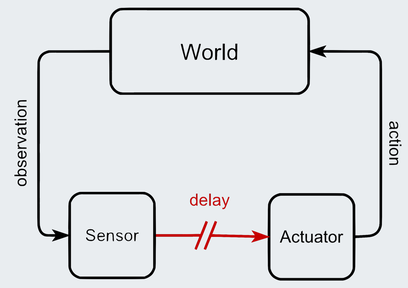
Model-Based Reinforcement Learning under Random Observation Delays

Armin Karamzade, Kyungmin Kim, JB Lanier, Davide Corsi, Roy Fox

UCI University of California, Irvine

1. Delays Matter in RL

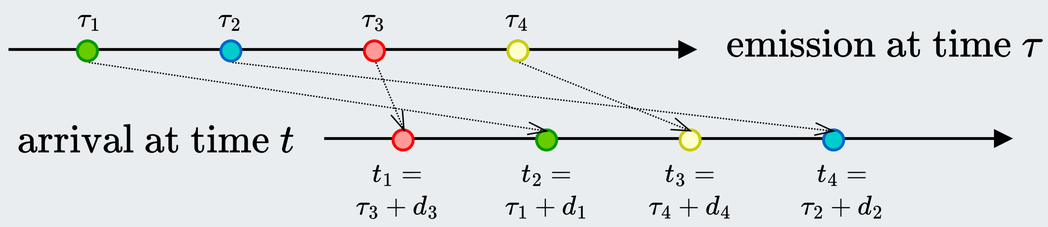
- Standard RL assumes instantaneous perception, yet sensing, processing, and communication delays are pervasive in robotics, autonomous driving, and distributed control.



- Waiting for delayed observations is often impractical or unsafe, e.g., a vehicle cannot pause while an obstacle approaches.
- Prior work assumes fully observable MDPs or fixed delays in POMDPs, neither captures real systems.

2. Setting

- Environment $\langle M, D \rangle$: a POMDP M and a delay distribution D. The observation emitted at time t arrives at $t + d_t$ where $d_t \sim D$.
- Observations can arrive out-of-sequence: unique to random delays under partial observability.



Reduction to a standard POMDP: augmenting the state with a buffer of pending observations yields an equivalent delay-free environment.

→ but state and observation spaces blow up exponentially, and the belief must also track pending observations.

3. Out-of-Sequence Filtering via World Model

Goal: the belief $\phi_t = p(x_t \mid \tilde{o}_{\leq t}, a_{<t})$ over the latent state, given observations $\tilde{o}_{\leq t}$ that have actually arrived.

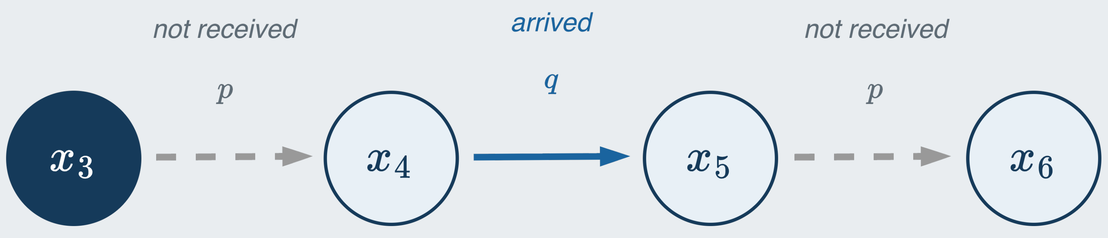
Auxiliary kernel ψ retroactively folds in late arrivals:

$$\psi(x_\tau \mid x_{\tau-1}, \tilde{o}_{\leq t}, a_{\tau-1}) = \begin{cases} q(x_\tau \mid x_{\tau-1}, o_\tau, a_{\tau-1}) & \text{if } o_\tau \text{ has been received} \\ p(x_\tau \mid x_{\tau-1}, a_{\tau-1}) & \text{otherwise} \end{cases}$$

Roll forward from the last fully received prefix:

$$\phi_t = E_q E_\psi \ldots E_\psi [\psi(x_t \mid x_{t-1}, \tilde{o}_{\leq t}, a_{t-1})]$$

this gives the exact Bayesian filter for the available history.



Update with the posterior q when an observation is available; fall back to the prior p when it is still pending. Belief approximated with K particles — in practice, K = 1 suffices thanks to the RSSM's deterministic path.

4. Delay-Aware MBRL

We introduce a Delay-Aware framework for incorporating belief inference under random observation delays into model-based RL.

Algorithm 1 Delay-Aware MBRL

Input: $\mathcal{A}$: Model-Based RL algorithm optimizing objective (1)
`// Inference Mode`
Initialize an empty observation buffer
for *time t in episode* **do**
  Receive newly arrived observations and add them to the buffer
  Roll forward using ψ_t from the last checkpoint to compute the belief ϕ_t
  Checkpoint the belief at the earliest missing observation; discard older buffer entries
  Execute action $a_t \sim \pi(\cdot \mid \phi_t)$
end
`// Training Mode`
foreach *update step* **do**
  Collect data with π using inference mode;
   store ordered trajectories with delays $d_{<T}$ in replay buffer B
  Use $\mathcal{A}$ to update world model (p, q) as in the undelayed setting
  Replay the delays to recompute beliefs $\phi_{<T}$
  Use $\mathcal{A}$ to update policy $\pi(\cdot \mid \phi_t)$ and critic $V(\phi_t)$, if applicable
end

Modifications to the standard pipeline are highlighted in blue.

