Delayed Reinforcement Learning by Imitation

Pierre Liotet [*1] **Davide Maran** [*1] **Lorenzo Bisi** [1] **Marcello Restelli** [1]

Abstract

When the agent's observations or interactions are delayed, classic reinforcement learning tools usually fail. In this paper, we propose a simple yet new and efficient solution to this problem. We assume that, in the undelayed environment, an efficient policy is known or can be easily learned, but the task may suffer from delays in practice and we thus want to take them into account. We present a novel algorithm, Delayed Imitation with Dataset Aggregation (DIDA), which builds upon imitation learning methods to learn how to act in a delayed environment from undelayed demonstrations. We provide a theoretical analysis of the approach that will guide the practical design of DIDA. These results are also of general interest in the delayed reinforcement learning literature by providing bounds on the performance between delayed and undelayed tasks, under smoothness conditions. We show empirically that DIDA obtains high performances with a remarkable sample efficiency on a variety of tasks, including robotic locomotion, classic control, and trading.

1. Introduction

In reinforcement learning (RL), it is generally assumed that the effect of an action over the environment is known instantaneously to the agent. However, in the presence of delays, this classic setting is challenged. The effect of a delayed action execution or state observation, if not accounted for, can have perilous effects in practice (Dulac-Arnold et al., 2019). It can induce a performance loss in trading (Wilcox, 1993), create instability in dynamic systems (Dugard & Verriest, 1998; Gu & Niculescu, 2003), be detrimental to the training of real-world robots (Mahmood et al., 2018). To further grasp the importance of delay, one may notice that most traffic laws around the world base safety distances on drivers' "reaction time", which is partly due to the perception of the event and partly to the implementation of the action (Droździel et al., 2020). These two types of delay have an exact correspondence in RL, where they are dubbed as state observation and action execution delays. There are many ways in which delays can further vary. They may be anonymous (i.e., not known to the agent), constant or stochastic, integer or non-integer. In this work, as in most of the literature, we focus on constant non-anonymous delays in the action execution or, equivalently (Katsikopoulos & Engelbrecht, 2003), in the state observation.

Previous research can be divided into three main directions. In *memoryless* approaches the agent's policy depends on the last observed state (Schuitema et al., 2010). *Augmented* approaches try to cast the problem into a Markov decision process (MDP) by building policies based on an augmented state, composed of the last observed state and on the actions that the agent knows it has taken since then (Bouteiller et al., 2020). A last line of research, which we refer to as *model-based* approach, considers using as a policy input any statistics about the current unknown state that can be computed from the augmented state (Walsh et al., 2009; Firoiu et al., 2018; Chen et al., 2021; Agarwal & Aggarwal, 2021; Derman et al., 2021; Liotet et al., 2021). The goal of this approach is to avoid the curse of dimensionality posed by the augmented state (Walsh et al., 2009; Bouteiller et al., 2020). We formalize the problem of delays in Section 3 and give a more in-depth description of the literature in Section 4.

We adopt the simple yet practically effective idea of learning a policy in a delayed environment by applying *imitation learning* to a policy learned in the undelayed environment, as described in Section 5. We propose to use DAGGER (Ross et al., 2011) as the imitation learning algorithm. It is particularly well suited to the task since, being one of the few algorithms to compute the loss under the learner's own distribution (Osa et al., 2018), it is able to account for the shift in distribution induced by the delay We provide a theoretical analysis (Section 6) which, under *smoothness* conditions over the MDP, bounds the performance lost by introducing delays. Finally, we provide an extensive experimental analysis (Section 7) where our algorithm is compared with state-of-the-art approaches on a variety of delayed problems and demonstrates great performances and sample efficiency.

[*]Equal contribution [1]Politecnico di Milano, Milan, Italy. Correspondence to: Pierre Liotet <pierre.liotet@polimi.it>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

2. Preliminaries

Reinforcement Learning A discrete-time discounted Markov Decision Process (MDP) (Puterman, 2014) is a 6-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, R, \gamma, \mu)$ where $\mathcal{S}$ and $\mathcal{A}$ are measurable sets of states and actions respectively, $p(s'|s,a)$ is the probability to transition to a state s' departing from state s and taking action a, $R(s,a)$ is a random variable defining the reward collected during such a transition. We denote by $r(s,a)$ its expected value. Finally, μ is the initial state distribution. The agent's goal is to find a policy π, which assigns probabilities to the actions given a state, to maximize the *expected discounted return* with discount factor $\gamma \in [0,1)$, defined as[1]

$$J(\pi) = \mathop{\mathbb{E}}_{\substack{s_{t+1}\sim p(\cdot|s_t,a_t)\\ a_t\sim\pi(\cdot|s_t)\\ s_0\sim\mu(\cdot)}}\left[\sum_{t=0}^{H}\gamma^t R(s_t,a_t)\right]. \quad (1)$$

We consider an infinite horizon setting, where $H = \infty$. Note that $\frac{1}{1-\gamma}$ can be seen as the effective horizon in this case. We restrict the set of policies to the *stationary Markovian* policies, Π, as it contains the optimal one (Puterman, 2014). RL analysis frequently introduces the concept of state-action value function, which quantifies the expected return obtained under some policy, starting from a given state and fixing the first action. Formally, this function is defined as

$$Q^\pi(s,a) = \mathop{\mathbb{E}}_{a_t\sim\pi(\cdot|s_t)}\left[\sum_{t=0}^{H}\gamma^t R(s_t,a_t)\Big|_{\substack{s_0=s,\\ a_0=a}}\right]. \quad (2)$$

Similarly we define the state value function as $V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^\pi(s,a)]$. Lastly, we consider the discounted visited state distribution under some policy π, starting from any initial distribution ρ, for some state $s \in \mathcal{S}$ as

$$d_\rho^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\,\mathbb{P}(s_t = s|\pi,\rho).$$

Lipschitz MDPs We now introduce notions that will allow us to characterize the smoothness of an MDP. Let $L > 0$ and let (X, d_X) and (Y, d_Y) be two metric spaces. A function $f : X \to Y$ is said to be L-Lipschitz continuous (L-LC) if, $\forall x, x' \in X$, $\mathrm{d}_Y(f(x), f(x')) \leqslant L\,\mathrm{d}_X(x,x')$. We denote the Lipschitz semi-norm of a function f as $\|f\|_L = \sup_{x,x'\in X, x\neq x'}\frac{\mathrm{d}_Y(f(x),f(x'))}{\mathrm{d}_X(x,x')}$. In real space $X \subset \mathbb{R}^n$, we use as distance the Euclidean one, i.e., $\mathrm{d}_X(x,x') = \|x - x'\|_2$. As for the probabilities, we use the L_1-Wasserstein distance, which for some probabilities μ, ν with sample space Ω is (Villani, 2009):

$$\mathcal{W}_1(\mu,\nu) = \sup_{\|f\|_L\leqslant 1}\left|\int_\Omega f(\omega)(\mu-\nu)(d\omega)\right|.$$

[1]In the sequel, we will tacit that $s_{t+1} \sim p(\cdot|s_t,a_t)$.

We now use those concepts to quantify the smoothness of an MDP.

Definition 2.1 (Lipschitz MDP)**.** *An MDP is said to be (L_P, L_r)-LC if, for all $(s,a),(s',a') \in \mathcal{S}\times\mathcal{A}$*

$$\mathcal{W}_1(p(\cdot|s,a),p(\cdot|s',a')) \leqslant L_P\left(\mathrm{d}_\mathcal{S}(s,s') + \mathrm{d}_\mathcal{A}(a,a')\right),$$
$$\left|r(s,a) - r(s',a')\right| \leqslant L_r\left(\mathrm{d}_\mathcal{S}(s,s') + \mathrm{d}_\mathcal{A}(a,a')\right).$$

Definition 2.2 (Lipschitz policy)**.** *A stationary Markovian policy π is said to be L_π-LC if, $\forall s, s' \in \mathcal{S}$*

$$\mathcal{W}_1(\pi(\cdot|s),\pi(\cdot|s')) \leqslant L_\pi\,\mathrm{d}_\mathcal{S}(s,s').$$

These concepts provide useful tools for theoretical analysis and have been extensively used in the field of RL (Rachelson & Lagoudakis, 2010). Under the assumption of (L_P, L_r)-LC MDP and L_π-LC policy π, provided that $\gamma L_P(1 + L_\pi) \leqslant 1$, then Q^π is L_Q-LC with $L_Q = \frac{L_r}{1-\gamma L_P(1+L_\pi)}$ (Rachelson & Lagoudakis, 2010, Theorem 1). This property can be useful to prove the Lipschitzness of the Q function.

Additionally, in the case of delays, the smoothness of trajectories (sequence of consecutive states and actions) is a key factor. Intuitively, smoother trajectories make the current unknown state more predictable. Therefore, we consider the concept of *time-Lipschitzness*, introduced by Metelli et al. (2020).

Definition 2.3 (Time-Lipschitz MDP)**.** *An MDP is said to be L_T-Time Lipschitz Continuous (L_T-TLC) if, $\forall s,a \in \mathcal{S}\times\mathcal{A}$*

$$\mathcal{W}_1(p(\cdot|s,a),\delta_s) \leqslant L_T,$$

where δ_s is the Dirac distribution with mass on s.

3. Problem Definition

A delayed MDP (DMDP) stems from an MDP endowed with a sequence of variables $(\Delta_t)_{t\in\mathbb{N}}$ corresponding to the delay at each step of the sequential process. The delay can affect the state observation, which implies that the agent has no access to the current state but only to a state visited Δ_t steps before. If it affects the action execution instead, the delay implies that the agent must select an action that will be executed in Δ_t steps from now. Lastly, reward collection delays may raise credit assignment issues and are outside the scope of this paper. In the first two cases, the DMDP violates the Markov assumption since the next observed state-reward couple does not depend only on the currently observable state and the chosen action. In the literature, the delay is usually assumed to be a Markovian process, that is $\Delta_t \sim P(\cdot|\Delta_{t-1}, s_{t-1}, a_{t-1})$. Note that this definition includes *state dependent delays* when $\Delta_t \sim P(\cdot|s_{t-1})$, *Markov chain delays* when $\Delta_t \sim P(\cdot|\Delta_{t-1})$ and *stochastic delays* when $(\Delta_t)_{t\in\mathbb{N}}$ are i.i.d.

In this work we consider *constant delays*, denoting the delay with the symbol Δ. When the delay is constant, the action execution delay and the state observation one are equivalent (Katsikopoulos & Engelbrecht, 2003), thus, it is sufficient to consider only the state observation delay. Furthermore, following Katsikopoulos & Engelbrecht (2003), we consider a reward collection delay equal to the state observation delay so as not to collect a reward on a yet unobserved state, which could result in some form of partial state information. Finally, we assume that the delay is known to the agent, placing ourselves in the *non-anonymous* delay framework.

Within this reduced framework, it is possible to introduce an important concept of DMDPs, the *augmented state*. Given the last observed state s and the sequence of actions $(a_1, \ldots, a_\Delta)$ which have been taken since then, but whose outcome has not yet been observed, the agent can construct an augmented state, i.e., a new state in $\mathcal{X} = \mathcal{S} \times \mathcal{A}^\Delta$ which casts the DMDP into an MDP (Bertsekas, 1987; Altman & Nain, 1992). Said alternatively, the augmented state contains all the information the agent needs to learn the optimal policy in the DMDP. From the augmented state, we can gather information on the current state. This information can be summarized by the *belief*, the probability distribution of the current unknown state s given the augmented state x as $b(s|x)$. More explicitly, given $x = (s_1, a_1, \ldots a_\Delta)$, one has

$$b(s|x) = \int_{S^{\Delta-1}} p(s|s_\Delta, a_\Delta) \prod_{i=2}^{\Delta} p(s_i|s_{i-1}, a_{i-1}) ds_i.$$

The delayed reward collected for playing action a on x is given by $\widetilde{r}(x, a) = \mathbb{E}_{s\sim b(\cdot|x)} [r(s, a)]$. To complete the DMDP framework, we define $\widetilde{\mu}$, the initial augmented state distribution. It samples the state contained in the augmented state under μ and samples the first action sequence under a distribution whose choice depends on the environment. We consider a uniform distribution on $\mathcal{A}$.

4. Related Works

The first proposed solution to the problem of delays is to use regular RL algorithms on the augmented-state MDP. Although the optimal delayed policy could potentially be obtained, this approach is affected by the exponential growth of the augmented state space, which becomes $|\mathcal{S}||\mathcal{A}|^\Delta$ (Walsh et al., 2009) and can be harmful for the performance in practice. Nonetheless, recent work by Bouteiller et al. (2020) revisits this approach and propose an ingenious way to resample trajectories without interacting with the environment by populating the augmented state with actions from a different policy, greatly improving the sample efficiency. They propose an algorithm, Delay-Correcting Actor-Critic (DCAC), which builds on SAC (Haarnoja et al., 2018) using the aforementioned resampling idea. DCAC has great experimental results and is sample efficient by design.

A second line of research focuses on memoryless policies, inspired by the partially observable MDP literature. It ignores the action queue to act according to the last observed state only. However, the delay can still be taken into account as in dSARSA (Schuitema et al., 2010), a modified version of SARSA (Sutton & Barto, 2018) which accounts for the delay during its update. Indeed, SARSA would credit the reward collected for applying action a on the augmented state x, containing the last observed state s, to the pair (s, a). Instead, dSARSA proposes to credit (s, a_1), where a_1 is the oldest action stored in x, the action actually applied on s. Despite being memoryless, dSARSA often achieves good performances in practice.

Finally, the most common line of research, the model-based approach, relies on computing statistics on the current state which are then used to select an action. The name model-based comes from the fact that those solutions usually learn a model of the environment to predict the current state, by simulating the effect of the actions stored in the augmented state on the last observed state. Walsh et al. (2009) learn the transition as a deterministic mapping so as to predict the most probable state, before selecting actions based on it. Derman et al. (2021) and Firoiu et al. (2018) propose a similar approach by learning the transitions with feed-forward and recurrent neural networks, respectively. Agarwal & Aggarwal (2021) estimate the transition probabilities and the undelayed Q function to select the action that gives the maximum Q under the estimated distribution of the current state. Chen et al. (2021) use a particle-based approach to produce potential outcomes for the current state and, interestingly, extend the predictions to collect better value estimates. Liotet et al. (2021) propose D-TRPO which learns a vectorial encoding of the belief of the current state which is then used as an input to the policy, the latter being trained with TRPO (Schulman et al., 2015a). The authors also propose another algorithm, L2-TRPO which, instead of the belief, learns the expected current state.

While most of these works assume that the delay is fixed, some consider the problem of stochastic delays (Bouteiller et al., 2020; Derman et al., 2021; Agarwal & Aggarwal, 2021). Only one of them considers the case of non-integer delays (Schuitema et al., 2010).

5. Imitation Learning for Delays

Our proposed approach is motivated by the limitations of two lines of research from the literature. Augmented approaches are affected by the increased dimensionality of the augmented state space that hinders the learning process, while model-based approaches require carefully designed models of the state transitions and usually come with a computational burden. Instead, we propose to learn a mapping from augmented state directly to undelayed expert actions, facilitating the learning process as opposed to augmented approaches and removing explicit approximation of transitions as opposed to model-based approaches. It implies however that training is split into two sub-problems: learning an expert undelayed policy and then imitating this policy in a DMDP.

5.1. Imitation Learning

It is usually easier to learn a behavior from demonstrations than learning from scratch using standard RL techniques. Imitation learning aims at learning a policy by mimicking the actions of an expert, bridging the gap between RL and *supervised learning*. Obviously, it requires that one can collect examples of an expert's behavior to learn from. For an expert policy π_E, most imitation learning approaches aim at finding a policy π that minimizes $\mathbb{E}_{s\sim d_\mu^{\pi_E}}[l(s,\pi)]$ (Ross et al., 2011) where $l(s,\pi)$ is a loss designed to make π closer to π_E. Note that this objective is defined under the state distribution induced by π_E. This can easily be problematic as, whenever the learner makes an error, it could end up in a state where its knowledge of the expert's behavior is poor and therefore errors could accumulate. Indeed, it has been shown (Xu et al., 2020, Theorem 1) that the error made by the learner potentially propagates as the squared effective horizon. This is consistent with other bounds found in the literature depending on H^2 in the finite horizon setting (Ross & Bagnell, 2010, Theorem 2.1).

