

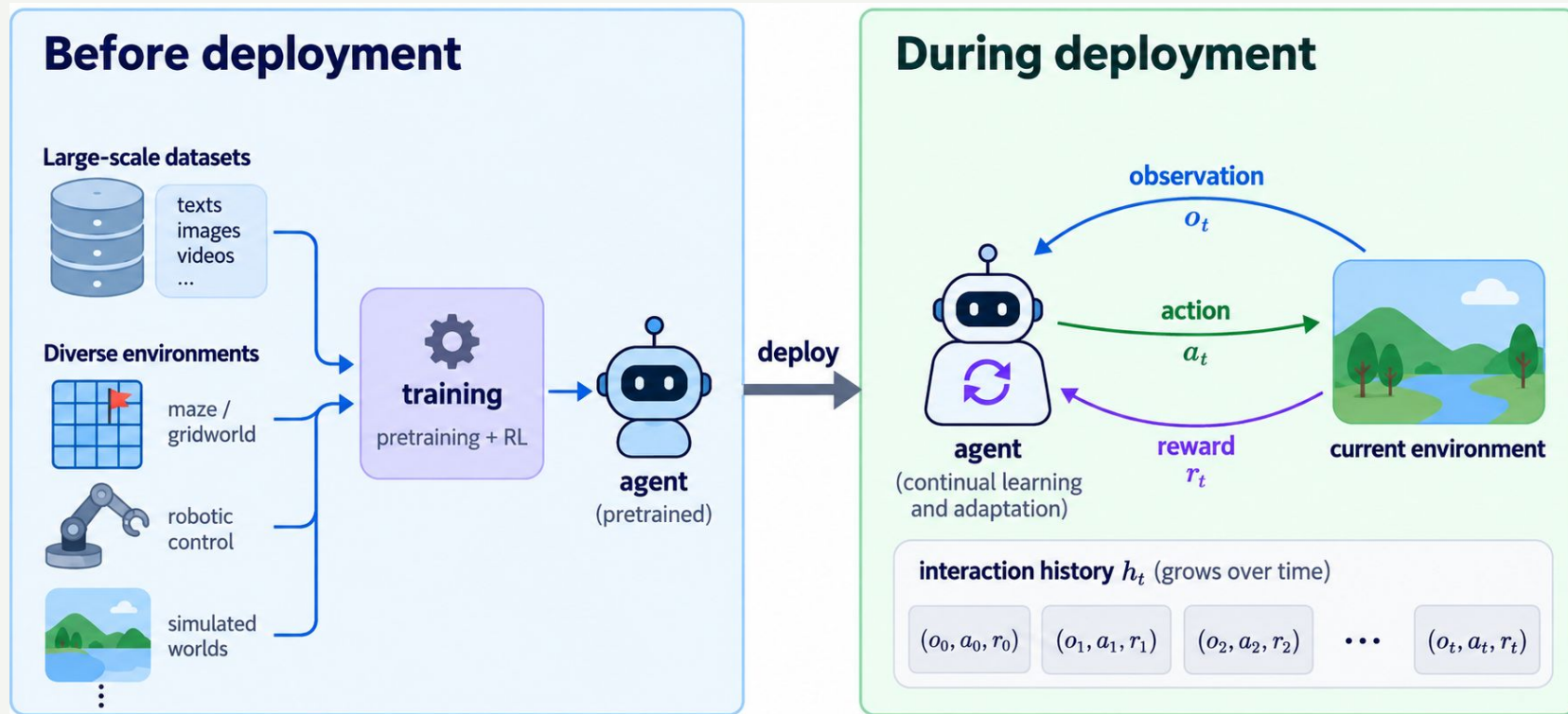
History-Dependent Reinforcement Learning: Memory, Representation, and Adaptability

Tianwei Ni's PhD defence

Mila & Université de Montréal

Sep 18, 2026

Modern agents adapt through interaction



History-dependent RL from scratch is hard

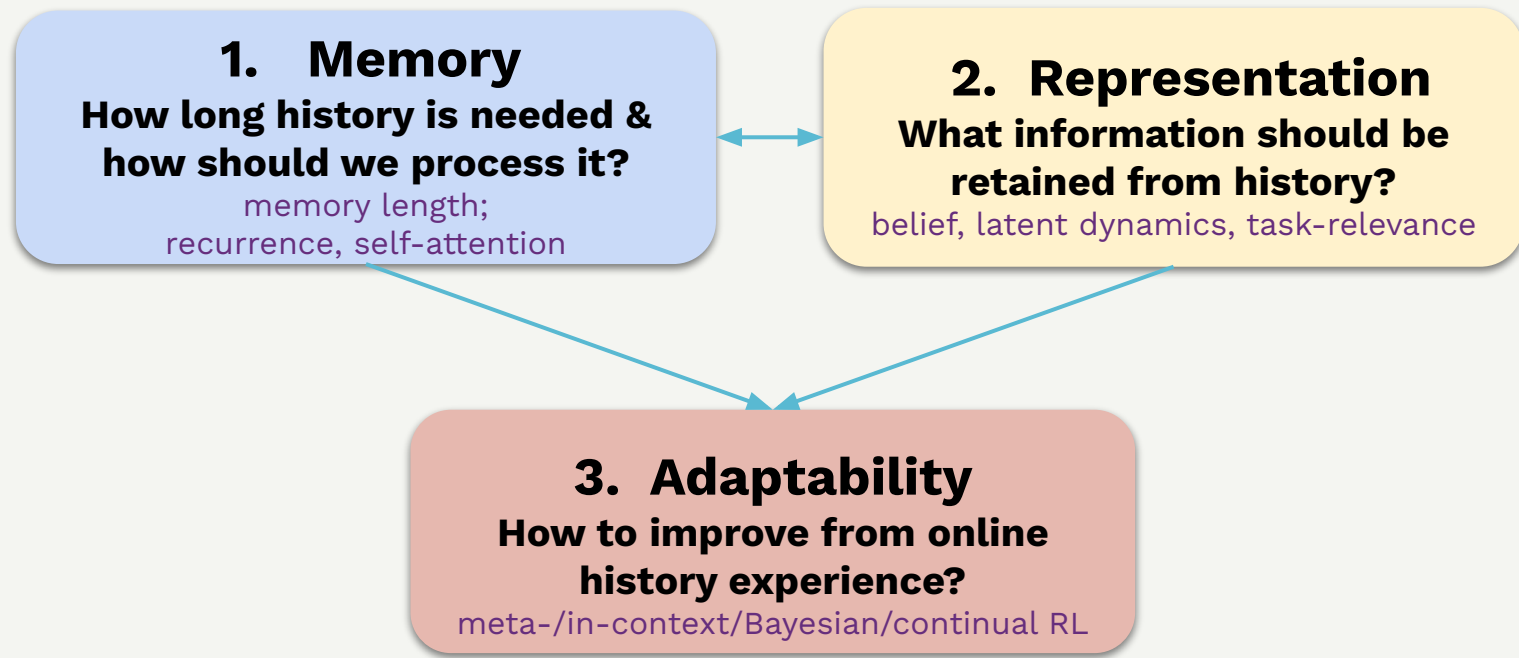
Planning and acting in partially observable
stochastic domains

Leslie Pack Kaelbling^{a,*,1,2}, Michael L. Littman^{b,3},
Anthony R. Cassandra^{c,1}

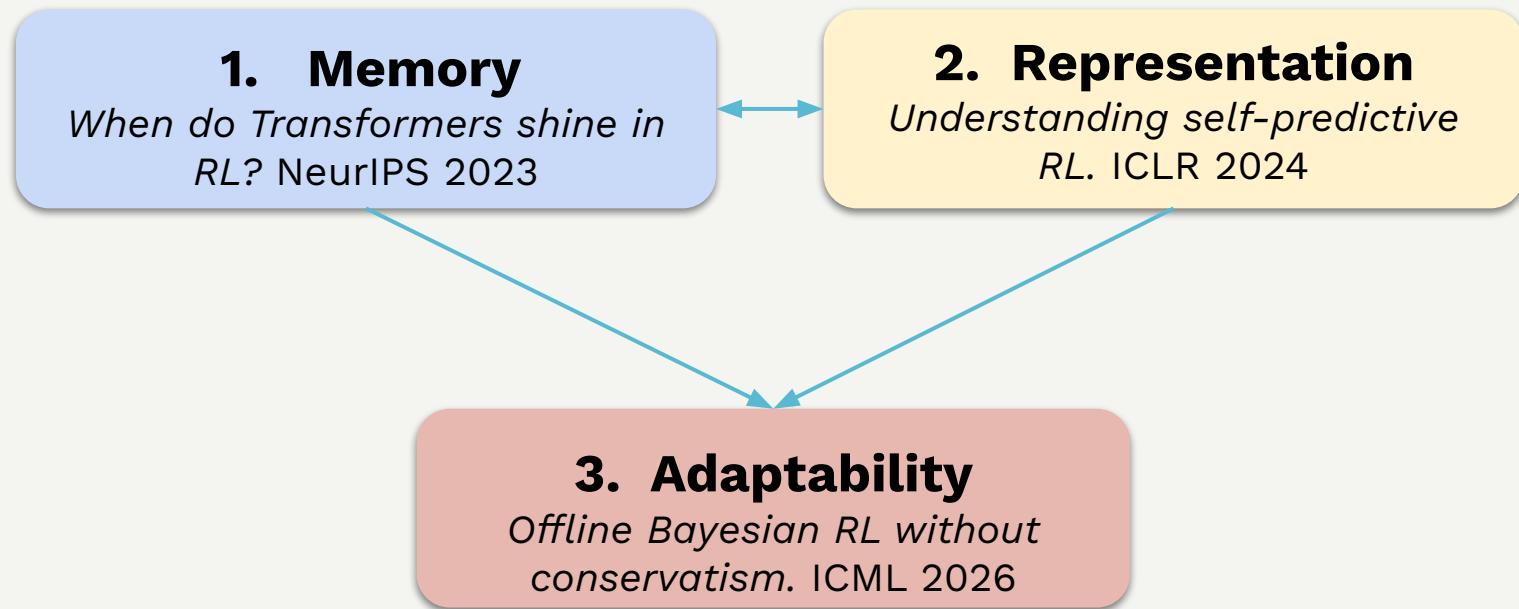
- Modern agents often use a simple recipe: Transformer + PPO.
 - But most capability comes from **pretraining**.
- Without pretraining, history-dependent RL is a classical challenge:
 - Under **partial observability**, the current observation is insufficient to determine the underlying state, so the agent must condition on its interaction history.
 - The history grows over time, making learning and planning difficult.

