Directly Forecasting Belief for Reinforcement Learning with Delays

Qingyuan Wu* [1] Yuhui Wang* [2] Simon Sinong Zhan* [3]
Yixuan Wang [3] Chung-Wei Lin [4] Chen Lv [5] Qi Zhu [3] Jürgen Schmidhuber [2, 6] Chao Huang [1]

[1]University of Southampton [2]Gen AI, KAUST [3]Northwestern University [4]National Taiwan University [5]Nanyang Technological University [6]The Swiss AI Lab IDSIA/USI/SUPSI
*Equal Contribution

TL;DR

We present the Directly Forecasting Belief Transformer (DFBT) for delayed RL, which can effectively reduce the compounding errors and improve performance. Specifically,

- We present DFBT, a novel directly forecasting belief method that effectively addresses compounding errors in recursively generated belief.
- We propose DFBT-SAC, a novel delayed RL method that further improves the learning efficiency via multi-step bootstrapping on the DFBT.
- We theoretically demonstrate that our DFBT significantly reduces compounding errors compared to the existing recursively forecasting belief approach.
- We empirically demonstrate that our DFBT method effectively forecasts state sequences with significantly higher prediction accuracy compared to baselines.
- We empirically show that our DFBT-SAC outperforms SOTAs in terms of sample efficiency and performance on the MuJoCo benchmark.

Background

A delay-free RL problem is formalized as an MDP represented as $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \rho, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ is the dynamic function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, ρ is the initial state distribution, and $\gamma \in (0,1)$ is the discount factor.

A delayed RL problem can be formalized as an augmented MDP. For instance, a delayed RL problem with constant delays Δ represented as $\langle \mathcal{X}, \mathcal{A}, \mathcal{P}_\Delta, \mathcal{R}_\Delta, \rho_\Delta, \gamma \rangle$, where

- Augmented state space $\mathcal{X} := \mathcal{S} \times \mathcal{A}^\Delta$
- Action space $\mathcal{A}$
- Delayed dynamic $\mathcal{P}_\Delta(x_{t+1}|x_t, a_t) := \mathcal{P}(s_{t-\Delta+1}|s_{t-\Delta}, a_{t-\Delta})\delta_{a_t}(a'_t) \prod_{i=1}^{\Delta-1} \delta_{a_{t-i}}(a'_{t-i})$
- Delayed reward function $\mathcal{R}_\Delta(x_t, a_t) := \mathbb{E}_{s_t \sim b(\cdot|x_t)} [\mathcal{R}(s_t, a_t)]$;
- Initial augmented state distribution $\rho_\Delta = \rho \prod_{i=1}^{\Delta} \delta_{a_{-i}}$
- Discount factor $\gamma \in (0,1)$
- **Belief representation** $b : \mathcal{X} \times \mathcal{S} \to [0,1]$

Research Problem

Specifically, belief representation is defined as follows:

$$b(s_t|x_t) := \int_{\mathcal{S}^\Delta} \prod_{i=0}^{\Delta-1} \mathcal{P}(s_{t-\Delta+i+1}|s_{t-\Delta+i}, a_{t-\Delta+i}) \mathrm{d}s_{t-\Delta+i+1}.$$

The belief representation can be viewed as the recursive forward prediction of the dynamics $\mathcal{P}$. With the belief representation, the agent can directly learn in the original state space $\mathcal{S}$.

However, the recursive process is evidently affected by the error accumulation of the approximate dynamic function across Δ steps.

The compounding errors grow exponentially with the delays Δ. This fundamental limitation of such recursive methodology for belief forecasting leads to significant performance degradation, especially in environments with long-delayed signals.

