Temporally Robust Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

Reinforcement learning (RL) agents face two kinds of distribution shift at deployment: changing *physics* (mass, friction, dynamics) and changing *clock* (delays, missed observations, asynchronous control). Although robust RL addresses the first, the second remains much less addressed, with existing work mostly confined to restrictive i.i.d. delay models. In this work, we introduce decision-time-robust semi-Markov decision processes (DTR-SMDPs), where the interaction time between the agent and the environment is state-dependent and chosen adversarially from an uncertainty set. We prove that the general problem is NP-hard, and propose a rectangularity assumption under which a policy iteration scheme converges. Our central insight is an equivalence between rectangular DTR-SMDPs and a transition-robust MDP class, with a physically plausible uncertainty set, free of the impossible transitions that standard norm-ball sets admit. Within this class, resilience to bad timing *is* resilience to bad physics. This equivalence has a practical consequence: an agent can achieve dynamics generalization by training against an adversarial clock, with no simulator-side physics randomization. We develop a practical algorithm and empirically confirm both its temporal robustness and its zero-shot transfer to physical perturbations. Finally, we characterize which robust-RL baselines do and do not inherit temporal robustness, a split our theory predicts.

1 Introduction

The safe and reliable deployment of reinforcement learning (RL) agents requires accounting for the discrepancy between the training simulation (sim) and the testing environment (real). For example, a robot trained under specific friction parameters may fail under changing surfaces or varying payloads. By design, robust RL addresses these modeling errors and external disturbances by provably maintaining performance when transition dynamics or rewards deviate from the nominal model (Tessler et al., 2019; Ashlag et al., 2025; Pinto et al., 2017).

Several axes of perturbation have been studied in robust RL: transition dynamics (Pinto et al., 2017), reward functions (Gadot et al., 2024), observations (Zhang et al., 2020), and actions (Tessler et al., 2019). In comparison, temporal perturbations have been underexplored. A detailed discussion of related work is deferred to Appx. A. Delayed and asynchronous MDPs extend RL to non-instantaneous interactions, but assume a fixed delay protocol (Derman et al., 2021; Valensi et al., 2024; Bouteiller et al., 2021). These known timing models do not protect the agent against mismatches between training and deployment. Alternative approaches rely on heuristics such as domain randomization (Bohlinger et al., 2024; Imai et al., 2022; Seo et al., 2025), providing no worst-case guarantee. Furthermore, even works that model temporal considerations typically treat the delay process as a state-independent, i.i.d. noise applied uniformly across the state space. Unfortunately, real-world delays systematically violate this assumption. For example, in chronic disease management, the time between clinical measurements depends on the patient's state: a patient is more likely to miss a check-up precisely when their condition deteriorates and monitoring matters most. Such state-correlated delays produce systemic vulnerabilities that an i.i.d. model cannot capture.

To address these gaps, we introduce *decision-time-robust semi-Markov decision processes* (DTR-SMDPs), in which the interaction time between agent and environment is state-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