**This thesis studies *foundations* of history-dependent RL,
without assuming a pretrained model.**

History-dependent RL has three core questions



Thesis roadmap



1. Memory and Transformers in Long-horizon RL

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

NeurIPS 2023 Oral

Tianwei Ni

Mila, Université de Montréal
tianwei.ni@mila.quebec

Michel Ma

Mila, Université de Montréal
michel.ma@mila.quebec

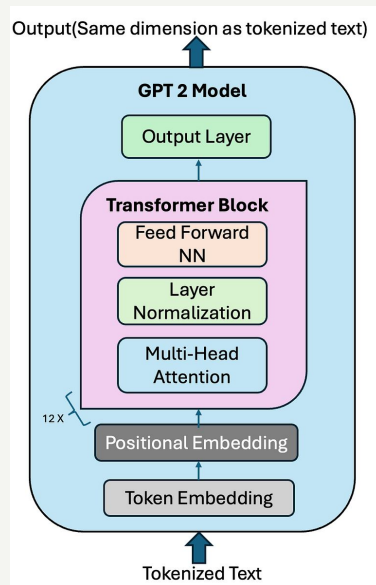
Benjamin Eysenbach

Princeton University
eysenbach@princeton.edu

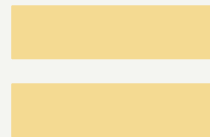
Pierre-Luc Bacon

Mila, Université de Montréal
pierre-luc.bacon@mila.quebec

Transformers have been successful in RL



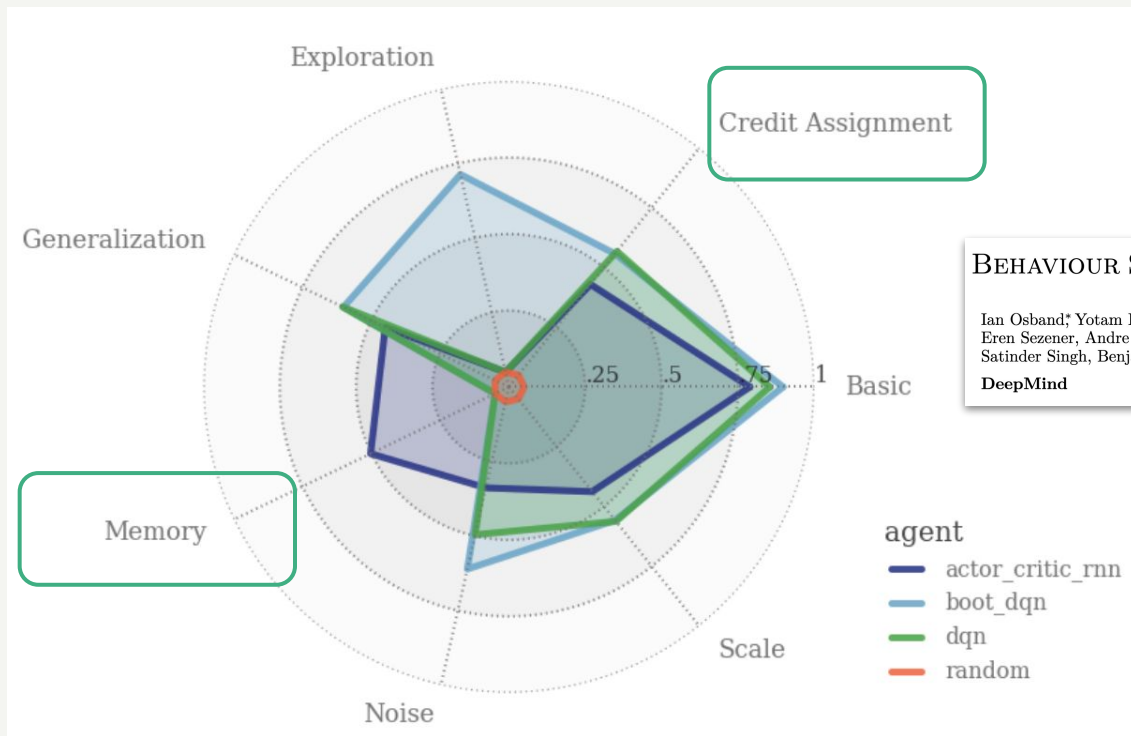
Any **RL**
algorithm
(PPO, DQN,
Dreamer)



Success?
Why successful?

Transformer (from NLP)

Why Transformers? Popular answer: Capture *temporal dependencies*.



BEHAVIOUR SUITE FOR REINFORCEMENT LEARNING

Ian Osband*, Yotam Doron, Matteo Hessel, John Aslanides
Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepesvari
Satinder Singh, Benjamin Van Roy, Richard Sutton, David Silver, Hado Van Hasselt
DeepMind

Memory vs Credit Assignment

TEMPORAL CREDIT ASSIGNMENT
IN REINFORCEMENT LEARNING

A Dissertation Presented
By
Richard S. Sutton

Memory

The ability to recall distant past events



Temporal credit assignment

The ability to determine *which past actions* earned the credit (Sutton, 1984)



Although we have intuition, we have not formalized these concepts in math.

Formalizing memory lengths

Intuition: How long is the minimal history required to predict current reward, observation, action, value?

Definitions

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

Theorem

For any optimal policy π^* with minimal policy memory length,

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

Optimal Policy memory length $\leq$ Optimal Value memory length
 $\leq \max(\text{reward memory length}, \text{transition memory length})$

Formalizing credit assignment lengths

Intuition: The optimal action $\mathbf{a}^*$ is better than any other action $\mathbf{a}'$ in terms of Q-value: $Q(s, \mathbf{a}^*) > Q(s, \mathbf{a}')$,

- where Q-value is the sum of n -step rewards with **$n=\text{infinity}$** .
- But we aim to find **the smallest n** to hold this.

Informal Definition

For an optimal action $a_t^* \in \operatorname{argmax}_{a_t} Q^*(h_{1:t}, a_t)$, 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).

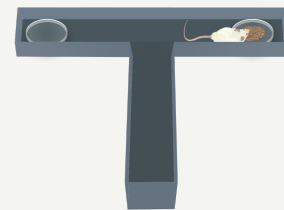
Characterizing existing RL *benchmarks*

	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	≤ 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	≤ 18
	Repeat First (Morad et al., 2023)	831	2	T	1
	Ballet (Lampinen et al., 2021)	1024	≥ 464	T	short
	Passive Visual Match (Sec. 5.1; Our experiment)	1030	T	T	≤ 18
	17 ² MiniGrid-Memory (Chevalier-Boisvert et al., 2018)	1445	≤ 51	T	≤ 51
	Passive T-Maze (Eg. 1; Ours)	1500	T	T	1
	15 ² Memory Maze (Pasukonis et al., 2022)	4000	long	long	≤ 225
	HeavenHell (Esslinger et al., 2022)	20	T	T	T
	T-Maze (Bakker, 2001)	70	T	T	T
Credit Assignment	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
	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

Many tasks entangle
memory and credit
assignment.