5. Results

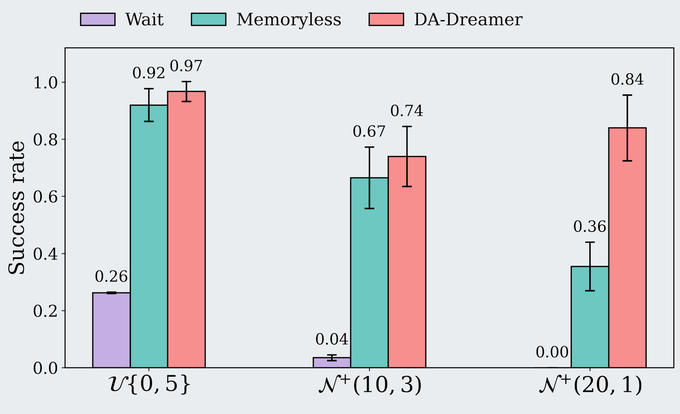
(a) **Outperforms delay-aware baselines.** DA-Dreamer wins in most environments and dominates on hard, high-dimensional tasks (Humanoid, HumanoidStandup); it is also the most robust as action noise increases.

(b) **Holds up where heuristics break.** Wait strategy fails almost everywhere; Memoryless approach collapses as delays grow.

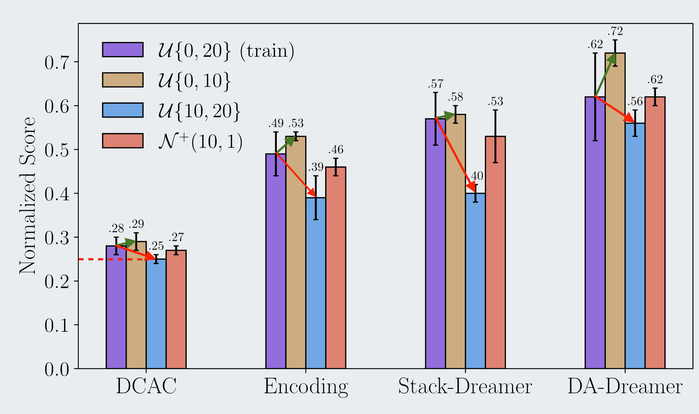
(c) **Generalizes to unseen delays.** Delay patterns are rarely known in advance. DA-Dreamer exploits shorter delays and degrades less under longer ones in test-time.

Delay	Environment	DCAC [1]	Encoding [2]	Stack-Dreamer	DA-Dreamer
$\mathcal{U}\{0, 10\}$	HalfCheetah-v4	3841.43 ± 802.96	4189.26 ± 401.81	1959.96 ± 839.58	**4985.40 ± 253.10**
	Hopper-v4	2394.67 ± 721.94	2373.70 ± 206.46	2694.22 ± 416.69	2251.36 ± 413.62
	Humanoid-v4	1062.81 ± 248.87	552.21 ± 31.89	522.13 ± 41.48	**1854.26 ± 205.04**
	HumanoidStandup-v4	145293.09 ± 4314.40	113240.31 ± 11024.15	93154.06 ± 17839.01	**220017.11 ± 23671.63**
	Reacher-v4	−6.36 ± 0.55	−6.79 ± 0.30	−6.81 ± 0.23	−6.37 ± 0.13
	Swimmer-v4	40.53 ± 1.55	121.56 ± 23.92	**346.58 ± 2.51**	**347.00 ± 5.64**
$\mathcal{U}\{0, 20\}$	HalfCheetah-v4	2144.86 ± 526.30	**4242.77 ± 213.47**	1045.15 ± 179.75	2958.56 ± 54.19
	Hopper-v4	6.48 ± 0.83	1707.41 ± 147.40	**2981.87 ± 275.14**	1713.85 ± 343.09
	Humanoid-v4	112.08 ± 76.92	545.76 ± 17.19	418.24 ± 42.17	**855.97 ± 95.77**
	HumanoidStandup-v4	139806.95 ± 20078.74	118456.86 ± 7541.70	101241.22 ± 4591.26	**195611.38 ± 14140.51**
	Reacher-v4	−6.59 ± 0.44	−7.17 ± 0.24	−6.71 ± 0.21	−7.06 ± 0.16
	Swimmer-v4	34.19 ± 3.38	117.84 ± 20.27	**348.32 ± 3.21**	**349.60 ± 3.53**

(a)



(b)



(c)

[1] Bouteiller, Yann, et al. "Reinforcement learning with random delays." International Conference on Learning Representations. 2020.

[2] Wang, Wei, et al. "Addressing signal delay in deep reinforcement learning." International Conference on Learning Representations. 2024.