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]

Abstract

Reinforcement learning (RL) with delays is challenging as sensory perceptions lag behind the actual events: the RL agent needs to estimate the real state of its environment based on past observations. State-of-the-art (SOTA) methods typically employ recursive, step-by-step forecasting of states. This can cause the accumulation of compounding errors. To tackle this problem, our novel belief estimation method, named Directly Forecasting Belief Transformer (DFBT), directly forecasts states from observations without incrementally estimating intermediate states step-by-step. We theoretically demonstrate that DFBT greatly reduces compounding errors of existing recursively forecasting methods, yielding stronger performance guarantees. In experiments with D4RL offline datasets, DFBT reduces compounding errors with remarkable prediction accuracy. DFBT's capability to forecast state sequences also facilitates multi-step bootstrapping, thus greatly improving learning efficiency. On the MuJoCo benchmark, our DFBT-based method substantially outperforms SOTA baselines. Code is available at https://github.com/QingyuanWuNothing/DFBT.

1. Introduction

Reinforcement learning (RL) has achieved impressive success in various scenarios, including board games (Silver et al., 2016; Schrittwieser et al., 2020), video games (Mnih et al., 2013; Berner et al., 2019) and cyber-physical systems (Wei et al., 2017; Wang et al., 2023b;c; Zhan et al., 2024). The timing factor is critical to enable RL in real-world applications, particularly the delays that occur in the interaction between the agent and the environment due to physical distance, computational processing, and signal transmission. Otherwise, delays fundamentally affect the system's safety (Mahmood et al., 2018), performance (Cao et al., 2020) and efficiency (Hwangbo et al., 2017). Unlike rewards-delays-arising credit assignment issues that have been well studied (Arjona-Medina et al., 2019; Wang et al., 2024), observation-delays and action-delays disrupt the Markovian property of the Markov Decision Process (MDP), posing more challenges in RL. Most of the literature (Kim et al., 2023; Wang et al., 2023a; Wu et al., 2024b) mainly focuses on observation-delays, which has been proved as a superset of action-delays (Katsikopoulos & Engelbrecht, 2003; Nath et al., 2021). Therefore, this work mainly focuses on the RL with delays Δ in observation: at time step t, the agent can not observe the real environment state s_t but only the history state $s_{t-\Delta}$ Δ steps ago.

To enable RL in environments with delayed observations, it is essential to restore the Markovian property (Altman & Nain, 1992; Katsikopoulos & Engelbrecht, 2003). Augmentation-based methods (Bouteiller et al., 2020; Kim et al., 2023) are based on an observation that the information state $x_t = \{s_{t-\Delta}, a_{t-\Delta:t-1}\}$, augmented from the last observable state and the sequential actions, carries the equivalent information with s_t (Bertsekas, 2012). Thus, the policy learning over the original state space $\mathcal{S}$ can be transferred to the policy learning over the augmented state space $\mathcal{X}$, which is memoryless and can be addressed by nominal RL. However, as delays Δ increase, the dimensionality of the augmented state space $\mathcal{X}$ expands significantly, leading to a drastic deterioration in learning efficiency due to the exponentially increased sample complexity (Wu et al., 2024b).

To improve learning efficiency in augmentation-based methods, belief-based methods (Walsh et al., 2009; Chen et al., 2021a) propose to conduct RL in the original state space $\mathcal{S}$, by estimating the instant unobservable state s_t from the information state x_t, of which the mapping is referred to as belief. The belief function is commonly forecasted recursively (Chen et al., 2021a; Karamzade et al., 2024): for $i = 1, \ldots, \Delta$, $s_{t-\Delta+i}$ is estimated by applying approximate dynamic function with the previous state $s_{t-\Delta+i-1}$ and the

[*]Equal contribution [1]University of Southampton [2]GenAI, King Abdullah University of Science and Technology [3]Northwestern University [4]National Taiwan University [5]Nanyang Technological University [6]The Swiss AI Lab IDSIA/USI/SUPSI. Correspondence to: Chao Huang <chao.huang@soton.ac.uk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

previous action $a_{t-\Delta+i-1}$. Obtaining the estimation of the instant state s_t, the delayed RL problem is effectively reduced to a delay-free RL problem without enlarging the state space, mitigating the curse of dimensionality. However, this 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.

This work aims to squarely address the compounding errors in existing belief-based techniques with recursive strategy, using a direct strategy in sequence modeling. Specifically, we present the Directly Forecasting Belief Transformer (DFBT), a novel directly forecasting belief method by reformulating belief forecasting as a sequence modeling problem. DFBT first represents the information state x_t and reward signals $r_{t-\Delta:t-1}$ to Δ tokens in the form $\{s_{t-\Delta}, a_{t-\Delta+i}, r_{t-\Delta+i}\}_{i=0}^{\Delta-1}$. **Unlike recursively forecasting belief methods that invoke the forward prediction process Δ times, DFBT simultaneously forecasts the states $s_{t-\Delta+1:t}$ without introducing accumulated errors, effectively addressing the issue of compounding errors.** We then integrate DFBT with the Soft Actor-Critic (SAC) method (Haarnoja et al., 2018) in online learning, incorporating multi-step bootstrapping on accurate state predictions generated from DFBT, which further improves learning efficiency. Theoretically, we demonstrate how the compounding errors in recursively forecasting belief affect the RL performance and how it is alleviated by our direct strategy. Empirically, using the D4RL benchmark (Fu et al., 2020), we show that DFBT achieves significantly higher prediction accuracy compared to other belief methods. On the MuJoCo benchmark (Todorov et al., 2012), across various delays settings, we demonstrate that our DFBT-SAC consistently outperforms SOTA augmentation-based and belief-based methods in both learning efficiency and overall performance.

In Section 3, we introduce the delayed RL problem and the concept of belief representation. Section 4 presents our proposed DFBT, which directly forecasts states from delayed observations, avoiding the need for recursive single-step predictions. Additionally, we develop DFBT-SAC by leveraging multi-step bootstrapping on the states predicted by DFBT. Through theoretical analysis in Section 5, we show that DFBT effectively mitigates the compounding errors associated with recursively forecasting belief methods, ensuring superior performance. Finally, in Section 6, we empirically demonstrate the superior prediction accuracy of DFBT on the D4RL datasets and the remarkable efficacy of DFBT-SAC compared to state-of-the-art approaches on MuJoCo across various delays settings. Overall, our key contributions are summarized as follows:

- We present Directly Forecasting Belief Transformer (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 directly forecasting belief significantly reduces compounding errors compared to existing recursively forecasting belief approaches, offering a more robust performance guarantee.
- We empirically demonstrate that our DFBT method effectively forecasts state sequences with significantly higher prediction accuracy compared to baseline approaches.
- We empirically show that our DFBT-SAC outperforms SOTAs in terms of sample efficiency and performance on the MuJoCo benchmark.

2. Related Works

In classical control theory, delays are modelled with delay–differential equations (DDEs) (Cooke, 1963). A rich toolbox is available for analysing their reachability (Fridman & Shaked, 2003; Xue et al., 2020), stability (Feng et al., 2019), and safety (Xue et al., 2021). These techniques, however, assume fully known dynamics and become impractical for high-dimensional systems. Delayed reinforcement learning (RL) is far closer to real deployments than the delay-free setting that dominates the literature. Latencies are intrinsic to robotics (Mahmood et al., 2018; Hwangbo et al., 2017), high-frequency trading (Hasbrouck & Saar, 2013), intelligent transportation (Cao et al., 2020), and many other domains. Depending on the methodology of retrieving the Markovian property, existing delayed RL approaches can be mainly categorized as augmentation-based and belief-based approaches.

Augmentation-based Approaches. Augmentation-based approaches retrieve the Markovian property by augmenting the original state with the sequence of actions within the delays window, optimizing the policy in the resulting augmented MDP to mitigate performance degradation caused by errors in belief representation approximation. Specifically, DIDA (Liotet et al., 2022) learns the delayed policy in the augmented MDP by imitating the behaviours from the delay-free expert policy in the original MDP; DC/AC (Bouteiller et al., 2020) suggest using the delays correction operator

to improve the learning efficiency in the augmented MDP; ADRL (Wu et al., 2024b) introduces the auxiliary delayed task with the smaller augmented MDP for bootstrapping the larger augmented MDP; VDPO (Wu et al., 2024a) proposes to learn the delayed policy through the lens of variational inference, improving the learning efficiency via extensive optimization tools. However, augmentation-based approaches still operate on an increasingly large augmented state space as delays grow, inevitably suffering from the curse of dimensionality and resulting in significant learning inefficiencies.

Belief-based Approaches. The belief-based approach retrieves the Markovian property by forecasting the unobservable state through belief representation, then optimizing policy within the original state space. Thus, correspondingly, the prediction accuracy of approximated belief representation is highly related to the overall performance. DATS (Chen et al., 2021a) is the first to propose approximating the belief representation using a Gaussian distribution. Inspired by the world models (Schmidhuber, 1990b;a; Ha & Schmidhuber, 2018), D-Dreamer (Karamzade et al., 2024) employ a recurrent neural network to capture temporal dependencies in the belief through forward prediction in the latent space, enabling adaptation to high-dimensional observations. D-SAC (Liotet et al., 2021) utilizes the attention mechanism in the causal transformer to integrate the information across the delays. Unfortunately, the approximation errors of recursively forecasting belief accumulated at each step finally lead to compounding errors, which hinder prediction accuracy and performance.

Time Series Forecasting in RL. Time series forecasting plays a critical role in predicting future states in RL, particularly within the context of world model learning (Ha & Schmidhuber, 2018), which simulates the dynamics of the environment, predicting future states and planning effectively without directly interacting with the real environment. World model learning often leverages time series forecasting techniques to capture temporal dependencies, enhancing the agent's understanding of the environment. Applications of world models include robotics, autonomous driving, and gaming scenarios (Hafner et al., 2019). Additionally, the RL problem can be treated as a sequence modeling problem, which can be effectively solved by the transformer (Vaswani et al., 2017). Directly predicting the action from the desired outcomes (Schmidhuber, 2019), decision transformer (Chen et al., 2021b) predict future actions based on historical trajectories and desired outcomes. Similarly, trajectory transformer (Janner et al., 2021) models trajectories as sequences, enabling policy optimization by leveraging autoregressive modeling. Therefore, in this paper, we investigate how to leverage the advanced time series forecasting approaches in addressing the issue of compounding errors existing in the recursively forecasting belief. We also present a comprehensive theoretical analysis of how compounding errors seriously degenerate performance as delays are increased.

3. Preliminaries

3.1. From Delay-free RL to Delayed RL

A conventional delay-free RL problem is usually formalized as a Markov Decision Process (MDP) represented as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \rho, \gamma \rangle$, 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. The agent selects an action $a_t \sim \pi(\cdot|s_t)$ according to the policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ based on the current state s_t at time step t. The agent will then observe the next state $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$ and the reward $r_t = \mathcal{R}(s_t, a_t)$ from the environment. The objective of the agent is to find the optimal policy π^* that maximizes the expected discounted return $G := \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)\right]$ where $p_\pi(\tau)$ represents the distribution of trajectories induced by policy π.

A delayed RL problem can be formalized as an augmented MDP by applying the augmentation technique (Altman & Nain, 1992; Katsikopoulos & Engelbrecht, 2003) to retrieve the Markovian property. For a delayed RL problem with constant delays Δ, the newly formed MDP is represented as $\langle \mathcal{X}, \mathcal{A}, \mathcal{P}_\Delta, \mathcal{R}_\Delta, \rho_\Delta, \gamma \rangle$, where $\mathcal{X} := \mathcal{S} \times \mathcal{A}^\Delta$ is the augmented state space, $\mathcal{A}$ is the original action space, the delayed dynamic function is $\mathcal{P}_\Delta(x_{t+1}|x_t, a_t) := \mathcal{P}(s_{t-\Delta+1}|s_{t-\Delta}, a_{t-\Delta})\delta_{a_t}(a'_t) \prod_{i=1}^{\Delta-1} \delta_{a_{t-i}}(a'_{t-i})$ where δ is the Dirac distribution, the delayed reward function is defined as $\mathcal{R}_\Delta(x_t, a_t) := \mathbb{E}_{s_t \sim b(\cdot|x_t)} [\mathcal{R}(s_t, a_t)]$ where $b : \mathcal{X} \times \mathcal{S} \to [0, 1]$ is the belief representation mapping from the augmented state space $\mathcal{X}$ to the original state space $\mathcal{S}$, $\rho_\Delta = \rho \prod_{i=1}^{\Delta} \delta_{a_{-i}}$ is the initial augmented state distribution.