Method

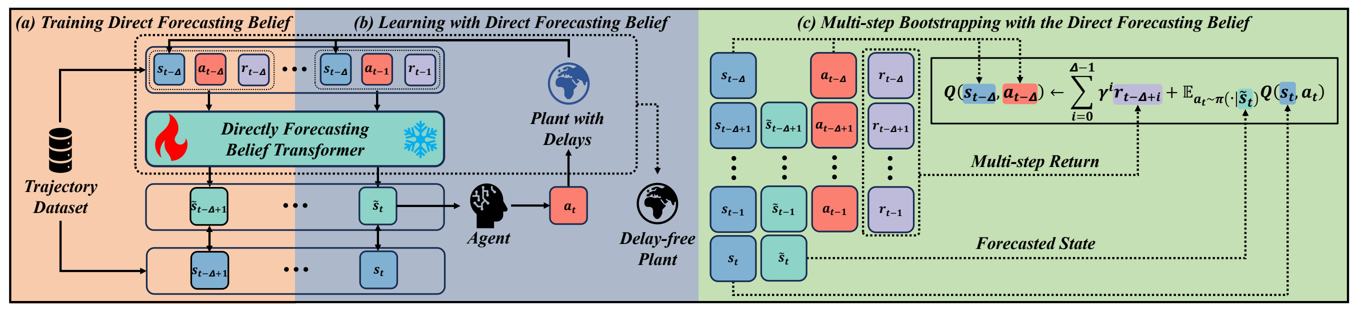


Figure 1. Pipeline of DFBT-SAC.

Given sub-trajectory with Δ timesteps $\{s_{t-\Delta+i}, a_{t-\Delta+i}, r_{t-\Delta+i}\}_{i=0}^{\Delta}$. We reform the representation of the augmented state to Δ tokens for sequence modeling: $x_t^{\text{tokens}} = \{s_{t-\Delta}, a_{t-\Delta+i}, r_{t-\Delta+i}\}_{i=0}^{\Delta-1}$. Then, DFBT predicts the unobserved Δ states $\{s_{t-\Delta+i}\}_{i=1}^{\Delta}$ via autoregressive modeling with loss:

$$\nabla_\theta \left[\sum_{i=1}^{\Delta} \left[-\log b_\theta^{(i)}(s_{t-\Delta+i}|x_t^{\text{tokens}}) \right] \right], \tag{1}$$

where $b_\theta^{(i)}(\cdot|x_t)$ represents the i-th prediction. The critic of DFBT-SAC is multi-step bootstrapped on the states predicted by the DFBT. Specifically, the critic Q_ψ parameterized by ψ is updated via:

$$\nabla_\psi \left[\frac{1}{2} \left(Q_\psi(s_{t-\Delta}, a_{t-\Delta}) - \mathbb{Y} \right)^2 \right], \tag{2}$$

where N-step $(N \leq \Delta)$ temporal difference target $\mathbb{Y}$ is defined as:

$$\mathbb{Y} := \sum_{i=0}^{N-1} \gamma^i r_{t-\Delta+i} + \gamma^N \mathop{\mathbb{E}}_{\substack{a \sim \pi(\cdot|\hat{s}_{t-\Delta+N}) \\ \hat{s}_{t-\Delta+N} \sim b_\theta^{(N)}(\cdot|x_t^{\text{tokens}})}} \left[Q(s_{t-\Delta+N}, a) + \log \pi(a|\hat{s}_{t-\Delta+N}) \right].$$

Main Theoretical Results

Performance Difference of Recursively Forecasting Belief

[Theorem 5.5] For the delay-free policy π and the delayed policy π_Δ. Given any $x_t \in \mathcal{X}$, the performance difference $I^{\text{recursive}}(x_t)$ of the recursively forecasting belief b_θ can be bounded as follows, respectively. For deterministic delays Δ, we have