One successful solution to this problem is *dataset aggregation* as proposed by Ross et al. (2011) in their DAGGER algorithm. The idea is to sample new data under the learned policy and query the expert on those new samples in order to match the learner's state distribution. DAGGER recursively builds a dataset $\mathcal{D}$ by sampling trajectories under policy $\pi_i = \beta_i \pi_E + (1-\beta_i)\hat{\pi}_i$ obtained from a β_i-weighted mixture of the expert policy and the previously imitated policy $\hat{\pi}_i$. One then queries the expert's policy on the states encountered in these trajectories and adds those tuples $(s, \pi_E(s))$ to $\mathcal{D}$. Finally, a new imitated policy $\hat{\pi}_{i+1}$ is trained on $\mathcal{D}$. The sequence $(\beta_i)_{i\in[\![1,N]\!]}$ is such that $\beta_1 = 1$, so as to sample initially only from π_E and $\beta_N = 0$ to sample only from the imitated policy in the end.

5.2. Duality of Trajectories

Once sampled either from a DMDP or its underlying MDP, the trajectories can be interpreted in either one of the processes when the delay is an integer number of steps. In a DMDP, the current state will eventually be observed by a delayed agent. In an MDP, the trajectories can be reorganized to simulate the effect of a delay. In particular, this means that one can collect trajectories with an undelayed environment and sample either from an undelayed policy or a delayed policy (by creating a synthetic augmented state). This is exactly what is required to adapt DAGGER to imitate an undelayed expert with a delayed learner.

Algorithm 1 Delayed Imitation with DAGGER (DIDA)

Inputs (un)delayed environment $\mathcal{E}$, undelayed expert π_E, β routine, number of steps N, empty dataset $\mathcal{D}$.
Outputs: delayed policy π_I

1: **for** β_i in β-routine **do**
2:   **for** j in $\{1,\dots,N\}$ **do**
3:     **if** New episode **then**
4:       Initialize state buffer $(s_1, s_2, \dots, s_{\Delta+1})$ and action buffer $(a_1, a_2 \dots, a_\Delta)$
5:     **end if**
6:     Sample $a_E \sim \pi_E(\cdot|s_{\Delta+1})$, set $a = a_E$
7:     **if** Random $u \sim U([0,1]) \geqslant \beta_i$ **then**
8:       Overwrite $a \sim \pi_I(\cdot|[s_1, a_1, \dots, a_\Delta])$
9:     **end if**
10:     Aggregate dataset:
$\mathcal{D} \leftarrow \mathcal{D} \cup ([s_1, a_1, \dots, a_{\Delta-1}], a_E)$
11:     Apply a in $\mathcal{E}$ and get new state s
12:     Update buffers:
$(s_1, \dots, s_{\Delta+1}) \leftarrow (s_2, \dots, s_{\Delta+1}, s)$
$(a_1, \dots, a_\Delta) \leftarrow (a_2, \dots, a_\Delta, a)$
13:   **end for**
14:   Train π_I on $\mathcal{D}$
15: **end for**

5.3. Imitating an Undelayed Policy

We follow the learning scheme of DAGGER with the slight difference that, if the expert is queried, then the current state is fed to π_E while if the imitator policy is queried, an augmented state is built from the past samples, considering the state Δ-steps before the current state and the sequence of actions taken since then. This implies that a buffer of the latest states and actions has to be built. We present our approach, which we call Delayed Imitation with DAGGER (DIDA), in Algorithm 1. In practice there is no need to store each augmented state in the dataset $\mathcal{D}$ but only trajectories of state-action pairs from the underlying MDP from which the augmented states can be rebuilt during training. Otherwise, each action would be repeated Δ times in $\mathcal{D}$, since an action is contained in Δ consecutive augmented states.

When sampling from the undelayed environment is prohibited, we apply a slight modification to DIDA as follows. Whenever the undelayed policy should be queried, one could instead sample from it in a memoryless fashion. That means taking the state contained in the augmented state as input to π_E, thus substituting Line 6 of Algorithm 1 for $a_E \sim \pi_E(\cdot|s_1)$. The action stored in the buffer must remain the undelayed expert one's on the current unobserved state however. This implies that the agent must wait Δ steps until the current state is observed before updating the buffer for the current augmented state. Thus delaying the execution of Line 10 of Algorithm 1. We call this variation memoryless-DIDA (M-DIDA).

What will be the policy learned by DIDA in practice? Given an augmented state x, DIDA learns to replicate the action taken by the expert on the current state s, unknown to the agent. However, the same augmented state can lead to different current states, which is summarized in the belief $b(s|x)$. Therefore, DIDA learns the following policy

$$\widetilde{\pi}_b(a|x) = \int_{\mathcal{S}} b(s|x)\pi_E(a|s)ds. \tag{3}$$

This policy is similar to the policies from model-based approaches, and, for this reason, may yield sub-optimal policies in some MDPs (Liotet et al., 2021, Proposition VI.1.). In practice, the class of functions of the imitated policy π_I and the loss chosen for training in step 15 of Algorithm 1 may slightly modify the policy learned by DIDA. For instance, a deterministic π_I would naturally forbid to learn the distribution given in Equation (3). This is discussed in Appendix E.1.

5.4. Extension to Non-Integer Delays

We now suppose that the delay is non-integer, yet still constant. For simplicity, we assume $\Delta \in (0, 1)$ but the general case follows from similar considerations. We consider a Δ-delay in the action execution (the case of state observation is similar).

DMDP with non-integer delays can be viewed as the result of two interleaved MDPs, $\mathcal{M}$ with time indexes $t, t+1, \dots$ and $\mathcal{M}_\Delta$ with indexes $t+\Delta, t+\Delta+1, \dots$. Those two discrete MDPs stem from a single continuous process, of which we observe only some fixed time steps, similarly to (Sutton et al., 1999). They share the same transition and reward functions. A delayed agent would see states from $\mathcal{M}$ while executing actions on states of $\mathcal{M}_\Delta$. In practice, the agent taking action a_t seeing state s_t would collect a reward $r(s_{t+\Delta}, a_t)$. The transition probabilities are also affected. We define $b_\Delta(s_{t+\Delta}|s_t, a)$ the probability of reaching $s_{t+\Delta}$ from s_t when action a is applied during time Δ and $b_{1-\Delta}(s_t|s_{t-1+\Delta}, a)$ the probability of reaching s_t from $s_{t-1+\Delta}$ when action a is applied during time $1-\Delta$. To make the definition consistent with the regular MDP, those probabilities must satisfy that, for all $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$

$$p(s'|s, a) = \int_{\mathcal{S}} b_{1-\Delta}(s'|z, a) b_\Delta(z|s, a)\, dz. \tag{4}$$

Clearly, even for $\Delta \in (0, 1)$, an augmented state $x_t = (s_t, a_{t-1}) \in \mathcal{S} \times \mathcal{A} =: \mathcal{X}$ is needed in order not to lose information about the state $s_{t+\Delta} \sim b_\Delta(\cdot|s_t, a_{t-1})$. The DMDP with augmented state can again be cast into an MDP as in Bertsekas (1987); Altman & Nain (1992), where the new transition is defined for $x_t = (s_t, a_{t-1}), x_{t+1} = (s_{t+1}, a_t) \in \mathcal{X}$, as,

$$\tilde{p}(x_{t+1}|x_t,a) := \delta_a(a_t) \int_{\mathcal{S}} b_{1-\Delta}(s_{t+1}|z,a) b_\Delta(z|s_t,a_{t-1})\, dz,$$

where the term $\delta_a(a_t)$ ensures that the new extended state contains the action that has been applied on x_t. For delays greater than 1, one needs to consider the augmented state in the space $\mathcal{S} \times \mathcal{A}^{\lceil \Delta \rceil}$ and the previous considerations hold by first considering the integer part of the delay and then its remaining non-integer part. In this setting, we propose to use DIDA by learning an undelayed policy in $\mathcal{M}_\Delta$ and imitating it by building an augmented state from the states in $\mathcal{M}$. For clarity we provide the extension to this setting in Appendix D.

To extend the theoretical analysis to the non-integer setting, we need to extend the concept of L_T-TLC. For $\Delta \in (0, 1)$, $\forall s, a \in \mathcal{S} \times \mathcal{A}$, a DMDP is L_T-TLC if

$$\mathcal{W}_1(b_\Delta(\cdot|s, a), \delta_s) \leqslant \Delta L_T.$$

6. Theoretical Analysis of the Approach

We will now provide a theoretical analysis of the approach proposed above. The role of this analysis is twofold. First, it gives insights into which expert undelayed policy is best suited to be imitated in a DMDP. Secondly, it provides general results on the value functions bounds between DMDPs and MDPs, when the latter has guarantees of smoothness, setting aside pathological counterexamples such as in (Liotet et al., 2021, Proposition VI.1.) while remaining realistic. Comparing the performances of delayed and undelayed policies via their state value functions is not trivial since they live on two different spaces ($\mathcal{S}$ and $\mathcal{X} = \mathcal{S} \times \mathcal{A}^\Delta$).

Different approaches were proposed to address this issue. In (Walsh et al., 2009, Theorem 3) the idea stems from the fact that a delayed policy in a deterministic MDP has same optimal performance as the undelayed one. Therefore, when a finite MDP is not deterministic but *mildly stochastic*, that is there exists ϵ such that $\forall (s, a) \in \mathcal{S} \times \mathcal{A}, \exists s', p(s'|s, a) \geqslant 1 - \epsilon$, then, one can use a deterministic approximation to the MDP and use it to learn a delayed policy. The distance between the value function of this delayed policy, $\widetilde{V}^\pi$ and the one of the real MDP, V^π as $\|\widetilde{V}^\pi - V^\pi\|_\infty \leqslant \frac{\gamma \epsilon R_{\max}}{(1-\gamma)^2}$, where $R_{\max}$ is the maximum absolute reward. The assumptions by Walsh et al. (2009) are quite strong and the bound grows quadratically with the effective time horizon. Another approach is proposed by Agarwal & Aggarwal (2021, Theorem 1), who compare the delayed value function $V^{\widetilde{\pi}}$ to $\mathbb{E}_{s \sim b(\cdot|x)}[V^\pi(s)]$, which corresponds to the expected value function of the undelayed policy averaged on the current unknown state given some augmented state. However, the authors make no assumptions about smoothness.

Instead, we base our analysis on smoothness assumptions to provide our main result on the difference in performance between delayed and undelayed policies in Theorem 6.1. To obtain this result, we must first derive a delayed version of the performance difference lemma (Kakade & Langford, 2002). Its proof, as for all other results in this section, is given in Appendix B and applies to any couple of delayed and undelayed policies. Note that these results hold for either integer or non-integer constant delays. For simplicity, we state the results with belief b but b_Δ is intended if the delay is non-integer.

Lemma 6.1. *[Delayed Performance Difference Lemma] Consider an undelayed policy π_E and a Δ-delayed policy $\widetilde{\pi}$, with $\Delta \in \mathbb{R}_{\geqslant 0}$. Then, for any $x \in \mathcal{X}$,*

$$\mathop{\mathbb{E}}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - V^{\widetilde{\pi}}(x) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{x'\sim d_x^{\widetilde{\pi}}}\left[\mathop{\mathbb{E}}_{s\sim b(\cdot|x')}[V^{\pi_E}(s)] - \mathop{\mathbb{E}}_{\substack{s\sim b(\cdot|x')\\ a\sim\widetilde{\pi}(\cdot|x')}}[Q^{\pi_E}(s,a)]\right].$$

We can then leverage the previous result to obtain a valuable result for DMDPs, which holds for delayed policies of the form of Equation (3).

Theorem 6.1. *Consider an (L_P, L_r)-LC MDP and a L_π-LC undelayed policy π_E, such that Q^{π_E} is L_Q -L.C.*[2]*. Let $\widetilde{\pi}_b$ be the Δ-delayed policy defined as in Equation (3), with $\Delta \in \mathbb{R}_{\geqslant 0}$. Then, for any $x \in \mathcal{X}$,*

$$\mathop{\mathbb{E}}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - V^{\widetilde{\pi}_b}(x) \leqslant \frac{L_Q L_\pi}{1-\gamma}\sigma_b^x,$$

where $\sigma_b^x = \mathop{\mathbb{E}}_{\substack{x'\sim d_x^{\widetilde{\pi}_b}\\ s,s'\sim b(\cdot|x')}}[\mathrm{d}_\mathcal{S}(s,s')]$.

Proof sketch. The first step of the proof involves applying the delayed performance difference lemma to the l.h.s.. This brings an upper bound depending on $Q^{\pi_E}(s,a_1) - Q^{\pi_E}(s,a_2)$ where $a_1 \sim \pi_E(\cdot|s)$ and $a_2 \sim \widetilde{\pi}_b(\cdot|x')$. It quantifies the effect of selecting an action under $\widetilde{\pi}_b$ and then following π_E instead of following π_E directly. Using the Lipschitzness of Q^{π_E}, this term can be bounded by a quantity in $\mathcal{W}_1(\pi_E(\cdot|s)\|\widetilde{\pi}_b(\cdot|x')$. From the definition of $\widetilde{\pi}_b(\cdot|x')$ which corresponds to applying π_E on the state sampled from the belief $b(\cdot|x')$, an upper bound of the previous Wasserstein distance can be obtained, involving the integration of $b(s'|x')\mathcal{W}_1(\pi_E(\cdot|s)\|\pi_E(\cdot|s'))$ for s'. This is precisely where the loss in performance of DIDA with respect to the expert might come from. Instead of knowing the current state s, it is constrained to select an action which performs well on the possible current states weighted by their belief given the augmented state. Using the Lipschitzness of π_E completes the proof. □

[2] In fact, only Lipschizness in the second argument is necessary (see proof).

However, this result seems difficult to grasp because of its dependence on the term σ_b^x. We suggest two ways to further bound this term. The first involves the time-Lipschitzness assumption of the MDP and yields Corollary 6.1.

Corollary 6.1. *Under the assumptions of Theorem 6.1, adding that the MDP is L_T-TLC, then, for any $x \in \mathcal{X}$,*

$$\mathop{\mathbb{E}}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - V^{\widetilde{\pi}_b}(x) \leqslant \frac{2\Delta L_T L_Q L_\pi}{1-\gamma}.$$

This first result clearly highlights the linear dependence on the delay Δ. However, the bound does not vanish (as expected) when the MDP is deterministic. This property is verified by a second result. This second result assumes a state space in $\mathbb{R}^n$ equipped with the Euclidean norm and yields Corollary 6.2.

Corollary 6.2. *Under the assumptions of Theorem 6.1 adding that $\mathcal{S} \subset \mathbb{R}^n$ is equipped with the Euclidean norm. Then, for any $x \in \mathcal{X}$,*

$$\mathop{\mathbb{E}}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - V^{\widetilde{\pi}_b}(x) \leqslant \frac{2L_Q L_\pi}{1-\gamma} \mathop{\mathbb{E}}_{x'\sim d_x^{\widetilde{\pi}_b}(\cdot)}\left[\sqrt{\mathop{\mathbb{V}\mathrm{ar}}_{s\sim b(\cdot|x')}(s|x')}\right].$$

Interestingly, we show that this second corollary matches a theoretical lower bound when the expert policy is optimal. We provide this lower bound in Theorem 6.2, which shows that a too irregular expert policy (with high Lipschitz constant) provides weaker guarantees.

Theorem 6.2. *For every $L_\pi > 0$, $L_Q > 0$, there exists an MDP such that the optimal policy is L_π-LC, its state action value function is L_Q-LC in the second argument, but for any Δ-delayed policy $\widetilde{\pi}$, with $\Delta \in \mathbb{R}_{\geqslant 0}$, and any $x \in \mathcal{X}$*

$$\mathop{\mathbb{E}}_{s\sim b(\cdot|x)}[V^*(s)] - V^{\widetilde{\pi}}(x) \geqslant \frac{\sqrt{2}}{\sqrt{\pi}}\frac{L_Q L_\pi}{1-\gamma} \mathop{\mathbb{E}}_{x'\sim d_x^{\widetilde{\pi}}(\cdot)}\left[\sqrt{\mathop{\mathbb{V}\mathrm{ar}}_{s\sim b(\cdot|x')}(s|x')}\right],$$

where V^* is the value function of the optimal undelayed policy.