3.2. Belief Representation in Delayed RL

Delayed RL can be viewed as a special form of a partially observable RL problem where the observation is contaminated by noise, instead of being delayed. Therefore, similar to partially observable RL, delayed RL also have the belief representation defined as follows:

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

The belief representation can retrieve the Markovian property via mapping the augmented state space $\mathcal{X}$ to $\mathcal{S}$, recasting the delayed RL problem in the original MDP without augmenting the state space. 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 $\mathcal{S}$. In this work, we use s and $\hat{s}$ to represent the true state of the environment and the predicted state of the approximate belief, respectively.

4. Our Approach

In this section, we present the Directly Forecasting Belief Transformer (DFBT), a new directly forecasting belief method. By framing belief forecasting as a sequence modeling problem, DFBT directly predicts the unobservable states. Instead of recursively predicting the next state step-by-step, DFBT can effectively capture the dependency across the delays via the attention mechanism of the transformer model. Therefore, our DFBT can effectively reduce the compounding errors of the recursively forecasting belief, especially in the long delays. Specifically, as summarized in Figure 1, we train the DFBT using a pre-collected trajectory dataset, latter deploying the trained DFBT in the environment with delays to reconstruct the delay-free environment for training. Furthermore, we integrate multi-step bootstrapping with the predicted states of DFBT to improve learning efficiency.

4.1. Directly Forecasting Belief

The belief representation learning problem can be viewed as a sequence modeling problem. Given a pre-collected delay-free trajectory $\{s_i, a_i, r_i\}_{i=0}^{T}$, we sample the sub-trajectory with Δ timesteps $\{s_{t-\Delta+i}, a_{t-\Delta+i}, r_{t-\Delta+i}\}_{i=0}^{\Delta}$. We model the past reward signals $r_{t-\Delta:t-1}$ into the augmented state $x_t = \{s_{t-\Delta}, a_{t-\Delta:t-1}\}$ for considering more dynamic information in belief. Then, we have reformed the representation of the augmented state to Δ tokens for sequence modeling: $x_t^{\text{tokens}} = \{s_{t-\Delta}, a_{t-\Delta+i}, r_{t-\Delta+i}\}_{i=0}^{\Delta-1}$. The Directly Forecasting Belief Transformer (DFBT) b_θ leverages the transformer architecture (Vaswani et al., 2017), utilizing the attention mechanism to effectively capture dependencies across long delays. Inputting with x_t^{tokens}, DFBT simultaneously predicts the unobserved Δ states $\{s_{t-\Delta+i}\}_{i=1}^{\Delta}$ via autoregressive modeling with loss:

$$\bigtriangledown_\theta \left[\sum_{i=1}^{\Delta} \left[-\log b_\theta^{(i)}(s_{t-\Delta+i} | x_t^{\text{tokens}}) \right] \right], \tag{2}$$

where $b_\theta^{(i)}(\cdot|x_t)$ represents the i-th prediction. In a deterministic environment with a deterministic belief function, we typically replace Equation (2) with the Mean-Square-Error (MSE) loss.

4.2. Multi-step Bootstrapping with DFBT

Next, we directly deploy the trained DFBT b_θ in the online environment with delayed signals to reconstruct the delay-free environment where the agent can directly learn with the original state space $\mathcal{S}$ instead of the augmented state space $\mathcal{X}$, thus maintaining the superior sample complexity in online learning.

Algorithm 1 DFBT-SAC

Input: offline dataset $\mathcal{D}$, DFBT b_θ, critic Q_ψ, actor π_ϕ;
# training DFBT on the offline dataset
for Epoch = $1, \dots$ **do**
  Update b_θ on the $\mathcal{D}$ via Equation (2)
end for
# learning with DFBT on the online environment
for Epoch = $1, \dots$ **do**
  Update Q_ψ on via Equation (3)
  Update π_ϕ on via Equation (4)
end for
Output: b_θ and π_ϕ

Then, we present our practical delayed RL method, named DFBT-SAC by incorporating the trained DFBT b_θ with Soft Actor-Critic (Haarnoja et al., 2018). To improve the learning efficiency with the DFBT, the critic of DFBT-SAC is bootstrapped on the states predicted by the DFBT with the multi-step learning (Sutton & Barto, 2018; Hessel et al., 2018) and delay-free training techniques (Wu et al., 2024b; Kim et al., 2023). Specifically, given multi-step data $(x_t^{\text{tokens}}, s_{t-\Delta+1:t})$, the critic Q_ψ parameterized by ψ is updated via:

$$\bigtriangledown_\psi \left[\frac{1}{2} \left(Q_\psi(s_{t-\Delta}, a_{t-\Delta}) - \mathbb{Y} \right)^2 \right], \tag{3}$$

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

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

and $N (\leq \Delta)$ is the bootstrapping steps on the DBFT. In this work, we set $N = 8$ as default, and we also conduct the ablation study (Section 6) to investigate bootstrapping steps settings. The actor π_ϕ parameterized by ϕ is updated by:

$$\bigtriangledown_\phi \mathop{\mathbb{E}}_{\substack{a \sim \pi(\cdot|\hat{s}_{t-\Delta+N}) \\ \hat{s}_{t-\Delta+N} \sim b_\theta^{(N)}(\cdot|x_t^{\text{tokens}})}} \left[\log \pi_\phi(a|\hat{s}_{t-\Delta+N}) - Q_\psi(s_{t-\Delta+N}, a) \right]. \tag{4}$$

The pseudo-code of DFBT-SAC is provided in Algorithm 1.

5. Theoretical Analysis

In this section, we theoretically illustrate that performance degeneration is rooted in compounding errors of recursively forecasting belief (Section 5.1), which can be effectively addressed by our DFBT, directly forecasting belief, thus achieving a better performance guarantee (Section 5.2).

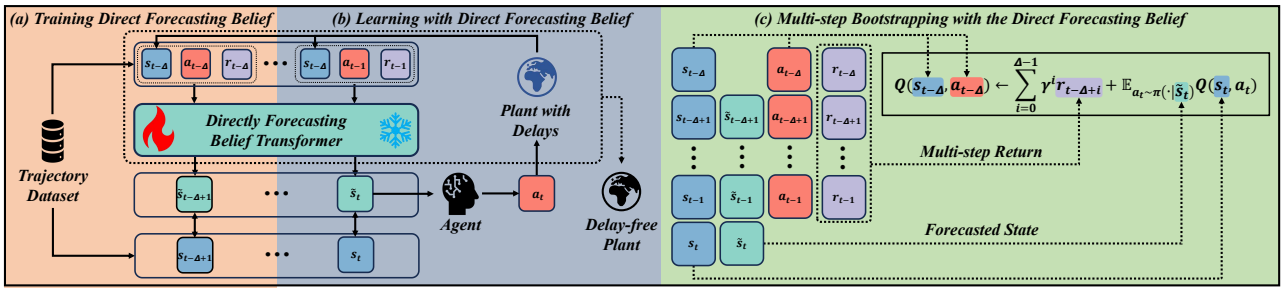


Figure 1. Pipeline of DFBT-SAC. (a) Training DFBT on the trajectory dataset. (b) The agent can interact and learn with the delay-free environment recovered by the DFBT as highlighted in the dashline box. (c) Multi-step bootstrapping on the forecasted states from DFBT.

Before starting the theoretical analysis, we introduce the definition of the performance degeneration of the ground-truth belief b as follows.

Definition 5.1 (Performance Degeneration of Ground-truth Belief (Liotet et al., 2022))**.** For policies π and π_Δ with delays Δ. Given any $x_t \in \mathcal{X}$, the performance difference of the ground-truth belief b is denoted as $I^{\text{true}}_\Delta(x_t)$

$$I^{\text{true}}_\Delta(x_t) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{\hat{s}\sim b(\cdot|\hat{x}) \\ \hat{a}\sim\pi_\Delta(\cdot|\hat{x}) \\ \hat{x}\sim d^\pi_{x_t}}} \left[V^\pi(\hat{s}) - Q^\pi(\hat{s},\hat{a})\right],$$

where $d^\pi_{x_t}$ is the augmented state distribution induced by policy π with the initial state x_t.

Furthermore, we also introduce the Lipschitz Continuity of MDP (Definition 5.2) and value function (Definition 5.3), which are common and mild assumptions in the literature.

Definition 5.2 (Lipschitz Continuity of MDP (Rachelson & Lagoudakis, 2010))**.** An MDP is $L_\mathcal{P}$-LC, if $\forall(s_1,a_1),(s_2,a_2) \in \mathcal{S}\times\mathcal{A}$, dynamic function $\mathcal{P}$ satisfies:

$$\mathcal{W}\left(\mathcal{P}(\cdot|s_1,a_1)||\mathcal{P}(\cdot|s_2,a_2)\right) \le L_\mathcal{P}(d_\mathcal{S}(s_1,s_2)+d_\mathcal{A}(a_1,a_2)),$$

where $\mathcal{W}$ is the L1-Wasserstein distance, $d_\mathcal{S}$ and $d_\mathcal{A}$ are distrance measure of the $\mathcal{S}$ and $\mathcal{A}$, repectively.

Definition 5.3 (Lipschitz Continuity of Value Function (Rachelson & Lagoudakis, 2010))**.** Consider a L_Q-LC Q-function Q^π of the L_π-LC policy π, value function V^π satisfies that

$$\left|\mathop{\mathbb{E}}_{\substack{s_1\sim\mu \\ s_2\sim\upsilon}} \left[V^\pi(s_1) - V^\pi(s_2)\right]\right| \le L_V \mathcal{W}\left(\mu||\upsilon\right),$$

where $L_V = L_Q(1+L_\pi)$ and μ,υ are two arbitrary distributions over $\mathcal{S}$.

5.1. Recursively Forecasting Belief: Compounding Errors Analysis

In this section, we show that the performance degeneration of recursively forecasting belief is influenced by compounding errors, which consist of the recursive error and are exponentially increased with delays. We assume that the recursively forecasting belief $\mathcal{P}_\theta$ has the approximated error bound as shown in Assumption 5.4.

Assumption 5.4 (Approximated Dynamic Difference Bound)**.** The distance between the approximated dynamic function $\mathcal{P}_\theta$ parameterized by θ and the ground-truth dynamic function $\mathcal{P}$ is bounded, it satisfies that $\forall(s,a) \in \mathcal{S}\times\mathcal{A}$,

$$\mathcal{W}(\mathcal{P}_\theta(\cdot|s,a)||\mathcal{P}(\cdot|s,a)) \le \epsilon_\mathcal{P}.$$

Then, we demonstrate that the performance difference of the recursively forecasting belief is determined by the compounding errors (Theorem 5.5).

Theorem 5.5 (Performance Difference of Recursively Forecasting Belief, Proof in Theorem B.1)**.** *For the delay-free policy π and the delayed policy π_Δ. Given any $x_t \in \mathcal{X}$, the performance difference $I^{recursive}(x_t)$ of the recursively forecasting belief b_θ can be bounded as follows, respectively.*

For deterministic delays Δ, we have

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

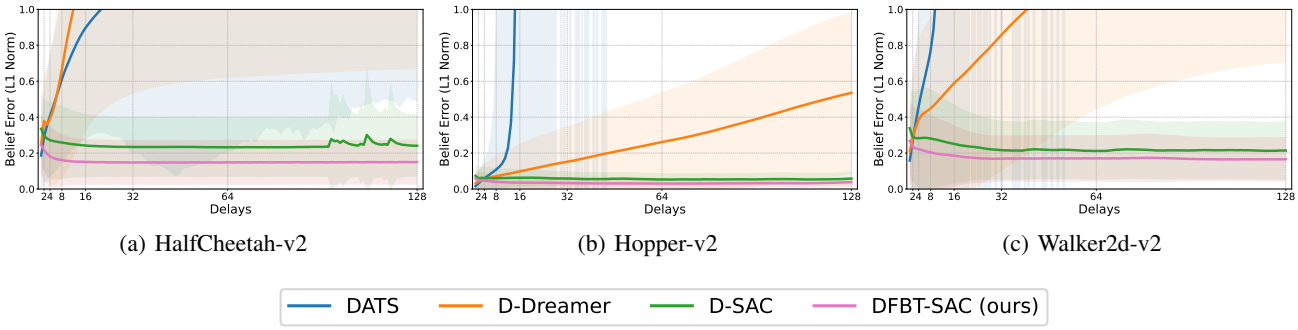
And for stochastic delays $\delta \sim d_\Delta(\cdot)$, we have

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

Theorem 5.5 tells that the compounding errors are exponentially increased with the delays, seriously degenerating the performance.

