Finding the FrameStack: Learning What to Remember for Non-Markovian Reinforcement Learning

Geraud Nangue Tasse*, Matthew Riemer, Benjamin Rosman, and Tim Klinger
University of the Witwatersrand, Johannesburg, South Africa

RLC · IBM · WITS · MIND Machine Intelligence and Neural Discovery Institute

Markov need not apply! RL agents can efficiently handle long-term dependencies by learning what to remember, reducing memory and compute costs while preserving optimality.

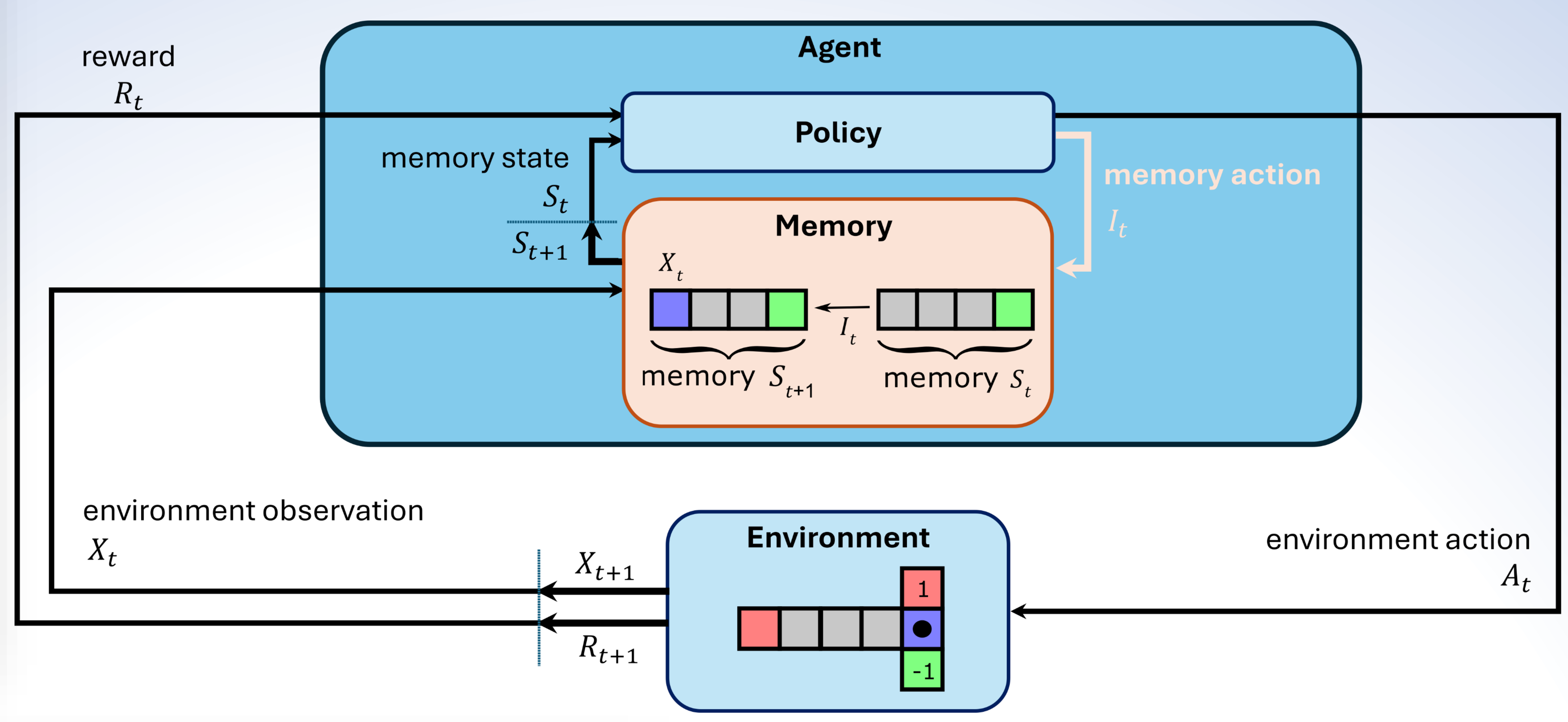
Motivation

Unlike a standard **Frame Stack**, which blindly retains recent observations (needs **full history** k^*), **we want** agents that learn only the **minimal number of observations** κ to retain based only on their **relevance for reward maximisation**.