Proof sketch. We build an MDP where such a bounds holds. We consider a state and action space in $\mathbb{R}$. Performing action

a, we transition from s to $s + a + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. The reward function is proportional to $-|s + a|$, so that the objective is to always move towards the origin In the undelayed case, knowing the current position, it is possible to achieve the optimal 0 return. Instead, when there is an observation delay, we know the state up to an error of distribution $\mathcal{N}(0, \Delta\sigma^2)$, which makes it impossible to have an expected reward smaller than $\mathbb{E}[|\mathcal{N}(0, \Delta\sigma^2)|]$. $\square$

We provide an alternative way to derive bounds in performance in Appendix C, which provide slightly different results as discussed in Appendix C.1.

We have bounded the performance of our perfectly imitated delayed policy $\widetilde{\pi}_b$ with respect to the undelayed expert π_E. However, two additional sources of performance loss have to be taken into account. First, the expert π_E may be sub-optimal in the undelayed MDP. Second, the imitated policy π_I may not learn exactly $\widetilde{\pi}_b$.

These theoretical results highlight two important trade-offs in practice. If the expert policy is smoother than the optimal undelayed policy, then we might miss out on some opportunities, but the delayed policy is likely to be more similar to the expert one, according to Theorem 6.1. The second trade-off concerns noisier policies. For them, the imitation step is likely to be easier, as it provides examples of how to recover from bad decisions (Laskey et al., 2017). Therefore, our imitated policy π_I is likely to be more similar to $\widetilde{\pi}_b$. However, this may decrease the performance of the expert compared to the optimal undelayed policy.

7. Experiments

7.1. Setting

As we have seen in the theoretical analysis, a smoother expert is beneficial for the performance bound of the imitated delayed policy. Therefore, in the following experiments, we consider expert policies learned with SAC. As reported in an extensive study about smooth policies (Mysore et al., 2021), the entropy-maximization framework of SAC is able to learn a smooth policy even without additional forms of regularization. To avoid ever-growing memory by storing all samples in the buffer as done in Algorithm 1, we use a maximum buffer size of 10 iterations for DIDA and overwrite the oldest iteration samples when this buffer is full. As suggested by Ross et al. (2011), we use $\beta_1 = 1, \beta_{i\geqslant 2} = 0$ as mixture weights for the sampling policy. The policy for DIDA is a simple feed-forward neural network. More details and all hyper-parameters are reported in Appendix E.2.

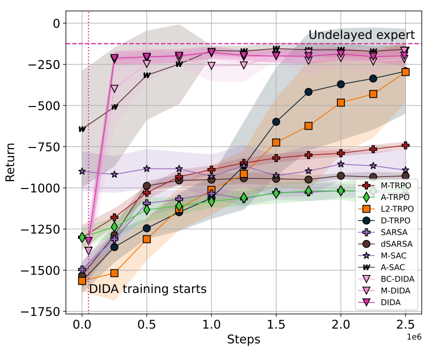
We will test DIDA and M-DIDA, along with some baselines from the state of the art, on the following environments.

Pendulum The task of the agent is to rotate a pendulum upward. It is a classic experiment in delayed RL as delays are highly impacting performance due to unstable equilibrium in the upward position. We use the version from the library `gym` (Brockman et al., 2016).

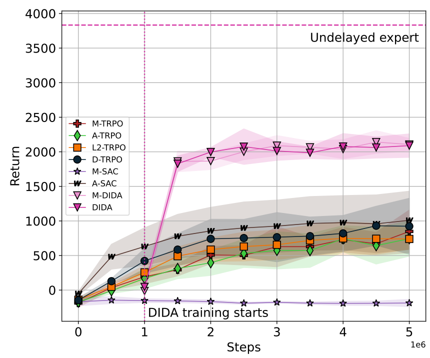
Mujoco Continuous robotic locomotion control tasks realized with an advanced physics simulator from the library `mujoco` (Todorov et al., 2012). Here the main difficulty lies in the complex dynamics and in the large state and action spaces. Among the possible environments, we consider the ones that are most affected by delays, namely Walker2d, HalfCheetah, Reacher, and Swimmer.

Trading The agent trades the EUR-USD (€/$) currency pair every 10 minutes and can either *buy*, *sell* or stay *flat* against a fixed amount of USD, following the framework of Bisi et al. (2020) and Riva et al. (2021). We assume trading is without fees, but we do take the spread into account. To this setting, we add a delay of 50 seconds to the action execution, thus placing the experiment in the non-integer delay framework. In this environment, we leverage the knowledge of an expert which is a policy trained on years 2016-2017 by Fitted Q-Iteration (FQI Ernst et al., 2005) with XGBoost (Chen & Guestrin, 2016) as a regressor for the Q function. Only for this task, we use Extra Trees (Geurts et al., 2006) as policy for DIDA. Due to the highly stochastic nature of the environment, we consider 4 experts trained with the same configuration but a different seed. For each of these expert, 5 seeds have been used for DIDA, resulting in 20 runs of DIDA on which mean and standard deviation of the results are computed. Finally, the expert has been selected by performing validation of its hyper-parameters on 2018, it is therefore possible to do validation on the delayed dataset of 2018 in order to select an expert which, albeit trained on undelayed data, performs well on delayed data. We refer to this expert as delayed expert. We compare the performance of the expert, undelayed expert and DIDA on year 2019.

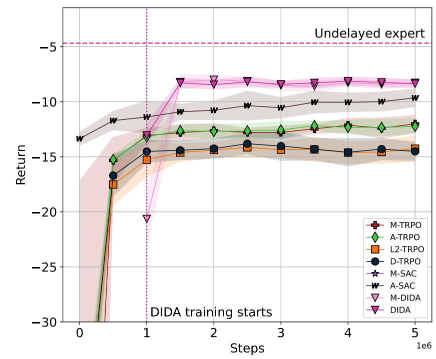
The baselines for comparison with our algorithm include a memoryless and an augmented version of TRPO (M-TRPO and A-TRPO respectively), D-TRPO and L2-TRPO (Liotet et al., 2021), SARSA (Sutton & Barto, 2018) and dSARSA (Schuitema et al., 2010). The last two algorithms involve state discretization and are thus tested on pendulum only. We consider also augmented SAC (A-SAC), considered also by Bouteiller et al. (2020), and memoryless SAC (M-SAC). Although SAC can be trained at every step as we do for Pendulum, we restrict training to every 50 steps on Mujoco to speed up the procedure and reduce memory usage. We have also considered adding DCAC (Bouteiller et al., 2020) but for computational reasons, we have decided not to include it. Early experimental results showed that its running time was more than 50 times the one of DIDA. For a fair comparison to the baselines, which learn a policy



(a) Pendulum.

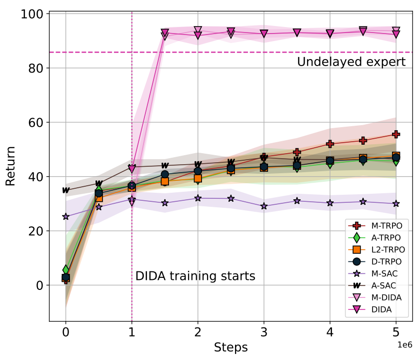


(b) HalfCheetah.

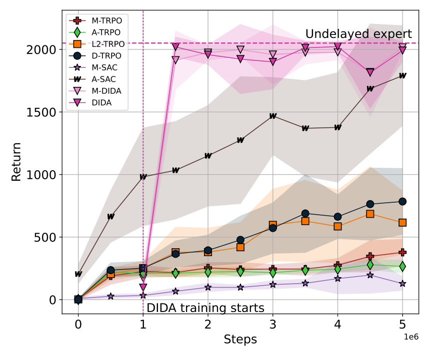


(c) Reacher.

Figure 1: For a 5-steps delay, mean return and one standard deviation (shaded) as a function of the number of steps sampled from the environment (10 seeds).



(a) Swimmer.



(b) Walker2d.

Figure 2: For a 5-steps delay, mean return and one standard deviation (shaded) as a function of the number of steps sampled from the environment (10 seeds).

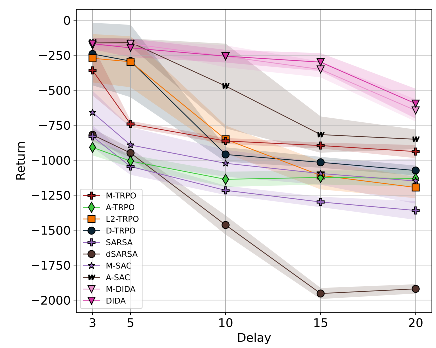


Figure 3: Mean return and its standard deviation (shaded) as a function delay (10 seeds).

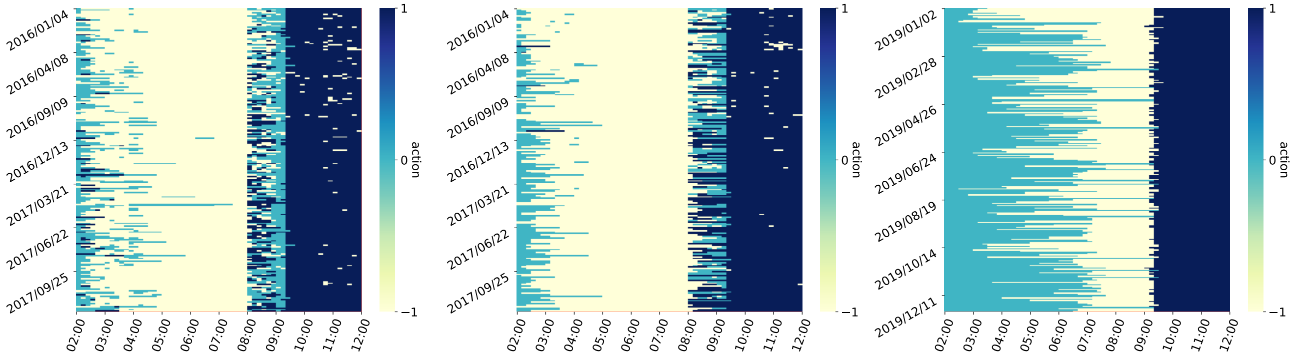
from scratch, we include the training steps of the expert in the step count of DIDA, as indicated by the vertical dotted line in the figures.

7.2. Results

As we can see from the results on pendulum and mujoco, Figures 1a to 1c, 2a and 2b, DIDA is able to converge much faster than the baselines in any environment, with the exception of A-SAC on the pendulum environment. In less than half a million steps on mujoco, and 250.000 steps on pendulum, DIDA almost reaches its final performance. In HalfCheetah and Reacher, we note that, although the best delayed algorithm, DIDA performs much worse than the expert. Surprisingly, in Swimmer, DIDA performs slightly better than the undelayed expert. All these phenomenons might actually come from a single cause. In our implementation of a delayed environment, we start from an undelayed environment to which is added a wrapper simulating the effect of the delay. The initialization of this process involves sampling a sequence of Δ random actions that are applied in the undelayed environment in order to sample the first augmented state. Depending on the environment, this random sequence of actions could cause the agent to start in disadvantageous or advantageous states. For instance[3], in HalfCheetah, the random action have put the agent head-down when the the latter is first allowed to control the environment. It must thus first get back on its feet before starting to move. On the contrary, in a simpler environment like Swimmer, the initial random action queue might give some initial speed to the agent, yielding higher rewards at the beginning than its undelayed counterpart.

We provide another experiment on Pendulum where we study the robustness of DIDA, as compared to baselines, against an increase in the delay for fixed hyper-parameters. We report the final mean return per episode for different values of the delay in Figure 3. Clearly, from all the baselines studied, DIDA is the most robust to the increase in the delay.

[3] We provide a visual interpretation of these phenomenons here.



(a) Expert 2016-2017. (b) DIDA 2016-2017. (c) DIDA 2019.

Figure 4: Comparison of the actions selected by the expert and the 10^{th} iteration of DIDA on the Trading environment, during train (2016-2017) and test (2019). Results show the action (colour) selected for each day (row) at each time of the day (column).

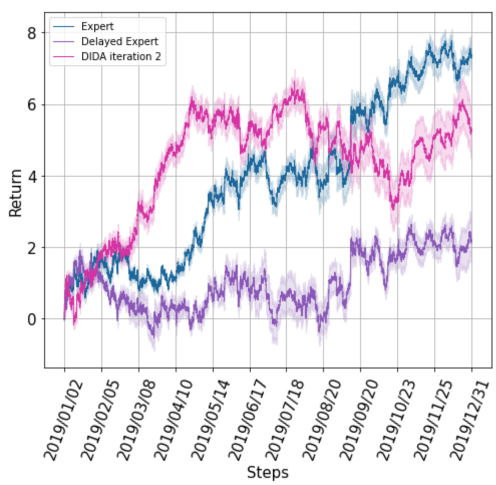


Figure 5: Evolution of the return of DIDA for the trading of the EUR-USD pair in 2019. Performance is computed in percentage w.r.t. the invested amount.

For the Trading, being a batch-RL task since the training dataset of 2016-2017 is a fixed set of historical exchange rates, it is necessary to select the hyperparameters of DIDA by validation on 2018 to avoid overfitting. The second iteration of DIDA has been selected by validation. The results on the test set of 2019 are shown in Figure 5. The results show the ability of DIDA to adapt to non-integer delays and maintain a positive return, which is not a simple when taking the spread into account in trading the EUR-USD. It clearly outperforms the delayed expert on this task. Surprisingly, the delayed policy is able to outperform the expert on the first period of the test. This could be explained by the fact that the expert undelayed policy may have overfitted the training set while the imitation learning of an undelayed policy acted as a regularization. We provide an analysis on the policy learned by DIDA with respect to the expert in Figure 4. It illustrates the specific overfitting problem of the Trading task caused by the batch-RL scenario by showing the policy of DIDA at the 10^{th} and last iteration compared to the expert policy. Interestingly, the allocations of all the presented policies follow peculiar daily patterns. Figure 4a-b, where allocations on the training set are presented, shows that DIDA's policy replicates quite faithfully expert policy temporal patterns. On the other hand in Figure 4c, where test set is shown, it can be noticed that DIDA's allocations start to shift away from the expert pattern.

Finally, we provide in Appendix E.3 additional experiments on different stochastic versions of pendulum.

8. Conclusion

In this paper, we explored the possibility of splitting delayed reinforcement learning into easier tasks, traditional undelayed reinforcement learning on the one hand, and imitation learning on the other one. We provided a theoretical analysis demonstrating bounds on the performance of a delayed policy compared to undelayed experts, both for integer or non-integer constant delays. These bounds apply in our particular setting but are also of interest in general for delayed policies. This guided us in the creation of our algorithm, DIDA, which learns a delayed policy by imitating an undelayed expert using DAGGER. We have empirically shown that this idea, although rather simple, provides excellent results in practice, achieving high performance with remarkable sample efficiency and light computations. We believe that our work paves the way for many possible generalizations, which include stochastic delays and particular situations in which an undelayed simulator is not available, but where an undelayed dataset can be artificially created from delayed trajectories in order to train an expert offline.

References

Agarwal, M. and Aggarwal, V. Blind decision making: Reinforcement learning with delayed observations. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, pp. 2–6, 2021.

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

Bertsekas, D. P. *Dynamic Programming: Determinist. and Stochast. Models*. Prentice-Hall, 1987.

Bisi, L., Liotet, P., Sabbioni, L., Reho, G., Montali, N., Restelli, M., and Corno, C. Foreign exchange trading: a risk-averse batch reinforcement learning approach. In *Proceedings of the First ACM International Conference on AI in Finance*, pp. 1–8, 2020.

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

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

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

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Derman, E., Dalal, G., and Mannor, S. Acting in delayed environments with non-stationary markov policies. *arXiv preprint arXiv:2101.11992*, 2021.

Droździel, P., Tarkowski, S., Rybicka, I., and Wrona, R. Drivers' reaction time research in the conditions in the real traffic. *Open Engineering*, 10(1):35–47, 2020.

Dugard, L. and Verriest, E. I. *Stability and control of time-delay systems*, volume 228. Springer, 1998.

Dulac-Arnold, G., Mankowitz, D., and Hester, T. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Ernst, D., Geurts, P., and Wehenkel, L. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(Apr):503–556, 2005.