5.2. Directly Forecasting Belief: Errors Analysis

Next, we theoretically show that our directly forecasting belief can effectively address the compounding errors. The



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

Figure 2. Belief errors comparison on (a) HalfCheetah-v2, (b) Hopper-v2, and (c) Walker2d-v2.

sequence-modeling method estimates the states $s_{t-\Delta+1:t}$ based on the augmented state x_t in parallel, instead of invoking Δ times iteratively, which effectively alleviates the source of compounding errors. Especially, the attention mechanism in the transformer can selectively capture the long-range relationships within the augmented state with long delays. We assume that the belief error between the directly forecasting belief and the ground-truth belief is bounded (Assumption 5.6).

Assumption 5.6 (Directly Forecasting Belief Difference Bound)**.** The distance between the directly forecasting belief belief b_θ parameterized by θ and the ground-truth belief b is bounded, it satisfies that $\forall x_t \in \mathcal{X}$, we have $\mathcal{W}(b(\cdot|x_t)||b_\theta(\cdot|x_t)) \leq \epsilon_{direct}$, where $\epsilon_{direct} := \max_{i=1,\dots,\Delta} \mathcal{W}(b^{(i)}(\cdot|x_t)||b_\theta^{(i)}(\cdot|x_t))$.

From Theorem 5.5, we can derive the performance degeneration bound of directly forecasting belief as follows.

Proposition 5.7 (Performance Degeneration Bound of Directly Forecasting Belief, Proof in Proposition B.2)**.** *For the delay-free policy π and the delayed policy π_Δ. Given any $x_t \in \mathcal{X}$, the performance degeneration I^{direct} of the directly forecasting belief b_θ can bounded as follows respectively.*

For deterministic delays Δ, we have

$$\left|I^{direct}(x_t)\right| \leq |I_\Delta^{true}(x_t)| + L_V \epsilon_{direct}.$$

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

$$\left|I^{direct}(x_t)\right| \leq \mathop{\mathbb{E}}_{\delta\sim d_\Delta(\cdot)} [|I_\delta^{true}(x_t)|] + L_V \epsilon_{direct}.$$

Then, we have the performance degeneration comparison between directly forecasting belief and recursively forecasting belief.

Proposition 5.8 (Performance Degeneration Comparison, Proof in Proposition B.3)**.** *Directly forecasting belief could achieve a better performance guarantee $\left|I^{direct}(x_t)\right| \leq \left|I^{recursive}(x_t)\right|$, if we have*

$$\epsilon_{direct} \leq \frac{1 - {L_\mathcal{P}}^\Delta}{1 - L_\mathcal{P}} \epsilon_\mathcal{P}$$

for deterministic delays Δ, and

$$\epsilon_{direct} \leq \mathop{\mathbb{E}}_{\delta\sim d_\Delta(\cdot)} \left[\frac{1 - {L_\mathcal{P}}^\delta}{1 - L_\mathcal{P}}\right] \epsilon_\mathcal{P}$$

for stochastic delays $\delta \sim d_\Delta(\cdot)$.

Remark 5.9 (Empirical Validation of Proposition 5.8). It is obvious that the belief errors of recursively forecasting grows much faster than the directly forecasting, which is not strictly related with the delay length. We also empirically show that Proposition 5.8 is always held in Section 6.2.1. In the future, we will theoretically investigate the sample complexity of recursive and direct forecasting beliefs.

Remark 5.10 (General Error Distribution Case). In the context of time forecasting, our theoretical results of performance degeneration comparison have the potential to extend to the variance analysis commonly discussed in related literature (Taieb & Atiya, 2015; Clements & Hendry, 1996; Chevillon, 2007).

6. Experiments

6.1. Experimental Setting

We adopt D4RL (Fu et al., 2020) and MuJoCo (Todorov et al., 2012) as the offline dataset and the benchmark respectively to evaluate our DFBT-SAC. For the baselines, we choose the SOTA augmentation-based methods (A-SAC (Haarnoja et al., 2018), BPQL (Kim et al., 2023), and ADRL (Wu et al., 2024b)) and belief-based methods (DATS (Chen et al., 2021a), D-Dreamer (Karamzade et al., 2024), and D-SAC (Liotet et al., 2021)). All of the belief-based baselines and our DFBT-SAC are trained on the same D4RL dataset. We first investigate the prediction accuracy of beliefs (Section 6.2.1), followed by the performance comparison with deterministic and stochastic delays (Section 6.2.2). Additionally, we conduct ablation studies on the multi-step bootstrapping of our DFBT-SAC (Section 6.2.3).

Table 1. Performance on MuJoCo with Deterministic Delays. The best performance is underlined, the best belief-based method is in red.

Task	Delays	Augmentation-based			Belief-based				
		A-SAC	BPQL	ADRL	DATS	D-Dreamer	D-SAC	DFBT-SAC(1) (ours)	DFBT-SAC (ours)
HalfCheetah-v2	8	$0.10_{\pm 0.01}$	$0.40_{\pm 0.04}$	$0.44_{\pm 0.03}$	$0.08_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.12_{\pm 0.06}$	$0.38_{\pm 0.03}$	$0.35_{\pm 0.12}$
	32	$0.02_{\pm 0.02}$	$0.40_{\pm 0.03}$	$0.26_{\pm 0.04}$	$0.11_{\pm 0.04}$	$0.08_{\pm 0.00}$	$0.08_{\pm 0.02}$	$0.40_{\pm 0.07}$	$0.42_{\pm 0.03}$
	128	$0.04_{\pm 0.06}$	$0.08_{\pm 0.13}$	$0.14_{\pm 0.02}$	$0.10_{\pm 0.08}$	$0.15_{\pm 0.05}$	$0.09_{\pm 0.04}$	$0.40_{\pm 0.06}$	$0.41_{\pm 0.03}$
Hopper-v2	8	$0.61_{\pm 0.31}$	$0.87_{\pm 0.09}$	$0.95_{\pm 0.16}$	$0.41_{\pm 0.31}$	$0.11_{\pm 0.01}$	$0.16_{\pm 0.05}$	$0.92_{\pm 0.28}$	$0.77_{\pm 0.18}$
	32	$0.11_{\pm 0.02}$	$0.89_{\pm 0.14}$	$0.73_{\pm 0.20}$	$0.07_{\pm 0.04}$	$0.11_{\pm 0.05}$	$0.11_{\pm 0.01}$	$0.60_{\pm 0.23}$	$0.68_{\pm 0.20}$
	128	$0.04_{\pm 0.01}$	$0.08_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.09_{\pm 0.03}$	$0.06_{\pm 0.01}$	$0.16_{\pm 0.02}$	$0.20_{\pm 0.03}$
Walker2d-v2	8	$0.44_{\pm 0.26}$	$1.07_{\pm 0.02}$	$0.97_{\pm 0.10}$	$0.13_{\pm 0.05}$	$0.11_{\pm 0.06}$	$0.09_{\pm 0.05}$	$0.95_{\pm 0.14}$	$0.99_{\pm 0.03}$
	32	$0.10_{\pm 0.02}$	$0.37_{\pm 0.25}$	$0.16_{\pm 0.08}$	$0.02_{\pm 0.03}$	$0.08_{\pm 0.05}$	$0.08_{\pm 0.02}$	$0.57_{\pm 0.21}$	$0.64_{\pm 0.10}$
	128	$0.06_{\pm 0.00}$	$0.07_{\pm 0.08}$	$0.08_{\pm 0.01}$	$0.02_{\pm 0.02}$	$0.08_{\pm 0.05}$	$0.11_{\pm 0.06}$	$0.38_{\pm 0.11}$	$0.40_{\pm 0.08}$

Table 2. Performance on MuJoCo with Stochastic Delays. The best performance is underlined, and the best belief-based method is in red.

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

6.2. Experimental Results

6.2.1. Belief Prediction Accuracy

We first evaluate the state prediction accuracy of our DFBT in D4RL offline datasets. Using the L1 norm as the belief error, We report the error curves increasing with delays of the belief of DATS, D-Dreamer, D-SAC and our DFBT-SAC in Figure 2. All methods are trained on the D4RL mixed dataset including random, medium and expert policy demonstrations. The implementation details are provided in Appendix A. From the results, we can tell that our DFBT can address the compounding errors effectively via directly forecasting delayed observations, thus maintaining the best prediction accuracy with increased delays, which is consistent with our theoretical results. We further provide the belief qualitative comparison in Appendix E.

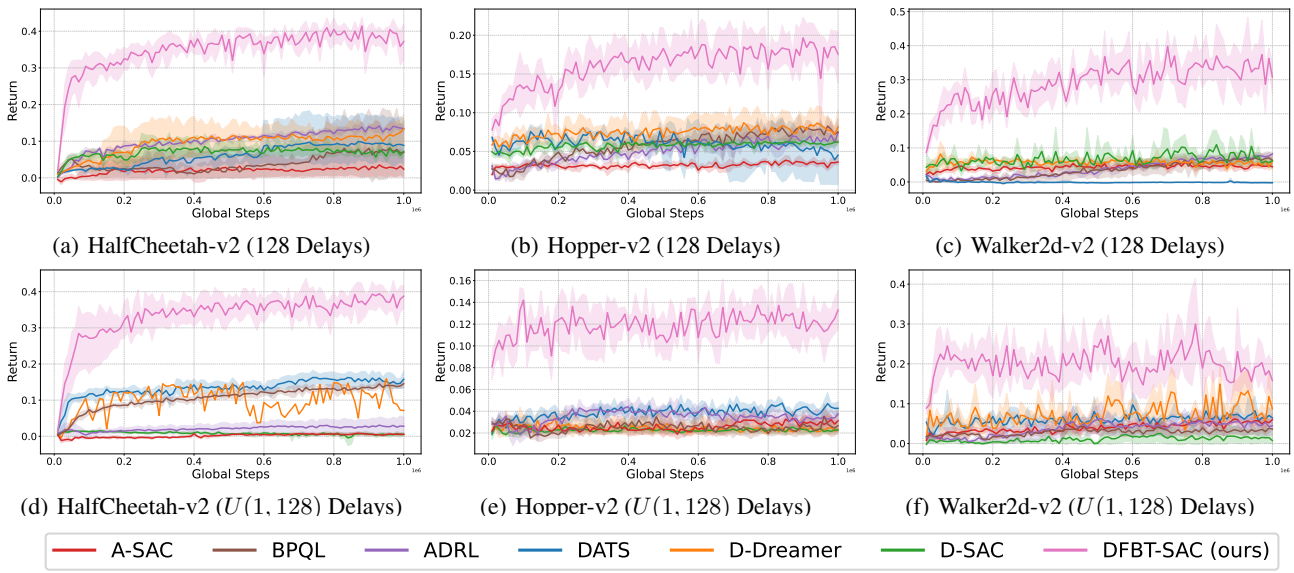
6.2.2. Performance Comparison

We report the normalized return $R_{\text{norm}} := \frac{R_{\text{alg}} - R_{\text{random}}}{R_{\text{sac}} - R_{\text{random}}}$, where $R_{\text{alg}}, R_{\text{sac}}, R_{\text{random}}$ are the return of the algorithm, delay-free SAC, and random returns, respectively. Each method is evaluated across 5 random seeds, and the implementation details are provided in the Appendix A. Additional experiment results and learning curves for different delays are provided in the Appendix C and Appendix D, respectively.