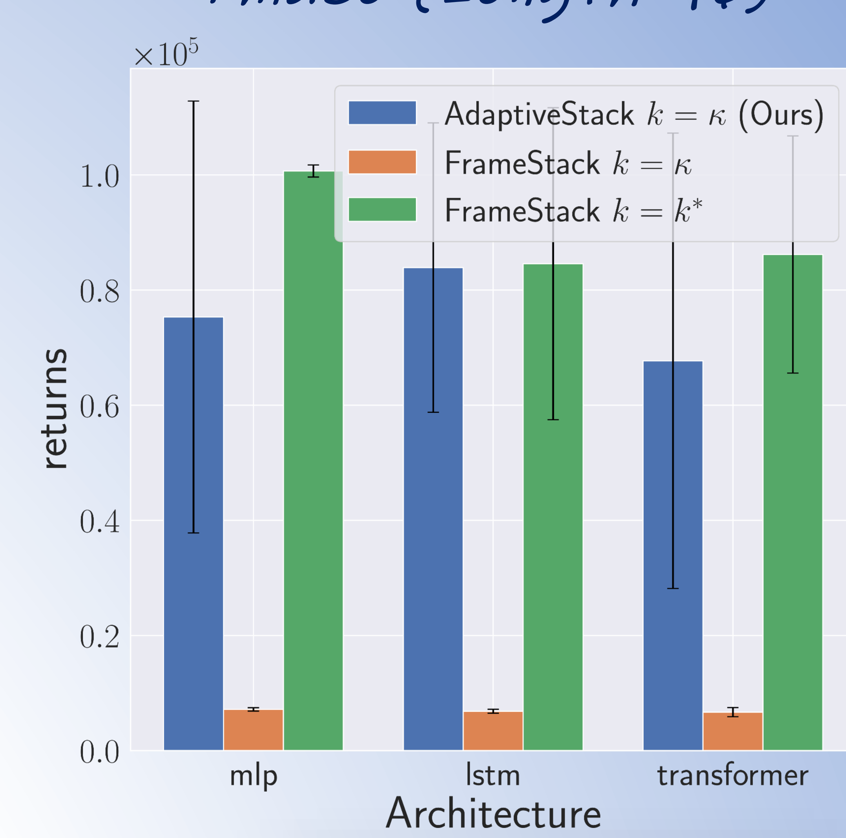
Frame Stack → **exponential increase** in observations, which impacts **compute c** and **memory w**

Architecture	Memory Type	$\lvert c\rvert_{a\sim\pi_\theta}$	$\lvert c\rvert_{\text{TD}}$	$\lvert w\rvert_{a\sim\pi_\theta}$	$\lvert w\rvert_{\text{TD}}$
MLP or LSTM	Frame Stack	$\Omega(k^*)$	$\Omega(k^*)$	$\Omega(k^*)$	$\Omega(k^*)$
MLP or LSTM	Adaptive Stack	$\Omega(\kappa)$	$\Omega(\kappa)$	$\Omega(\kappa)$	$\Omega(\kappa)$
Transformer	Frame Stack	$\Omega(k^{*2})$	$\Omega(k^*)$	$\Omega(k^{*2})$	$\Omega(k^*)$
Transformer	Adaptive Stack	$\Omega(\kappa^2)$	$\Omega(\kappa)$	$\Omega(\kappa^2)$	$\Omega(\kappa)$