Most memory tasks
evaluate **memory length** $\leq$
1000

Proposing configurable toy tasks: Passive and Active T-Mazes



Passive T-Maze



Mem len: T
CA len: 1

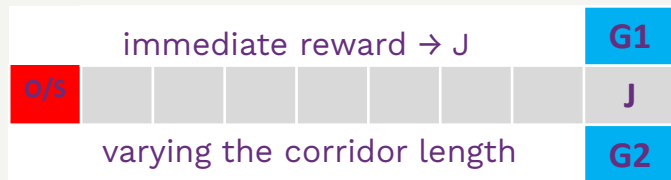
- Agent starts at state **S**
- Goal state is randomized between **G1** and **G2**
- State **O** tells the agent where the goal state is.
- At state **J**, it must make a choice

Active T-Maze

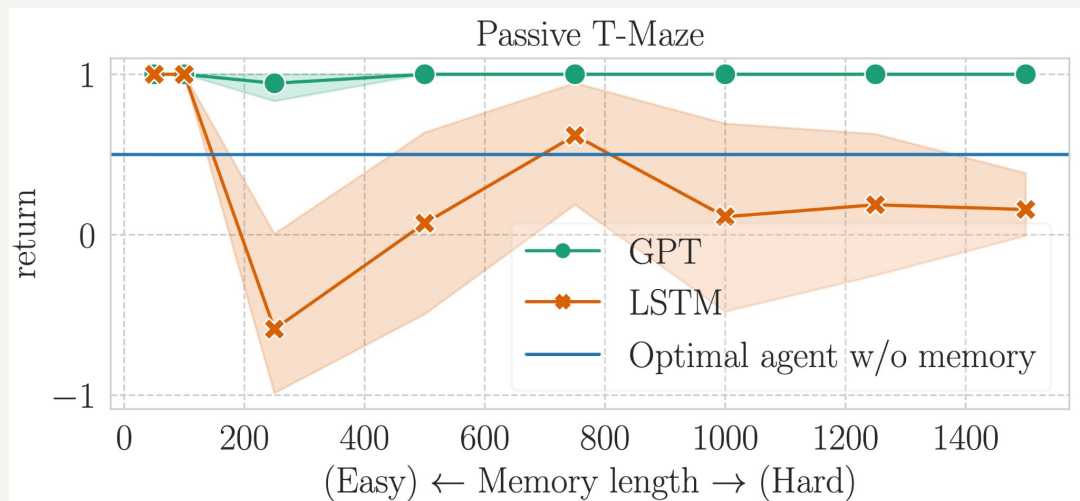


Mem len: T
CA len: T
(the first action of S → O is only rewarded at the final step)

Transformer-based RL excels at long-memory tasks.

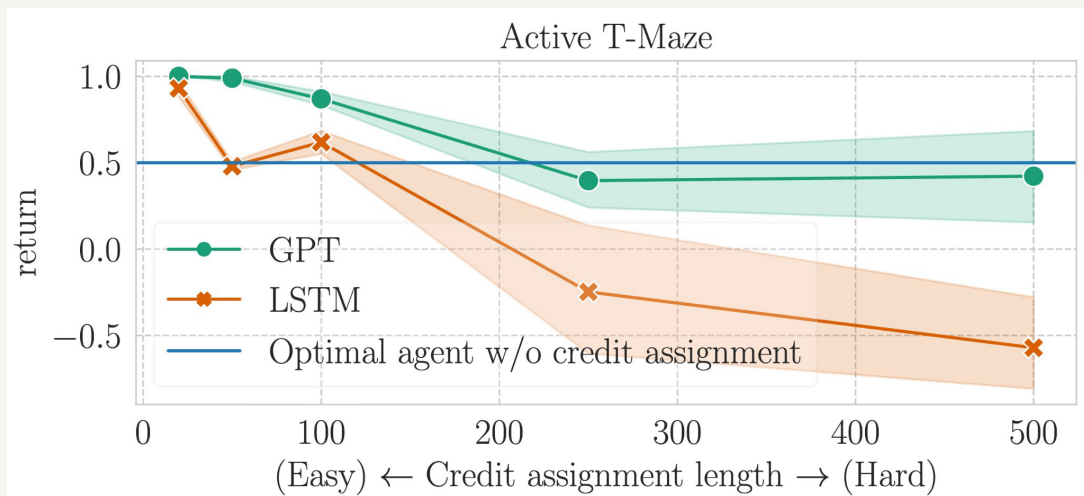
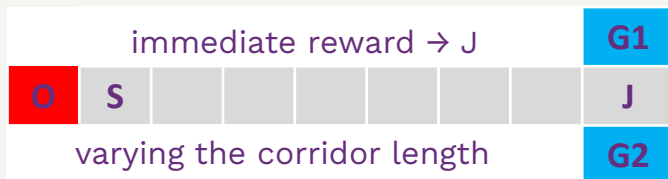


RL algorithm: DDQN w/ eps greedy
Sequence model: LSTM or Decoder-only Transformer (GPT-2)
Context length: full episode length



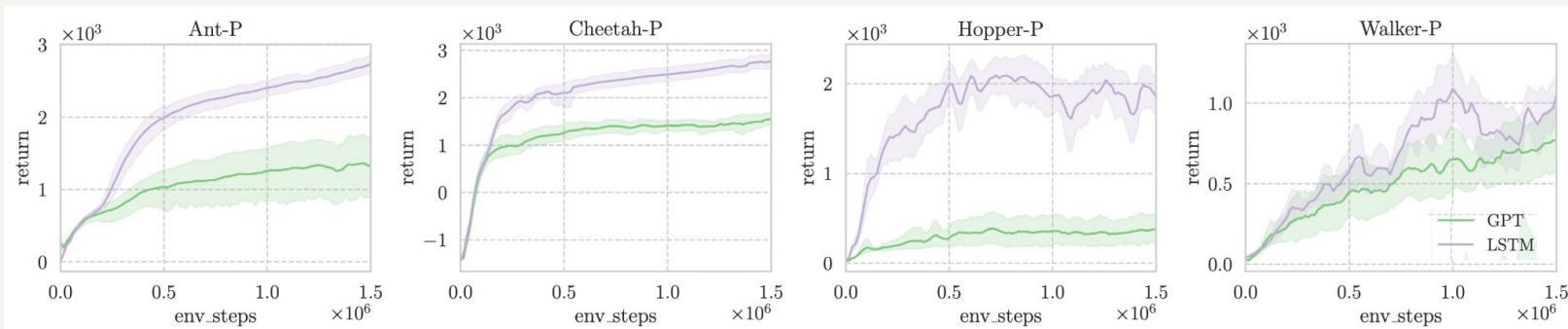
Perfectly solved by
GPT-2

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



In Active T-Maze, Transformers can reach the Junction. Transformers help, **but degrade severely when the CA length ≥ 250 .**

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



Using model-free TD3 in occluded continuous control tasks with *dense* rewards.

Contributions

- Memory and credit assignment lengths are mathematically separable.
- Transformers' success in RL mainly stems from memory, not credit assignment.
- Transformer-based RL cannot solve core RL challenges (credit assignment) alone and remains data-hungry.

2. What to Abstract from History in RL?

BRIDGING STATE AND HISTORY REPRESENTATIONS:
UNDERSTANDING SELF-PREDICTIVE RL

ICLR 2024

Tianwei Ni♣, Benjamin Eysenbach♣, Erfan Seyedsalehi◇*, Michel Ma♣, Clement Gehring♣,
Aditya Mahajan◇, Pierre-Luc Bacon♣

♣Mila, Université de Montréal, ♣Princeton University, ◇Mila, McGill University

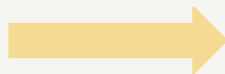
Is reward enough for RL?

Reward is enough

David Silver*, Satinder Singh, Doina Precup, Richard S. Sutton

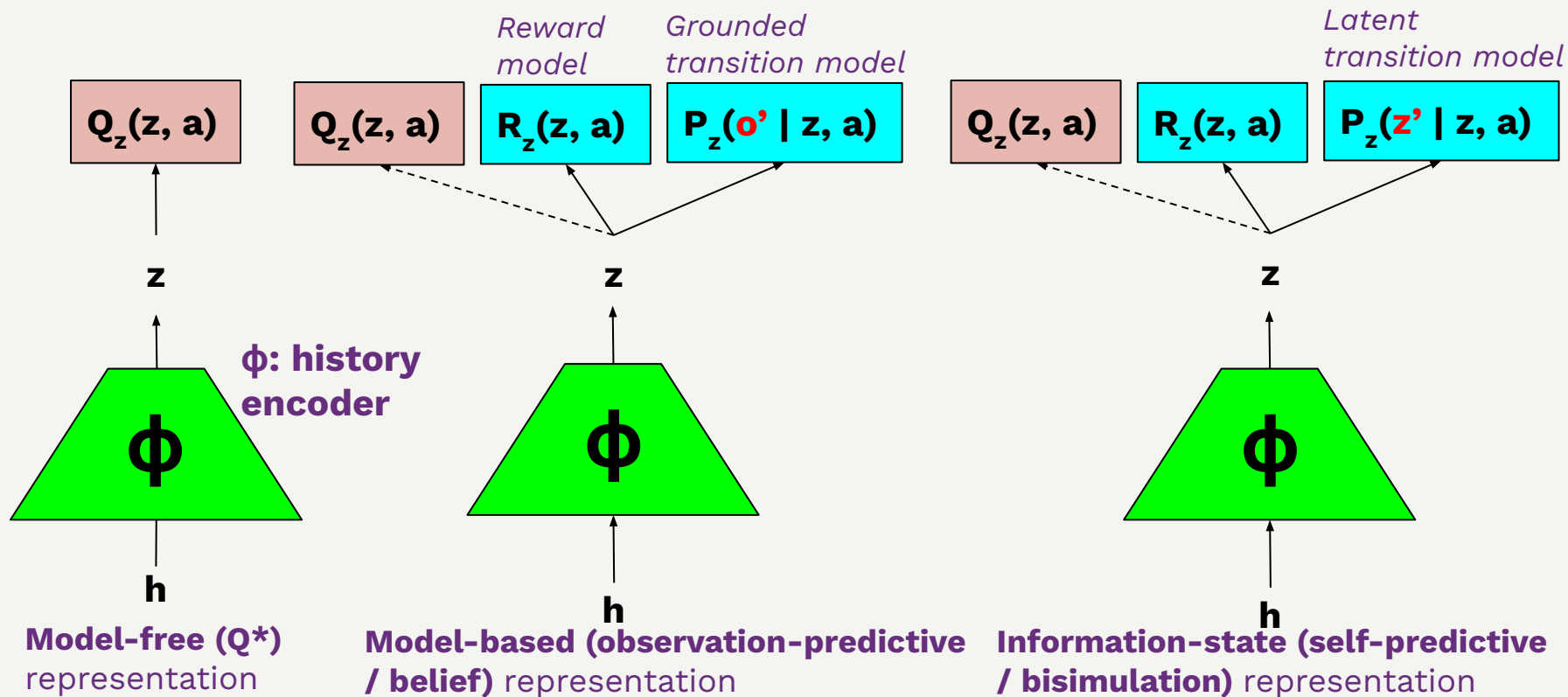
RL is terrible as it “sucks supervision through a straw”.