Deterministic Delays. The performance of DFBT-SAC and baselines are evaluated on MuJoCo with deterministic delays (8, 32, and 128) reported in Table 1, showing that our DFBT-SAC overall outperforms other belief-based methods significantly. Specifically, DFBT-SAC shows comparable performance with the SOTA augmentation-based methods (BPQL and ADRL) on tasks with relatively short delays (8). However, our DFBT-SAC yields a leading performance on challenging tasks when the delays increase to long delays (32 and 128). Additionally, we also compare DFBT-SAC(1), the single-step bootstrapping version of DFBT-SAC. The results imply that the multi-step bootstrapping technique can further improve the performance effectively, validating our statement in Section 4.2. We will investigate the bootstrapping steps in the later ablation study.

Stochastic Delays. We also evaluate DFBT-SAC on MuJoCo with stochastic delays which follow the uniform distribution U. As shown in Table 2, our DFBT-SAC remarkably outperforms all belief-based baselines on all tasks with all delays settings ($U(1,8)$, $U(1,32)$, and $U(1,128)$). Especially for $U(1,128)$ delays, DFBT-SAC also performs approximately 144%, 180%, and 100% better than the second best baselines on HalfCheetah-v2, Hopper-v2, and Walker2d-v2, respectively.

Learning Curves on MuJoCo with Challenging Delays. As summarized in Figure 3, we report the learning curves on MuJoCo with 128 and $U(1,128)$ delays. Our DFBT-



(a) HalfCheetah-v2 (128 Delays) (b) Hopper-v2 (128 Delays) (c) Walker2d-v2 (128 Delays)

(d) HalfCheetah-v2 ($U(1, 128)$ Delays) (e) Hopper-v2 ($U(1, 128)$ Delays) (f) Walker2d-v2 ($U(1, 128)$ Delays)

Figure 3. Learning Curves on MuJoCo with 128 Delays and $U(1, 128)$ Delays.

SAC exhibits a leading learning efficiency across all these challenging delays. Especially in the MuJoCo with 128 delays, DFBT-SAC can learn remarkably faster than all baselines, resulting in 173.3% (HalfCheetah-v2), 122.2% (Hopper-v2), and 263.6% (Walker2d-v2) final performance improvement better than the second-best baselines.

Results Analysis and Discussion. Based on the above experimental results, we can observe a general trend in both deterministic and stochastic delays that DFBT-SAC shows comparable performance with SOTA augmentation-based methods under short delays scenarios, in which augmentation-based methods generally have better performance than other belief-based approaches. Since when delays are short, the dimensionality of the augmented state space is proper for learning, and without the approximation errors from belief representation. However, as delays increase in both scenarios, DFBT-SAC unleashes its advantages. This empirical finding validates our theoretical analysis in Section 5. For augmentation-based methods, as the delays increase, the augmented state in augmentation-based methods grows too rapidly, making efficient policy optimization infeasible. For recursively forecasting belief methods, the compounding errors in belief function approximation grow exponentially with increasing delays, as demonstrated in Figure 2 and Section 6.2.1. This results in inaccurate predictions of delayed observations and undermines subsequent policy training. In contrast, DFBT's prediction error remains largely unaffected by the length of delays, enabling DFBT-SAC to maintain strong performance even in scenarios with long delays.

6.2.3. Ablation Study on Bootstrapping Steps

As mentioned in Section 4.2, we conduct the ablation study on the bootstrapping steps in DFBT-SAC. The result presented in Table 3 tells us that DFBT-SAC with 8 bootstrapping steps achieves the averaged best performance compared to other choices. It is also confirmed that bootstrapping with the states predicted by the trained DFBT can effectively improve performance.

Table 3. Final Performance on Walker2d-v2 of DFBT-SAC with different bootstrapping steps. The best performance is underlined.

Delays	Bootstrapping Steps N			
	1	2	4	8
8	$0.86_{\pm 0.07}$	$0.96_{\pm 0.07}$	$1.00_{\pm 0.14}$	$\underline{1.11_{\pm 0.10}}$
16	$0.84_{\pm 0.24}$	$0.76_{\pm 0.21}$	$\underline{1.02_{\pm 0.12}}$	$0.99_{\pm 0.06}$
32	$0.63_{\pm 0.22}$	$0.53_{\pm 0.15}$	$0.67_{\pm 0.20}$	$\underline{0.67_{\pm 0.15}}$
64	$0.34_{\pm 0.24}$	$0.28_{\pm 0.22}$	$0.29_{\pm 0.15}$	$\underline{0.41_{\pm 0.10}}$
128	$0.24_{\pm 0.03}$	$0.29_{\pm 0.07}$	$0.27_{\pm 0.14}$	$\underline{0.30_{\pm 0.13}}$

Table 4. Performance comparison of DFBT-SAC with different belief training methods on MuJoCo tasks with 32 delays. The best performance is underlined.

Task	Online	Offline	Offline + Fine-tuning
HalfCheetah-v2	0.11 ± 0.39	$\underline{0.42 \pm 0.12}$	0.39 ± 0.04
Hopper-v2	0.10 ± 0.58	0.68 ± 0.20	$\underline{0.84 \pm 0.04}$
Walker2d-v2	0.09 ± 0.27	0.64 ± 0.10	$\underline{0.96 \pm 0.32}$

6.2.4. Evaluation on Additional MuJoCo tasks.

we conducted experiments on the Pusher-v2, Reacher-v2, and Swimmer-v2 with deterministic 32 delays. The offline datasets (500k samples) are collected from a SAC policy, with other settings unchanged. The results are shown in Table 5, showing that DFBT-SAC achieves superior performance on these tasks.

Table 5. Performance comparison on additional MuJoCo tasks with 32 deterministic delays. The best performance is underlined.

Task	A-SAC	BPQL	ADRL	DATS	D-Dreamer	D-SAC	DFBT-SAC (ours)
Pusher-v2	0.05 ± 0.00	0.93 ± 0.58	0.95 ± 0.24	0.92 ± 0.10	0.87 ± 0.18	0.94 ± 0.04	$\underline{1.04 \pm 0.24}$
Reacher-v2	0.89 ± 0.08	0.83 ± 0.06	0.85 ± 0.01	0.82 ± 0.13	0.84 ± 0.02	0.88 ± 0.07	$\underline{0.93 \pm 0.06}$
Swimmer-v2	0.27 ± 0.05	0.80 ± 0.14	0.60 ± 0.06	0.25 ± 0.05	0.21 ± 0.07	0.30 ± 0.07	$\underline{1.01 \pm 0.27}$

Table 6. Performance on stochastic MuJoCo with deterministic 128 delays. The best performance is underlined.

Task	A-SAC	BPQL	ADRL	DATS	D-Dreamer	D-SAC	DFBT-SAC (ours)
HalfCheetah-v2	0.00 ± 0.03	0.01 ± 0.05	0.13 ± 0.04	0.13 ± 0.03	0.07 ± 0.01	0.00 ± 0.04	$\underline{0.35 \pm 0.04}$
Hopper-v2	0.03 ± 0.04	0.08 ± 0.05	0.06 ± 0.04	0.05 ± 0.06	0.04 ± 0.05	0.04 ± 0.05	$\underline{0.13 \pm 0.22}$
Walker2d-v2	0.06 ± 0.02	0.04 ± 0.01	0.08 ± 0.01	0.06 ± 0.03	0.10 ± 0.03	0.05 ± 0.02	$\underline{0.30 \pm 0.07}$

6.2.5. Ablation Results of Belief Training.

Some task-specific information may be missing if the belief is frozen in the RL process, leading to limited performance improvement. This issue can be mitigated by fine-tuning the belief within the RL process. The results, shown in Table 4, demonstrate that fine-tuning helps the DFBT capture the task-specific information with better performance. Note that there are many potential methods for capturing task-specific information, not limited to fine-tuning DFBT. Belief learning from scratch in the online RL process always suffers from instability issues. Therefore, in this paper, we separate belief learning from the online RL process, which allows us to investigate the belief component solely, eliminate potential influences from the RL side.

6.2.6. Stochastic MuJoCo.

We conducted additional experiments on the stochastic MuJoCo tasks with a probability of 0.001 for the unaware noise and deterministic 128 delays. As shown in Table 6, the results demonstrate that our DFBT-SAC achieves superior performance in these stochastic MuJoCo tasks.

6.2.7. Inference Speed Comparison.

We conducted additional experiments on computational efficiency. As shown in Table 7, the results demonstrate that directly forecasting belief maintains a consistent and stable inference speed (around 4 ms) across different delays. In contrast, the recursively forecasting belief experiences inference speed issues as delays increase. In HalfCheetah-v2 with 128 delays, the training times of DATS and D-Dreamer are around 10 hours and 15 hours, respectively, while those of D-SAC and DFBT-SAC are both around 6 hours.

Table 7. Inference speed (ms) comparison in HalfCheetah-v2.

Delays	DATS	D-Dreamer	D-SAC	DFBT-SAC
8	1.10 ± 0.02	1.85 ± 0.01	4.03 ± 0.03	4.18 ± 0.04
32	3.85 ± 0.06	6.80 ± 0.04	4.03 ± 0.04	4.18 ± 0.04
128	14.97 ± 0.22	26.51 ± 0.19	4.03 ± 0.03	4.15 ± 0.05

6.3. Limitations and Challenges

In this work, we empirically validate that our approach can address the compounding errors of recursively forecasting belief with significant performance improvement. However, there remain some limitations and challenges as follows.

Online Belief Learning. In this paper, we mainly consider learning DFBT from the offline dataset, which means the performance of the belief is subject to the quality and quantity of the dataset. However, direct learning belief in the online environment will introduce an auxiliary representation task, destabilizing the learning process.

Sample Complexity of Directly Forecasting Belief. Although we have empirically demonstrated that directly forecasting belief can effectively reduce the compounding errors of recursively forecasting belief. However, it is worth theoretically analysing the sample complexity of belief learning, especially in online scenarios.

7. Conclusion

This work investigates the challenges of RL in environments where inherent delays exist between actions and their corresponding outcomes. Existing belief-based approaches usually suffer from the compounding errors issue of the recursively forecasting belief as the delays are increased, seriously hindering the performance. To resolve this issue, we present DFBT, a new directly forecasting belief representation method. Furthermore, we present DFBT-SAC, which facilitates multi-step bootstrapping in learning the value function via the state prediction from the DFBT, effectively improving the sample efficiency. We demonstrate that our DFBT greatly reduces compounding errors, yielding stronger performance guarantees. Our empirical results validate that DFBT has remarkable prediction accuracy in the D4RL datasets. We also empirically show that our DFBT-SAC not only effectively enhance the learning efficiency with superior performance in the MuJoCo benchmark.

Acknowledgement

We sincerely appreciate the insightful suggestions provided by ICML 2025 reviewers, which help us improve the quality of this paper. We sincerely acknowledge the support by the grant EP/Y002644/1 under the EPSRC ECR International Collaboration Grants program, funded by the International Science Partnerships Fund (ISPF) and the UK Research and Innovation. We also sincerely acknowledge the support by Taiwan NSTC under Grant Number NSTC-112-2221-E-002-168-MY3, and from King Abdullah University of Science and Technology (KAUST) - Center of Excellence for Generative AI, under award number 5940. Simon Sinong Zhan and Qi Zhu are partially supported by the US National Science Foundation grants 2324936 and 2328973.

Impact Statement

This paper aims to advance the field of Machine Learning. While acknowledging there are many potential societal consequences of our work, we believe that none need to be specially highlighted here.

References

Altman, E. and Nain, P. Closed-loop control with delayed information. *ACM sigmetrics performance evaluation review*, 20(1):193–204, 1992.

Arjona-Medina, J. A., Gillhofer, M., Widrich, M., Unterthiner, T., Brandstetter, J., and Hochreiter, S. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32, 2019.

Asadi, K., Misra, D., and Littman, M. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 264–273. PMLR, 2018.

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Bertsekas, D. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific, 2012.