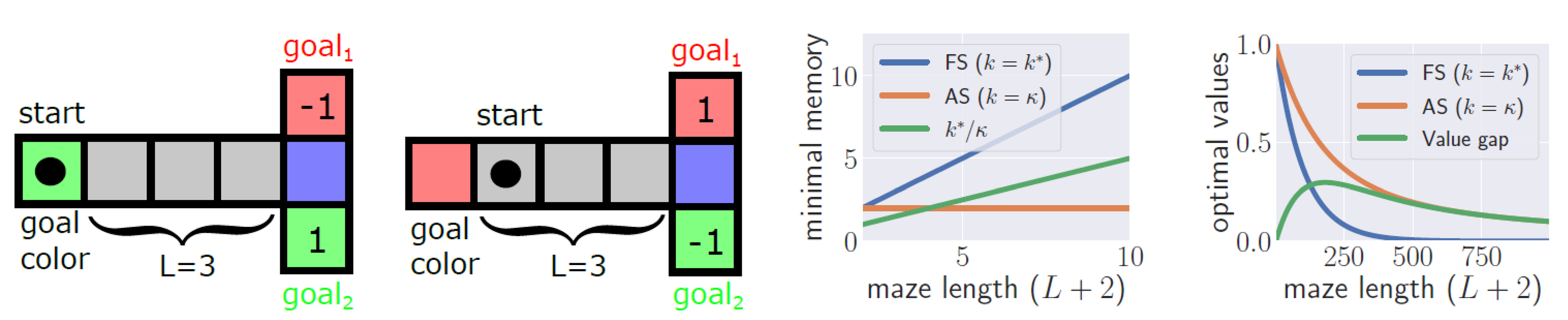
Adaptive Stacking



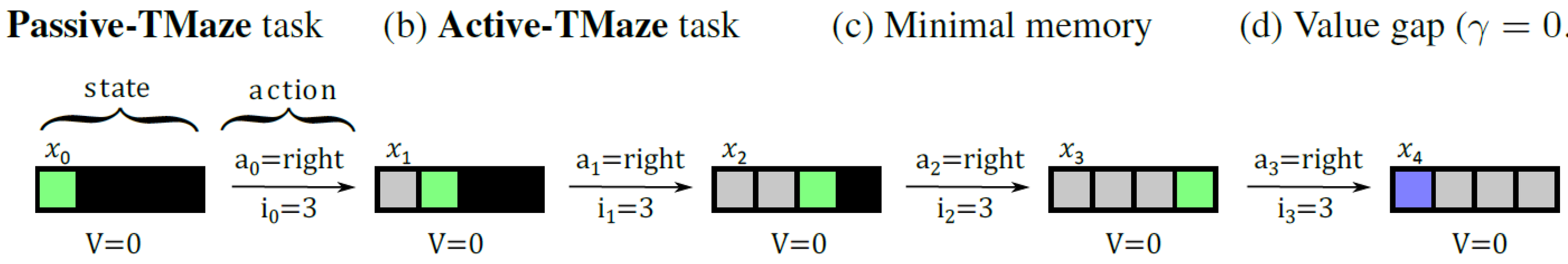
Tmaze (Length=16)