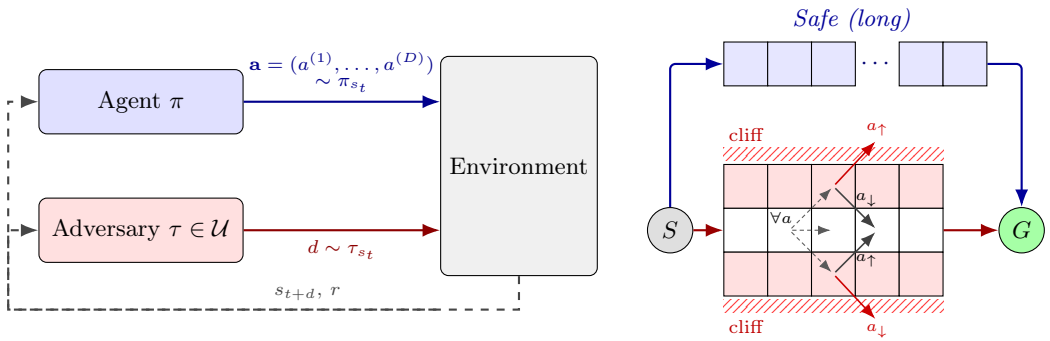

Figure 1: **(a) DTR-SMDP setup.** At decision time t in state s_t, the agent emits an action chunk $\mathbf{a} = (a^{(1)}, \ldots, a^{(D)}) \sim \pi_{s_t}$ and the adversary draws a holding time $d \sim \tau_{s_t}$, where the timing rule τ is chosen from an uncertainty set $\mathcal{U}$. The environment executes the first d actions of the chunk and returns s_{t+d} together with the cumulative reward over that epoch. **(b) Illustrative example.** Two routes reach the goal: a long safe corridor (top) and a short three-lane tube between two cliffs (bottom). The agent has two actions, $a_\uparrow$ and $a_\downarrow$; dashed arrows mark stochastic transitions, solid ones deterministic. Under a deterministic clock $\tau \equiv 1$, the short tube is safe: the agent sees each drift and steers back on the next step, so that route wins. Under any uncertain τ, it must commit to an action chunk while the drift goes unobserved, and the committed step may carry it over the cliff. Small *physical* uncertainty in the edge-lane transitions fails the same way, so DTR-trained and dynamics-robust policies both prefer the corridor, while nominal-optimal ones take the tube.

dependent and chosen adversarially from a fixed uncertainty set (see Fig. 1(a)). Rather than committing to a single delay protocol, this formulation guarantees worst-case performance over an entire set of clocks, and captures state-dependent temporal patterns by construction. Moreover, we prove that resilience to adversarial timing translates into robustness against a physically interpretable class of dynamics perturbations, and can therefore serve as a cheap alternative to simulator-side physics randomization.

Our contributions are organized as follows. Sec. 3 formally introduces our DTR framework and provides a theoretical analysis of its properties. **Hardness and tractability.** In Sec. 3.1, we show that the problem is NP-hard for general uncertainty sets, highlighting the intrinsic difficulty of temporal robustness. To render the problem tractable, we adapt rectangularity from robust MDPs to the temporal setting, assume the adversary's delay choices are independent across states, and propose a provably converging policy iteration scheme in Sec. 3.2. **Worst-case adversaries.** We characterize the worst-case temporal perturbations in Sec. 3.3, where we show that optimal adversaries are "sparse", assigning positive probability to at most two interaction-time values. This theoretical insight directly motivates our adversarial architectures and worst-case experimental design. **Equivalence to robust MDPs.** Sec. 3.4 establishes a formal equivalence between rectangular DTR and a class of transition-robust MDPs. The uncertainty sets resulting from this equivalence are interpretable and significantly easier to design than standard transition-kernel uncertainty sets. **Algorithm and experiments.** In Sec. 4, we develop a practical algorithm to solve DTR problems. We empirically demonstrate that DTR agents achieve superior performance against worst-case temporal perturbations while maintaining competitive nominal returns. Finally, we test both directions: DTR agents inherit zero-shot robustness to dynamics perturbations they never train on, and transition-robust agents inherit temporal robustness to the extent that their uncertainty sets are built from nominally reachable transitions.

2 Preliminaries

Markov decision processes. A Markov decision process (MDP) is defined by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ where $\mathcal{S}$ and $\mathcal{A}$ denote finite state and action spaces, $P : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is the transition kernel, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, and $\gamma \in [0, 1)$ the discount factor

(Puterman, 1994). Throughout the paper, a state or an action in subscript indicates that this argument is held fixed; for instance, $P_a = P(\cdot|\cdot, a) \in (\Delta_{\mathcal{S}})^{\mathcal{S}}$ and $R_a = R(\cdot, a) \in \mathbb{R}^{\mathcal{S}}$. A policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ maps states to action probabilities. We consider two standard performance criteria: (i) The *discounted return*: $J_\gamma(\pi, P, R) := \mathbb{E}_P^\pi \left[\sum_{t=0}^\infty \gamma^t R(s_t, a_t)\right] = \langle d_\gamma^\pi, R\rangle$, where $d_\gamma^\pi(s, a) = (1-\gamma)\sum_{t=0}^\infty \gamma^t \mathbb{P}_P^\pi(s_t = s, a_t = a)$ is the discounted occupancy measure; (ii) The *average return*: $J_{\text{avg}}(\pi, P, R) := \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_P^\pi \left[\sum_{t=0}^T R(s_t, a_t)\right] = \langle d^\pi, R\rangle$. Under standard ergodicity assumptions, the average return is well-defined and independent of the initial state, with $d^\pi(s, a) = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^T \mathbb{P}_P^\pi(s_t = s, a_t = a)$ the unique stationary state-action distribution (Puterman, 1994). In the remainder of this paper, we omit the subscript and use J to denote the return when the distinction between the two criteria is irrelevant.