Bouteiller, Y., Ramstedt, S., Beltrame, G., Pal, C., and Binas, J. Reinforcement learning with random delays. In *International conference on learning representations*, 2020.

Cao, Z., Guo, H., Song, W., Gao, K., Chen, Z., Zhang, L., and Zhang, X. Using reinforcement learning to minimize the probability of delay occurrence in transportation. *IEEE transactions on vehicular technology*, 69(3):2424–2436, 2020.

Chen, B., Xu, M., Li, L., and Zhao, D. Delay-aware model-based reinforcement learning for continuous control. *Neurocomputing*, 450:119–128, 2021a.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021b.

Chevillon, G. Direct multi-step estimation and forecasting. *Journal of Economic Surveys*, 21(4):746–785, 2007.

Clements, M. P. and Hendry, D. F. Multi-step estimation for forecasting. *Oxford Bulletin of Economics and Statistics*, 58(4):657–684, 1996.

Cooke, K. L. Differential—difference equations. In *International symposium on nonlinear differential equations and nonlinear mechanics*, pp. 155–171. Elsevier, 1963.

Feng, S., Chen, M., Zhan, N., Fränzle, M., and Xue, B. Taming delays in dynamical systems: Unbounded verification of delay differential equations. In *International Conference on Computer Aided Verification*, pp. 650–669. Springer, 2019.

Fridman, E. and Shaked, U. On reachable sets for linear systems with delay and bounded peak inputs. *Automatica*, 39(11):2005–2010, 2003.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Hasbrouck, J. and Saar, G. Low-latency trading. *Journal of Financial Markets*, 16(4):646–679, 2013.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Huang, S., Dossa, R. F. J., Ye, C., Braga, J., Chakraborty, D., Mehta, K., and AraÃšjo, J. G. Cleanrl: High-quality

single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

Hwangbo, J., Sa, I., Siegwart, R., and Hutter, M. Control of a quadrotor with reinforcement learning. *IEEE Robotics and Automation Letters*, 2(4):2096–2103, 2017.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34: 1273–1286, 2021.

Karamzade, A., Kim, K., Kalsi, M., and Fox, R. Reinforcement learning from delayed observations via world models. *arXiv preprint arXiv:2403.12309*, 2024.

Katsikopoulos, K. V. and Engelbrecht, S. E. Markov decision processes with delays and asynchronous cost collection. *IEEE transactions on automatic control*, 48(4): 568–574, 2003.

Kim, J., Kim, H., Kang, J., Baek, J., and Han, S. Belief projection-based reinforcement learning for environments with delayed feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Liotet, P., Venneri, E., and Restelli, M. Learning a belief representation for delayed reinforcement learning. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2021.

Liotet, P., Maran, D., Bisi, L., and Restelli, M. Delayed reinforcement learning by imitation. In *International Conference on Machine Learning*, pp. 13528–13556. PMLR, 2022.

Mahmood, A. R., Korenkevych, D., Komer, B. J., and Bergstra, J. Setting up a reinforcement learning task with a real-world robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4635–4640. IEEE, 2018.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Nath, S., Baranwal, M., and Khadilkar, H. Revisiting state augmentation methods for reinforcement learning with stochastic delays. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 1346–1355, 2021.

Rachelson, E. and Lagoudakis, M. G. On the locality of action domination in sequential decision making. 2010.

Schmidhuber, J. An on-line algorithm for dynamic reinforcement learning and planning in reactive environments. In *1990 IJCNN international joint conference on neural networks*, pp. 253–258. IEEE, 1990a.

Schmidhuber, J. *Making the world differentiable: on using self supervised fully recurrent neural networks for dynamic reinforcement learning and planning in non-stationary environments*, volume 126. Inst. für Informatik, 1990b.

Schmidhuber, J. Reinforcement learning upside down: Don't predict rewards–just map them to actions. *arXiv preprint arXiv:1912.02875*, 2019.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Taieb, S. B. and Atiya, A. F. A bias and variance analysis for multistep-ahead time series forecasting. *IEEE transactions on neural networks and learning systems*, 27(1): 62–76, 2015.

Tarasov, D., Nikulin, A., Akimov, D., Kurenkov, V., and Kolesnikov, S. CORL: Research-oriented deep offline reinforcement learning library. In *3rd Offline RL Workshop: Offline RL as a "Launchpad"*, 2022. URL https://openreview.net/forum?id=SyAS49bBcv.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Walsh, T. J., Nouri, A., Li, L., and Littman, M. L. Learning and planning in environments with delayed feedback. *Autonomous Agents and Multi-Agent Systems*, 18:83–105, 2009.

Wang, W., Han, D., Luo, X., and Li, D. Addressing signal delay in deep reinforcement learning. In *The Twelfth*

International Conference on Learning Representations, 2023a.

Wang, Y., Zhan, S., Wang, Z., Huang, C., Wang, Z., Yang, Z., and Zhu, Q. Joint differentiable optimization and verification for certified reinforcement learning. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, pp. 132–141, 2023b.

Wang, Y., Zhan, S. S., Jiao, R., Wang, Z., Jin, W., Yang, Z., Wang, Z., Huang, C., and Zhu, Q. Enforcing hard constraints with soft barriers: Safe reinforcement learning in unknown stochastic environments. In *International Conference on Machine Learning*, pp. 36593–36604. PMLR, 2023c.

Wang, Y., Strupl, M., Faccio, F., Wu, Q., Liu, H., Grudzień, M., Tan, X., and Schmidhuber, J. Highway reinforcement learning. *arXiv preprint arXiv:2405.18289*, 2024.

Wei, T., Wang, Y., and Zhu, Q. Deep reinforcement learning for building hvac control. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, June 2017. doi: 10.1145/3061639.3062224.

Wu, Q., Zhan, S. S., Wang, Y., Wang, Y., Lin, C.-W., Lv, C., Zhu, Q., and Huang, C. Variational delayed policy optimization. *Advances in neural information processing systems*, 2024a.

Wu, Q., Zhan, S. S., Wang, Y., Wang, Y., Lin, C.-W., Lv, C., Zhu, Q., Schmidhuber, J., and Huang, C. Boosting reinforcement learning with strongly delayed feedback through auxiliary short delays. In *Forty-first International Conference on Machine Learning (ICML 2024)*, 2024b.

Xue, B., Wang, Q., Feng, S., and Zhan, N. Over-and under-approximating reach sets for perturbed delay differential equations. *IEEE Transactions on Automatic Control*, 66 (1):283–290, 2020.

Xue, B., Bai, Y., Zhan, N., Liu, W., and Jiao, L. Reach-avoid analysis for delay differential equations. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 1301–1307. IEEE, 2021.

Zhan, S., Wang, Y., Wu, Q., Jiao, R., Huang, C., and Zhu, Q. State-wise safe reinforcement learning with pixel observations. In *6th Annual Learning for Dynamics & Control Conference*, pp. 1187–1201. PMLR, 2024.

A. Implementation Details

The implementation of DFBT and DFBT-SAC is based on CORL (Tarasov et al., 2022) and CleanRL (Huang et al., 2022). The codebase for reproducing our experimental results is also provided in the Supplementary Material. We detail the hyperparameter settings of DFBT and DFBT-SAC in Table 8 and Table 9, respectively.

Table 8. Hyper-parameters table of DFBT.

Hyper-parameter	Value
Epoch	1e3
Batch Size	256
Attention Heads Num	4
Layers Num	10
Hidden Dim	256
Attention Dropout Rate	0.1
Residual Dropout Rate	0.1
Hidden Dropout Rate	0.1
Learning Rate	1e-4
Optimizer	AdamW
Weight Decay	1e-4
Betas	(0.9, 0.999)

Table 9. Hyper-parameters table of DFBT-SAC.

Hyper-parameter	Value
Bootstrapping Steps N	8
Learning Rate (Actor)	3e-4
Learning Rate (Critic)	1e-3
Learning Rate (Entropy)	1e-3
Train Frequency (Actor)	2
Train Frequency (Critic)	1
Soft Update Factor (Critic)	5e-3
Batch Size	256
Neurons	[256, 256]
Layers	3
Hidden Dim	256
Activation	ReLU
Optimizer	Adam

B. Theoretical Analysis

Theorem B.1 (Performance Difference of Recursively Forecasting Belief)**.** *For the delay-free policy π and the delayed policy π_Δ. Given any $x_t \in \mathcal{X}$, the performance difference $I^{recursive}(x_t)$ of the recursively forecasting belief b_θ can be bounded as follows, respectively.*

For deterministic delays Δ, we have

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

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

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

Proof. For deterministic delays Δ, the performance difference $I^{\text{recursive}}$ can be written as:

$$\begin{aligned} I^{\text{recursive}}(x_t) &= \mathop{\mathbb{E}}_{s_t\sim b_\theta(\cdot|x_t)}[V^\pi(s_t)] - V^{\pi_\Delta}(x_t) \\ &= I_\Delta^{\text{true}}(x_t) + \mathop{\mathbb{E}}_{s_t\sim b_\theta(\cdot|x_t)}[V^\pi(s_t)] - \mathop{\mathbb{E}}_{s_t\sim b(\cdot|x_t)}[V^\pi(s_t)]. \end{aligned}$$

And recall that we have the Lipschitz continuity of the value function V^π:

$$\left|\mathop{\mathbb{E}}_{s_t\sim b_\theta(\cdot|x_t)}[V^\pi(s_t)] - \mathop{\mathbb{E}}_{s_t\sim b(\cdot|x_t)}[V^\pi(s_t)]\right| \leq L_V \mathcal{W}(b_\theta(\cdot|x_t)||b(\cdot|x_t)).$$

For $\mathcal{W}(b_\theta(\cdot|x_t)||b(\cdot|x_t))$, we follow the proof sketch of (Asadi et al., 2018).

We use $b^{(i)}(\cdot|x_t)(i=1,\ldots,\Delta)$ to note that the belief function with the specific delays i. For instance, $b^{(\Delta)}(\cdot|x_t) = b(\cdot|x_t)$ and $b^{(1)}(\cdot|x_t) = \mathcal{P}(\cdot|s_t,a_t)$.

Then, we have

$$\begin{aligned} &\mathcal{W}(b_\theta^{(\Delta)}(\cdot|x_t)||b^{(\Delta)}(\cdot|x_t)) \\ &= \mathcal{W}(\mathcal{P}_\theta(\cdot|b_\theta^{(\Delta-1)}(\cdot|x_t),a_{t-1})||\mathcal{P}(\cdot|b^{(\Delta-1)}(\cdot|x_t),a_{t-1})) \\ &\leq \mathcal{W}(\mathcal{P}_\theta(\cdot|b_\theta^{(\Delta-1)}(\cdot|x_t),a_{t-1})||\mathcal{P}(\cdot|b_\theta^{(\Delta-1)}(\cdot|x_t),a_{t-1})) + \mathcal{W}(\mathcal{P}(\cdot|b_\theta^{(\Delta-1)}(\cdot|x_t),a_{t-1})||\mathcal{P}(\cdot|b^{(\Delta-1)}(\cdot|x_t),a_{t-1})) \\ &\leq \epsilon_\mathcal{P} + L_\mathcal{P}\mathcal{W}(b_\theta^{\Delta-1}(\cdot|x_t)||b^{\Delta-1}(\cdot|x_t)) \\ &\leq (1+L_\mathcal{P})\epsilon_\mathcal{P} + {L_\mathcal{P}}^2\mathcal{W}(b_\theta^{(\Delta-2)}(\cdot|x_t)||b^{(\Delta-2)}(\cdot|x_t)) \\ &\leq \cdots \\ &\leq (1+\cdots+{L_\mathcal{P}}^{\Delta-2})\epsilon_\mathcal{P} + {L_\mathcal{P}}^{\Delta-1}\mathcal{W}(b_\theta^{(1)}(\cdot|x_t)||b^{(1)}(\cdot|x_t)) \\ &\leq (1+\cdots+{L_\mathcal{P}}^{\Delta-1})\epsilon_\mathcal{P} \\ &= \frac{1-{L_\mathcal{P}}^\Delta}{1-L_\mathcal{P}}\epsilon_\mathcal{P}. \end{aligned}$$