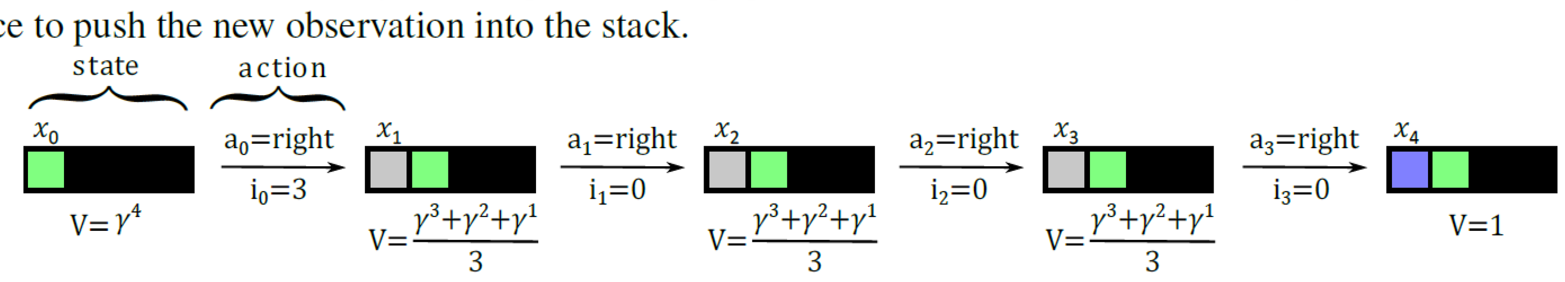
RL with Internal Memory Decisions



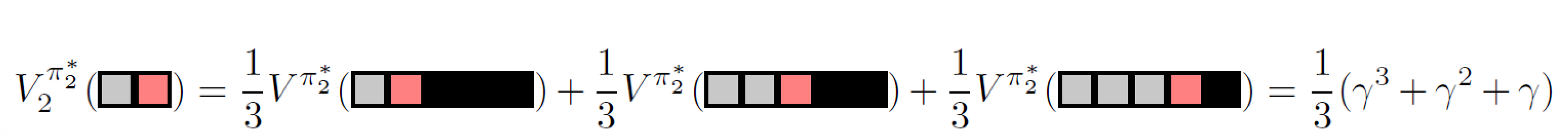
(a) **Passive-TMaze** task (b) **Active-TMaze** task (c) Minimal memory (d) Value gap ($\gamma = 0.99$)



(a) **Frame Stacking**. At every time step, the agent pops the last observation in the memory stack in order free up space to push the new observation into the stack.



(b) **Adaptive Stacking**. At every time step, the agent chooses which observation in the memory stack to pop in order to free up space to push the new observation into the stack.



Adaptive Stacking

Get initial observation $x_0 \in \mathcal{X}$
Initialise observation stack $s_0 \leftarrow [x_0]_k$
foreach *timestep* $t = 0, 1, \dots, T$ *while episode is not done* **do**

$$\langle a_t, i_t\rangle \leftarrow \begin{cases} \arg\max_{\langle a,i\rangle} Q(s_t, \langle a, i\rangle) & \text{w.p. } 1-\varepsilon \\ \text{a random action} & \text{w.p. } \varepsilon \end{cases}$$

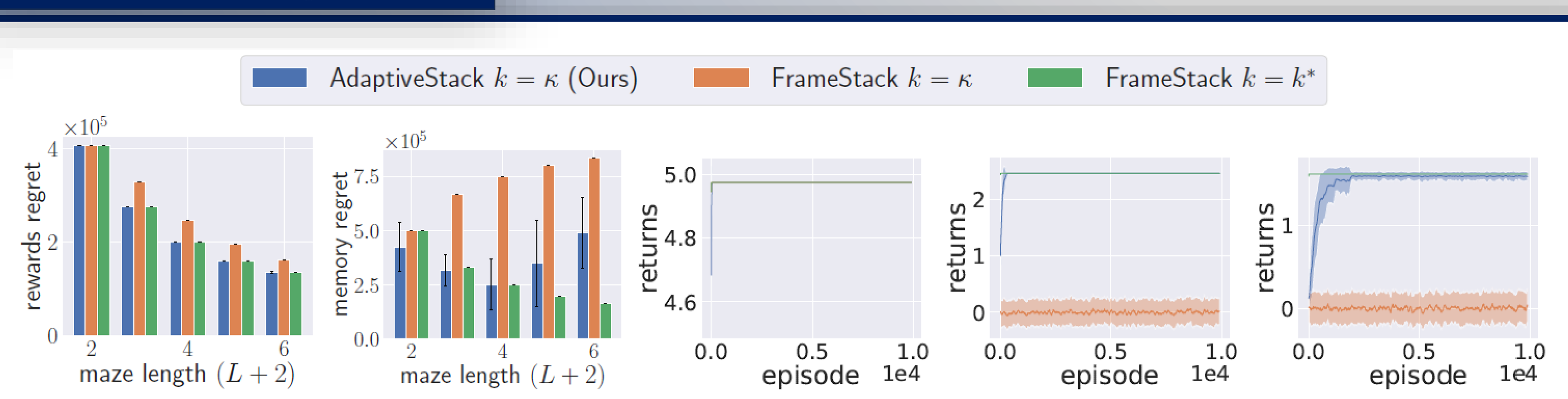
Execute a_t, get reward r_{t+1} and next observation x_{t+1}
Remove observation from stack $s_{t+1} \leftarrow pop(s_t, i_t)$
Push observation into stack $s_{t+1} \leftarrow push(s_{t+1}, x_{t+1})$