Firoiu, V., Ju, T., and Tenenbaum, J. At human speed: Deep reinforcement learning with action delay. *arXiv preprint arXiv:1810.07286*, 2018.

Geurts, P., Ernst, D., and Wehenkel, L. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

Gu, K. and Niculescu, S.-I. Survey on recent results in the stability and control of time-delay systems. *J. Dyn. Sys., Meas., Control*, 125(2):158–165, 2003.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.

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.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Laskey, M., Lee, J., Fox, R., Dragan, A., and Goldberg, K. Dart: Noise injection for robust imitation learning. In *Conference on robot learning*, pp. 143–156. PMLR, 2017.

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.

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.

Metelli, A. M., Mazzolini, F., Bisi, L., Sabbioni, L., and Restelli, M. Control frequency adaptation via action persistence in batch reinforcement learning. In *International Conference on Machine Learning*, pp. 6862–6873. PMLR, 2020.

Mysore, S., Mabsout, B., Mancuso, R., and Saenko, K. Regularizing action policies for smooth control with reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1810–1816. IEEE, 2021.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., and Peters, J. An algorithmic perspective on imitation learning. *arXiv preprint arXiv:1811.06711*, 2018.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. *arXiv preprint arXiv:1705.07057*, 2017.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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

Riva, A., Bisi, L., Liotet, P., Sabbioni, L., Vittori, E., Pinciroli, M., Trapletti, M., and Restelli, M. Learning fx trading strategies with fqi and persistent actions. 2021.

Ross, S. and Bagnell, D. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 661–668. JMLR Workshop and Conference Proceedings, 2010.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Schuitema, E., Buşoniu, L., Babuška, R., and Jonker, P. Control delay in reinforcement learning for real-time dynamic systems: a memoryless approach. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3226–3231. IEEE, 2010.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015a.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

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

Sutton, R. S., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211, 1999.

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.

Villani, C. *Optimal transport: old and new*, volume 338. Springer, 2009.

Walsh, T. J. et al. Learning and planning in environments with delayed feedback. *Autonomous Agents and Multi-Agent Systems*, 18(1):83, 2009.

Wilcox, J. W. The effect of transaction costs and delay on performance drag. *Financial Analysts Journal*, 49(2): 45–54, 1993.

Xu, T., Li, Z., and Yu, Y. Error bounds of imitating policies and environments. *Advances in Neural Information Processing Systems*, 33, 2020.

A. General Results

A.1. Bounds involving the Wasserstein distance

Proposition A.1. *Let X,Y be two random variables on $\mathbb{R}$ with distribution π_0, π_1 respectively. Then,*

$$|\mathbb{E}[X] - \mathbb{E}[Y]| \leqslant \mathcal{W}_1(\pi_0 \| \pi_1).$$

Proof. One has

$$\mathbb{E}[X] - \mathbb{E}[Y] = \int_{\mathbb{R}} x(\pi_0(x) - \pi_1(x))dx \leqslant \mathcal{W}_1(\pi_0 \| \pi_1),$$

since $x \mapsto x$ is 1-LC. The same holds for $\mathbb{E}[Y] - \mathbb{E}[X]$, since the Wasserstein distance is symmetric. □

The next result asserts that if one applies a L-LC function to two random variables, one gets two random variables with distribution whose Wasserstein distance is bounded by the original Wasserstein distance multiplied by a factor L.

Proposition A.2. *Let $g : \Omega \to \mathbb{R}$ be an L-LC function and π_0, π_1 two probability measures over the metric space Ω. Note g_π the distribution of the random variable $g(X)$ where X is distributed according to π. Then,*

$$\mathcal{W}_1(g_{\pi_0} \| g_{\pi_1}) \leqslant L\mathcal{W}_1(\pi_0 \| \pi_1).$$

Proof. By definition of Wasserstein distance,

$$\begin{aligned}\mathcal{W}_1(g_{\pi_0} \| g_{\pi_1}) &= \sup_{\|f\|_L \leqslant 1} \left| \int_{\mathbb{R}} f(x)(g_{\pi_0}(x) - g_{\pi_1}(x))dx \right| \\ &= \sup_{\|f\|_L \leqslant 1} \left| \int_{\mathbb{R}} f(x) g_{\pi_0}(x)dx - \int_{\mathbb{R}} f(x) g_{\pi_1}(x)dx \right|.\end{aligned}$$

We can then use the definitions of g_{π_0} and g_{π_1} to rewrite the previous formula in terms of expected values

$$\begin{aligned}\mathcal{W}_1(g_{\pi_0} \| g_{\pi_1}) &= \sup_{\|f\|_L \leqslant 1} \left| \int_{\mathbb{R}} f(g(x))\pi_0(x)dx - \int_{\mathbb{R}} f(g(x))\pi_1(x)dx \right| \\ &= \sup_{\|f\|_L \leqslant 1} \left| \int_{\mathbb{R}} f(g(x))(\pi_0(x) - \pi_1(x))dx \right|.\end{aligned}$$

Since g is L_g-LC by assumption, the composition $f(g(x))$ is still L_g-LC, so

$$\mathcal{W}_1(g_{\pi_0} \| g_{\pi_1}) \leqslant L_g \mathcal{W}_1(\pi_0 \| \pi_1).$$

□

Proposition A.3. *Consider an MDP with π a policy such that its state-action value function is Lipschitz with constant L_Q in the second argument, i.e. it satisfies, for all $s \in \mathcal{S}$ and $a, a' \in \mathcal{A}$*

$$\left| Q^\pi(s,a) - Q^\pi(s,a') \right| \leqslant L_Q \, \mathrm{d}_{\mathcal{A}}(a,a'),$$

then, for every couple of probability distributions η, ν over $\mathcal{A}$, one has that

$$\left| \mathop{\mathbb{E}}_{\substack{X\sim\eta \\ Y\sim\nu}} [Q^\pi(s,X) - Q^\pi(s,Y)] \right| \leqslant L_Q \mathcal{W}_1(\eta(\cdot) \| \nu(\cdot)).$$

Proof. We note g_η and g_ν the respective distributions of $Q^\pi(s,X)$ and $Q^\pi(s,Y)$. First of all, we can apply Proposition A.1 to say that

$$\left| \mathop{\mathbb{E}}_{\substack{X\sim\eta \\ Y\sim\nu}} [Q^\pi(s,X) - Q^\pi(s,Y)] \right| \leqslant \mathcal{W}_1(g_\eta \| g_\nu).$$

For a fixed $s \in \mathcal{S}$, the two random variables $Q^\pi(s, X)$ and $Q^\pi(s, Y)$ can be seen as the application of the L_Q-Lipschitz function $Q^\pi(s, \cdot) : \mathcal{A} \to \mathbb{R}$ to X and Y, respectively. This satisfies the assumptions of Proposition A.2, therefore

$$\mathcal{W}_1(Q^\pi(s, \eta(\cdot)) \| Q^\pi(s, \nu(\cdot))) \leqslant L_Q \mathcal{W}_1(\eta(\cdot) \| \nu(\cdot)).$$

□

Proposition A.4. *Consider an L_P transition function p and an L_π policy π in some MDP $\mathcal{M}$. Then, for any $f : S \to \mathbb{R}$ which is 1-LC, we have that the function $g : S \to \mathbb{R}$ given by*

$$g(s) := \int_{\mathcal{S}} f(s') \int_A p(s'|s, a)\pi(a|s) \, da \, ds'$$

is Lipschitz with constant $L_p(1 + L_\pi)$

Proof. Let $s, z \in S$, one has

$$\begin{aligned}
|g(s) - g(z)| &= \left| \int_{\mathcal{S}} f(s') \int_A p(s'|s, a)\pi(a|s) - p(s'|z, a)\pi(a|z) \, da \, ds' \right| \\
&\leqslant \left| \int_{\mathcal{S}} f(s') \int_A p(s'|s, a) \left(\pi(a|s) - \pi(a|z)\right) \, da \, ds' \right| \\
&\quad + \left| \int_{\mathcal{S}} f(s') \int_A \big(p(s'|s, a) - p(s'|z, a)\big) \pi(a|z) \, da \, ds' \right| \qquad (5) \\
&\leqslant \underbrace{\left| \int_A (\pi(a|s) - \pi(a|z)) \int_{\mathcal{S}} f(s') p(s'|s, a) \, ds' \, da \right|}_{A} \\
&\quad + \underbrace{\left| \int_A \pi(a|z) \int_{\mathcal{S}} f(s') \big(p(s'|s, a) - p(s'|z, a)\big) \, ds' \, da \right|}_{B}, \qquad (6)
\end{aligned}$$

where we add and remove the quantity $p(s'|s, a)\pi(a|z)$ in Equation (5) and use Fubini's theorem in Equation (6).

By Lipschitzness of p, we have that $a \mapsto \int_{\mathcal{S}} f(s')p(s'|s, a) \, ds'$ is L_p-LC. Thus,

$$\begin{aligned}
A &= L_P \left| \int_A (\pi(a|s) - \pi(a|z)) \frac{\int_{\mathcal{S}} f(s')p(s'|s, a) \, ds'}{L_P} \, da \right| \\
&\leqslant L_\pi L_P \, \mathrm{d}_{\mathcal{S}}(s, z)
\end{aligned}$$

For the second term, again, by Lipschitzness of p, we have

$$\begin{aligned}
B &\leqslant \left| \int_A L_P d_{\mathcal{S}}(s, z) \pi(a|z) \, da \right| \\
&= L_P d_{\mathcal{S}}(s, z).
\end{aligned}$$

Overall,

$$|g(s) - g(z)| \leqslant L_p(L_\pi + 1) \, \mathrm{d}_{\mathcal{S}}(s, z).$$

□

A.2. Bounding σ_b^ρ

We provide two bounds for $\sigma_b^\rho = \mathbb{E}_{\substack{x' \sim d_\rho^{\tilde{\pi}} \\ s, s' \sim b(\cdot|x')}} [\mathrm{d}_{\mathcal{S}}(s, s')]$ with ρ a distribution on $\mathcal{S}$. The first uses the assumption that the state space in is $\mathbb{R}^n$ and is equipped with the Euclidean norm while the second assumes that the MDP is TLC.

Lemma A.1 (Euclidean bound)**.** *Consider an MDP such that $\mathcal{S} \subset \mathbb{R}^n$ is equipped with the Euclidean norm. Then one has*

$$\sigma_b^\rho \leqslant \mathop{\mathbb{E}}_{x' \sim d_\rho^{\tilde{\pi}}(\cdot)} \left[\sqrt{\mathop{\mathbb{V}\mathrm{ar}}_{s \sim b(\cdot|x')}(s|x')} \right].$$

Proof. We derive the following results which intermediate steps are detailed after.

$$\begin{aligned}
\sigma_b^\rho &= \mathop{\mathbb{E}}_{\substack{x' \sim d_\rho^{\tilde{\pi}} \\ s,s' \sim b(\cdot|x')}} \left[\mathrm{d}_{\mathcal{S}}(s, s') \right] \\
&= \mathop{\mathbb{E}}_{\substack{x' \sim d_\rho^{\tilde{\pi}} \\ s,s' \sim b(\cdot|x')}} \left[\sqrt{(s'-s)^2} \right] && (7) \\
&= \mathop{\mathbb{E}}_{x' \sim d_\rho^{\tilde{\pi}}} \sqrt{\mathop{\mathbb{E}}_{s,s' \sim b(\cdot|x')} \left[(s'-s)^2 \right]}. && (8)
\end{aligned}$$

Equation (7) follows from the definition of the Euclidean norm and Equation (8) is obtained by applying Jensen's inequality. To conclude, since s' and s are i.i.d., one has that $\mathbb{E}_{s,s' \sim b(\cdot|x')} \left[(s'-s)^2 \right] = 2\mathbb{V}ar_{s \sim b(\cdot|x')}[s]$. ☐

The following proposition is involved in the proof of the bound of σ_b^ρ when the MDP is TLC.

Proposition A.5. *Consider an L_T-TLC MDP. Consider any augmented state $x = (s_1, a_1, \ldots, a_\Delta) \in \mathcal{S} \times \mathcal{A}^\Delta$ for a given $\Delta \in \mathbb{R}_{\geqslant 0}$. Then*

$$\mathcal{W}_1 \left(b(\cdot|x) \| \delta_{s_1} \right) \leqslant \Delta L_T$$

Proof. First, consider $\Delta \in \mathbb{N}$ We proceed by induction. The case case $\Delta = 0$ is true since the current state is known exactly without delay. The case $\Delta = 1$ is true by the assumption of L_T-TLC. Assume that the statement is true for $\Delta \in \mathbb{N}$, then

$$\begin{aligned}
\mathcal{W}_1 \left(b(\cdot|x) \| \delta_{s_1} \right) &= \sup_{\|f\|_L \leqslant 1} \left| \int_{\mathcal{S}} f(s') \left(b(s'|x) - \delta_{s_1}(s') \right) ds' \right| \\
&= \sup_{\|f\|_L \leqslant 1} \left| \int_{\mathcal{S}} p(s_2|s_1, a_1) \int_{\mathcal{S}} f(s') \left(b(s'|s_2, a_2, \cdots, a_\Delta) - \delta_{s_1}(s') \right) ds' \right| && (9) \\
&= \sup_{\|f\|_L \leqslant 1} \left| \int_{\mathcal{S}} p(s_2|s_1, a_1) \int_{\mathcal{S}} f(s') \left(b(s'|s_2, a_2, \cdots, a_\Delta) - \delta_{s_2}(s') \right) ds' \right. \\
&\quad \left. + \int_{\mathcal{S}} p(s_2|s_1, a_1) \int_{\mathcal{S}} f(s') \left(\delta_{s_2}(s) - \delta_{s_1}(s') \right) ds' \right| && (10) \\
&\leqslant \underbrace{\sup_{\|f\|_L \leqslant 1} \left| \int_{\mathcal{S}} p(s_2|s_1, a_1) \int_{\mathcal{S}} f(s') \left(b(s'|s_2, a_2, \cdots, a_\Delta) - \delta_{s_2}(s') \right) ds' \right|}_{A} \\
&\quad + \underbrace{\mathcal{W}_1 \left(P(|s_1, a_1) \| \delta_{s_1} \right)}_{B},
\end{aligned}$$

where (9) hols by conditioning on the second visited state s_2 and Equation (10) holds by adding and subtracting δ_{s_2}. The reader may have recognized that the statement at $\Delta - 1$ can be used to bound A while B can be bounded with the L_T-TLC assumption. Therefore

$$\mathcal{W}_1 \left(b(\cdot|x) \| \delta_{s_1} \right) \leqslant \Delta L_T,$$

and the statement holds for any $\Delta \in \mathbb{N}$.