Therefore, we have

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

The above theoretical results can be extended to the stochastic delays $\delta \sim d_\Delta(\cdot)$ easily. The performance difference of the ground-truth belief I_δ^{true} is defined as:

$$I_\delta^{\text{true}}(x_t) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{\hat{s}\sim b_\delta(\cdot|\hat{x}) \\ \hat{a}\sim\pi_\delta(\cdot|\hat{x}) \\ \hat{x}\sim d_\delta^\pi(\cdot|x_t)}} \left[V^\pi(\hat{s}) - Q^\pi(\hat{s},\hat{a})\right].$$

Finally, we have

$$\left|I^{\text{recursive}}(x_t)\right| \le \sum_{\delta=1}^{\Delta} d_\Delta(\delta) \left[\left|I_\delta^{\text{true}}(x_t)\right| + L_V \frac{1-{L_\mathcal{P}}^\delta}{1-L_\mathcal{P}} \epsilon_\mathcal{P}\right].$$

□

Proposition B.2 (Performance Degeneration Bound of Directly Forecasting Belief)**.** *For the delay-free policy π and the delayed policy π_Δ. Given any $x_t \in \mathcal{X}$, the performance degeneration I^{direct} of the directly forecasting belief b_θ can bounded as follows respectively.*

For deterministic delays Δ, we have

$$\left|I^{direct}(x_t)\right| \le |I_\Delta^{true}(x_t)| + L_V \epsilon_{direct}.$$

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

$$\left|I^{direct}(x_t)\right| \le \mathop{\mathbb{E}}_{\delta\sim d_\Delta(\cdot)} [|I_\delta^{true}(x_t)|] + L_V \epsilon_{direct}.$$

Proof. Applying Assumption 5.6 and the proof of Theorem B.1. □

Proposition B.3 (Performance Degeneration Comparison)**.** *Directly forecasting belief could achieve a better performance guarantee $\left|I^{direct}(x_t)\right| \le \left|I^{recursive}(x_t)\right|$, if we have*

$$\epsilon_{direct} \le \frac{1-{L_\mathcal{P}}^\Delta}{1-L_\mathcal{P}} \epsilon_\mathcal{P}$$

for deterministic delays Δ, and

$$\epsilon_{direct} \le \mathop{\mathbb{E}}_{\delta\sim d_\Delta(\cdot)} \left[\frac{1-{L_\mathcal{P}}^\delta}{1-L_\mathcal{P}}\right] \epsilon_\mathcal{P}$$

for stochastic delays $\delta \sim d_\Delta(\cdot)$.

Proof. For deterministic delays Δ, if we have

$$\epsilon_{direct} \le \frac{1-{L_\mathcal{P}}^\Delta}{1-L_\mathcal{P}} \epsilon_\mathcal{P},$$

then it is obvious that we have

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

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

$$\epsilon_{direct} \le \mathop{\mathbb{E}}_{\delta\sim d_\Delta(\cdot)} \left[\frac{1-{L_\mathcal{P}}^\delta}{1-L_\mathcal{P}} \epsilon_\mathcal{P}\right],$$

then it is obvious that we have

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

□

C. Additional Results on MuJoCo.

We report additional experimental results on MuJoCo, including more different patterns of deterministic and stochastic delays in Table 10 and Table 11, respectively.

Table 10. Performance on MuJoCo with Deterministic Delays. The best performance is underlined, the best belief-based method is in red.

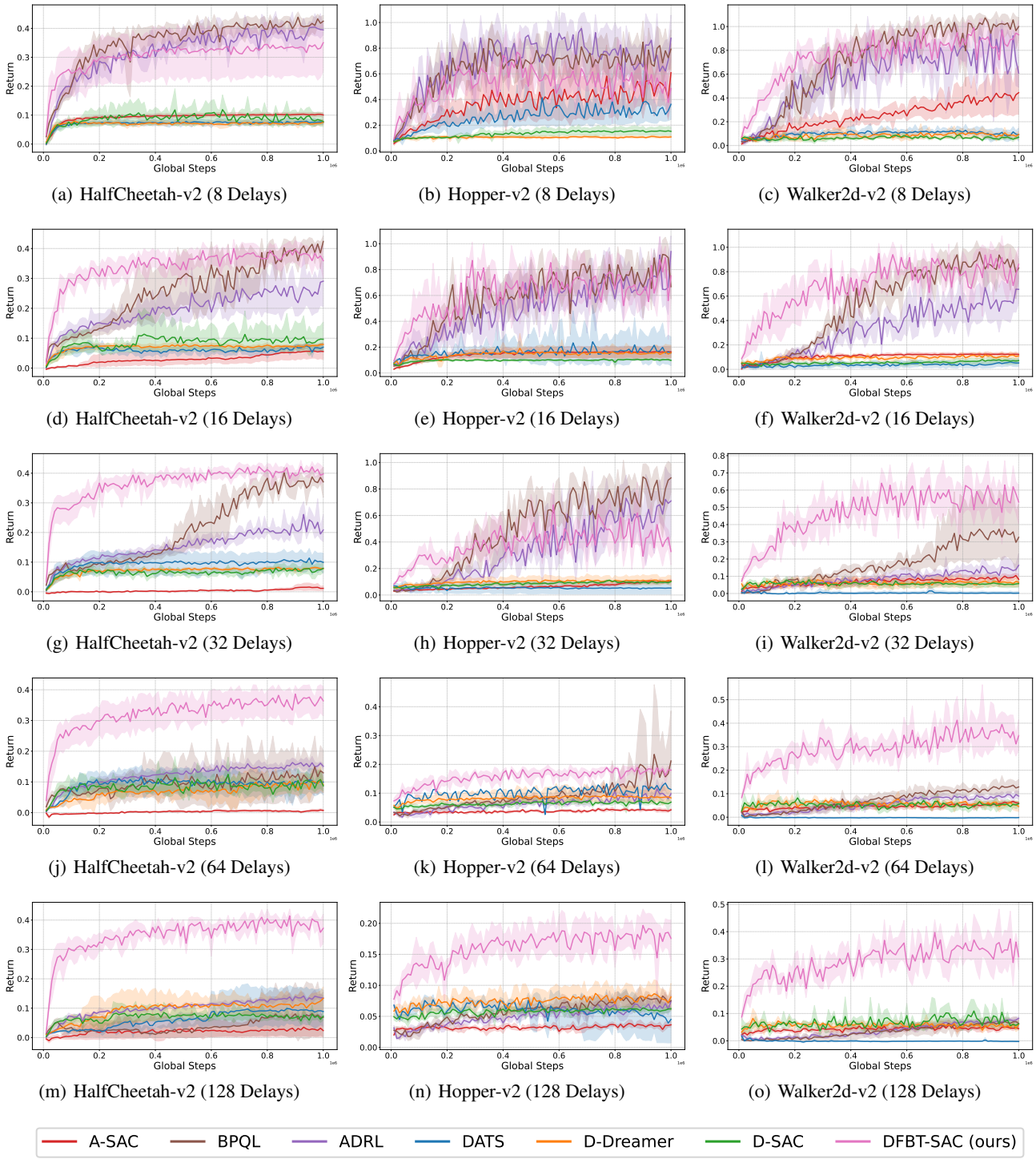
Task	Delays	Augmentation-based			Belief-based			
		A-SAC	BPQL	ADRL	DATS	D-Dreamer	D-SAC	DFBT-SAC (ours)
HalfCheetah-v2	8	$0.10_{\pm 0.01}$	$0.40_{\pm 0.04}$	$0.44_{\pm 0.03}$	$0.08_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.12_{\pm 0.06}$	$0.35_{\pm 0.12}$
	16	$0.06_{\pm 0.03}$	$0.42_{\pm 0.02}$	$0.29_{\pm 0.10}$	$0.08_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.11_{\pm 0.10}$	$0.40_{\pm 0.05}$
	32	$0.02_{\pm 0.02}$	$0.40_{\pm 0.03}$	$0.26_{\pm 0.04}$	$0.11_{\pm 0.04}$	$0.08_{\pm 0.00}$	$0.08_{\pm 0.02}$	$0.42_{\pm 0.03}$
	64	$0.01_{\pm 0.01}$	$0.15_{\pm 0.12}$	$0.16_{\pm 0.02}$	$0.12_{\pm 0.05}$	$0.11_{\pm 0.05}$	$0.12_{\pm 0.07}$	$0.39_{\pm 0.06}$
	128	$0.04_{\pm 0.06}$	$0.08_{\pm 0.13}$	$0.14_{\pm 0.02}$	$0.10_{\pm 0.08}$	$0.15_{\pm 0.05}$	$0.09_{\pm 0.04}$	$0.41_{\pm 0.03}$
Hopper-v2	8	$0.61_{\pm 0.31}$	$0.87_{\pm 0.09}$	$0.95_{\pm 0.16}$	$0.41_{\pm 0.31}$	$0.11_{\pm 0.01}$	$0.16_{\pm 0.05}$	$0.77_{\pm 0.18}$
	16	$0.17_{\pm 0.06}$	$0.92_{\pm 0.16}$	$0.94_{\pm 0.17}$	$0.24_{\pm 0.31}$	$0.19_{\pm 0.13}$	$0.11_{\pm 0.01}$	$0.89_{\pm 0.13}$
	32	$0.11_{\pm 0.02}$	$0.89_{\pm 0.14}$	$0.73_{\pm 0.20}$	$0.07_{\pm 0.04}$	$0.11_{\pm 0.05}$	$0.11_{\pm 0.01}$	$0.68_{\pm 0.20}$
	64	$0.05_{\pm 0.00}$	$0.23_{\pm 0.30}$	$0.11_{\pm 0.03}$	$0.13_{\pm 0.00}$	$0.09_{\pm 0.05}$	$0.08_{\pm 0.02}$	$0.19_{\pm 0.02}$
	128	$0.04_{\pm 0.01}$	$0.08_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.09_{\pm 0.03}$	$0.06_{\pm 0.01}$	$0.20_{\pm 0.03}$
Walker2d-v2	8	$0.44_{\pm 0.26}$	$1.07_{\pm 0.02}$	$0.97_{\pm 0.10}$	$0.13_{\pm 0.05}$	$0.11_{\pm 0.06}$	$0.09_{\pm 0.05}$	$0.99_{\pm 0.03}$
	16	$0.13_{\pm 0.02}$	$0.96_{\pm 0.05}$	$0.67_{\pm 0.21}$	$0.06_{\pm 0.10}$	$0.12_{\pm 0.03}$	$0.08_{\pm 0.04}$	$0.95_{\pm 0.11}$
	32	$0.10_{\pm 0.02}$	$0.37_{\pm 0.25}$	$0.16_{\pm 0.08}$	$0.02_{\pm 0.03}$	$0.08_{\pm 0.05}$	$0.08_{\pm 0.02}$	$0.64_{\pm 0.10}$
	64	$0.07_{\pm 0.01}$	$0.14_{\pm 0.03}$	$0.10_{\pm 0.01}$	$0.01_{\pm 0.02}$	$0.08_{\pm 0.03}$	$0.08_{\pm 0.04}$	$0.41_{\pm 0.14}$
	128	$0.06_{\pm 0.00}$	$0.07_{\pm 0.03}$	$0.08_{\pm 0.01}$	$0.02_{\pm 0.02}$	$0.08_{\pm 0.05}$	$0.11_{\pm 0.06}$	$0.40_{\pm 0.08}$

Table 11. Performance on MuJoCo with Stochastic Delays. The best performance is underlined, and the best belief-based method is in red.