By Andrej Karpathy

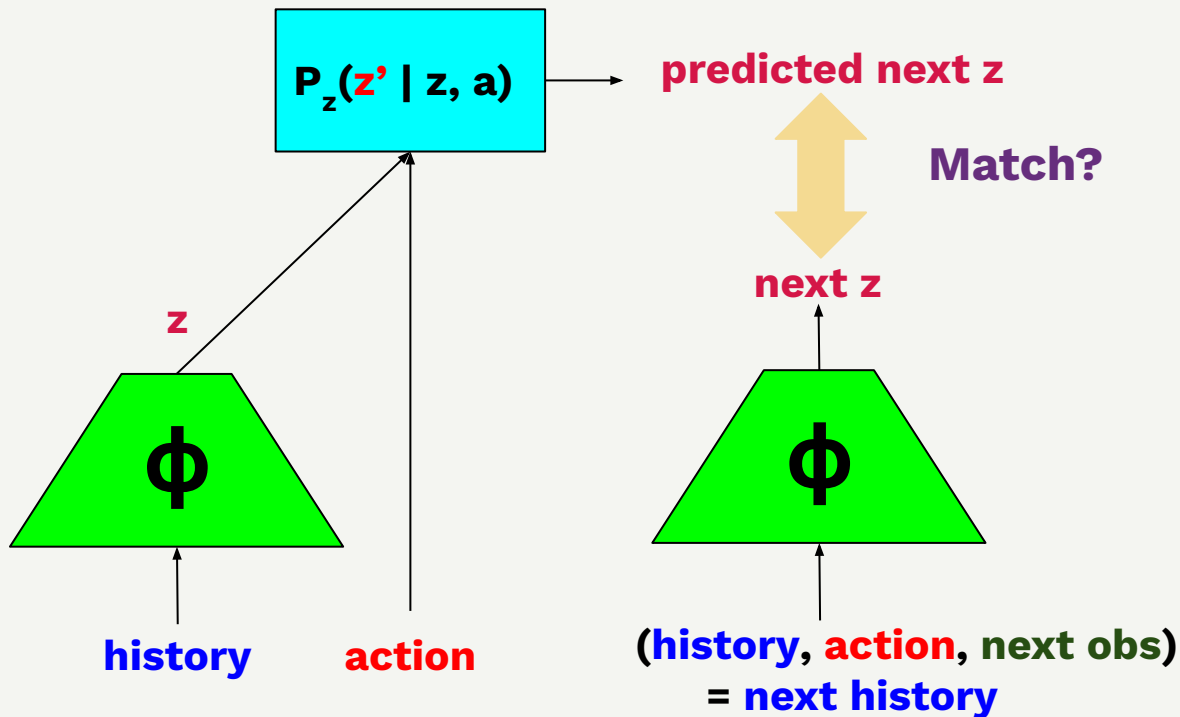


What *auxiliary* information should a learning agent abstract from histories?