The proof for a general delay $\Delta \in \mathbb{R}_{\geqslant 0}$ start by consider the fractional part of the delay in a similar way as described above. The bound will be the sum of a term like A where the statement for integer delays can be applied and a term $\{\Delta\} L_T$ where $\{\Delta\}$ is fractional part of Δ. ☐

Lemma A.2 (Time-Lipschitz bound)**.** *Consider an L_T-Lipschitz MDP with delay Δ. Then, one has*

$$\sigma_b^\rho \leqslant 2\Delta L_T$$

Proof. Call s_x the state contained in x. By triangular inequality, one has

$$\begin{aligned}
\sigma_b^\rho &= \mathop{\mathbb{E}}_{\substack{x\sim d_\rho^{\widetilde{\pi}} \\ s,s'\sim b(\cdot|x)}} \left[\mathrm{d}_{\mathcal{S}}(s,s')\right] \\
&\leqslant \mathop{\mathbb{E}}_{\substack{x\sim d_\rho^{\widetilde{\pi}} \\ s\sim b(\cdot|x')}} \left[\mathrm{d}_{\mathcal{S}}(s,s_x)\right] + \mathop{\mathbb{E}}_{\substack{x\sim d_\rho^{\widetilde{\pi}} \\ s'\sim b(\cdot|x)}} \left[\mathrm{d}_{\mathcal{S}}(s_x,s')\right] \\
&= 2 \mathop{\mathbb{E}}_{x\sim d_\rho^{\widetilde{\pi}}} \left[\int_{\mathcal{S}} d_{\mathcal{S}}(s,s_x) b(s|x)\, ds\right] \\
&= 2 \mathop{\mathbb{E}}_{x\sim d_\rho^{\widetilde{\pi}}} \left[\int_{\mathcal{S}} \mathrm{d}_{\mathcal{S}}(s,s_x)\left(b(s|x) - \delta_{s_x}(s)\right)\, ds\right] &(11)\\
&\leqslant 2 \mathop{\mathbb{E}}_{x\sim d_\rho^{\widetilde{\pi}}} \left[\mathcal{W}_1\left(b(\cdot|x)\|\delta_{s_x}\right)\right], &(12)
\end{aligned}$$

where Equation (11) holds because $\int_{\mathcal{S}} d_{\mathcal{S}}(s,s_x)\delta_{s_x}(s)ds = 0$ and Equation (12) follows by recognizing the Wasserstein distance. One can then use Proposition A.5 on each of the two terms to conclude. ☐

B. Bounding the Value Function via Performance Difference Lemma

Lemma 6.1. *[Delayed Performance Difference Lemma] Consider an undelayed policy π_E and a Δ-delayed policy $\widetilde{\pi}$, with $\Delta \in \mathbb{R}_{\geqslant 0}$. Then, for any $x \in \mathcal{X}$,*

$$\mathop{\mathbb{E}}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - V^{\widetilde{\pi}}(x) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{x'\sim d_x^{\widetilde{\pi}}}\left[\mathop{\mathbb{E}}_{s\sim b(\cdot|x')}[V^{\pi_E}(s)] - \mathop{\mathbb{E}}_{\substack{s\sim b(\cdot|x') \\ a\sim\widetilde{\pi}(\cdot|x')}}[Q^{\pi_E}(s,a)]\right].$$

Proof. We first prove the result for integer delay $\Delta \in \mathbb{N}$. We start by adding and subtracting $\mathbb{E}_{\substack{s\sim b(\cdot|x') \\ a\sim\widetilde{\pi}(\cdot|x')}}\left[r(s,a) + \gamma\,\mathbb{E}_{s'\sim p(\cdot|s,a)}[V^{\pi_E}(s')]\right]$ to the quantity of interest $I(x) = \mathbb{E}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - V^{\widetilde{\pi}}(x)$.

This yields:

$$\begin{aligned}
I(x) = &\underbrace{\mathop{\mathbb{E}}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - \mathop{\mathbb{E}}_{\substack{s\sim b(\cdot|x') \\ a\sim\widetilde{\pi}(\cdot|x')}}\left[r(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim p(\cdot|s,a)}[V^{\pi_E}(s')]\right]}_{A} \\
&+ \underbrace{\mathop{\mathbb{E}}_{\substack{s\sim b(\cdot|x') \\ a\sim\widetilde{\pi}(\cdot|x')}}\left[r(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim p(\cdot|s,a)}[V^{\pi_E}(s')]\right] - V^{\widetilde{\pi}}(x)}_{B}.
\end{aligned}$$

The first term is

$$A = \mathop{\mathbb{E}}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - \mathop{\mathbb{E}}_{\substack{s\sim b(\cdot|x') \\ a\sim\widetilde{\pi}(\cdot|x')}}[Q^{\pi_E}(s,a)].$$

For the second term, note that $V^{\widetilde{\pi}}(x) = \mathbb{E}_{\substack{s\sim b(\cdot|x') \\ a\sim\widetilde{\pi}(\cdot|x')}}[r(s,a)] + \gamma\,\mathbb{E}_{\substack{x'\sim\tilde{p}(\cdot|x,a) \\ a\sim\widetilde{\pi}(\cdot|x')}}[V^{\widetilde{\pi}}(x')]$. Therefore,

$$B = \gamma \mathop{\mathbb{E}}_{\substack{s\sim b(\cdot|x') \\ a\sim\widetilde{\pi}(\cdot|x')}}\left[\mathop{\mathbb{E}}_{s'\sim p(\cdot|s,a)}[V^{\pi_E}(s')]\right] - \gamma \mathop{\mathbb{E}}_{\substack{x'\sim\tilde{p}(\cdot|x,a) \\ a\sim\widetilde{\pi}(\cdot|x')}}[V^{\widetilde{\pi}}(x')].$$

By observing that $\int_{\mathcal{S}} b(s|x)p(s'|s,a)ds = \int_{\mathcal{X}} \tilde{p}(x'|x,a)b(s'|x')dx'$, that is, the probability of the next state conditioned on the current augmented state x and the action a can be computed by either conditioning on the probability of the current unobserved state s or conditioning on the next augmented state x'.

$$\begin{aligned} B &= \gamma \mathop{\mathbb{E}}_{\substack{x'\sim\tilde{p}(\cdot|x,a) \\ a\sim\widetilde{\pi}(\cdot|x')}}\left[\mathop{\mathbb{E}}_{s\sim b(\cdot|x')}[V^{\pi_E}(s')] - V^{\widetilde{\pi}}(x')\right] \\ &= \gamma \mathop{\mathbb{E}}_{\substack{x'\sim\tilde{p}(\cdot|x,a) \\ a\sim\widetilde{\pi}(\cdot|x')}}\left[I(x')\right], \end{aligned}$$

where we have recognised the quantity of interest I taken at another augmented state. One can thus iterate as in the original performance difference lemma to get

$$\begin{aligned} I(x) &= \sum_{t=0}^{\infty}\gamma^t \mathbb{E}^{\widetilde{\pi}}\left[\mathop{\mathbb{E}}_{s\sim b(\cdot|x_t)}[V^{\pi_E}(s)] - \mathop{\mathbb{E}}_{\substack{s\sim b(\cdot|x_t) \\ a\sim\widetilde{\pi}(\cdot|x_t)}}[Q^{\pi_E}(s,a)]\middle| x_0 = x\right] \\ &= \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{x'\sim d_x^{\widetilde{\pi}}}\left[\mathop{\mathbb{E}}_{s\sim b(\cdot|x')}[V^{\pi_E}(s)] - \mathop{\mathbb{E}}_{\substack{s\sim b(\cdot|x') \\ a\sim\widetilde{\pi}(\cdot|x')}}[Q^{\pi_E}(s,a)]\right], \end{aligned} \tag{13}$$

where Equation (13) is obtained by recognising the discounted state distribution under policy $\widetilde{\pi}$. This completes the proof.

We now assume non-integer delay, setting $\Delta \in (0,1)$ but the proof for $\Delta \in \mathbb{R}$ follows easily. The proof is the same as above except for substituting b with b_Δ. One step which might not be evident is that

$$\int_{\mathcal{S}} b_\Delta(s_{t+\Delta}|x_t)p(s_{t+1+\Delta}|s_{t+\Delta},a)\; ds_{t+\Delta} = \int_{\mathcal{X}} \tilde{p}(x_{t+1}|x_t,a)b_\Delta(s_{t+1+\Delta}|x_{t+1})\; dx_{t+1}.$$

This is true because, for $x_t = (s_t, a_{t-1}), x_{t+1} = (s_{t+1}, a_t) \in \mathcal{X}$,

$$\begin{aligned} &\int_{\mathcal{S}} b_\Delta(s_{t+\Delta}|x_t)p(s_{t+1+\Delta}|s_{t+\Delta},a)\; ds_{t+\Delta} \\ &\quad= \int_{\mathcal{S}} b_\Delta(s_{t+\Delta}|x_t)\int_{\mathcal{S}} b_\Delta(s_{t+1+\Delta}|s_{t+1},a)b_{1-\Delta}(s_{t+1}|s_{t+\Delta},a)\; ds_{t+1}\; ds_{t+\Delta} \qquad (14) \\ &\quad= \int_{\mathcal{S}} b_\Delta(s_{t+1+\Delta}|s_{t+1},a)\int_{\mathcal{S}} b_\Delta(s_{t+\Delta}|x_t)b_{1-\Delta}(s_{t+1}|s_{t+\Delta},a)\; ds_{t+\Delta}\; ds_{t+1} \\ &\quad= \int_{\mathcal{S}} b_\Delta(s_{t+1+\Delta}|s_{t+1},a)\int_{\mathcal{S}} b_\Delta(s_{t+\Delta}|s_t,a_{t-1})b_{1-\Delta}(s_{t+1}|s_{t+\Delta},a)\; ds_{t+\Delta}\; ds_{t+1} \\ &\quad= \int_{\mathcal{A}}\int_{\mathcal{S}} b_\Delta(s_{t+1+\Delta}|s_{t+1},a_t)\delta_a(a_t)\int_{\mathcal{S}} b_\Delta(s_{t+\Delta}|s_t,a_{t-1})b_{1-\Delta}(s_{t+1}|s_{t+\Delta},a)\; ds_{t+\Delta}\; ds_{t+1}\; da_t \qquad (15) \\ &\quad= \int_{\mathcal{X}} b_\Delta(s_{t+1+\Delta}|x_{t+1})\tilde{p}(x_{t+1}|x_t,a)\; dx_{t+1} \end{aligned}$$

where Equation (14) holds by replacing the transition p as in Equation (4) and Equation (15) holds by definition of the transition in the augmented MDP. □

Theorem 6.1. *Consider an (L_P, L_r)-LC MDP and a L_π-LC undelayed policy π_E, such that Q^{π_E} is L_Q -L.C.*[4]*. Let $\widetilde{\pi}_b$ be*

[4] In fact, only Lipschizness in the second argument is necessary (see proof).

the Δ-delayed policy defined as in Equation (3), with $\Delta \in \mathbb{R}_{\geqslant 0}$. Then, for any $x \in \mathcal{X}$,

$$\mathbb{E}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - V^{\widetilde{\pi}_b}(x) \leqslant \frac{L_Q L_\pi}{1-\gamma}\sigma_b^x,$$

where $\sigma_b^x = \mathbb{E}_{\substack{x'\sim d_x^{\widetilde{\pi}_b} \\ s,s'\sim b(\cdot|x')}}[\mathrm{d}_{\mathcal{S}}(s,s')]$.

Proof. We first prove the result for integer delay $\Delta \in \mathbb{N}$. The first step in this proof is to use the results of Lemma 6.1, which yields, for any $x \in \mathcal{X}$:

$$\mathbb{E}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - V^{\widetilde{\pi}_b}(x) \leqslant \frac{1}{1-\gamma}\mathbb{E}_{x'\sim d_x^{\widetilde{\pi}_b}}\left[\underbrace{\mathbb{E}_{s\sim b(\cdot|x')}[V^{\pi_E}(s)] - \mathbb{E}_{\substack{s\sim b(\cdot|x') \\ a\sim \widetilde{\pi}_b(\cdot|x')}}[Q^{\pi_E}(s,a)]}_{A}\right].$$

We consider the term inside the expectation over x', called A. We reformulate this term to highlight how we then apply Proposition A.3.

$$\begin{aligned} A &= \mathbb{E}_{s\sim b(\cdot|x')}\left[\mathbb{E}_{\substack{a_1\sim\pi_E(\cdot|s) \\ a_2\sim\widetilde{\pi}_b(\cdot|x')}}[Q^{\pi_E}(s,a_1) - Q^{\pi_E}(s,a_2)]\right] \\ &\leqslant L_Q \mathbb{E}_{s\sim b(\cdot|x')}\left[\mathcal{W}_1(\pi_E(\cdot|s)\|\widetilde{\pi}_b(\cdot|x'))\right]. \end{aligned} \tag{16}$$

To finish the proof, it remains to upper bound $\mathbb{E}_{s\sim b(\cdot|x')}[\mathcal{W}_1(\pi_E(\cdot|s)\|\widetilde{\pi}_b(\cdot|x'))]$ with $\sigma_b^x \coloneqq \mathbb{E}_{\substack{x'\sim d_x^{\widetilde{\pi}_b} \\ s,s'\sim b(\cdot|x')}}[\mathrm{d}_{\mathcal{S}}(s,s')]$.

One has

$$\begin{aligned} \mathcal{W}_1(\pi_E(\cdot|s)\|\widetilde{\pi}_b(\cdot|x')) &= \sup_{\|f\|_L\leqslant 1}\left|\int_{\mathcal{A}} f(a)(\pi_E(a|s) - \widetilde{\pi}_b(a|x'))da\right| \\ &= \sup_{\|f\|_L\leqslant 1}\left|\int_{\mathcal{A}} f(a)(\pi_E(a|s) - \int_{\mathcal{S}}\pi_E(a|s')b(s'|x')ds')da\right| && (17) \\ &\leqslant \int_{\mathcal{S}} b(s'|x')\sup_{\|f\|_L\leqslant 1}\left|\int_{\mathcal{A}} f(a)(\pi_E(a|s) - \pi_E(a|s'))da\right|ds' && (18) \\ &\leqslant \int_{\mathcal{S}} b(s'|x')\mathcal{W}_1(\pi_E(\cdot|s)\|\pi_E(\cdot|s'))ds' \\ &\leqslant L_\pi\int_{\mathcal{S}} b(s'|x')\,\mathrm{d}_{\mathcal{S}}(s,s')ds', && (19) \end{aligned}$$

where Equation (17) holds by definition of the optimal imitated delayed policy (see Equation (3)), Equation (18) holds by application of Fubini-Tonelli's theorem and Equation (19) holds by Lipschitzness of the expert undelayed policy. By re-injecting this result into Equation (16) we get the desired result.

We now assume non-integer delay, setting $\Delta \in (0,1)$ but the proof for $\Delta \in \mathbb{R}$ follows easily. In this case, the optimal policy learnt by DIDA is

$$\widetilde{\pi}_b(a|x) = \int_{\mathcal{S}} b_\Delta(s|x)\pi_E(a|s)ds. \tag{20}$$

The proof remains the same except for the first step, where Lemma 6.1 is of course applied in its non-integer delay version. □

Corollary 6.2. *Under the assumptions of Theorem 6.1 adding that $\mathcal{S} \subset \mathbb{R}^n$ is equipped with the Euclidean norm. Then, for any $x \in \mathcal{X}$,*

$$\mathop{\mathbb{E}}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - V^{\widetilde{\pi}_b}(x) \leqslant \frac{2L_Q L_\pi}{1-\gamma} \mathop{\mathbb{E}}_{x'\sim d_x^{\widetilde{\pi}_b}(\cdot)}\left[\sqrt{\mathop{\mathbb{V}\mathrm{ar}}_{s\sim b(\cdot|x')}(s|x')}\right].$$

Proof. The result follows from application of Lemma A.1 to Theorem 6.1. □

Corollary 6.1. *Under the assumptions of Theorem 6.1, adding that the MDP is L_T-TLC, then, for any $x \in \mathcal{X}$,*

$$\mathop{\mathbb{E}}_{s\sim b(\cdot|x)}[V^{\pi_E}(s)] - V^{\widetilde{\pi}_b}(x) \leqslant \frac{2\Delta L_T L_Q L_\pi}{1-\gamma}.$$

Proof. The result follows from application of Lemma A.2 to Theorem 6.1. □

B.1. Lower Bounding the Value Function

Theorem 6.2. *For every $L_\pi > 0$, $L_Q > 0$, there exists an MDP such that the optimal policy is L_π-LC, its state action value function is L_Q-LC in the second argument, but for any Δ-delayed policy $\widetilde{\pi}$, with $\Delta \in \mathbb{R}_{\geqslant 0}$, and any $x \in \mathcal{X}$*

$$\mathop{\mathbb{E}}_{s\sim b(\cdot|x)}[V^*(s)] - V^{\widetilde{\pi}}(x) \geqslant \frac{\sqrt{2}}{\sqrt{\pi}} \frac{L_Q L_\pi}{1-\gamma} \mathop{\mathbb{E}}_{x'\sim d_x^{\widetilde{\pi}}(\cdot)}\left[\sqrt{\mathop{\mathbb{V}\mathrm{ar}}_{s\sim b(\cdot|x')}(s|x')}\right],$$

where V^* is the value function of the optimal undelayed policy.