Task	Delays	Augmentation-based			Belief-based			
		A-SAC	BPQL	ADRL	DATS	D-Dreamer	D-SAC	DFBT-SAC (ours)
HalfCheetah-v2	$U(1,8)$	$0.09_{\pm 0.01}$	$0.21_{\pm 0.07}$	$0.17_{\pm 0.07}$	$0.09_{\pm 0.03}$	$0.02_{\pm 0.01}$	$0.03_{\pm 0.01}$	$0.37_{\pm 0.12}$
	$U(1,16)$	$0.04_{\pm 0.04}$	$0.31_{\pm 0.08}$	$0.24_{\pm 0.04}$	$0.13_{\pm 0.03}$	$0.03_{\pm 0.02}$	$0.01_{\pm 0.01}$	$0.37_{\pm 0.06}$
	$U(1,32)$	$0.01_{\pm 0.00}$	$0.33_{\pm 0.07}$	$0.23_{\pm 0.02}$	$0.11_{\pm 0.04}$	$0.02_{\pm 0.00}$	$0.01_{\pm 0.01}$	$0.31_{\pm 0.16}$
	$U(1,64)$	$0.06_{\pm 0.11}$	$0.23_{\pm 0.06}$	$0.17_{\pm 0.02}$	$0.16_{\pm 0.03}$	$0.04_{\pm 0.03}$	$0.01_{\pm 0.00}$	$0.40_{\pm 0.06}$
	$U(1,128)$	$0.01_{\pm 0.01}$	$0.03_{\pm 0.03}$	$0.15_{\pm 0.02}$	$0.16_{\pm 0.03}$	$0.16_{\pm 0.00}$	$0.02_{\pm 0.00}$	$0.39_{\pm 0.04}$
Hopper-v2	$U(1,8)$	$0.17_{\pm 0.05}$	$0.20_{\pm 0.04}$	$0.18_{\pm 0.04}$	$0.04_{\pm 0.01}$	$0.07_{\pm 0.05}$	$0.14_{\pm 0.04}$	$0.86_{\pm 0.18}$
	$U(1,16)$	$0.08_{\pm 0.02}$	$0.11_{\pm 0.11}$	$0.07_{\pm 0.04}$	$0.04_{\pm 0.01}$	$0.03_{\pm 0.01}$	$0.04_{\pm 0.02}$	$0.89_{\pm 0.17}$
	$U(1,32)$	$0.05_{\pm 0.01}$	$0.07_{\pm 0.09}$	$0.05_{\pm 0.01}$	$0.05_{\pm 0.01}$	$0.04_{\pm 0.01}$	$0.03_{\pm 0.01}$	$0.43_{\pm 0.21}$
	$U(1,64)$	$0.03_{\pm 0.01}$	$0.03_{\pm 0.01}$	$0.03_{\pm 0.01}$	$0.05_{\pm 0.01}$	$0.03_{\pm 0.01}$	$0.03_{\pm 0.01}$	$0.17_{\pm 0.05}$
	$U(1,128)$	$0.03_{\pm 0.01}$	$0.04_{\pm 0.01}$	$0.04_{\pm 0.02}$	$0.05_{\pm 0.00}$	$0.03_{\pm 0.01}$	$0.03_{\pm 0.00}$	$0.14_{\pm 0.01}$
Walker2d-v2	$U(1,8)$	$0.36_{\pm 0.24}$	$0.40_{\pm 0.32}$	$0.41_{\pm 0.15}$	$0.07_{\pm 0.01}$	$0.07_{\pm 0.05}$	$0.12_{\pm 0.04}$	$1.11_{\pm 0.10}$
	$U(1,16)$	$0.19_{\pm 0.10}$	$0.27_{\pm 0.17}$	$0.24_{\pm 0.10}$	$0.08_{\pm 0.02}$	$0.13_{\pm 0.08}$	$0.07_{\pm 0.02}$	$0.99_{\pm 0.06}$
	$U(1,32)$	$0.12_{\pm 0.03}$	$0.16_{\pm 0.04}$	$0.11_{\pm 0.05}$	$0.09_{\pm 0.04}$	$0.12_{\pm 0.04}$	$0.05_{\pm 0.02}$	$0.67_{\pm 0.15}$
	$U(1,64)$	$0.08_{\pm 0.02}$	$0.09_{\pm 0.08}$	$0.06_{\pm 0.01}$	$0.08_{\pm 0.04}$	$0.15_{\pm 0.05}$	$0.06_{\pm 0.03}$	$0.41_{\pm 0.10}$
	$U(1,128)$	$0.06_{\pm 0.01}$	$0.06_{\pm 0.06}$	$0.04_{\pm 0.02}$	$0.10_{\pm 0.04}$	$0.15_{\pm 0.07}$	$0.03_{\pm 0.04}$	$0.30_{\pm 0.13}$

D. Learning Curves on MuJoCo.

We report learning curves on MuJoCo with different patterns of deterministic and stochastic delays in Figure 4 and Figure 5, respectively.



(a) HalfCheetah-v2 (8 Delays) (b) Hopper-v2 (8 Delays) (c) Walker2d-v2 (8 Delays)

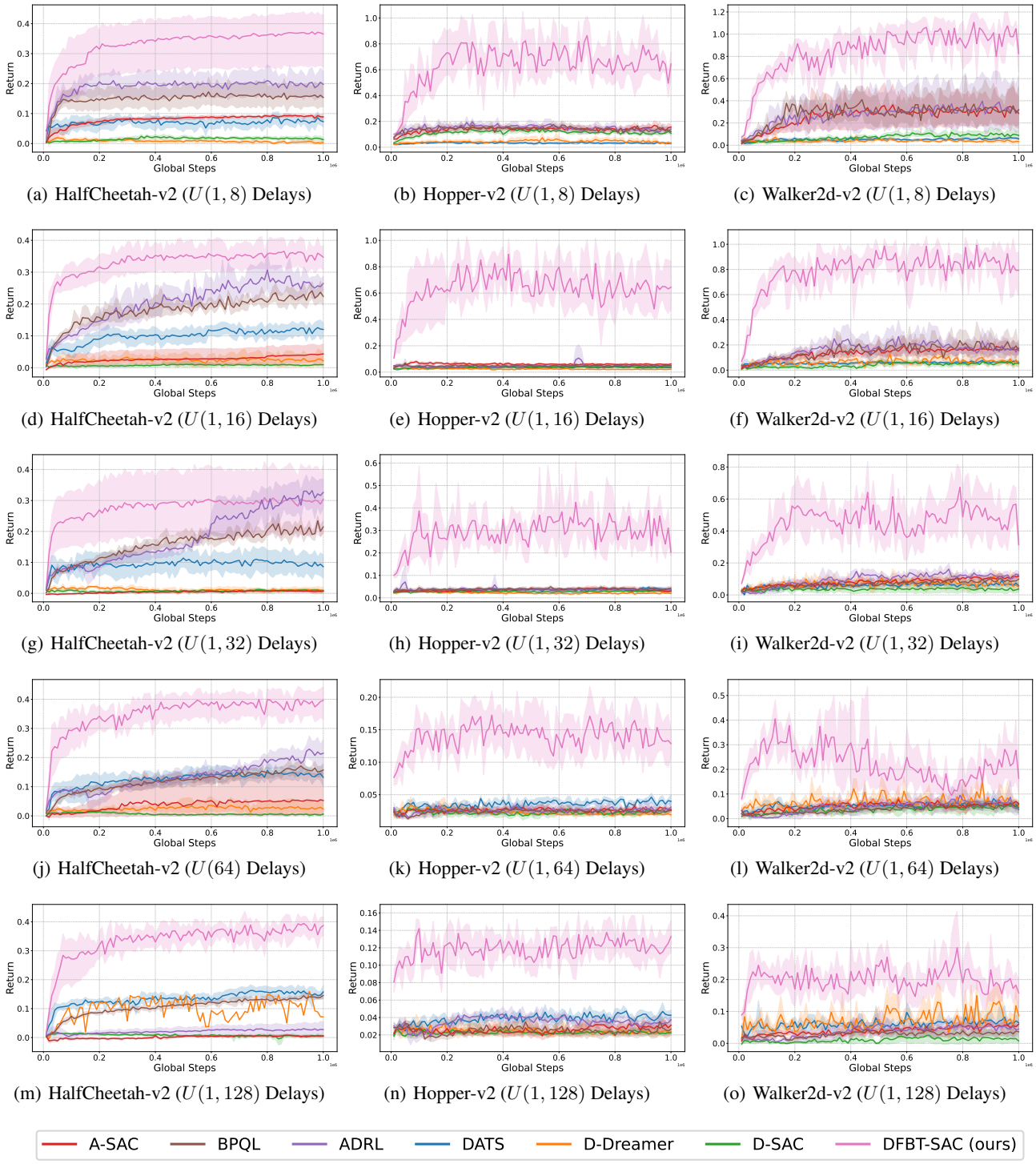
(d) HalfCheetah-v2 (16 Delays) (e) Hopper-v2 (16 Delays) (f) Walker2d-v2 (16 Delays)

(g) HalfCheetah-v2 (32 Delays) (h) Hopper-v2 (32 Delays) (i) Walker2d-v2 (32 Delays)

(j) HalfCheetah-v2 (64 Delays) (k) Hopper-v2 (64 Delays) (l) Walker2d-v2 (64 Delays)

(m) HalfCheetah-v2 (128 Delays) (n) Hopper-v2 (128 Delays) (o) Walker2d-v2 (128 Delays)

Figure 4. Learning Curves on MuJoCo with Deterministic Delays.



(a) HalfCheetah-v2 ($U(1,8)$ Delays) (b) Hopper-v2 ($U(1,8)$ Delays) (c) Walker2d-v2 ($U(1,8)$ Delays)

(d) HalfCheetah-v2 ($U(1,16)$ Delays) (e) Hopper-v2 ($U(1,16)$ Delays) (f) Walker2d-v2 ($U(1,16)$ Delays)

(g) HalfCheetah-v2 ($U(1,32)$ Delays) (h) Hopper-v2 ($U(1,32)$ Delays) (i) Walker2d-v2 ($U(1,32)$ Delays)

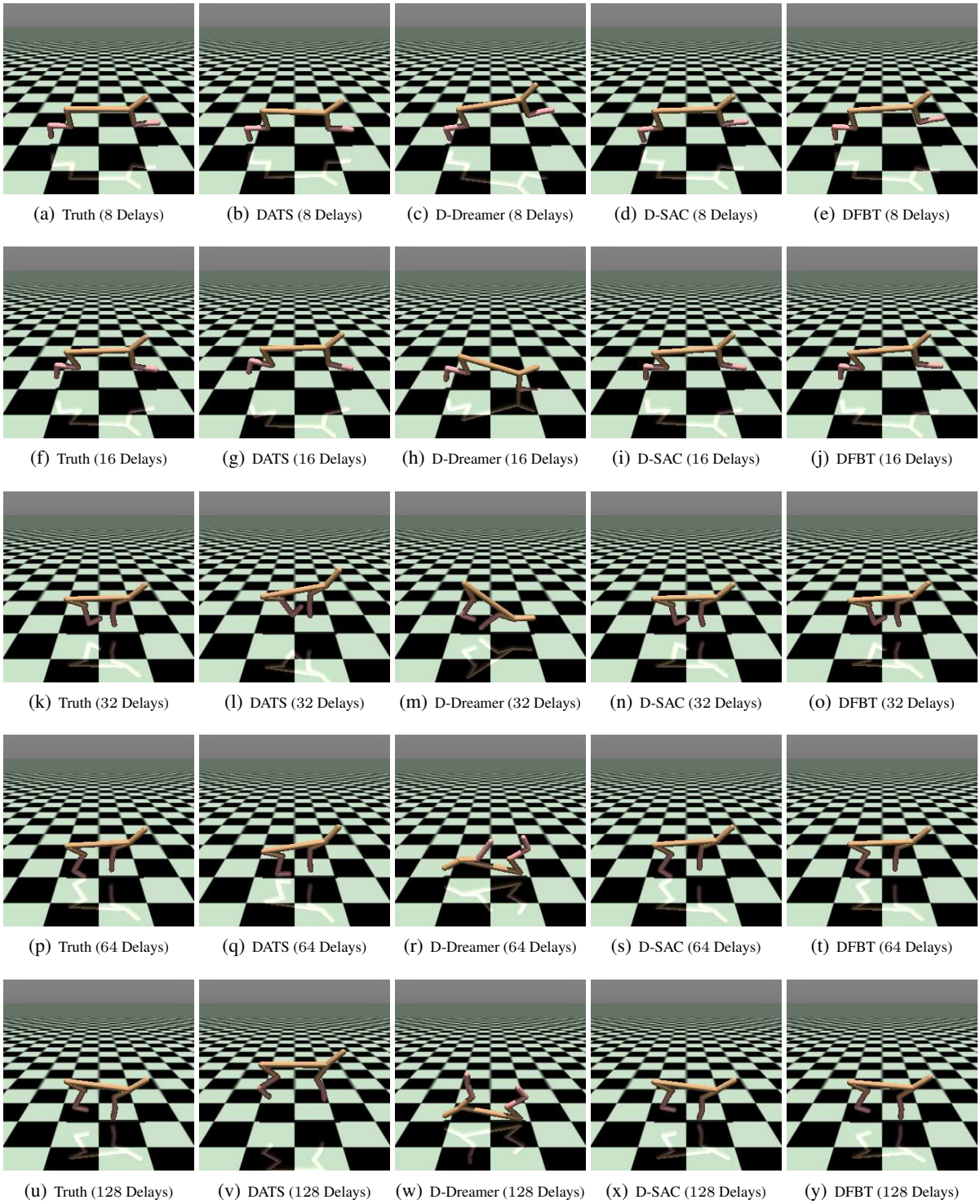
(j) HalfCheetah-v2 ($U(64)$ Delays) (k) Hopper-v2 ($U(1,64)$ Delays) (l) Walker2d-v2 ($U(1,64)$ Delays)

(m) HalfCheetah-v2 ($U(1,128)$ Delays) (n) Hopper-v2 ($U(1,128)$ Delays) (o) Walker2d-v2 ($U(1,128)$ Delays)

Figure 5. Learning Curves on MuJoCo with Stochastic Delays.