Three representation paradigms

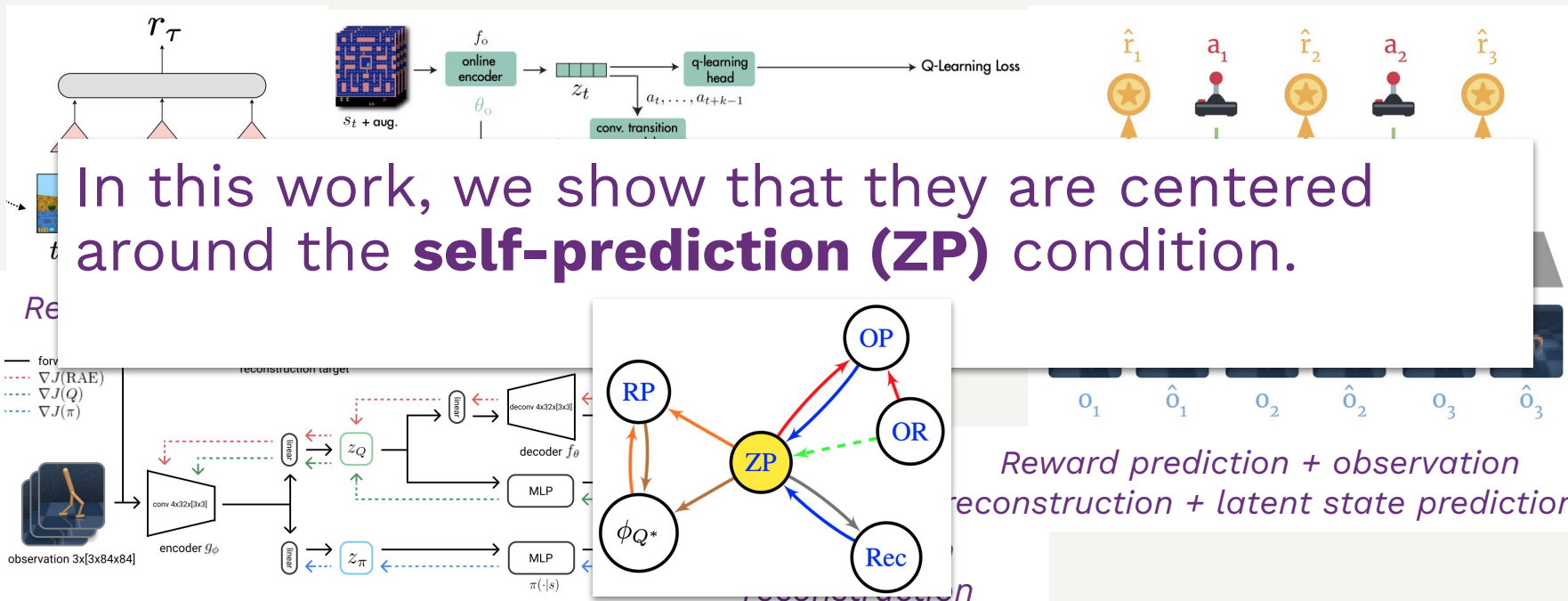


Self-prediction (ZP): predict next latent states without reconstruction

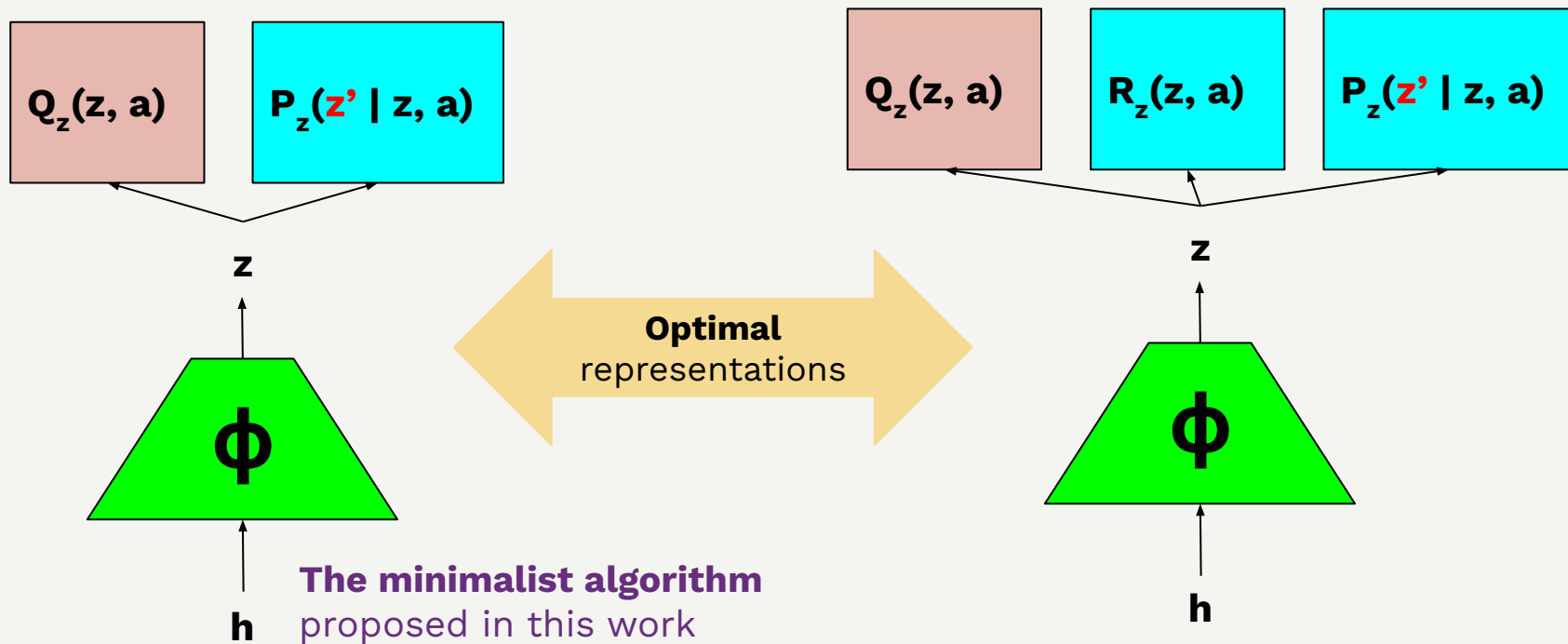


$$\exists P_z : \mathcal{Z} \times \mathcal{A} \rightarrow \Delta(\mathcal{Z}), \quad \text{s.t.} \quad P(\mathbf{z}' | h, a) = P_z(\mathbf{z}' | \phi_L(h), a), \quad \forall h, a, \mathbf{z}' \quad (\text{ZP})$$

Many representation learning methods share one principle



Key finding: $Q^* + ZP \rightarrow$ implicit reward modeling (RP)



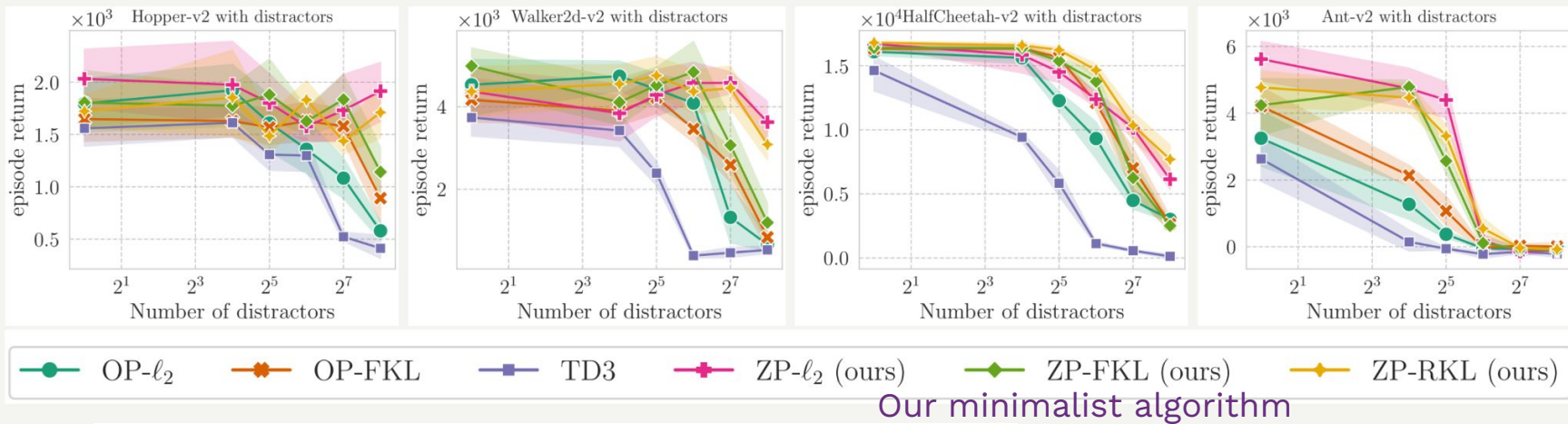
A spectrum of abstractions

Representation learning	Abstraction	Compute	Good at what tasks (See next slides)
Vanilla RL (Q^*) (e.g., PPO/SAC)	Coarse	Cheap	Dense rewards
RL + self-prediction (ZP) (e.g., SPR, TD-MPC)	Medium	Medium	High-dim/noisy observation
RL + observation-prediction (OP) (e.g., Dreamer)	Fine	Costly	Sparse rewards & Clean observation

Self-prediction makes RL robust to distractions

mujoco state

Gaussian noise from 1 to 2^8

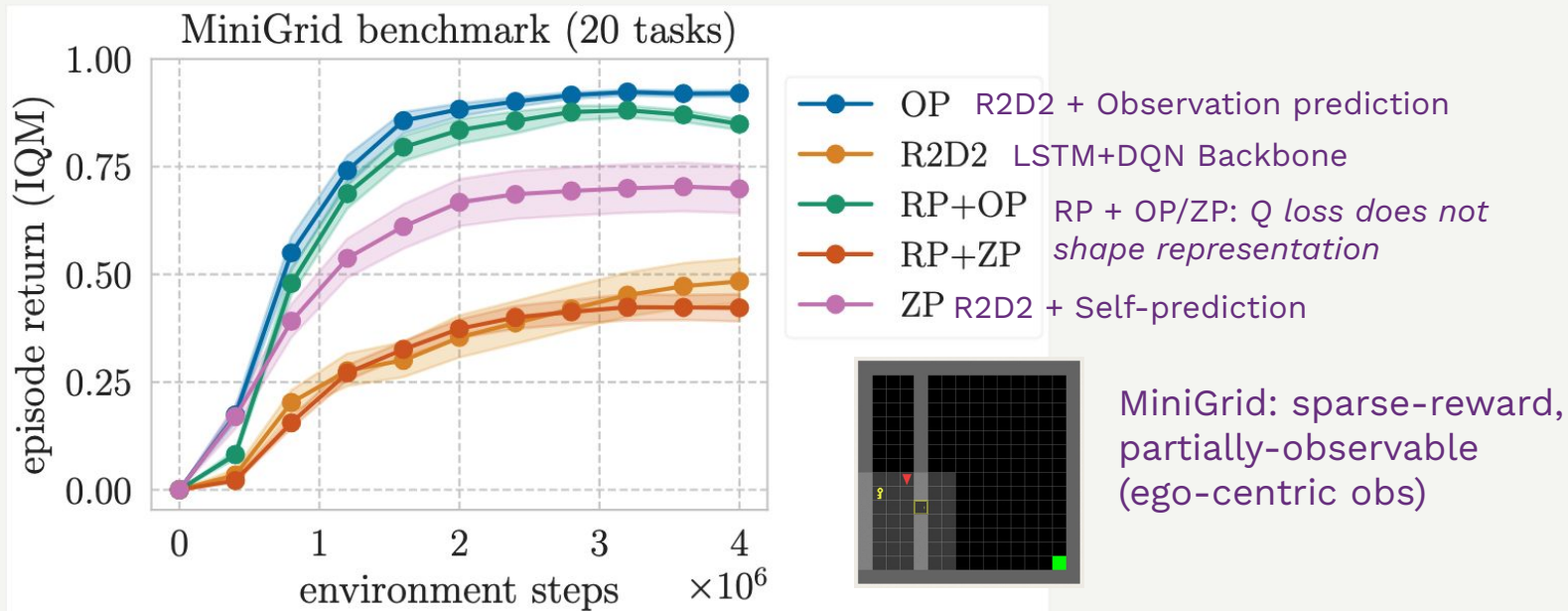


$$\mathbb{E}_{o' \sim P(|h, a)} \left[\|g_{\theta}(f_{\phi}(h), a) - f_{\tilde{\phi}}(h')\|_2^2 \right], \quad \text{l2 loss}$$

$$\mathbb{E}_{z \sim \mathbb{P}_{\phi}(|h), o' \sim P(|h, a)} \left[D_{\mathbf{f}} \left(\mathbb{P}_{\tilde{\phi}}(z' | h') \parallel \mathbb{P}_{\theta}(z' | z, a) \right) \right],$$

Forward or
Reverse KL

Self-prediction and observation prediction are critical for sparse rewards



Contributions

- We position the self-prediction condition (ZP) as the foundation uniting modern representation learning methods in RL.
 - Derive a minimalist algorithm for self-predictive learning.
- Optimal representation learning in RL is task-dependent:
 - Dense rewards: “reward is enough”
 - Noisy observations: self-prediction (ZP) is robust
 - Clean observations + Sparse rewards: observation-prediction (OP) is most sample-efficient

3. In-context Adaptability Trained with Offline RL

**Long-Horizon Model-Based Offline Reinforcement Learning
Without Explicit Conservatism**

ICML 2026

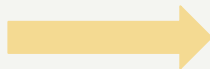
**Tianwei Ni^{1,2} Esther Derman^{1,2} Vineet Jain^{1,3} Vincent Taboga^{1,2}
Siamak Ravanbakhsh^{1,3} Pierre-Luc Bacon^{1,2}**

From passive prompting to active in-context adaptation

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

1	Translate English to French:	← task description
2	sea otter => loutre de mer	← examples
3	peppermint => menthe poivrée	←
4	plush girafe => girafe peluche	←
5	cheese =>	← prompt

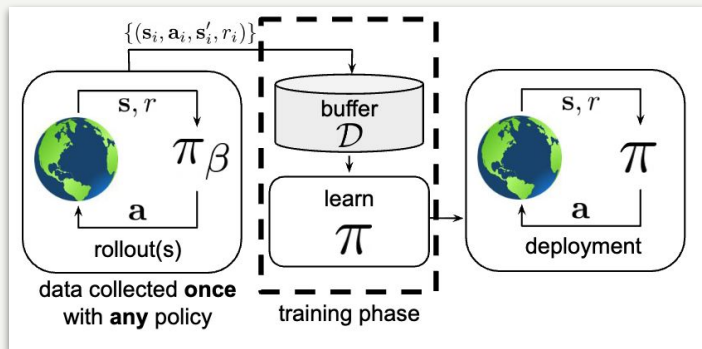


The deployment itself will involve a learning trial-and-error period.
By Ilya Sutskever

GPT-3 popularized few-shot in-context learning at test time, notably without parameter update.
But the context is given by prompt.