Robust MDPs. In robust MDPs, the transition kernel P and reward function R are unknown but lie in an *uncertainty set*, namely, $(P, R) \in \mathcal{U}$ (Wiesemann et al., 2013). There, the agent seeks to maximize performance under the worst model realization within this set, namely, solve $\max_\pi J^{\mathcal{U}}(\pi)$, where $J^{\mathcal{U}}(\pi) := \min_{(P,R)\in\mathcal{U}} J(\pi, P, R)$. Solving this max-min optimization problem is generally NP-hard (Bagnell et al., 2001). To render the problem tractable, it is standard to assume the uncertainty set is *rectangular* (Wiesemann et al., 2013): $\mathcal{U} = \bigtimes_{s\in\mathcal{S}} \mathcal{U}_s$ (s-rectangularity). This assumption keeps the model uncertainty local and admits robust analogues of the Bellman operators that can be solved efficiently.

Semi-Markov decision processes. A semi-Markov decision process (SMDP) generalizes an MDP by letting actions span multiple timesteps. Given a base MDP that evolves in discrete time $t = 1, 2, \ldots$, the agent in the SMDP acts only at a subset of timesteps called *decision times*. If the k-th decision time occurs at state $s_k \in \mathcal{S}$, the number of base timesteps until the next decision, the *transition time*, is drawn from $\tau_{s_k} \in \Delta_{[D]}$, where $\tau : \mathcal{S} \to \Delta_{[D]}$ is the transition-time process and D bounds the interval between consecutive decisions. At each decision time, the agent selects a temporally extended action, which in this work takes one of two forms: *(i)* **Action chunking.** The agent commits to an action sequence $\mathbf{a} = (a^{(1)}, \ldots, a^{(D)}) \in \mathcal{A}^D$, resulting in a policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}^D}$. The base-MDP effectively executes the first $d \sim \tau$ actions $(a^{(1)}, \ldots, a^{(d)})$ until the next decision-time, where the agent updates its augmented action commitment; denote by Π^{AC} the corresponding policies. *(ii)* **Action repetition.** The agent selects a single primitive action $a \in \mathcal{A}$, which it repeats until the next decision time. We denote the corresponding policy space by Π^{AR}. Clearly, $\Pi^{\text{AR}} \subseteq \Pi^{\text{AC}}$. In both cases, during a transition time d, the state evolves according to the base transition kernel P and the executed primitive actions. Consequently, the expected return in the SMDP setting depends on the transition-time process as well; we write it $J(\pi, P, R, \tau)$, abbreviated $J(\pi, \tau)$ wherever the base model (P, R) is fixed.

3 Decision-Time-Robust SMDPs

We introduce decision-time-robust semi-MDPs (DTR-SMDPs), where instead of following a known distribution, the transition time process is adversarially selected within an uncertainty set $\mathcal{U}$. Formally, $\mathcal{U}$ is a set of mappings $\tau : \mathcal{S} \to \Delta_{[D]}$. Then, we aim to find a policy that maximizes the expected return under the worst-case decision-time process, which formalizes as a solution to the max-min objective: $\max_\pi \min_{\tau\in\mathcal{U}} J(\pi, \tau)$. We establish in Sec. 3.1 that similarly to robust MDPs, the above problem is NP-hard. We introduce a rectangularity assumption that renders the problem tractable, and propose a convergent policy iteration scheme in Sec. 3.2. The structure of an optimal adversary is characterized in Sec. 3.3. Finally, we show a formal equivalence between DTR-SMDPs and transition-kernel-robust MDPs in Sec. 3.4.

3.1 General DTR-SMDPs are NP-Hard

A natural question arising from the DTR-SMDP framework is whether the problem is tractable for general uncertainty sets. As we now demonstrate, even the primary task of policy evaluation can be NP-hard.

Theorem 1 (Informal)**.** *Approximating the DTR-SMDP problem within any constant factor is generally NP-Hard.*

The full proof is detailed in Appx. B.1 and employs a similar technique to Wiesemann et al. (2013). We construct a polynomial-time reduction from the NP-hard problem of integer programming. The intuition is that an unrestricted adversary can couple its perturbations across states, encoding hard combinatorial constraints. Temporal robustness is therefore infeasible without structural assumptions, even in the relatively constrained SMDP setting where observations arrive fresh at each decision time, as opposed to more general delayed MDPs. This motivates imposing rectangularity: the adversary must choose independently across states.