Proof. We consider an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, R, \mu, \gamma)$ such that $\mathcal{S} = \mathbb{R}$ and $\mathcal{A} = \mathbb{R}$, its state transition is given by $s_{t+1} = s_t + \frac{a}{L_\pi} + \varepsilon_t$, where $\varepsilon_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$. The transition distribution can be written as

$$p(s'|s,a) = \mathcal{N}\left(s';\ s + \frac{a}{L_\pi}, \sigma^2\right).$$

Defining $L_r := L_Q L_\pi$, the reward is given by $r(s,a) = -L_r \left|s + \frac{a}{L_\pi}\right|$. Note that the reward is always negative, yet the policy $\pi^*(\cdot|s) = \delta_{-L_\pi s}(s')$ always yields 0 reward and is therefore optimal. Clearly, its value function satisfies $V^*(s) = 0$ for every $s \in \mathcal{S}$. Its Q function is

$$\begin{aligned} Q^*(s,a) &= -L_r \left|s + \frac{a}{L_\pi}\right| + \gamma \int_{\mathbb{R}} V^*(s')\, p(s'|s,a)\, ds' \\ &= -L_r \left|s + \frac{a}{L_\pi}\right| \\ &= -L_Q \left|L_\pi s + a\right|. \end{aligned}$$

Therefore, Q^* is indeed L_Q-LC in the second argument.

Consider now any Δ-delayed policy $\widetilde{\pi}$. At each time step t, the current state s_t can be decomposed in this way

$$s_t = \underbrace{s_{t-\Delta-1} + \sum_{\tau=t-\Delta-1}^{t-1} \frac{a_\tau}{L_\pi}}_{=:\phi(x_t)} + \underbrace{\sum_{\tau=t-\Delta-1}^{t-1} \varepsilon_\tau}_{\epsilon},$$

where the first quantity is a deterministic function of the augmented state, while the second is distributed under $\mathcal{N}(0, d\sigma^2)$. The expected value of the instantaneous reward is then given by

$$\mathbb{E}[r(x_t, a)] = -L_r\,\mathbb{E}\left[\left|s_t + \frac{a}{L_\pi}\right|\right]$$
$$= -L_r\,\mathbb{E}\left[\left|\phi(x_t) + \frac{a}{L_\pi} + \sum_{\tau=t-\Delta-1}^{t-1} \varepsilon_\tau\right|\right].$$

The function $f : y \mapsto \mathbb{E}\left[|\mathcal{N}(y, \sigma)|\right]$ has minimum at 0 by symmetry of the normal distribution. Its value is the mean of a half-normal distribution, that is $\mathbb{E}\left[|\mathcal{N}(0, \sigma)|\right] = \frac{\sqrt{2}}{\sqrt{\pi}}\sigma$. Therefore

$$\mathbb{E}[r(x_t, a)] \leqslant -L_r \frac{\sqrt{2}}{\sqrt{\pi}}\sqrt{d}\sigma,$$

which implies

$$V^{\tilde{\pi}}(x_t) \leqslant -\frac{L_r}{1-\gamma}\frac{\sqrt{2}}{\sqrt{\pi}}\sqrt{d}\sigma$$
$$= -\frac{L_Q L_\pi}{1-\gamma}\frac{\sqrt{2}}{\sqrt{\pi}}\sqrt{\underset{s\sim b(\cdot|x')}{\mathbb{V}\mathrm{ar}}(s|x')},$$

by noticing that $L_r = L_Q L_\pi$ and that $\mathbb{V}ar_{s\sim b(\cdot|x)}(s|x) = d\sigma^2$. Note that $\sqrt{\mathbb{V}\mathrm{ar}_{s\sim b(\cdot|x)}(s|x)}$ is the same for each $x \in \mathcal{X}$ so we can replace it with $\mathbb{E}_{x'\sim d_x^{\tilde{\pi}}(\cdot)}\left[\sqrt{\mathbb{V}\mathrm{ar}_{s\sim b(\cdot|x')}(s|x')}\right]$ to have a result more similar to Theorem 6.1.

Recalling that the optimal value function had value 0 at any state concludes the proof. □

C. Bounding the Value Function via the State Distribution

For this bound, we wish to use the difference in state distribution between the delayed and the undelayed expert to grasp their difference. Obviously, they do not share the same state space since the DMDP is handled by augmenting the state. However, it is possible to define a unifying framework with the following object. In this section, we consider $\Delta \in \mathbb{N}$.

Definition C.1 (Δ^{th}-order MDP). *Given an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, R, \mu)$, we define its correspondent Δ^{th}-order MDP, $\Delta \in \mathbb{N}$, as the MDP $\overline{\mathcal{M}}$ (a "$^-$" will be used to refer to an element of an Δ^{th}-order MDP) with*

- *State space $\bar{\mathcal{S}} = \mathcal{S}^{\Delta+1} \times \mathcal{A}^\Delta$, whose states $\bar{s}$ are composed of the last $\Delta + 1$ states and Δ actions of the MDP, namely $\bar{s} = (s_1, a_1, s_2, a_2, \dots, s_\Delta, a_\Delta, s_{\Delta+1})$.*
- *Unchanged action space $\mathcal{A}$.*
- *Reward function R, overwriting the undelayed notation but using as input a Δ^{th}-order state, such that $R(\bar{s}_t, a) = R(s_t, a)$. The overwriting is justified by this equality.*
- *Transition function $\bar{p}$ given by*

$$\bar{p}(\bar{s}'|\bar{s}, a) = p(s'_{\Delta+1}|s_{\Delta+1}, a)\delta_a(a'_\Delta)\prod_{i=1}^{\Delta}\delta_{s_{i+1}}(s'_i)\prod_{i=1}^{d-1}\delta_{a_{i+1}}(a'_i), \tag{21}$$

  where $\bar{s} = (s_1, a_1, \dots, a_\Delta, s_{\Delta+1})$ and $\bar{s}' = (s'_1, a'_1, \dots, a'_\Delta, s'_{\Delta+1})$.
- *The initial state distribution $\bar{\mu}$ is such that the inital action queue x is distributed as in the delayed MDP while the states of the states queue are distributed as $s_{i+1} \sim p(\cdot|s_i, a_i)$.*

This definition is inspired from the concept of Δ^{th}-order Markov chain. In this definition, $s_{\Delta+1}$ is intended to be the current state, while s_1 is the Δ-delayed state, $a_{1:\Delta}$ is the action queue. Therefore, from the state of the Δ^{th}-order MDP, one can either extract an extended state and query a delayed policy or the extract the current state and query an undelayed policy. This implies that one can define the state distribution on the Δ^{th}-order MDP for both an undelayed and a delayed policy. We overwrite the notations and write respectively d^π and $d^{\bar{\pi}}$ the distribution of state on the Δ^{th}-order MDP. The fact that the distribution concerns a Δ^{th}-order state and not a state from the underlying MDP or DMDP will be clear from the notation of the variable which is sampled from this distribution. For instance, in $s \sim d^\pi$, d^π is a distribution defined by applying π on the undelayed MDP while $\tilde{s} \sim d^\pi$ assumes a distribution under π on the Δ^{th}-order MDP.

Before deriving bounds on the Δ^{th}-order state probability distribution, we first prove a Lipschitzness result concerning the Δ^{th}-order MDP which be used in later proofs.

Lemma C.1. *Consider an (L_P, L_r)-LC MDP and its Δ^{th}-order MDP counterpart. Let $f : \bar{\mathcal{S}} \to \mathbb{R}$ such that $\|f\|_L \leqslant 1$ w.r.t. to the L2-norm on $\bar{\mathcal{S}}$. Then, the function*

$$\begin{aligned} g_f : \bar{\mathcal{S}} \times \mathcal{A} &\longrightarrow \mathbb{R} \\ (\bar{s}, a) &\longmapsto \int_{\bar{\mathcal{S}}} f(\bar{s}')\bar{p}(\bar{s}'|\bar{s}, a)\, d\bar{s}', \end{aligned}$$

is L_P-LC w.r.t. the second variable.

Proof. Let $\bar{s} \in \bar{\mathcal{S}}$ such that $\bar{s} = (s_1, a_1, \ldots, a_\Delta, s_{\Delta+1})$ and $a, b \in \mathcal{A}$. Then,

$$\begin{aligned} |g_f(\bar{s}, a) - g_f(\bar{s}, b)| &= \left| \int_{\bar{\mathcal{S}}} f(\bar{s}') \left(\bar{p}(\bar{s}'|\bar{s}, a) - \bar{p}(\bar{s}'|\bar{s}, b)\right)\, d\bar{s}' \right| \\ &= \left| \int_{\mathcal{S}\times\mathcal{A}} f(s_2, a_2, \ldots, s_{\Delta+1}, a'_{\Delta+1}, s'_{\Delta+1}) \right. && (22) \\ &\quad \left. \left(p(s'_{\Delta+1}|s_{\Delta+1}, a)\delta_a(a'_{\Delta+1}) - p(s'_{\Delta+1}|s_{\Delta+1}, b)\delta_b(a'_{\Delta+1})\right)\, ds'_{\Delta+1}\, da'_{\Delta+1} \right| && (23) \end{aligned}$$

where in Equation (23) we integrate over the elements of $\bar{s}$ fixed by the Dirac distributions of Equation (21) except for the last action. Note that $h := (s, a) \mapsto f(s_2, a_2, \ldots, s_{\Delta+1}, a, s)$ is 1-LC because f is 1-LC. We add and substract the quantity $p(s'_{\Delta+1}|s_{\Delta+1}, b)\delta_a(a_{\Delta+1})$ inside the integral to get

$$\begin{aligned} |g_f(\bar{s}, a) - g_f(\bar{s}, b)| &\leqslant \left| \int_{\mathcal{S}\times\mathcal{A}} \delta_a(a'_{\Delta+1}) h(a'_{\Delta+1}, s'_{\Delta+1}) \left(p(s'_{\Delta+1}|s_{\Delta+1}, a) - p(s'_{\Delta+1}|s_{\Delta+1}, b)\right)\, ds'_{\Delta+1}\, da'_{\Delta+1} \right| \\ &\quad + \left| \int_{\mathcal{S}\times\mathcal{A}} h(a'_{\Delta+1}, s'_{\Delta+1}) p(s'_{\Delta+1}|s_{\Delta+1}, b) \left(\delta_a(a'_{\Delta+1}) - \delta_b(a'_{\Delta+1})\right)\, ds'_{\Delta+1}\, da'_{\Delta+1} \right| \\ &\leqslant (L_P + 1)\, \mathrm{d}_{\mathcal{A}}(a, b), \end{aligned}$$

where the last inequality follows by integrating over $s'_{\Delta+1}$ for the first integral and $a'_{\Delta+1}$ for the second, before taking the supremum over functions h and recognising the Wasserstein distance. $\square$

We can now prove a first important intermediary result which bounds the Wasserstein divergence in discounted state distribution in the Δ^{th}-order MDP between the undelayed and the delayed policy.

Theorem C.1. *Consider an (L_P, L_r)-LC MDP $\mathcal{M}$ and its $(L_{\bar{P}}, L_r)$-LC Δ^{th}-order MDP counterpart $\overline{\mathcal{M}}$. Let π_E be a L_π-LC undelayed policy and assume that $\gamma L_P(1 + L_\pi) \leqslant 1$. Let $\widetilde{\pi}_b$ be a delayed policy as defined in Equation (3). Then, the two discounted state distributions $d_{\bar{\mu}}^{\pi_E}, d_{\bar{\mu}}^{\widetilde{\pi}_b}$ defined on $\overline{\mathcal{M}}$ satisfy*

$$\mathcal{W}_1\left(d_{\bar{\mu}}^{\pi_E} \| d_{\bar{\mu}}^{\widetilde{\pi}_b}\right) \leqslant \gamma L_Q (1 + L_P) \sigma_b^{\widetilde{\mu}}$$

where $\sigma_b^{\widetilde{\mu}} = \mathbb{E}_{\substack{x \sim d_{\widetilde{\mu}}^{\widetilde{\pi}_b} \\ s, s' \sim b(\cdot|x)}} [\mathrm{d}_{\mathcal{S}}(s, s')]$ and $d_{\widetilde{\mu}}^{\widetilde{\pi}_b}$ is defined on the DMDP.

Proof. We start by developing the term $\mathcal{W}_1\left(d_{\bar{\mu}}^{\pi_E}\|d_{\bar{\mu}}^{\tilde{\pi}_b}\right)$, using the supremum over the space of functions $f:\bar{\mathcal{S}}\to\mathbb{R}$ such that $\|f\|_L\leqslant 1$ w.r.t. to the L2-norm on $\bar{\mathcal{S}}$.

$$\mathcal{W}_1\left(d_{\bar{\mu}}^{\pi_E}\|d_{\bar{\mu}}^{\tilde{\pi}_b}\right)=\sup_{\|f\|_L\leqslant 1}\left|\int_{\bar{\mathcal{S}}}f(\bar{s})\left(d_{\bar{\mu}}^{\pi_E}(\bar{s})-d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s})\right)\,d\bar{s}\right|.$$

We then use the fact that for some policy $\pi\in\Pi$, $d_{\bar{\mu}}^{\pi}(\bar{s})=(1-\gamma)\bar{\mu}(\bar{s})+\gamma\int_{\bar{\mathcal{S}}}\bar{p}^{\pi}(\bar{s}|\bar{s}')d_{\bar{\mu}}^{\pi}(\bar{s}')\,d\bar{s}'$, where $\bar{p}^{\pi}(\bar{s}|\bar{s}')=\int_{\mathcal{A}}p(\bar{s}'|\bar{s},a)\pi(a|\bar{s})\,da$ to yield

$$\begin{aligned}\mathcal{W}_1\left(d_{\bar{\mu}}^{\pi_E}\|d_{\bar{\mu}}^{\tilde{\pi}_b}\right)&=\gamma\sup_{\|f\|_L\leqslant 1}\left|\int_{\bar{\mathcal{S}}}f(\bar{s})\int_{\bar{\mathcal{S}}}\left(\bar{p}^{\pi_E}(\bar{s}|\bar{s}')d_{\bar{\mu}}^{\pi_E}(\bar{s}')-\bar{p}^{\tilde{\pi}_b}(\bar{s}|\bar{s}')d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s}')\right)\,d\bar{s}'\,d\bar{s}\right|\\&\leqslant\gamma\underbrace{\sup_{\|f\|_L\leqslant 1}\left|\int_{\bar{\mathcal{S}}}f(\bar{s})\int_{\bar{\mathcal{S}}}\left(d_{\bar{\mu}}^{\pi_E}(\bar{s}')-d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s}')\right)\bar{p}^{\pi_E}(\bar{s}|\bar{s}')\,d\bar{s}'\,d\bar{s}\right|}_{A}\\&\quad+\gamma\underbrace{\sup_{\|f\|_L\leqslant 1}\left|\int_{\bar{\mathcal{S}}}f(\bar{s})\int_{\bar{\mathcal{S}}}\left(\bar{p}^{\pi_E}(\bar{s}|\bar{s}')-\bar{p}^{\tilde{\pi}_b}(\bar{s}|\bar{s}')\right)d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s}')\,d\bar{s}'\,d\bar{s}\right|}_{B},\end{aligned}\tag{24}$$

where Equation (24) follows by adding and subtracting the term $\bar{p}^{\pi_E}(\bar{s}|\bar{s}')d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s}')$ and using the triangular inequality.