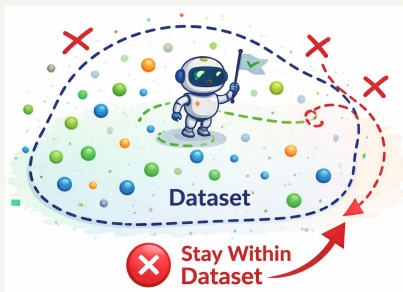
Context must be discovered by agent interaction with test-time environment.

Offline RL is not ready for in-context adaptation



Why study offline RL?

- Closest RL analogue to supervised pretraining
- Leverage diverse, static data without costly online training



However, the dominant principle, **conservatism**, suppressed this active in-context learnability.

- Penalize out-of-dataset / uncertain actions
- Designed for risk-averse static deployment, not for test-time exploration

Examples: behavioral cloning (TD3+BC), conservative Q-learning (CQL, IQL), uncertainty penalty (MOPQ)

Conservatism vs Bayesian principle

$$\max_{\pi} \min_{\mathcal{M} \in \mathfrak{M}_{\mathcal{D}}} \mathbb{E}_{\pi, \mathcal{M}} \left[\sum_t r_t \right]$$



$$\max_{\pi} \mathbb{E}_{\pi, \mathcal{M} \sim p(\mathcal{M}|\mathcal{D})} \left[\sum_t r_t \right]$$

History
dependence

MDP
posterior

Conservatism optimizes the (near) worst-case of possible MDPs

- Many MDPs can explain the offline data and disagree on unseen state-actions

Bayesianism optimizes the average case of possible MDPs (known as *epistemic POMDP* in prior work)

- Bayes-adaptive policy uses **online history** to infer the current MDP
- Requires a **Bayesian world model**

Bandit example for why Bayesian



$R(\text{arm}=0)$
 $\sim \text{Bern}(\mathbf{p0}^*)$

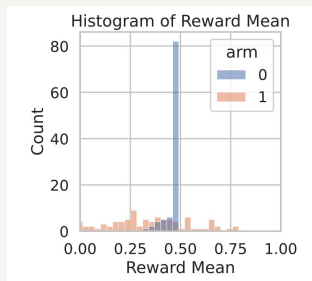


$R(\text{arm}=1)$
 $\sim \text{Bern}(\mathbf{p1}^*)$

- Horizon $T = 100$
- **Dataset D: full of trajectories with arm 0**
 - Traj: (arm 0, rew 0, arm 0, rew 0, ...)
 - Arm 1 is unobserved, uncertain

At deployment, should the agent avoid Arm 1 forever or try it?

Uncertainty penalty blocks adaptation



- Uncertainty penalty: ensemble disagreement
- Sufficient conservatism: heavily penalize arm 1 → **always choose arm 0, never try**
- Bayesian RL: infer the underlying arm 1 at test time → **adapt to the better arm**

exploration cost

adaptation to unseen, better action

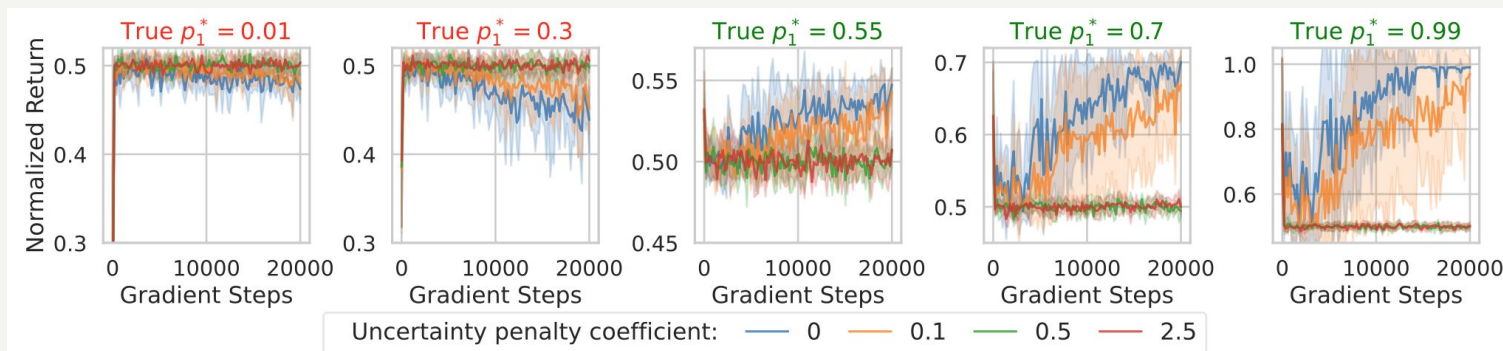


Figure 2. Average return (normalized by T) on **test-time bandits** with different p_1^* . Since the observed arm has $p_0^* = 0.5$, cases with $p_1^* < 0.5$ are worse and those with $p_1^* > 0.5$ are better. Zero penalty: Bayesian agent; nonzero penalty: conservative agent.

Dilemma: Removing conservatism causes value overestimation

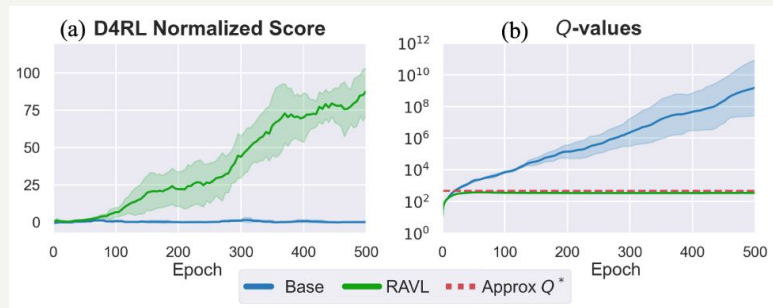
Algorithm 1 Framework for Model-based Offline Policy Optimization (MOPO) with Reward Penalty

Require: Dynamics model $\hat{T}$ with admissible error estimator $u(s, a)$; constant λ .

1: Define $\tilde{r}(s, a) = r(s, a) - \lambda u(s, a)$. Let $\tilde{M}$ be the MDP with dynamics $\hat{T}$ and reward $\tilde{r}$.

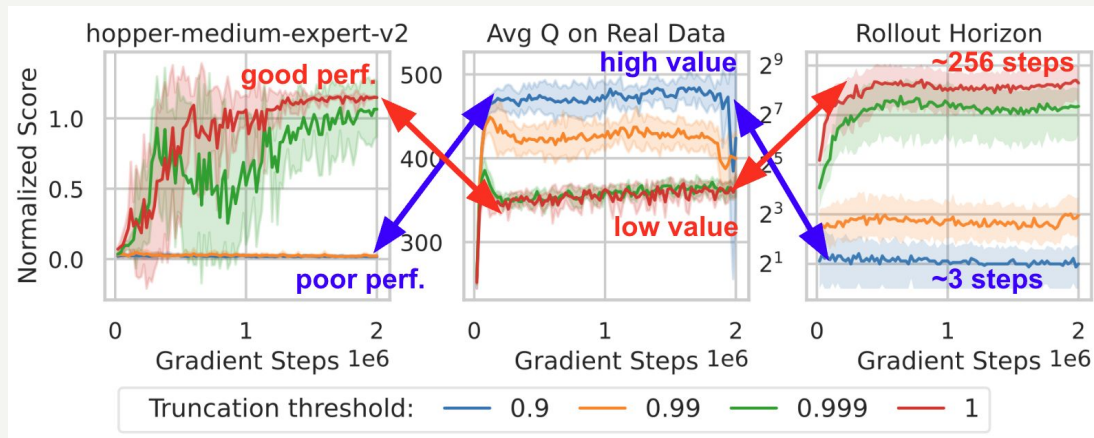
2: Run any RL algorithm on $\tilde{M}$ until convergence to obtain $\hat{\pi} = \operatorname{argmax}_{\pi} \eta_{\tilde{M}}(\pi)$

Fig src: The Edge-of-Reach Problem in Offline Model-Based Reinforcement Learning. Sims et al. NeurIPS 2024.