Remark 1 *Uncertainty in history may harm value expectations,* $|V^*(x_{t:t-k^*}) - V_k^{\pi_k^*}(s_t)| > 0$, *but it does not necessarily harm policy optimality as long as the uncertain differences are irrelevant for optimal decision making:* $V^*(x_{t:t-k^*}) = V^{\pi_k^*}(x_{t:t-k^*})$.

Definition 1 *Define* κ *to be the smallest memory length such that there exists a policy* π_κ^* *satisfying* $V^{\pi_k^*}(x_{t:t-k^*}) = V^*(x_{t:t-k^*})$ *for all* t.

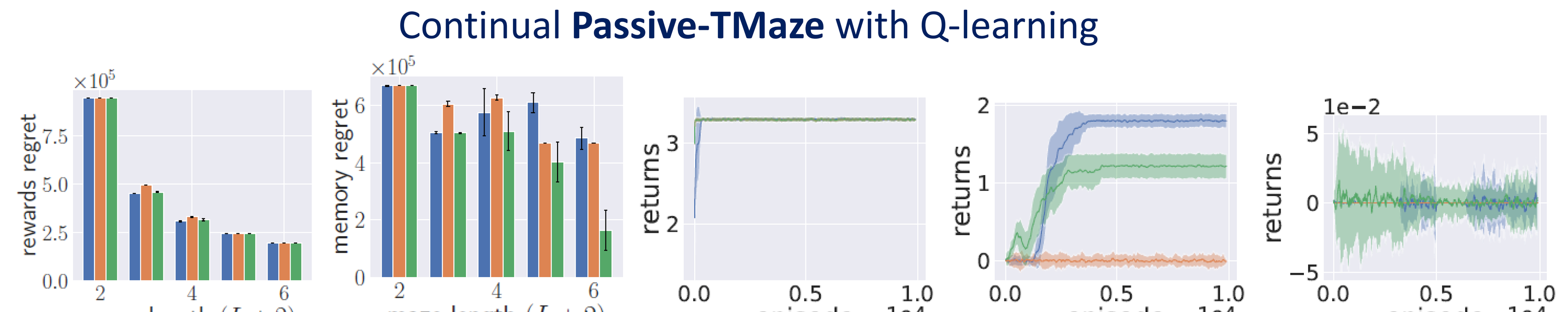
Theorem 1 *Let* $\mathbb{A}$ *be an RL algorithm that converges under Frame Stacking with* $k \geq k^*$. *If* $\mathbb{A}$ *uses unbiased value estimates to learn optimal policies, then it also converges under Adaptive Stacking with* $k \geq \kappa$ *observations, assuming the policy class is sufficiently expressive.*

