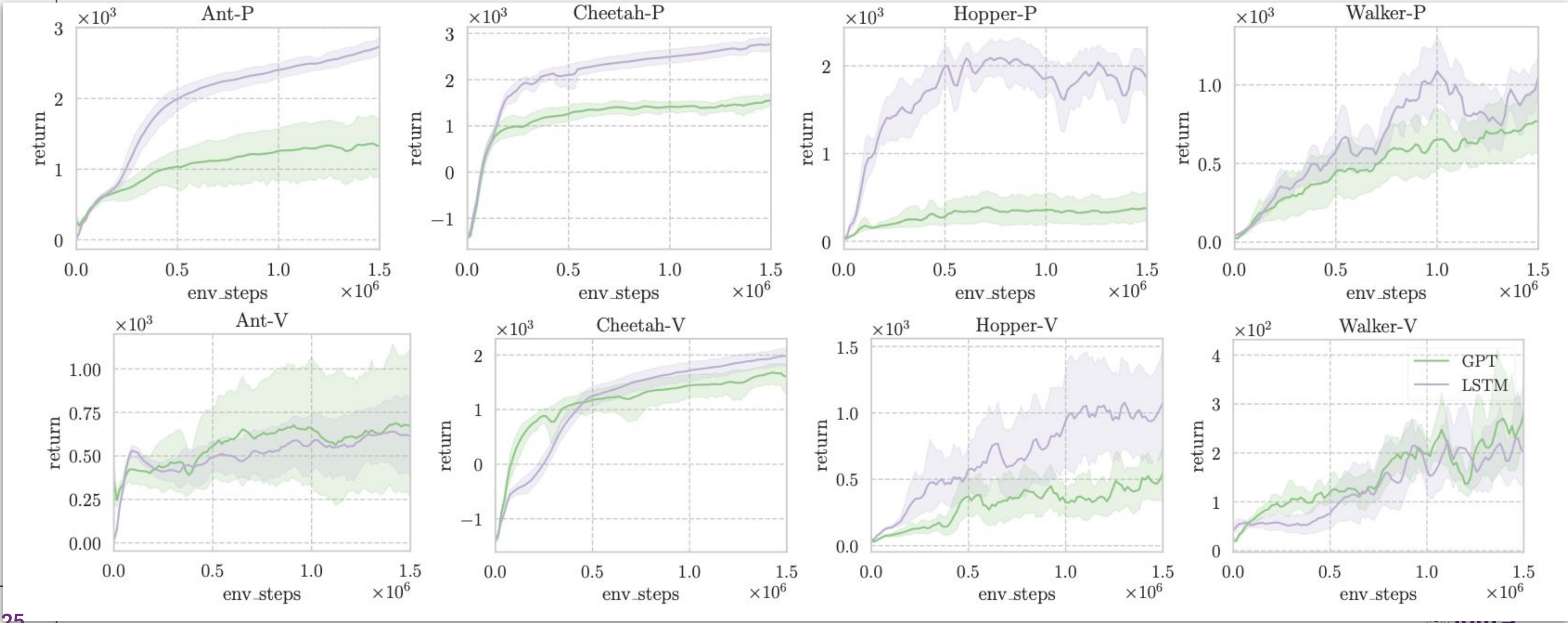




Mila

# Transformers are less sample-efficient than LSTMs in short-term dependency tasks.

Using model-free TD3 in PyBullet occlusion continuous control tasks.



# Future work

- Scalable RL systems for high-dim, long-term memory tasks
- Scaling laws in Transformer-based RL
- How can we use Transformers for long-term credit assignment?

# Future work

- Scalable RL systems for high-dim, long-term memory tasks
- How can we use Transformers for long-term credit assignment?
- Transformers have yet to replace RL algorithms!

# Questions?
# Meet us at #1425 at 5-7pm!

Code