remove
penalty

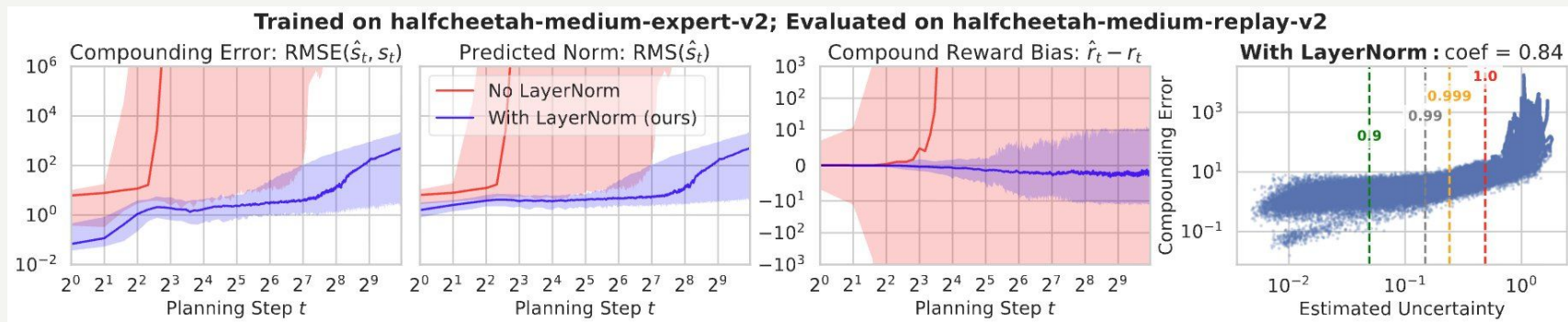
Mechanism: Long-horizon rollouts reduce overestimation



$$Q^{\text{Bayes}}(h_t, \hat{a}_t) \leftarrow \sum_{j=0}^{H-1} \underbrace{\gamma^j \hat{r}_{t+j+1}}_{\text{lower bias}} + \underbrace{\gamma^H}_{\text{discounted}} \underbrace{Q^{\text{Bayes}}(\hat{h}_{t+H}, \pi(\hat{h}_{t+H}))}_{\text{higher bias}}.$$

- Short rollouts rely heavily on **value bootstrapping** (overestimation)
- Long rollouts rely more on lower-biased model-generated rewards

Controlling long-horizon prediction errors



- (1) Simple LayerNorm can largely mitigate compounding errors
- (2) An uncertainty threshold (based on training data) for rollout truncation

Ensuring *reliably long* rollouts

NEUBAY algorithm

Algorithm 1 NEUBAY: Full training loop

Require: Offline dataset $\mathcal{D}$, Online buffer $\mathcal{B} \leftarrow \emptyset$. **World ensemble:** $\mathbf{m}_\theta$, **Recurrent actor:** π_ν , **Recurrent critic:** Q_ω .

```
1: Train  $\mathbf{m}_\theta$  on  $\mathcal{D}$  until convergence
2: while gradient steps  $\leq$  max gradient steps do
3:   for  $k = 1$  to  $K$  rollouts do  $\triangleright$  Parallelized in practice
4:     Sample  $m_\theta \sim \mathbf{m}_\theta$ , fixed for the rollout
5:     Sample initial history  $h_t \sim \mathcal{D}$ 
6:     Append ROLLOUT( $h_t, m_\theta, \pi_\nu$ ) to  $\mathcal{B}$ 
7:   end for
8:   for  $g = 1$  to  $G$  gradient steps do
9:     Optimize RL loss  $L(Q_\omega, \pi_\nu; \tau, \kappa)$  with  $\tau \sim \mathcal{B}$ 
10:   end for
11: end while
```

Algorithm 2 NEUBAY: Rollout function

Require: Terminal function f_{term} , Max episode length T , Uncertainty threshold $\mathcal{U}(\zeta)$, **Rollout horizon H**

```
1: function ROLLOUT( $h_t, m_\theta, \pi_\nu$ )
2:   Set  $\hat{h}_t = h_t$  and done to False
3:   while done is False do
4:      $\hat{a}_t \sim \pi_\nu(\hat{h}_t)$ ,  $(\hat{r}_{t+1}, \hat{s}_{t+1}) \sim m_\theta(\hat{s}_t, \hat{a}_t)$ 
5:      $\hat{d}_{t+1} = f_{\text{term}}(\hat{s}_t, \hat{a}_t, \hat{s}_{t+1})$ 
6:     trunc =  $(U_\theta(\hat{s}_t, \hat{a}_t) > \mathcal{U}(\zeta))$ 
7:     done =  $\hat{d}_{t+1} \vee \text{trunc} \vee (t + 1 \geq T)$ 
8:      $\hat{h}_{t+1} = (\hat{h}_t, \hat{a}_t, \hat{r}_{t+1}, \hat{s}_{t+1})$ ,  $t \leftarrow t + 1$ 
9:   end while
10:  return Trajectory  $\tau$  (i.e.,  $h_t$  plus imagined rollout)
11: end function
```

Eliminating two important hyperparameters in prior work: **(short) rollout length H and uncertainty penalty (λ)**, which hinder test-time in-context adaptability.

D4RL locomotion benchmark

Table 1. Comparison of offline RL methods on the **D4RL locomotion** benchmark. We report mean normalized scores for all baselines, with $\pm$ std for competitive baselines. The **best mean score** is bolded, and **marked** methods are statistically similar under a t -test. Our results use 6 seeds, each evaluated at the final step with 20 episodes.

	Model-free				Conservative model-based							Bayesian-inspired				Ours
Dataset	CQL	EDAC	COMBO	RAMBO	MOPO	LEQ	MOBILE	MoMo	ADMPO	SUMO	VIPO	APE-V	MAPLE	MoDAP	CBOP	NEUBAY
hc-random	61.3	28.4	38.8	40.0	38.5	30.8±3.3	39.3±3.0	39.6±3.7	45.4±2.8	34.9±2.1	42.5±0.2	29.9	41.5	36.5±1.8	32.8±0.4	37.0±3.3
hp-random	5.3	25.3	17.9	21.6	31.7	32.4±0.3	31.9±0.6	18.3±2.8	32.7±0.2	30.8±0.9	33.4±1.9	31.3	10.7	8.9±1.1	31.4±0.0	24.5±28.5
wk-random	5.4	16.6	7.0	11.5	7.4	21.5±0.1	17.9±6.6	26.8±3.3	22.2±0.2	27.9±2.0	20.0±0.1	15.5	22.1	23.1±1.6	17.8±0.4	34.1±6.8
hc-med-rep	45.3	61.3	55.1	77.6	72.1	65.5±1.1	71.7±1.2	72.9±1.8	67.6±3.4	76.2±1.3	77.2±0.4	64.6	69.5	67.3±3.4	66.4±0.3	72.1±2.4
hp-med-rep	86.3	101.0	89.5	92.8	103.5	103.9±1.3	103.9±1.0	104.0±1.8	104.4±0.4	109.9±1.4	109.6±0.9	98.5	85.0	94.2±4.8	104.3±0.4	110.6±0.7
wk-med-rep	76.8	87.1	56.0	86.9	85.6	98.7±6.0	89.9±1.5	90.4±7.7	95.6±2.1	78.2±1.5	98.4±0.3	82.9	75.4	88.4±4.2	92.7±0.9	99.3±19.3
hc-medium	46.9	65.9	54.2	68.9	73.0	71.7±4.4	74.6±1.2	77.1±0.9	72.2±0.6	84.3±2.4	80.0±0.4	69.1	48.5	77.3±1.1	74.3±0.2	78.6±1.6
hp-medium	61.9	101.6	97.2	96.6	62.8	103.4±0.3	106.6±0.6	110.8±2.3	107.4±0.6	104.8±2.1	107.7±1.0	—	44.1	106.6±1.9	102.6±0.1	54.2±7.2
wk-medium	79.5	92.5	81.9	85.0	84.1	74.9±26.9	87.7±1.1	95.0±1.4	93.2±1.1	94.1±2.5	93.1±1.8	90.3	81.3	81.1±6.5	95.5±0.4	106.4±23.0
hc-med-exp	95.0	106.3	90.0	93.7	90.8	102.8±0.4	108.2±2.5	107.9±1.9	103.7±0.2	106.6±2.4	110.0±0.4	101.4	55.4	103.4±4.3	105.4±1.6	109.5±8.7
hp-med-exp	96.9	110.7	111.1	83.3	81.6	109.4±1.8	112.6±0.2	109.1±0.4	112.7±0.3	107.8±0.7	113.2±0.1	105.7	95.3	94.5±7.8	111.6±0.2	114.8±0.5
wk-med-exp	109.1	114.7	103.3	68.3	112.9	108.2±1.3	115.2±0.7	118.4±0.9	114.9±0.3	122.8±0.4	117.7±1.0	110.0	107.0	112.2±2.8	117.2±0.5	120.7±1.3
AVG	61.6	76.0	66.8	68.9	70.3	76.9	80.0	80.9	81.0	81.5	83.6	—	61.3	74.5	79.3	80.1

- Matching leading conservative methods (**new SoTA** on 7 D4RL tasks)
- Better at **lower-quality** and **moderate-coverage** offline datasets
- Also evaluate on D4RL manipulation, navigation, and NeoRL benchmarks

Contributions

- Identify the root cause of why offline (model-based) RL cannot perform test-time in-context adaptation:
 - Short horizon rollouts
 - Conservatism on out-of-dataset actions
- Propose a simple and effective algorithm to have reliably long rollouts without explicit conservatism.