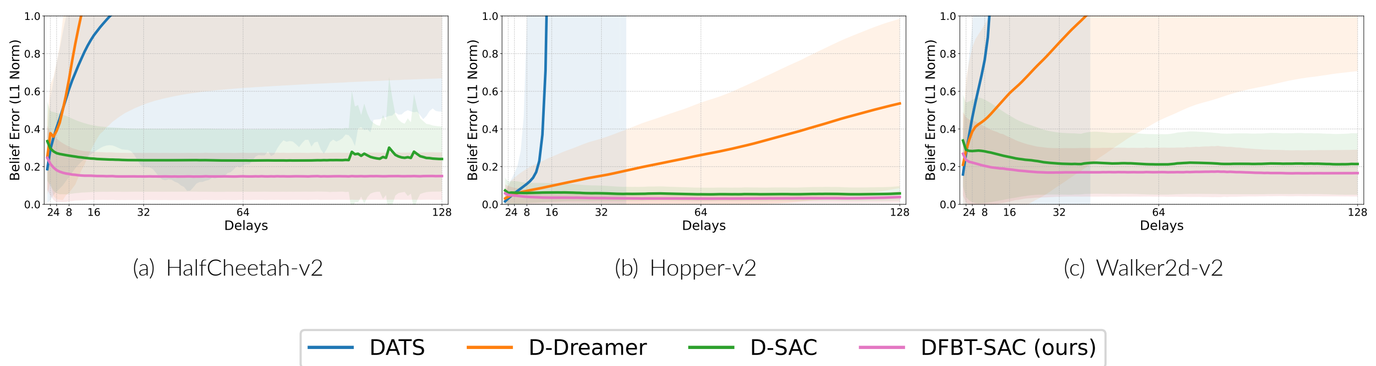
$$\left| I^{\text{recursive}}(x_t) \right| \leq \left| I_\Delta^{\text{true}}(x_t) \right| + L_V \underbrace{\frac{1 - {L_\mathcal{P}}^\Delta}{1 - L_\mathcal{P}} \epsilon_\mathcal{P}}_{\text{compounding errors}}.$$

For stochastic delays $\delta \sim d_\Delta(\cdot)$, we have

$$\left| I^{\text{recursive}}(x_t) \right| \leq \mathop{\mathbb{E}}_{\delta \sim d_\Delta(\cdot)} \left[\left| I_\delta^{\text{true}}(x_t) \right| + L_V \underbrace{\frac{1 - {L_\mathcal{P}}^\delta}{1 - L_\mathcal{P}} \epsilon_\mathcal{P}}_{\text{compounding errors}} \right].$$

Main Experimental Results

Belief Errors Comparison



(a) HalfCheetah-v2 (b) Hopper-v2 (c) Walker2d-v2

Figure 2. Belief errors comparison.

Performance Comparison

The best performance is underlined, the best belief-based method is in red.

Table 1. Performance on MuJoCo with Deterministic Delays.

Task	Delays	Augmentation-based A-SAC	Augmentation-based BPQL	Augmentation-based ADRL	Belief-based DATS	Belief-based D-Dreamer	Belief-based D-SAC	Belief-based DFBT-SAC (ours)
HalfCheetah-v2	8	$0.10_{\pm 0.01}$	$0.40_{\pm 0.04}$	$\underline{0.44_{\pm 0.03}}$	$0.08_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.12_{\pm 0.06}$	$0.35_{\pm 0.12}$
	32	$0.02_{\pm 0.02}$	$0.40_{\pm 0.03}$	$0.26_{\pm 0.04}$	$0.11_{\pm 0.04}$	$0.08_{\pm 0.00}$	$0.08_{\pm 0.02}$	$\underline{0.42_{\pm 0.03}}$
	128	$0.04_{\pm 0.06}$	$0.08_{\pm 0.13}$	$0.14_{\pm 0.02}$	$0.10_{\pm 0.08}$	$0.15_{\pm 0.05}$	$0.09_{\pm 0.04}$	$\underline{0.41_{\pm 0.03}}$
Hopper-v2	8	$0.61_{\pm 0.31}$	$0.87_{\pm 0.09}$	$\underline{0.95_{\pm 0.16}}$	$0.41_{\pm 0.31}$	$0.11_{\pm 0.01}$	$0.16_{\pm 0.05}$	$0.77_{\pm 0.18}$
	32	$0.11_{\pm 0.02}$	$\underline{0.89_{\pm 0.14}}$	$0.73_{\pm 0.20}$	$0.07_{\pm 0.04}$	$0.11_{\pm 0.05}$	$0.11_{\pm 0.01}$	$0.68_{\pm 0.20}$
	128	$0.04_{\pm 0.01}$	$0.08_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.09_{\pm 0.03}$	$0.06_{\pm 0.01}$	$\underline{0.20_{\pm 0.03}}$
Walker2d-v2	8	$0.44_{\pm 0.26}$	$\underline{1.07_{\pm 0.02}}$	$0.97_{\pm 0.10}$	$0.13_{\pm 0.05}$	$0.11_{\pm 0.06}$	$0.09_{\pm 0.05}$	$0.99_{\pm 0.03}$
	32	$0.10_{\pm 0.02}$	$0.37_{\pm 0.25}$	$0.16_{\pm 0.08}$	$0.02_{\pm 0.03}$	$0.08_{\pm 0.05}$	$0.08_{\pm 0.02}$	$\underline{0.64_{\pm 0.10}}$
	128	$0.06_{\pm 0.00}$	$0.07_{\pm 0.03}$	$0.08_{\pm 0.01}$	$0.02_{\pm 0.02}$	$0.08_{\pm 0.05}$	$0.11_{\pm 0.06}$	$\underline{0.40_{\pm 0.08}}$