3.2 Policy iteration in rectangular DTR-SMDPs

Building upon semi-Markov games, we aim to show that rectangular DTR-SMDPs can be solved using policy iteration. Therefore, in the remainder of this study, we focus exclusively on s-rectangular uncertainty sets, and denote $\mathcal{U}_D := \bigtimes_{s\in\mathcal{S}} \mathcal{U}_s$, where $\mathcal{U}_s$ contains valid transition-time distributions $\tau_s \in \Delta_{[D]}$.

In addition to the uncertainty set being rectangular, we wish to: (i) Prevent the adversary from trivially applying the maximum transition time D at every step; (ii) Enable rigorous comparisons against natural baselines that model observation drops exogenously from the executing agent. Performance gains over such baselines would then reflect robustness to adversarial timing rather than a mere difference in the total delay incurred. A convenient way to achieve this double objective is to impose a fixed-mean constraint on the transition time process. Specifically, we define:

$$\mathcal{U}_{c,D} := \bigtimes_{s\in\mathcal{S}} \mathcal{U}_{c,D,s}, \qquad \mathcal{U}_{c,D,s} := \{\tau_s \in \Delta_{[D]} : \mathbb{E}_{d\sim\tau_s}[d] = c\}.$$

While we use a scalar c for convenience, our analysis generalizes with minimal modification to a state-dependent mean delay c_s for every state $s \in \mathcal{S}$. We now define several quantities. For a fixed interaction time $d \in [D]$, the expected cumulative discounted return obtained from state $s \in \mathcal{S}$ under policy $\pi \in \Pi^{\mathrm{AC}}$, and the transition distribution over this time period are respectively:

$$\bar{R}^\pi(s,d) := \mathbb{E}^\pi\left[\textstyle\sum_{t=0}^{d-1} \gamma^t R(s_t,a_t) \;\middle|\; s_0 = s\right],$$
$$\bar{P}^\pi_{(d)}(s'|s) := \mathbb{P}^\pi(s_d = s' \mid s_0 = s) = \sum_{\mathbf{a}\in\mathcal{A}^D} \pi(\mathbf{a}|s) P_{a^{(1)}} P_{a^{(2)}} \cdots P_{a^{(d)}}(s'|s),\ \forall (s',d) \in \mathcal{S}\times[D].$$

These enable us to define the Bellman backup T^π for a fixed state-delay pair, the robust evaluation operator B^π that integrates out the adversarial transition time, and the robust Bellman optimality operator T that maximizes over policies:

$$[T^\pi V](s,d) := \bar{R}^\pi(s,d) + \gamma^d \textstyle\sum_{s'\in\mathcal{S}} \bar{P}^\pi_{(d)}(s'|s) V(s'), \qquad (s,d) \in \mathcal{S}\times[D],$$
$$[B^\pi V](s) := \min_{\tau_s \in \mathcal{U}_{c,D,s}} \textstyle\sum_{d=1}^D \tau_s(d)[T^\pi V](s,d), \qquad [TV](s) := \max_{\pi\in\Pi^{\mathrm{AC}}} [B^\pi V](s).$$

A first key step toward solving DTR-SMDPs is to establish the contracting properties of T, which directly leads to converging value iteration.

Proposition 1. *The optimal Bellman DTR operator is a γ-contraction under the supremum-norm. It thus admits a unique fixed point denoted by V^*.*

All proofs in this subsection are deferred to Appx. B.2. We now present a policy-iteration scheme that solves DTR-SMDPs, called DTR-PI. As shown in Algo. 1, DTR-PI proceeds in two stages, similarly to zero-sum Markov games: (i) Given a fixed policy, compute the worst-case return; (ii) Improve the current policy by acting greedily with respect to the computed return. The following result shows that DTR-PI converges to the optimum.

Theorem 2. *Denote $V_k := V^{\pi^k}$ for all $k \in \mathbb{N}$ in Algo. 1. Then, V_k converges towards V^* at rate γ, that is: $\|V_k - V^*\|_\infty \le \gamma \|V_{k-1} - V^*\|_\infty$.*

Under the additional fixed-mean constraint, the adversary can no longer concentrate its mass on the shortest delays and the contraction improves accordingly. This is formalized next.