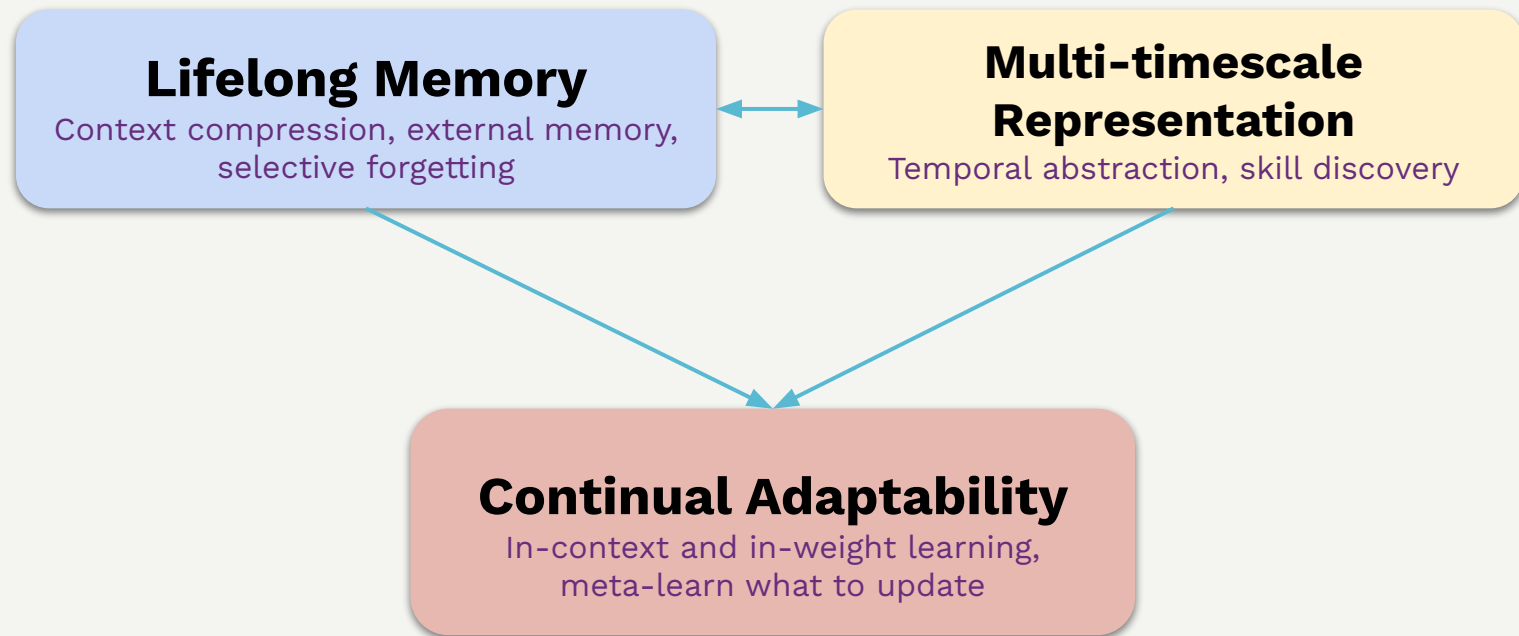
Future Directions

Toward lifelong agents

Challenges shift: learning from a transient history to a lifetime of experience.

- **Lifelong memory:** relevant experience spans the agent's entire lifetime.
 - Challenges: context explosion, memory pollution, length extrapolation
- **Representations across timescales:** low-level and high-level outcomes
 - Challenges: compounding errors with myopic representation
- **Continual adaptation:**
 - Challenges: plasticity loss, general-capability forgetting

Possible directions toward lifelong agents



Acknowledgments

Thank you to my Advisor, Collaborators & Committee

PhD Advisor



Pierre-Luc Bacon

Professors



Ben Eysenbach
(Princeton CS)



Aditya Mahajan
(McGill ECE)



Siamak Ravanbakhsh
(McGill CS)

Students



Michel Ma



Erfan Seyedsalehi



Vineet Jain

Postdocs



Clement Gehring



Esther Derman



Vincent Taboga

Thesis Committee



Glen Berseth



Pablo Samuel
Castro

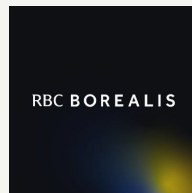


Amy Zhang

Acknowledgments: Community & Funding



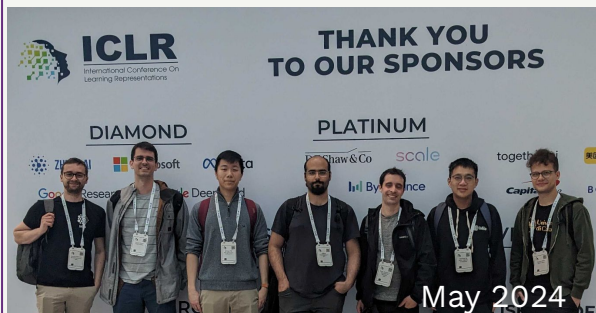
and more!



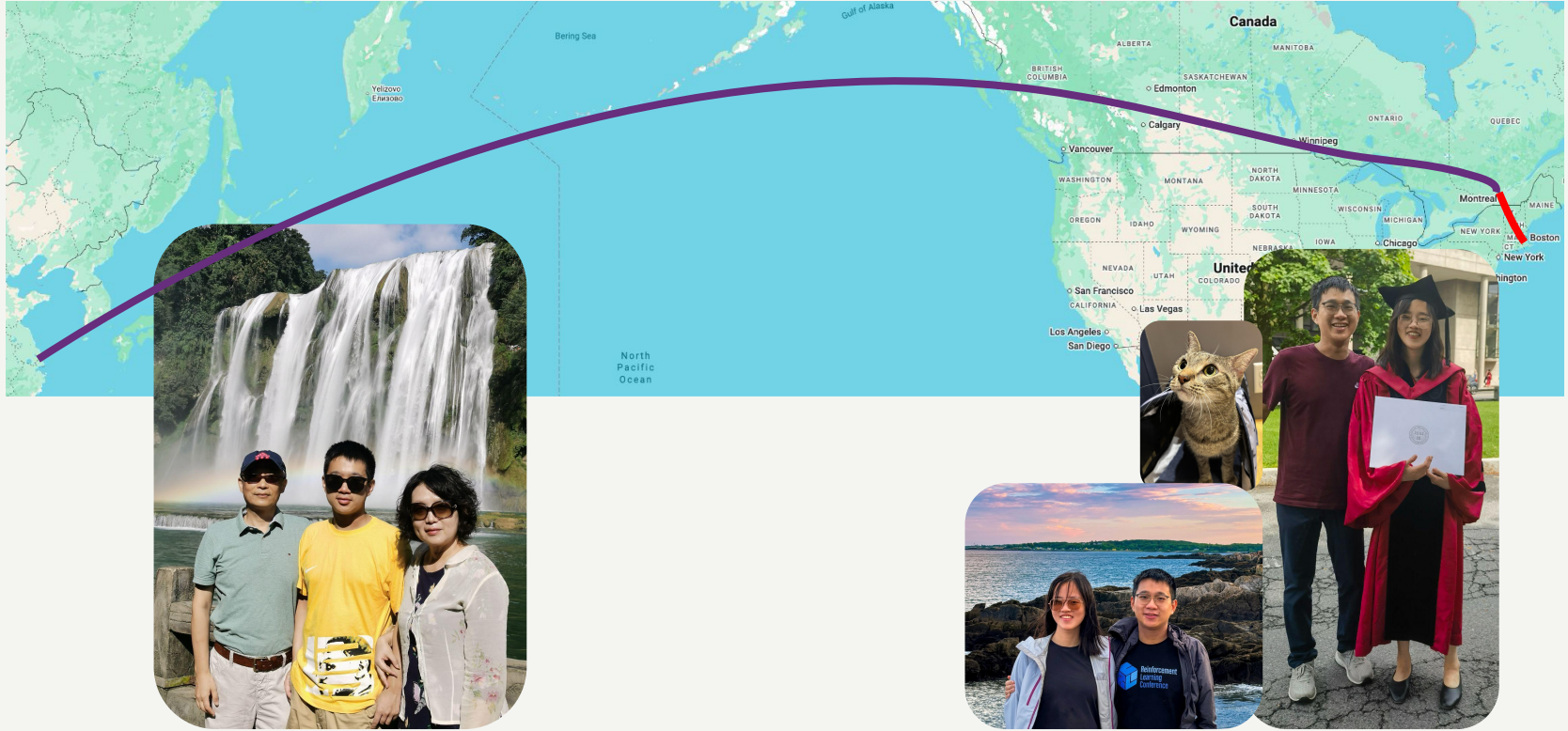
Memories with Plurl lab



Montréal 



Thanks to my family



Q&A