Training



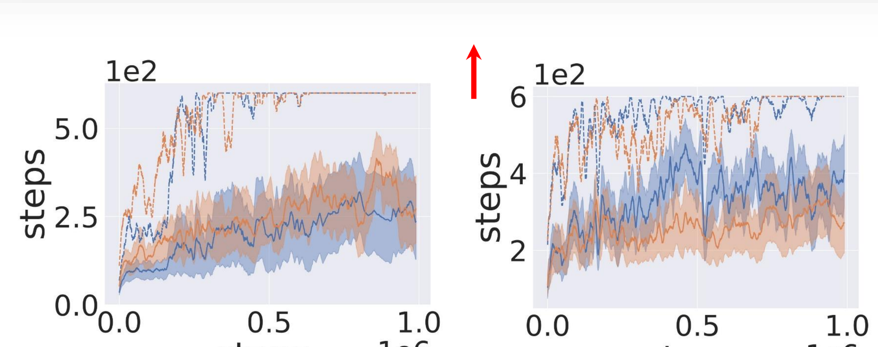
(a) **Rewards regret** (b) **Memory regret** (c) **Returns** $L=0$ (d) **Returns** $L=2$ (e) **Returns** $L=4$

Continual **Passive-TMaze** with Q-learning

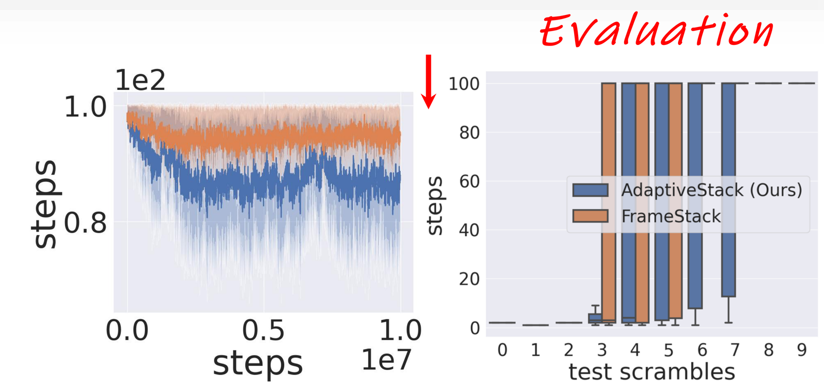


(a) **Rewards regret** (b) **Memory regret** (c) **Returns** $L=0$ (d) **Returns** $L=2$ (e) **Returns** $L=4$