Table 2. Performance on MuJoCo with Stochastic Delays.

Task	Delays	Augmentation-based A-SAC	Augmentation-based BPQL	Augmentation-based ADRL	Belief-based DATS	Belief-based D-Dreamer	Belief-based D-SAC	Belief-based DFBT-SAC (ours)
HalfCheetah-v2	$U(1,8)$	$0.09_{\pm 0.01}$	$0.21_{\pm 0.07}$	$0.17_{\pm 0.07}$	$0.09_{\pm 0.03}$	$0.02_{\pm 0.01}$	$0.03_{\pm 0.01}$	$\underline{0.37_{\pm 0.12}}$
	$U(1,32)$	$0.01_{\pm 0.00}$	$\underline{0.33_{\pm 0.07}}$	$0.23_{\pm 0.02}$	$0.11_{\pm 0.04}$	$0.02_{\pm 0.00}$	$0.01_{\pm 0.01}$	$0.31_{\pm 0.16}$
	$U(1,128)$	$0.01_{\pm 0.01}$	$0.03_{\pm 0.03}$	$0.15_{\pm 0.02}$	$0.16_{\pm 0.03}$	$0.16_{\pm 0.00}$	$0.02_{\pm 0.00}$	$\underline{0.39_{\pm 0.04}}$
Hopper-v2	$U(1,8)$	$0.17_{\pm 0.05}$	$0.20_{\pm 0.04}$	$0.18_{\pm 0.04}$	$0.04_{\pm 0.01}$	$0.07_{\pm 0.05}$	$0.14_{\pm 0.04}$	$\underline{0.86_{\pm 0.18}}$
	$U(1,32)$	$0.05_{\pm 0.01}$	$0.07_{\pm 0.09}$	$0.05_{\pm 0.01}$	$0.05_{\pm 0.01}$	$0.04_{\pm 0.01}$	$0.03_{\pm 0.01}$	$\underline{0.43_{\pm 0.21}}$
	$U(1,128)$	$0.03_{\pm 0.01}$	$0.04_{\pm 0.01}$	$0.04_{\pm 0.02}$	$0.05_{\pm 0.00}$	$0.03_{\pm 0.01}$	$0.03_{\pm 0.00}$	$\underline{0.14_{\pm 0.01}}$
Walker2d-v2	$U(1,8)$	$0.36_{\pm 0.24}$	$0.40_{\pm 0.32}$	$0.41_{\pm 0.15}$	$0.07_{\pm 0.01}$	$0.07_{\pm 0.05}$	$0.12_{\pm 0.04}$	$\underline{1.11_{\pm 0.10}}$
	$U(1,32)$	$0.12_{\pm 0.03}$	$0.16_{\pm 0.04}$	$0.11_{\pm 0.05}$	$0.09_{\pm 0.04}$	$0.12_{\pm 0.04}$	$0.05_{\pm 0.02}$	$\underline{0.67_{\pm 0.15}}$
	$U(1,128)$	$0.06_{\pm 0.01}$	$0.06_{\pm 0.06}$	$0.04_{\pm 0.02}$	$0.10_{\pm 0.04}$	$0.15_{\pm 0.07}$	$0.03_{\pm 0.04}$	$\underline{0.30_{\pm 0.13}}$