Proposition 2. *Let $D \geq 2$ and $c \in [1, D]$. Define the worst-case effective discount*

$$\tilde{\gamma}_{c,D} := \max_{\tau_s \in \mathcal{U}_{c,D,s}} \mathbb{E}_{d \sim \tau_s}\left[\gamma^d\right] = \frac{(D-c)\,\gamma + (c-1)\,\gamma^D}{D-1} \in [\gamma^c, \gamma]\,.$$

Then, the operator T is a $\tilde{\gamma}_{c,D}$-contraction, and DTR-PI converges at rate $\tilde{\gamma}_{c,D}$, with $\tilde{\gamma}_{c,D} < \gamma$ whenever $c > 1$; namely: $\|V_k - V^*\|_\infty \leq \tilde{\gamma}_{c,D}\|V_{k-1} - V^*\|_\infty$.

3.3 Characterizing worst-case temporal perturbations

Now that we have established that DTR-SMDPs are well-defined and solvable through robust policy iteration, we analyze the structure of the worst-case adversarial perturbations. Understanding this structure is crucial, as it directly motivates the worst-case temporal evaluations and architectural designs we introduce in Sec. 4.

Algorithm 1 DTR-PI

Initialize: $k = 0,\ \pi^k = \pi^0$
repeat
  Evaluate: Solve $V^{\pi^k} = B^{\pi^k} V^{\pi^k}$
  Improve: $\pi^{k+1} \in \arg\max_\pi B^\pi V^{\pi^k}$
  $k \leftarrow k + 1$
until no change
Return: π^k

We begin our analysis by focusing on the simpler case of the unconstrained rectangular uncertainty set $\mathcal{U}_D$. It is a foundational result in game theory that zero-sum games generally require stochastic (mixed) policies to guarantee a well-defined value. Interestingly, and similarly to standard MDPs, when the temporal adversary is bounded only by the unconstrained set $\mathcal{U}_D$, a corollary of our previous convergence analysis is that the optimal adversary collapses to a purely deterministic strategy. All proofs in this subsection are in Appx. B.3.

Corollary 1. *For unconstrained uncertainty sets $\mathcal{U}_D$, there exists an optimal adversarial transition time τ^* that is deterministic. That is, for every state $s \in \mathcal{S}$, the adversary selects a single, discrete transition time $d \in [D]$ with probability 1.*

Introducing the fixed-mean constraint under $\mathcal{U}_{c,D}$, however, fundamentally alters the adversary's optimal strategy. Due to this constraint, we cannot generally expect the worst-case policy to remain deterministic. Nevertheless, we can bound the complexity of this stochastic behavior, guaranteeing the existence of an optimal adversary whose support is limited to at most two transition times at any given state. More specifically:

Corollary 2. *Under the fixed-mean constrained uncertainty set $\mathcal{U}_{c,D}$, there exists an optimal adversarial policy τ^* such that for all states $s \in \mathcal{S}$, the support of τ_s^* has a cardinality of at most two. Specifically, τ_s^* either selects the mean c with probability 1, or is a Bernoulli distribution supported on two distinct values $d_1 < c < d_2$, with a parameter p satisfying the mean constraint.*

The proof hinges on casting the adversary's per-state problem as a linear program, then invoking the Fundamental Theorem of Linear Programming.

3.4 DTR-SMDPs are equivalent to robust MDPs

While the DTR-SMDP framework is fundamentally designed to address temporal uncertainty, we now demonstrate that it can also be interpreted as a specific instance of a transition-kernel robust MDP in the original state and action spaces. This equivalence provides a formal connection between resilience to "bad timing" and robustness to "bad physics", a connection foreshadowed by a shared geometric intuition: both robustness regimes reward trajectories whose *neighbors* are also high-value, since under either kind of perturbation the agent's realized trajectory drifts from the nominal one and the policy pays for whatever lies adjacent. Fig. 1(b) illustrates this on a minimal navigation example, where the risky route's neighbors include cliffs and both DTR and dynamics-robust policies prefer the safe corridor.