The first term A can be bounded by using Fubini's theorem first and then leveraging Proposition A.4 which implies that $\int_{\bar{\mathcal{S}}}f(\bar{s})\bar{p}^{\pi_E}(\bar{s}|\bar{s}')\,d\bar{s}'$ is $L_P(1+L_\pi)$-LC. Therefore

$$A\leqslant L_P(1+L_\pi)\mathcal{W}_1\left(d_{\bar{\mu}}^{\pi_E}\|d_{\bar{\mu}}^{\tilde{\pi}_b}\right)$$

Now looking at the second term B, we will develop the term $\bar{p}^{\pi_E}$ and $\bar{p}^{\tilde{\pi}_b}$ to highlight the influence of the policy and leverage Equation (3). We note $\bar{s}=(s_1,a_1,\dots,a_\Delta,s_{\Delta+1})$ and $\bar{s}'=(s_1',a_1',\dots,a_\Delta',s_{\Delta+1}')$. Moreover, we overwrite the notation of the belief to use it on a Δ^{th}-order state such that $b(z|\bar{s})=b(z|s_1,a_1,\dots,a_\Delta)$, that is, the belief is based on the augmented state constructed from the oldest state inside $\bar{s}$ and the sequence of action it contains.

$$\begin{aligned}B&=\sup_{\|f\|_L\leqslant 1}\left|\int_{\bar{\mathcal{S}}}f(\bar{s})\int_{\bar{\mathcal{S}}}\left(\int_{\mathcal{A}}\bar{p}(\bar{s}|\bar{s}',a)\pi_E(a|s_{\Delta+1}')\,da-\int_{\mathcal{A}}\int_{z\in\mathcal{S}}\bar{p}(\bar{s}|\bar{s}',a)b(z|\bar{s}')\pi_E(a|z)\,dz\,da\right)d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s}')\,d\bar{s}'\,d\bar{s}\right|\\&=\sup_{\|f\|_L\leqslant 1}\left|\int_{\bar{\mathcal{S}}}f(\bar{s})\left(\int_{\bar{\mathcal{S}}}\int_{\mathcal{A}}\underbrace{\left(\pi_E(a|s_{\Delta+1}')-\int_{z\in\mathcal{S}}b(z|\bar{s}')\pi_E(a|z)\,dz\right)}_{C}\bar{p}(\bar{s}|\bar{s}',a)\,da\,d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s}')\,d\bar{s}'\,d\bar{s}\right)\right|,\end{aligned}$$

where the term C in this equation accounts for the difference in taking an action with the undelayed policy π_E instead of the belief-based policy. Fubini's theorem yields

$$B=\sup_{\|f\|_L\leqslant 1}\left|\int_{\bar{\mathcal{S}}}\int_{\mathcal{A}}\left(\pi_E(a|s_{\Delta+1}')-\int_{z\in\mathcal{S}}b(z|\bar{s}')\pi_E(a|z)\,dz\right)\underbrace{\left(\int_{\bar{\mathcal{S}}}f(\bar{s})\bar{p}(\bar{s}|\bar{s}',a)\,d\bar{s}\right)}_{g_f(\bar{s}',a)}\,da\,d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s}')\,d\bar{s}'\right|,$$

where we note $g_f(\bar{s}',a):=\int_{\bar{\mathcal{S}}}f(\bar{s})\bar{p}(\bar{s}|\bar{s}',a)\,d\bar{s}$. This function is $(1+L_P)$-LC in a by Lemma C.1. Noting also that $\pi_E(a|\bar{s}_{\Delta+1}')=\int_{z\in\mathcal{S}}\pi_E(a|\bar{s}_{\Delta+1}')b(z|\bar{s}')dz$, one gets

$$B=\sup_{\|f\|_L\leqslant 1}\left|\int_{\bar{\mathcal{S}}}\int_{z\in\mathcal{S}}\int_{\mathcal{A}}\left(\pi_E(a|s_{\Delta+1}')-\pi_E(a|z)\right)g_f(\bar{s}',a)\,dab(z|\bar{s}')\,dz\,d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s}')\,d\bar{s}'\right|$$

Then, by Lipschitzness of π_E,

$$B \leqslant L_\pi(1+L_P)\int_{\bar{\mathcal{S}}}\int_{z\in\mathcal{S}} \mathrm{d}_{\mathcal{S}}(s'_{\Delta+1},z)b(z|\bar{s}')\,dz\,d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s}')\,d\bar{s}'.$$

Now, one can observe that $s\mapsto\int_{\bar{\mathcal{S}}}\delta(s=s_{\Delta+1})d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s})\,d\bar{s}$ and $s\mapsto\int_{\mathcal{X}} b(s|x)d_{\tilde{\mu}}^{\tilde{\pi}_b}(x)\,dx$ define the same distribution over $\mathcal{S}$. Note that the discounted state distributions $d_{\bar{\mu}}^{\tilde{\pi}_b}(\bar{s})$ is over the Δ^{th}-order MDP's state space while $d_{\tilde{\mu}}^{\tilde{\pi}_b}(x)$ is over the DMDP's state space. This yields

$$\begin{aligned}
B &\leqslant L_\pi(1+L_P)\int_{\mathcal{X}}\int_{s'\in\mathcal{S}}\int_{s\in\mathcal{S}} d_{\mathcal{S}}(s,s')b(s|x)b(s'|x)\,ds\,ds'\,d_{\tilde{\mu}}^{\tilde{\pi}_b}(x)\,dx\\
&= L_\pi(1+L_P)\mathop{\mathbb{E}}_{\substack{x\sim d_{\tilde{\mu}}^{\tilde{\pi}_b}\\ s,s'\sim b(\cdot|x)}}\left[\mathrm{d}_{\mathcal{S}}(s,s')\right]\\
&= L_\pi(1+L_P)\sigma_b^{\tilde{\mu}}.
\end{aligned}$$

One can now resume at Equation (24), and recording that $\gamma L_P(1+L_\pi)\leqslant 1$, one gets

$$\begin{aligned}
\mathcal{W}_1\left(d_{\bar{\mu}}^{\pi_E}\|d_{\bar{\mu}}^{\tilde{\pi}_b}\right) &\leqslant \gamma L_P(1+L_\pi)\mathcal{W}_1\left(d_{\bar{\mu}}^{\pi_E}\|d_{\bar{\mu}}^{\tilde{\pi}_b}\right)+\gamma L_\pi(1+L_P)\sigma_b^{\mu}\\
&\leqslant \frac{\gamma L_\pi(1+L_P)}{1-\gamma L_P(1+L_\pi)}\sigma_b^{\tilde{\mu}}.
\end{aligned}$$

Finally, recalling that $L_Q=\frac{L_r}{1-\gamma L_P(1+L_\pi)}$ finishes the proof. □

the previous results will be useful to prove a bound on the value function. However, before providing this result, we need a last intermediary result.

Proposition C.1. *The Δ^{th}-order MDP $\overline{\mathcal{M}}$ can be reduced to an Δ^{th}-order MDP where the mean reward is redefined as $\bar{r}(\bar{s},a)=r(s_1,a_1)$. Because it doesn't depend upon a, we equivalently write $\bar{r}(\bar{s}):=\bar{r}(\bar{s},a)$*

Proof. We derive a similar proof as Katsikopoulos & Engelbrecht (2003). Let V^π be the value function of any stationary Markovian policy π on $\overline{\mathcal{M}}$ and $\bar{V}^\pi$ its value function based on the reward $\bar{r}$. We show that there exist a quantity $I(\bar{s})$ such that $\bar{V}^\pi(\bar{s})=I(\bar{s})+V^\pi(\bar{s})$. Since the quantity $I(\bar{s})$ does not depend on π, it means that the ordering of the policies in terms of value function and thus performance is preserved by using this new reward function. Note that we can express the t^{th} state in $\overline{\mathcal{M}}$ as a tuple of element of the underlying MDP as $\bar{s}_t=(s_{t-\Delta},a_{t-\Delta},\dots,a_{t-1},s_t)$. We allow for negative indexing in the underlying MDP for the first term which are fixed by $\bar{\mu}$.

We now proceed to the write $\bar{V}^\pi$ to introduce as a function of $V^\pi(\bar{s})$.

$$\begin{aligned}
\bar{V}^\pi(\bar{s}) &= \mathop{\mathbb{E}}_{\substack{\bar{s}_{t+1}\sim p(\cdot|\bar{s}_t,a_t)\\ a_t\sim\pi(\cdot|\bar{s}_t)}}\left[\sum_{t=0}^{\infty}\gamma^t\bar{r}(\bar{s}_t,a_t)\middle|\bar{s}_0=\bar{s}\right]\\
&= \mathop{\mathbb{E}}_{\substack{\bar{s}_{t+1}\sim p(\cdot|\bar{s}_t,a_t)\\ a_t\sim\pi(\cdot|\bar{s}_t)}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_{t-\Delta},a_{t-\Delta})\middle|\bar{s}_0=\bar{s}\right]\\
&= \underbrace{\mathop{\mathbb{E}}_{\bar{s}_{t+1}\sim p(\cdot|\bar{s}_t,a_t)}\left[\sum_{t=0}^{\Delta-1}\gamma^t r(s_{t-\Delta},a_{t-\Delta})\middle|\bar{s}_0=\bar{s}\right]}_{I(\bar{s})}+\mathop{\mathbb{E}}_{\substack{\bar{s}_{t+1}\sim p(\cdot|\bar{s}_t,a_t)\\ a_t\sim\pi(\cdot|\bar{s}_t)}}\left[\sum_{t=\Delta}^{\infty}\gamma^t r(s_{t-\Delta},a_{t-\Delta})\middle|\bar{s}_0=\bar{s}\right].
\end{aligned}$$

Two things have to be noted in the previous equation, first, the term on the left does not depend on π anymore since the actions involved in the reward collection are already contained in $\bar{s}$. It is our sought-after quantity I. Second, the reward in the term of the right can be interpreted are regular reward of a Δ^{th}-order MDP since $r(\bar{s}_t,a_t)=r(s_t,a_t)$.

Therefore,

$$\begin{aligned}\bar{V}^{\pi}(\bar{s}) &= I(\bar{s}) + \mathop{\mathbb{E}}_{\substack{\bar{s}_{t+1}\sim p(\cdot|\bar{s}_t,a_t)\\ a_t\sim\pi(\cdot|\bar{s}_t)}}\left[\sum_{t=\Delta}^{\infty}\gamma^t r(\bar{s}_t,a_t)\middle|\bar{s}_0=\bar{s}\right]\\ &= I(\bar{s}) + V^{\pi}(\bar{s}).\end{aligned}$$

We found such a quantity I to link $\bar{V}^{\pi}$ and V^{π}, proving the statement. □

We are now able to prove a bound between delayed and undelayed value functions.

Theorem C.2. *Consider an (L_P, L_r)-LC MDP $\mathcal{M}$ and its Δ^{th}-order MDP counterpart $\overline{\mathcal{M}}$. Let π_E be a L_π-LC undelayed policy and assume that $\gamma L_P(1+L_\pi) \leqslant 1$. Let $\widetilde{\pi}_b$ be a delayed policy as defined in Equation (3). Then, for any $\bar{s}\in\bar{\mathcal{S}}$,*

$$\left|V^{\pi_E}(\bar{s}) - V^{\widetilde{\pi}_b}(\bar{s})\right| \leqslant \frac{\gamma}{1-\gamma}L_\pi L_Q(1+L_P)\sigma_b^x,$$

where V^{π_E} and $V^{\widetilde{\pi}_b}$ are defined on $\overline{\mathcal{M}}$ and $\sigma_b^x = \mathop{\mathbb{E}}_{\substack{x\sim d_x^{\widetilde{\pi}_b}\\ s,s'\sim b(\cdot|x)}}[\mathrm{d}_{\mathcal{S}}(s,s')]$ for x the augmented state contained in $\bar{s}$ and $d_x^{\widetilde{\pi}_b}$ being defined on the DMDP.

Proof. By (Puterman, 2014), the two state value functions can be written as follows.

$$V^{\pi_E}(\bar{s}) = \frac{1}{1-\gamma}\int_{\bar{\mathcal{S}}}\int_{\bar{\mathcal{A}}} r(\bar{s}',a)\pi_E(a|\bar{s})d_{\bar{s}}^{\pi_E}(\bar{s}')\;da\;d\bar{s}'.$$

$$V^{\widetilde{\pi}_b}(\bar{s}) = \frac{1}{1-\gamma}\int_{\bar{\mathcal{S}}}\int_{\bar{\mathcal{A}}} r(\bar{s}',a)\widetilde{\pi}_b(a|\bar{s})d_{\bar{s}'}^{\widetilde{\pi}_b}(\bar{s})\;da\;d\bar{s}'.$$

Writing their difference gives

$$V^{\pi_E}(\bar{s}) - V^{\widetilde{\pi}_b}(\bar{s}) = \frac{1}{1-\gamma}\int_{\bar{\mathcal{S}}}\int_{\bar{\mathcal{A}}} r(\bar{s}',a)\left(\pi_E(a|\bar{s})d_{\bar{s}}^{\pi_E}(\bar{s}') - \widetilde{\pi}_b(a|\bar{s})d_{\bar{s}}^{\widetilde{\pi}_b}(\bar{s}')\right)\;da\;d\bar{s}'.$$

We now use Proposition C.1 to remove the integral over the action space.

$$\begin{aligned}V^{\pi_E}(\bar{s}) - V^{\widetilde{\pi}_b}(\bar{s}) &= \frac{1}{1-\gamma}\int_{\bar{\mathcal{S}}}\int_{\bar{\mathcal{A}}} \bar{r}(\bar{s}')\left(\pi_E(a|\bar{s})d_{\bar{s}'}^{\pi_E}(\bar{s}) - \widetilde{\pi}_b(a|\bar{s})d_{\bar{s}}^{\widetilde{\pi}_b}(\bar{s}')\right)\;da\;d\bar{s}'\\ &= \frac{1}{1-\gamma}\int_{\bar{\mathcal{S}}} \bar{r}(\bar{s}')\left(d_{\bar{s}}^{\pi_E}(\bar{s}') - d_{\bar{s}}^{\widetilde{\pi}_b}(\bar{s}')\right)\;d\bar{s}'.\end{aligned}$$

One can now use the fact that $\bar{r}$ is L_r-LC on $\bar{\mathcal{S}}$ because r is L_r-LC on $\mathcal{S}\times\mathcal{A}$. One thus gets

$$\left|V^{\pi_E}(\bar{s}) - V^{\widetilde{\pi}_b}(\bar{s})\right| \leqslant \frac{L_r}{1-\gamma}\mathcal{W}_1\left(d_{\bar{s}}^{\pi_E}\|d_{\bar{s}}^{\widetilde{\pi}_b}\right).$$

Applying Theorem C.1 yields

$$\left|V^{\pi_E}(\bar{s}) - V^{\widetilde{\pi}_b}(\bar{s})\right| \leqslant \frac{\gamma}{1-\gamma}L_\pi L_Q(1+L_P)\sigma_b^x,$$

where x is the augmented state contained in $\bar{s}$. This concludes the proof. □

As for Theorem 6.1, we can now use additional assumptions to bound σ_b^μ.

Corollary C.1. *Under the conditions of Theorem C.2 and adding that $\mathcal{S}\subset\mathbb{R}^n$ is equipped with the Euclidean norm. Let $\widetilde{\pi}_b$ be a Δ-delayed policy as defined in Equation (3). Then, for any $\bar{s}\in\bar{\mathcal{S}}$,*

$$\left|V^{\pi_E}(\bar{s}) - V^{\widetilde{\pi}_b}(\bar{s})\right| \leqslant \frac{\gamma}{1-\gamma}L_\pi L_Q(1+L_P)\mathop{\mathbb{E}}_{x'\sim d_x^{\widetilde{\pi}}(\cdot)}\left[\sqrt{\mathop{\mathbb{V}\mathrm{ar}}_{s\sim b(\cdot|x')}(s|x')}\right].$$

Proof. The result follows from application of Lemma A.1 to Theorem C.2. □

Corollary C.2. *Under the conditions of Theorem C.2 and adding that the MDP is L_T-TLC. Let $\widetilde{\pi}_b$ be a Δ-delayed policy as defined in Equation (3). Then,*

$$\left\| V^{\pi_E} - V^{\widetilde{\pi}_b} \right\|_\infty \leqslant \frac{2\Delta\gamma}{1-\gamma} L_T L_Q L_\pi (1 + L_P)$$

where V^{π_E} and $V^{\widetilde{\pi}_b}$ are defined on $\bar{\mathcal{S}}$.

Proof. First, one applies Lemma A.2 to Theorem C.2 to obtain

$$\left| V^{\pi_E}(\bar{s}) - V^{\widetilde{\pi}_b}(\bar{s}) \right| \leqslant \frac{2\Delta\gamma}{1-\gamma} L_\pi L_Q (1 + L_P),$$

for some $\bar{s} \in \bar{\mathcal{S}}$. Then, taking the maximum over $\bar{\mathcal{S}}$ gives the result since the *rhs* doesn't depend on $\bar{s}$. □

C.1. Comparison of the Two Bounds

As stated in Section 6, there are several choices for the quantities to consider when comparing undelayed to delayed performance. In this paper, Theorem C.2 provides a bound on the space of Δ^{th}-order MDP ($\bar{\mathcal{S}}$) while Theorem 6.1 compares a value function on the space of the augmented MDP ($\mathcal{X}$) to a value function on the classic state space ($\mathcal{S}$).