Continual **Active-TMaze** with Q-learning

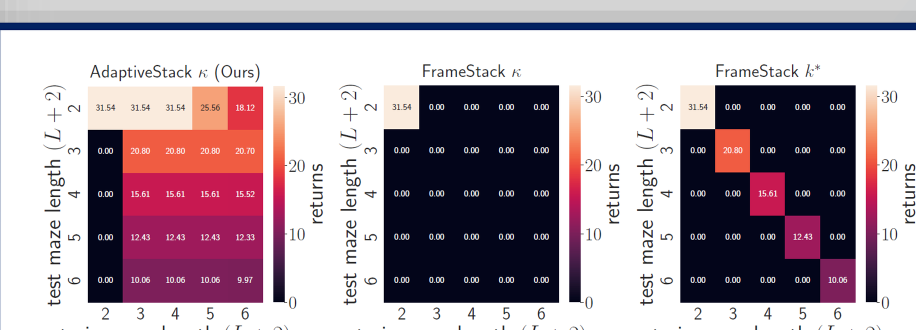


POPGym-CartPole PositionOnly - VelocityOnly

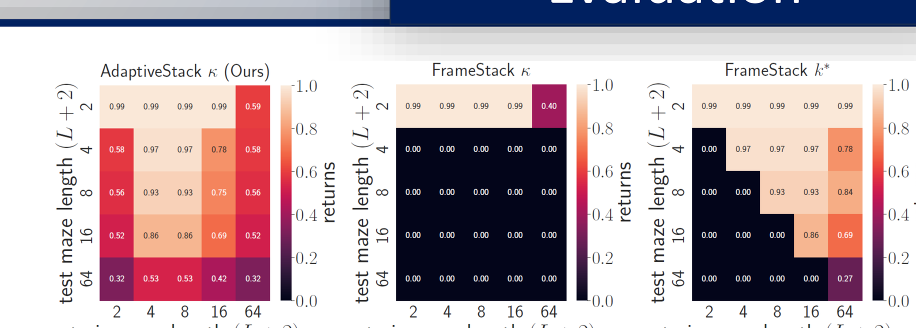


Rubik's-Cube 2x2x2 Front-view Camera

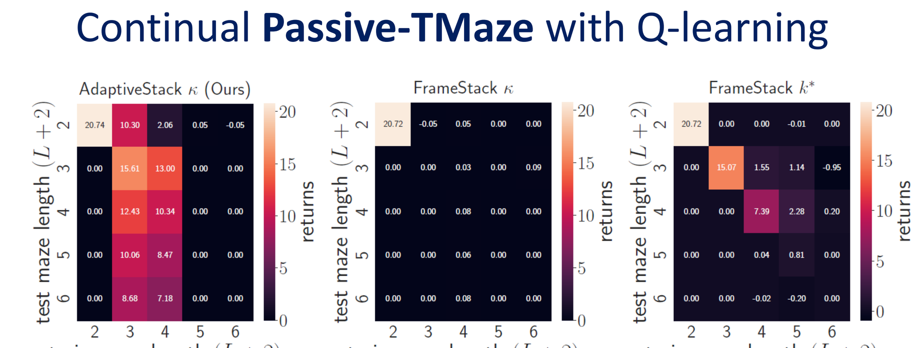
Evaluation



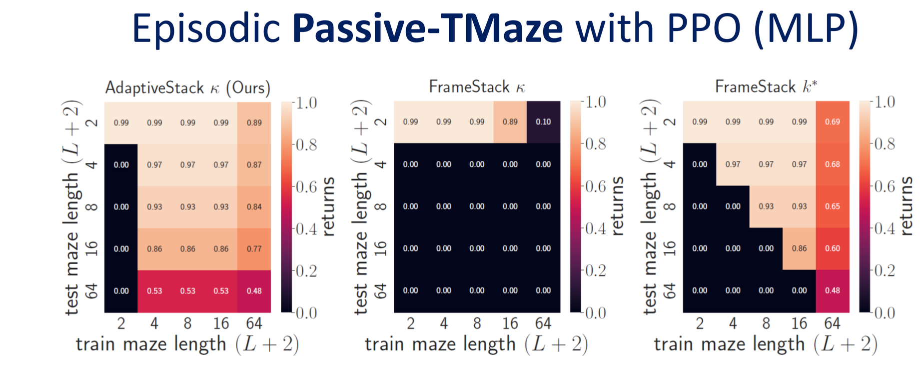
Continual **Passive-TMaze** with Q-learning



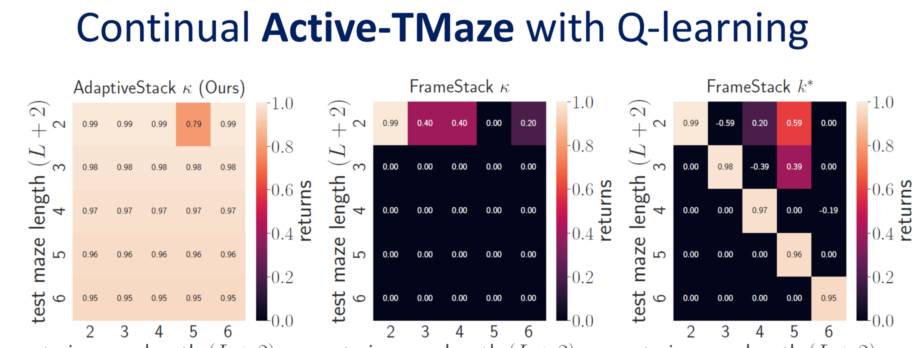
Episodic **Passive-TMaze** with PPO (MLP)