To clearly introduce this equivalence, we focus our analysis on the average reward criterion and the action repetition setting. We note that this duality also holds for the more complex

action chunking setting and the discounted return criterion, which we analyze in Appx. B.4. However, we focus here on analyzing the average reward under action repetition as we believe this yields uncertainty sets that are highly interpretable and cleanly isolate the structure of the equivalence between temporal and transition robustness. Here $\bar{R}(s,a,d) := \mathbb{E}[\sum_{t=0}^{d-1} R(s_t,a_t) \mid s_0 = s,\, a_t \equiv a]$ is the reward accumulated while a is repeated for d steps: the quantity $\bar{R}^\pi$ of Sec. 3.2 with π fixed to the repeated action and, as befits the average-reward criterion, undiscounted.

Theorem 3. *For $p \in \Delta_{[D]}$, let $\tilde{P}_p(\cdot|s,a) := \sum_{d=1}^{D} p_d\,(P_a)^d(\cdot|s)$ and $\tilde{R}_p(s,a) := \frac{1}{c}\sum_{d=1}^{D} p_d\,\bar{R}(s,a,d)$, and define $\mathcal{U} := \bigtimes_{s\in\mathcal{S}} \mathcal{U}_s$ where*

$$\mathcal{U}_s := \Big\{ \big(\tilde{P}_p(\cdot|s,\cdot),\, \tilde{R}_p(s,\cdot)\big) \;\Big|\; p \in \Delta_{[D]},\; \mathbb{E}_{d\sim p}[d] = c \Big\}.$$

If $\pi^* \in \Pi^{AR}$ is optimal for the DTR-SMDP under average reward, then it is also optimal in the robust MDP defined by the above uncertainty set $\mathcal{U}$, that is, $\pi^* \in \arg\max_\pi J^{\mathcal{U}}_{avg}(\pi)$.

The proof, deferred to Appx. B.4, marginalizes the SMDP dynamics over decision times to construct the effective kernel. We denote by $\mathcal{P}_s$ and $\mathcal{R}_s$ the marginal projections of $\mathcal{U}_s$ onto its kernel and reward components.

This mathematical equivalence exposes three critical advantages of our framework: (i) **Physically plausible uncertainty:** The sets $\mathcal{P}_s$ make the neighborhood intuition above precise: each effective kernel is a convex combination of multi-step transitions $(P_a)^d$ for $d \le D$, so perturbations are confined to states the dynamics can actually reach within D timesteps. This elegantly addresses the conservatism problem inherent in standard Robust RL (Kumar et al., 2025), where abstract distance measures (e.g., L_p norm balls) force the agent to guard against mathematically valid but physically impossible "jumps" in the state space. (ii) **Simulation simplicity:** In standard robust RL, designing uncertainty sets or engineering complex physical perturbations (e.g., varying mass, friction, or aerodynamics) in a simulator is labor-intensive and prone to sim-to-real gaps. Our equivalence proves that an agent can achieve meaningful dynamics generalization simply by training against an adversarial "clock". Varying timing intervals requires virtually zero simulation engineering, offering a computationally cheap and highly effective path to robust control. (iii) **Tractable rectangularity:** The fixed-mean constraint c ensures that the resulting uncertainty sets factorize across states, maintaining s-rectangularity. This simplifies the optimization landscape and allows for fair, direct theoretical and empirical comparisons against classical robust RL methods.

Finally, we note that this perspective offers a rigorous theoretical lens for understanding the empirical success of common practices such as delay randomization. While a DTR-SMDP optimizes worst-case performance over the uncertainty set $\mathcal{P}_s$, delay randomization optimizes the average performance over this exact same set. Crucially, both methods prevent the policy from overfitting to the nominal transition kernel P. By forcing the agent to be resilient to the state-blurring inherent in $\mathcal{P}_s$, delay randomization acts as a form of temporal data augmentation, inducing implicit robustness to perturbed dynamics that share similar structural properties.

4 Experiments

We evaluate the DTR-SMDP framework on continuous control benchmarks to test three hypotheses: (i) *temporal resilience* — our method achieves strong performance under worst-case timing perturbations while remaining competitive on the nominal clock; (ii) *forward duality* — temporally robust agents inherit zero-shot robustness to the *physically plausible* perturbations as identified in Thm. 3; and (iii) *reverse duality* — conversely, a transition-robust agent inherits temporal robustness to the extent that its own uncertainty set is constructed from transitions realizable by the nominal model.

4.1 Experimental Setup

Domains, methods, and base algorithm. We use four MuJoCo continuous control environments: HalfCheetah, Walker2d, Hopper, and InvertedPendulum (Todorov et al., 2012).