E. Belief Qualitative Comparison

We report the qualitative comparison of the beliefs on HalfCheetah-v2, Hopper-v2, and Walker2d-v2 in Figure 6, Figure 7, and Figure 8, respectively.



(a) Truth (8 Delays) (b) DATS (8 Delays) (c) D-Dreamer (8 Delays) (d) D-SAC (8 Delays) (e) DFBT (8 Delays)

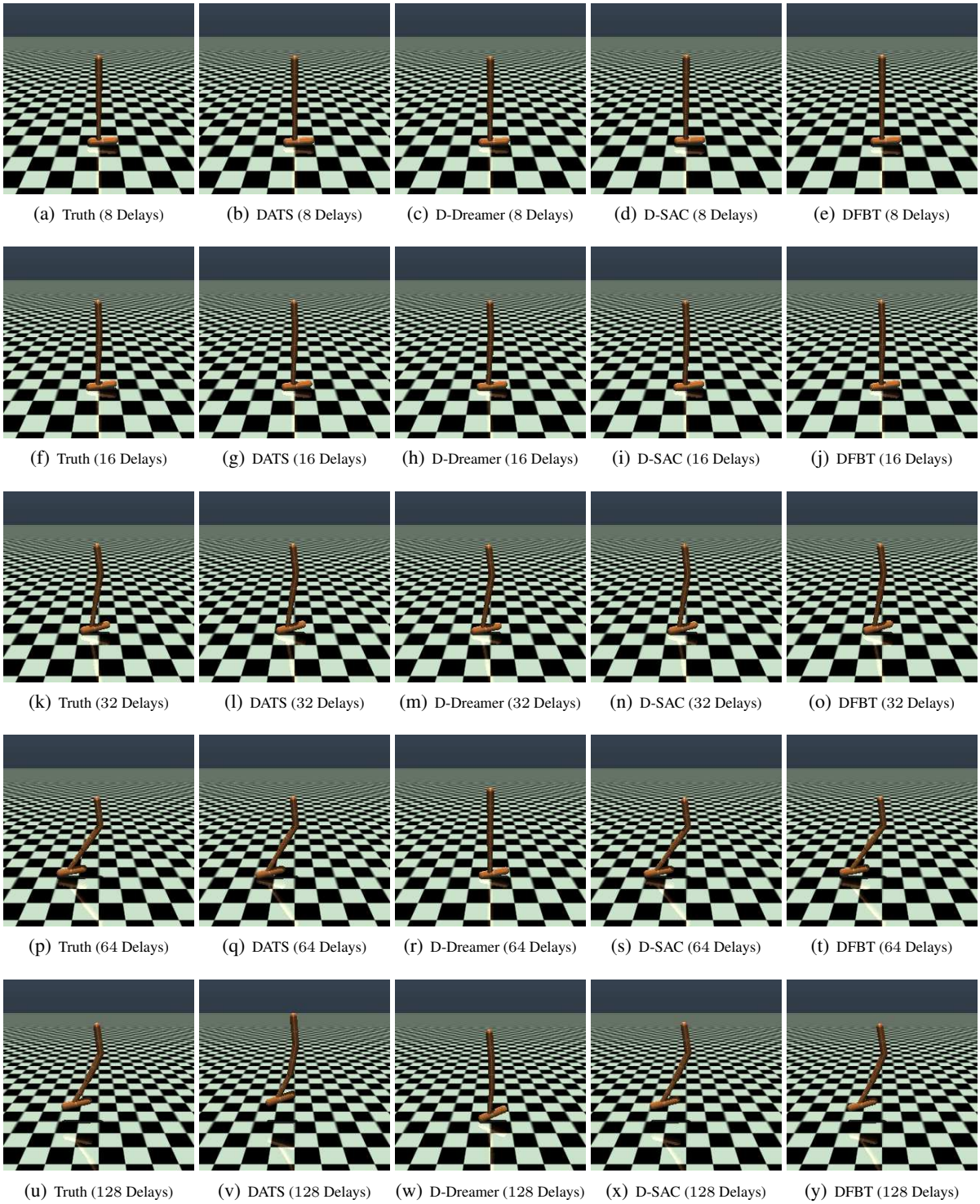
(f) Truth (16 Delays) (g) DATS (16 Delays) (h) D-Dreamer (16 Delays) (i) D-SAC (16 Delays) (j) DFBT (16 Delays)

(k) Truth (32 Delays) (l) DATS (32 Delays) (m) D-Dreamer (32 Delays) (n) D-SAC (32 Delays) (o) DFBT (32 Delays)

(p) Truth (64 Delays) (q) DATS (64 Delays) (r) D-Dreamer (64 Delays) (s) D-SAC (64 Delays) (t) DFBT (64 Delays)

(u) Truth (128 Delays) (v) DATS (128 Delays) (w) D-Dreamer (128 Delays) (x) D-SAC (128 Delays) (y) DFBT (128 Delays)

Figure 6. Belief qualitative comparison on HalfCheetah-v2 with different delays.



(a) Truth (8 Delays) (b) DATS (8 Delays) (c) D-Dreamer (8 Delays) (d) D-SAC (8 Delays) (e) DFBT (8 Delays)

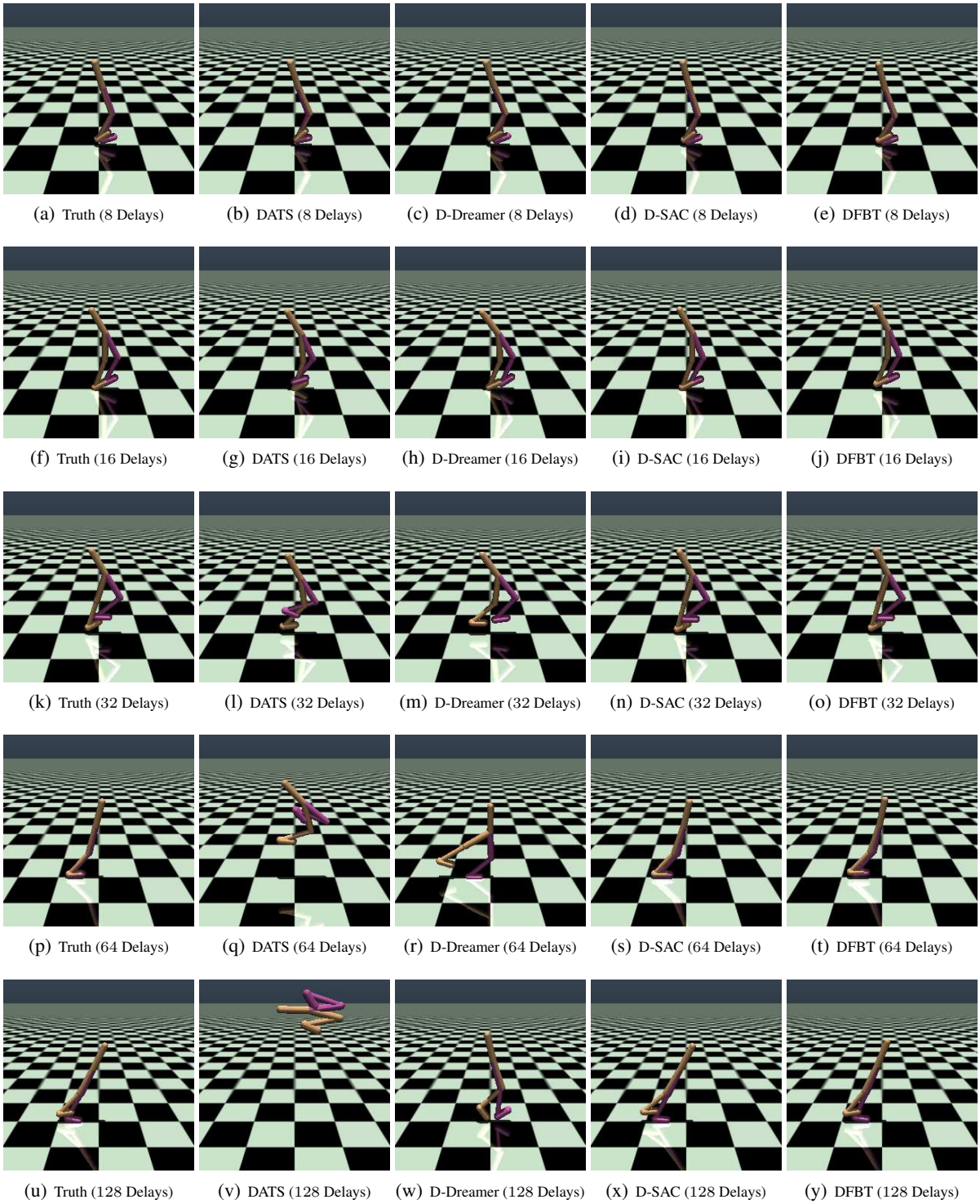
(f) Truth (16 Delays) (g) DATS (16 Delays) (h) D-Dreamer (16 Delays) (i) D-SAC (16 Delays) (j) DFBT (16 Delays)

(k) Truth (32 Delays) (l) DATS (32 Delays) (m) D-Dreamer (32 Delays) (n) D-SAC (32 Delays) (o) DFBT (32 Delays)

(p) Truth (64 Delays) (q) DATS (64 Delays) (r) D-Dreamer (64 Delays) (s) D-SAC (64 Delays) (t) DFBT (64 Delays)

(u) Truth (128 Delays) (v) DATS (128 Delays) (w) D-Dreamer (128 Delays) (x) D-SAC (128 Delays) (y) DFBT (128 Delays)

Figure 7. Belief qualitative comparison on Hopper-v2 with different delays.



(a) Truth (8 Delays) (b) DATS (8 Delays) (c) D-Dreamer (8 Delays) (d) D-SAC (8 Delays) (e) DFBT (8 Delays)

(f) Truth (16 Delays) (g) DATS (16 Delays) (h) D-Dreamer (16 Delays) (i) D-SAC (16 Delays) (j) DFBT (16 Delays)

(k) Truth (32 Delays) (l) DATS (32 Delays) (m) D-Dreamer (32 Delays) (n) D-SAC (32 Delays) (o) DFBT (32 Delays)

(p) Truth (64 Delays) (q) DATS (64 Delays) (r) D-Dreamer (64 Delays) (s) D-SAC (64 Delays) (t) DFBT (64 Delays)

(u) Truth (128 Delays) (v) DATS (128 Delays) (w) D-Dreamer (128 Delays) (x) D-SAC (128 Delays) (y) DFBT (128 Delays)

Figure 8. Belief qualitative comparison on Walker2d-v2 with different delays.