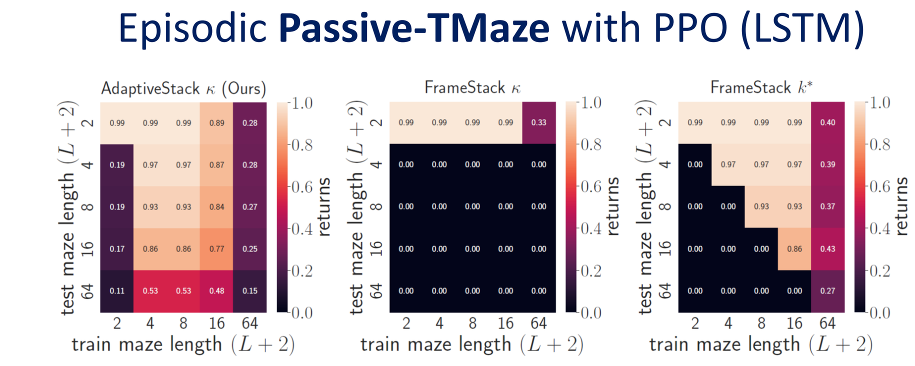
Continual **Active-TMaze** with Q-learning



Episodic **Passive-TMaze** with PPO (LSTM)

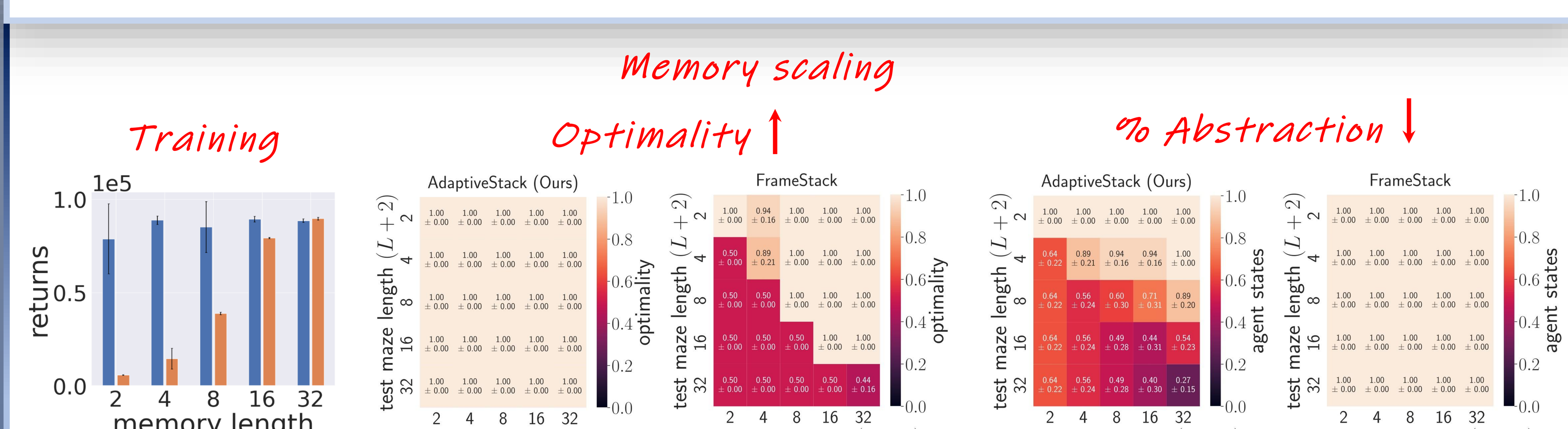


Episodic (Fix L) **Passive-TMaze** with PPO (MLP)



Episodic **Passive-TMaze** with PPO (Transformer)

Memory scaling



Episodic **Passive-TMaze** with PPO (MLP)