All agents share a single-file CleanRL (Huang et al., 2022) PPO backbone (Schulman et al., 2017) at its default hyperparameters and the same compute budget, isolating performance differences to the training objective; each worst-case baseline's robustness radius was swept separately. While our framework naturally accommodates action chunking, we focus on *action repetition* as the SMDP mechanism: it is standard in the asynchronous control literature and admits a fair, direct comparison against robust-RL baselines, which assume Markovian policies. Appx. C.5 reports a diffusion action-chunking variant, whose results are qualitatively similar to those presented below in Sec. 4.2. We fix the expected transition time to $c = 2$ and the maximum to $D = 4$ across all delayed methods. Appx. C.3 sweeps both, finds the conclusions of Sec. 4.2 to hold throughout, and isolates the effect of misspecifying the uncertainty set itself. We compare ten methods. Five vary only the clock: **Vanilla PPO** ($\tau \equiv 1$), **Fixed delay** ($\tau \equiv c$), **Uniform delay** ($\tau \sim \mathrm{Unif}\{1, \ldots, 2c-1\}$), and our two adversarial-clock variants **DTR (single-head)** and **DTR (two-head)**. The other five are robust-RL baselines, chosen to span distinct uncertainty-set structures. **PR-MDP** hands control to an adversary with probability $\alpha = 0.1$; its uncertainty set is an α-mixture of kernels the nominal simulator itself executes (Tessler et al., 2019, Sec. 3.1), hence physically plausible, like $\mathcal{P}_s$. **RNAC-IPM** (Zhou et al., 2023) is transition-robust and (s, a)-rectangular, but its set is a metric ball in the critic's feature geometry: abstract proximity rather than realizable dynamics. **SA-PPO** (Zhang et al., 2020) is observation-robust, perturbing the policy's input rather than the kernel. **DR-seen** and **DR-unseen** (Peng et al., 2018) randomize simulator physics per episode, over the axes perturbed at evaluation and over held-out ones respectively. DR-seen is in-distribution for Sec. 4.3 and is an upper reference rather than a zero-shot comparison; we mark it † throughout. Aggregates are IQMs with stratified bootstrap CIs (Agarwal et al., 2021). Training and evaluation details are in Appx. C.7; hyperparameters and per-baseline implementation details are in Appx. C.8.

Adversary implementation. We model the temporal adversary as a secondary policy trained jointly with the agent via PPO (Algo. 2). Because the game is zero-sum, the agent and adversary share critic networks and advantage estimates. The central design challenge is enforcing the mean-delay constraint $\mathbb{E}_{d\sim\tau_s}[d] = c$ at every state. The *single-head* variant relaxes this into a soft penalty $\lambda\,(\mathbb{E}_{d\sim\tau_s}[d] - c)^2$ on the adversary's loss, with a single coefficient λ tuned once on HalfCheetah and held fixed across environments (full tuning protocol in Appx. C.8). Note that we calculate this penalty using the exact expectation over the adversary's output distribution rather than discrete samples, avoiding high-variance gradient estimates. The *two-head* variant, motivated by Cor. 2 (optimal adversaries have support of size at most two), splits the adversary into two heads outputting distributions τ_s^L and τ_s^H over $[1, c]$ and $[c+1, D]$ respectively, and combines them as $\tau_s = \alpha_s \tau_s^L + (1 - \alpha_s)\tau_s^H$, where $\alpha_s \in [0, 1]$ is computed in closed form from the head outputs to enforce $\mathbb{E}_{d\sim\tau_s}[d] = c$ exactly. The two-head construction yields exact per-state constraint satisfaction and removes the need to tune the λ hyperparameter entirely. Fig. 3 verifies this empirically: the two-head's expected delay tracks $c = 2$ exactly throughout training on Walker2d, while the single-head exhibits a small but consistent positive bias. To our knowledge, the two-head architecture is novel, and we expect it to be useful beyond DTR to other problems with fixed-mean constrained policies. Cor. 2 also yields a parameter-free prediction at $c = 2$, $D = 4$, where the mean constraint leaves exactly three admissible per-state distributions: δ_2, $\mathrm{Bern}(1, 3)$, and the extremal $\mathrm{Bern}(1, 4)$ (where δ is a Dirac mass). Indeed, Appx. C.2 finds that at 87.7% of visited states — for both architectures — the trained adversary places at least 90% of its mass on one of these three, most often the extremal one.