Recall the bounds for $\bar{s} \in \bar{\mathcal{S}}$ of Theorem C.2

$$\left| V^{\pi_E}(\bar{s}) - V^{\widetilde{\pi}_b}(\bar{s}) \right| \leqslant \frac{\gamma}{1-\gamma} L_\pi L_Q (1 + L_P) \sigma_b^x, \tag{25}$$

and for $x \in \mathcal{X}$ of Theorem 6.1

$$\mathop{\mathbb{E}}_{s \sim b(\cdot|x)} [V^{\pi_E}(s)] - V^{\widetilde{\pi}_b}(x) \leqslant \frac{L_Q L_\pi}{1-\gamma} \sigma_b^x, \tag{26}$$

where $\sigma_b^x = \mathop{\mathbb{E}}_{\substack{x' \sim d_x^{\widetilde{\pi}_b} \\ s,s' \sim b(\cdot|x')}} [\mathrm{d}_{\mathcal{S}}(s, s')]$.

As opposed to what the notations may suggest, the Lipschitz constant L_Q doesn't have exactly the same value. In Equation (25), we recognised in the proof the L_Q of the Q function as given in Rachelson & Lagoudakis (2010) under the assumption that $\gamma L_P (1 + L_\pi) \leqslant 1$. In Equation (26) however, we only assume that there exists such constant L_Q for which the Q function is L_Q-LC. That includes the case when $\gamma L_P (1 + L_\pi) \leqslant 1$ but is a more general result.

The other difference between the bounds lies in the factor $\gamma(L_P + 1)$ of Equation (25). Depending on the task, this factor may be smaller or greater than 1, changing the order of the bounds.

D. DIDA for non-integer delays

We provide a midification of Algorithm 1 to account for non-integer delays in Algorithm 2. Note that we consider a delay $\Delta \in \mathbb{R}_{\geqslant 0}$ and we note $\lfloor \Delta \rfloor$ the integer part of Δ and $\{\Delta\} := \Delta - \lfloor \Delta \rfloor$ its fractional part. We consider a continuous MDP which yields $\mathcal{M}$ and $\mathcal{M}_{\{\Delta\}}$.

E. Experimental details

E.1. Imitation Loss and DIDA's Policy

As stated in the main paper, the function learnt by DIDA can drift from Equation (3) depending on the class of policies of π_I and the loss function that is used for the imitation step. We derive the policy learnt by DIDA for the two following cases.

Mean squared error loss DIDA is trained on

$$\arg\min_\theta \int_{\mathcal{S}} \int_{\mathcal{A}} (a - \widetilde{\pi}_\theta(x))^2 \pi_E(a|s) b(s|x) da\, ds$$

Algorithm 2 DIDA for non-integer delays ($\Delta \in [0, 1]$)

Inputs $\{\Delta\} := \Delta - \lfloor\Delta\rfloor$, continuous MDP yielding $\mathcal{M}$ and $\mathcal{M}_{\{\Delta\}}$, undelayed expert π_E trained on $\mathcal{M}_{\{\Delta\}}$, β routine, number of steps N, empty dataset $\mathcal{D}$.
Outputs: delayed policy π_I

for β_i in β-routine **do**
  for j in $\{1, \dots, N\}$ **do**
    if New episode **then**
      Initialize state buffer $(s_1, s_{1+\{\Delta\}}, s_2, \dots, s_{\lceil\Delta\rceil}, s_{\Delta+1})$ and action buffer $(a_1, a_2 \dots, a_{\lceil\Delta\rceil})$
    end if
    Sample $a_E \sim \pi_E(\cdot|s_\Delta)$, set $a = a_E$
    if Random $u \sim U([0, 1]) \geqslant \beta_i$ **then**
      Overwrite $a \sim \pi_I(\cdot|[s_1, a_1, \dots, a_{\lceil\Delta\rceil}])$
    end if
    Aggregate dataset:
    $\mathcal{D} \leftarrow \mathcal{D} \cup ([s_1, a_1, a_2, \dots, a_{\lceil\Delta\rceil}], a_E)$
    Apply a at s_Δ and get new states $(s_{\lfloor\Delta\rfloor+2}, s_{\Delta+2})$
    Update buffers:
    $(s_1, s_{1+\{\Delta\}}, s_2, \dots, s_{\lceil\Delta\rceil}, s_{\Delta+1}) \leftarrow (s_2, s_{2+\{\Delta\}}, s_2, \dots, s_{\lceil\Delta\rceil+1}, s_{\Delta+2})$
    $(a_1, \dots, a_{\lceil\Delta\rceil}) \leftarrow (a_2, \dots, a_{\lceil\Delta\rceil}, a)$
  end for
  Train π_I on $\mathcal{D}$
end for

which is minimized for θ^* such that

$$\widetilde{\pi}_{\theta*}(x) = \int_{\mathcal{S}} \mathop{\mathbb{E}}_{a \sim \pi_E(\cdot|s)}[a] b(s|x) ds.$$

That means that the policy learnt by DIDA outputs the mean value of the action given the belief and the expert policy distribution.

Kullback-Leibler loss DIDA is trained on

$$\begin{aligned}
&\arg\min_\theta \int_{\mathcal{S}} D_{KL}(\pi_E(\cdot|s) \| \widetilde{\pi}_\theta(\cdot|x)) b(s|x) ds \\
&= \arg\min_\theta \int_{\mathcal{S}} \int_{\mathcal{A}} \pi_E(a|s) \log \pi_E(a|x)) b(s|x) da\ ds - \int_{\mathcal{S}} \int_{\mathcal{A}} \pi_E(a|s) \log \widetilde{\pi}_\theta(a|x)) b(s|x) da\ ds \\
&= \arg\min_\theta - \int_{\mathcal{S}} \int_{\mathcal{A}} \pi_E(a|s) \log \widetilde{\pi}_\theta(a|x)) b(s|x) da\ ds \qquad (27) \\
&= \arg\min_\theta - \int_{\mathcal{A}} \int_{\mathcal{S}} \pi_E(a|s) \log \widetilde{\pi}_\theta(a|x)) b(s|x) ds\ da \qquad (28) \\
&= \arg\min_\theta \int_{\mathcal{A}} \int_{\mathcal{S}} \pi_E(a|s) b(s|x) \log \left(b(s|x) \pi_E(a|x)\right)) ds\ da - \int_{\mathcal{A}} \int_{\mathcal{S}} \pi_E(a|s) \log \widetilde{\pi}_\theta(a|x)) b(s|x) ds\ da \\
&= \arg\min_\theta D_{KL}\left(\int_{\mathcal{S}} \pi_E(\cdot|s) b(s|x) ds, \widetilde{\pi}_\theta(\cdot|x)\right),
\end{aligned}$$

where Equation (27) holds because the first integral does not depend on θ and Equation (28) holds by Fubini's theorem since the functions inside the integral are always negative.

E.2. Hyper-parameters

For pendulum, the test performance are obtained from interacting 1000 steps with the environment, with maximum episode length of 200. For mujoco environments, the number of steps is 1000 as well but the maximum episode length is 500. The other of hyper-parameter are given for each approach, for each environment in the following tables.

Hyper-parameter	Pendulum	Mujoco	Trading
Policy type	Feed-forward	Feed-forward	Extra Trees
Iterations	245	400	10
Steps per iteration	10,000	10,000	2 years of data
β sequence	$\beta_1 = 1, \beta_{i \geqslant 2} = 0$	$\beta_1 = 1, \beta_{i \geqslant 2} = 0$	$\beta_1 = 1, \beta_{i \geqslant 2} = 0$
Max buffer size	10 iterations	10 iterations	Unlimited
Policy neurons	$[100, 100, 10]$	$[100, 100, 100, 10]$	$\varnothing$
Activations	ReLU (Nair & Hinton, 2010)	ReLU	$\varnothing$
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$) (Kingma & Ba, 2014)	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)	$\varnothing$
Learning rate	$1e-3$	$1e-3$	$\varnothing$
Batch size	64	64	$\varnothing$
Min samples split	$\varnothing$	$\varnothing$	100
n estimators	$\varnothing$	$\varnothing$	100

Table 1: Hyper-parameters for DIDA

Hyper-parameter	Pendulum	Mujoco
Epochs	2000	1000
Steps per epoch	5000	5000
Pre-training epochs	25	25
Pre-training steps	10000	10000
Backtracking line search iterations	10	10
Backtracking line search step	0.8	0.8
Conjugate gradient iterations	10	10
Discount γ	0.99	0.99
Max KL divergence	0.001	0.001
λ for GAE (Schulman et al., 2015b)	0.97	0.97
Value function neurons	$[64]$	$[64]$
Value function iterations	3	3
Value function learning rate	0.01	0.01
Policy neurons	$[64, 64]$	$[64, 64]$
Activations	ReLU	ReLU
Encoder feed-forward neurons	$[8]$	$[8]$
Encoder learning rate	0.01	0.01
Encoder iterations	2	2
Encoder dimension	64	64
Encoder heads	2	2
Encoder optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
Layers of MAF (Papamakarios et al., 2017)	5	5
Maf neurons	$[16]$	$[16]$
MAF learning rate	0.01	0.01
MAF optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
Epochs of training the belief	200	200
Batch size belief learning	10000	1000
Prediction buffer size	100000	100000
Belief representation dimension	8	32

Table 2: Hyper-parameters for D-TRPO

Hyper-parameter	Pendulum
Discount γ	0.99
Initial replay size	64
Buffer size	50000
Batch size	64
Actor μ neurons	256
Actor σ neurons	256
Actor optimizer	Adam $(\beta_1 = 0.9, \beta_2 = 0.999)$
Warmup transitions	100
Polyak update τ	0.005
Entropy learning rate	$3e-4$
Train frequency	50

Table 3: Hyper-parameters for M-SAC and A-SAC

Hyper-parameter	Pendulum	Mujoco
Epochs	2000	1000
Steps per epoch	5000	5000
Pre-training epochs	2	2
Pre-training steps	10000	10000
Backtracking line search iterations	10	10
Backtracking line search step	0.8	0.8
Conjugate gradient iterations	10	10
Discount γ	0.99	0.99
Max KL divergence	0.001	0.001
λ for GAE	0.97	0.97
Value function neurons	$[64]$	$[64]$
Value function iterations	3	3
Value function learning rate	0.01	0.01
Policy neurons	$[64, 64]$	$[64, 64]$
Activations	ReLU	ReLU
Encoder feed-forward neurons	$[8]$	$[8]$
Encoder learning rate	$5e-3$	$5e-3$
Encoder iterations	1	1
Encoder dimension	64	64
Encoder heads	2	2
Encoder optimizer	Adam $(\beta_1 = 0.9, \beta_2 = 0.999)$	Adam $(\beta_1 = 0.9, \beta_2 = 0.999)$
Batch size belief learning	10000	1000
Prediction buffer size	100000	100000
Belief representation dimension	8	32

Table 4: Hyper-parameters for L2-TRPO

Hyper-parameter	Pendulum	Mujoco
Epochs	2000	1000
Steps per epoch	5000	5000
Pre-training epochs	2	2
Pre-training steps	10000	10000
Backtracking line search iterations	10	10
Backtracking line search step	0.8	0.8
Conjugate gradient iterations	10	10
Discount γ	0.99	0.99
Max KL divergence	0.001	0.001
λ for GAE	0.97	0.97
Value function neurons	$[64]$	$[64]$
Value function iterations	3	3
Value function learning rate	0.01	0.01
Policy neurons	$[64, 64]$	$[64, 64]$
Activations	ReLU	ReLU

Table 5: Hyper-parameters for M-TRPO and A-TRPO

Hyper-parameter	Pendulum
Epochs	2000
Steps per epoch	5000
Discount γ	0.99
Eligibility trace λ	0.9
Learning rate	0.1
ϵ-greedy parameter	0.2
$\mathcal{S}$ discretization size	15
$\mathcal{A}$ discretization size	3

Table 6: Hyper-parameters for SARSA

Hyper-parameter	Pendulum
Epochs	2000
Steps per epoch	5000
Discount γ	0.99
Eligibility trace λ	0.9
Learning rate	0.1
ϵ-greedy parameter	0.2
$\mathcal{S}$ discretization size	15
$\mathcal{A}$ discretization size	3

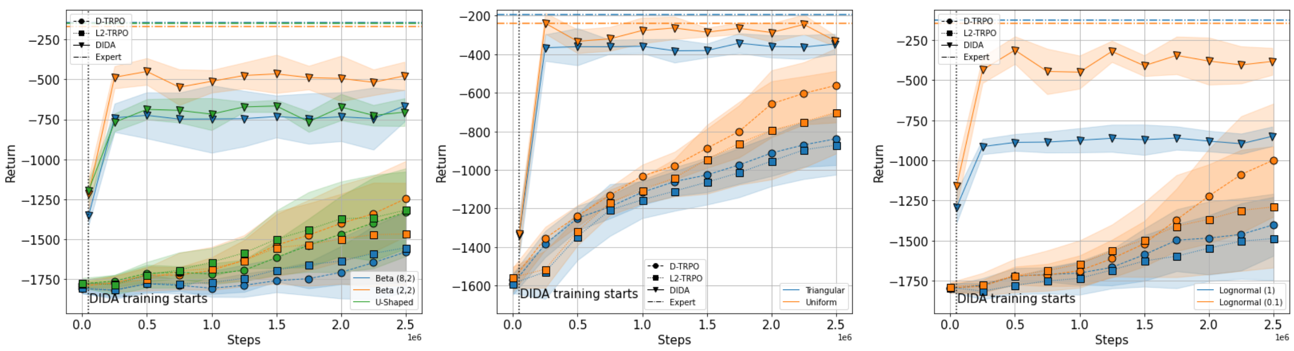
Table 7: Hyper-parameters for dSARSA

E.3. Further experiments

Stochastic Pendulum In this experiment, we evaluate DIDA on a stochastic environment. We follow Liotet et al. (2021) and add stochasticity to the pendulum environment. In order to do so, to the action selected by the agent, we add an i.i.d. noise of the form $\epsilon = \text{scale}(\eta + \text{shift})$ where η is some probability distribution. We construct 6 such noises reported in Table 8. For readability of the plots, we group noises by similarity. We build an additional noise, referred to as *uniform* noise, which follows the action of the agent with probability 0.9 and otherwise samples an action uniformly at random inside the action space. We place this noise in group 2. The results are obtained with the hyper-parameters given in Appendix E.2 for the pendulum environment.

Noise	Distribution η	Shift	Scale	Group
Beta (8,2)	$\beta(8, 2)$	0.5	2	1
Beta (2,2)	$\beta(2, 2)$	0.5	2	1
U-Shaped	$\beta(0.5, 0.5)$	0.5	2	1
Triangular	Triangular$(-2, 1, 2)$	0	1	2
Lognormal (1)	Lognormal$(0, 1)$	-1	1	3
Lognormal (0.1)	Lognormal$(0, 0.1)$	-1	1	3

Table 8: Distributions for the noise added to the action in the stochastic Pendulum.



(a) Noises of group 1.

(b) Noises of group 2.

(c) Noises of group 3.

Figure 6: Mean return and one standard deviation (shaded) as a function of the number of steps sampled from the environment for stochastic Pendulum with different noises (10 seeds).

- **Group 1**: In all these cases, we are considering noises based on beta distributions, with different parameters. We can see in Figure 6a that our algorithm is able to achieve a much better performance compared to the ones of the baselines, even with a fraction of the training samples. For every algorithm, the most favourable case seems to be the second one, based on a beta $(2, 2)$ noise. This may be because of the features of the other two noises. The first, beta $(8, 2)$, is non-zero mean, so that the action is affected, on average, by a translation in one direction. The third one, based on a $\beta(0.5, 0.5)$ distribution, is zero-mean, but is characterized by a higher variance than the second $(0.5$ vs $0.2)$.
- **Group 2**: Here again, we see in Figure 6b that DIDA is able to get the best performance, even if the two baselines seem to learn much faster than for group 1. This suggest that group 2 contains easier tasks, even though the triangular noise is not symmetric. Still, note that even if the probability of the random action in the first case is small, in the situation of delay it accumulates so that the probability of having a random action inside the action queue of $\Delta = 5$ is $1 - 0.9^5 \approx 0.41$. Nonetheless, DIDA seems to deal very well with this situation.
- **Group 3**: Being strongly asymmetric and unbounded, theses noises pose more challenge to the algorithms. We report the results in Figure 6c. For the Lognormal (1) noise, no algorithm reaches a satisfactory performance. However, again, DIDA obtains the best performance for each noise.