4.2 Robustness to Temporal Perturbations

We first test the core methodological claim: training against an adversarial clock should produce policies that resist worst-case timing without sacrificing nominal performance. For each agent we report two metrics: the *nominal* return under $\tau \equiv 1$, and the *worst-case* return, defined as the per-seed minimum across three evaluation distributions sharing the same mean delay $c = 2$: a constant delay ($\tau \equiv 2$), a uniform distribution on $\{1, 2, 3\}$, and a boundary distribution alternating between $d = 1$ and $d = 4$ to maintain $\mathbb{E}[d] = 2$. The boundary distribution is motivated by Cor. 2, which shows that worst-case adversaries are

Table 1: Aggregate temporal robustness over 4 environments × 20 seeds: interquartile mean (IQM) of min–max normalized scores and probability of improvement, with 95% CIs in brackets (Appx. C.7). The two DTR variants lie within each other's intervals and both are marked. Per-environment returns are in Appx. C.1.

	Nominal	Worst case		
Method	IQM	IQM	$P(\succ$ Vanilla)	P(DTR two-head $\succ$)
Vanilla PPO	0.628 [0.576, 0.683]	0.075 [0.059, 0.094]	—	1.000 [1.000, 1.000]
Fixed delay	0.715 [0.654, 0.773]	0.176 [0.155, 0.197]	0.825 [0.762, 0.884]	0.989 [0.972, 0.999]
Uniform delay	**0.808 [0.765, 0.843]**	0.397 [0.372, 0.424]	1.000 [1.000, 1.000]	0.880 [0.818, 0.936]
PR-MDP	0.510 [0.452, 0.569]	0.185 [0.164, 0.207]	0.898 [0.847, 0.942]	0.998 [0.992, 1.000]
SA-PPO	0.488 [0.458, 0.521]	0.113 [0.099, 0.128]	0.469 [0.393, 0.542]	1.000 [1.000, 1.000]
RNAC-IPM	0.625 [0.573, 0.679]	0.079 [0.065, 0.097]	0.420 [0.339, 0.501]	1.000 [1.000, 1.000]
DR-unseen	0.481 [0.438, 0.524]	0.097 [0.078, 0.118]	0.406 [0.327, 0.484]	1.000 [1.000, 1.000]
DR-seen†	0.285 [0.236, 0.339]	0.115 [0.098, 0.133]	0.652 [0.568, 0.736]	1.000 [1.000, 1.000]
DTR (single-head)	0.705 [0.648, 0.758]	**0.727 [0.697, 0.756]**	1.000 [1.000, 1.000]	0.505 [0.414, 0.596]
DTR (two-head)	0.770 [0.712, 0.817]	**0.723 [0.691, 0.758]**	1.000 [1.000, 1.000]	—

supported on at most two values and concentrate mass at the extremes of the temporal budget.

Both DTR variants attain by far the highest interquartile mean (IQM ≈ 0.72) against the strongest baseline, Uniform delay ($0.397\,[0.372, 0.424]$), with $P(\text{DTR two-head} \succ \text{Uniform}) = 0.880\,[0.818, 0.936]$ and $P(\succ \text{Vanilla}) = 1.000$ for both. Among non-DTR methods, Uniform delay is consistently the strongest, indicating that delay-randomization training already provides meaningful temporal resilience. The benefits are most pronounced on Inverted-Pendulum, where every non-DTR baseline collapses to near-zero returns under worst-case timing while DTR agents continue to solve the task, which we attribute to the task's stability-critical nature, as a single mistimed action can be unrecoverable. We note that all three evaluation schedules are fixed in advance and applied independently of the state, whereas DTR-SMDPs are defined by *state-dependent* timing. Appx. C.4 therefore reports a complementary evaluation against an adversary trained from scratch on each frozen checkpoint, drawn from the same $\mathcal{U}_{c,D}$ for every method; the picture above is unchanged (IQM 0.641 and 0.630 for the two DTR variants, against Uniform's 0.329 and Vanilla's 0.055). This comes at a modest nominal cost: DTR's nominal IQM ($0.770\,[0.712, 0.817]$) is second only to Uniform delay ($0.808\,[0.765, 0.843]$), from which it is statistically indistinguishable ($P = 0.430\,[0.356, 0.506]$), and above Vanilla PPO ($0.628\,[0.576, 0.683]$).

Sec. 3.4 predicts which robust methods should inherit temporal robustness: a robust agent transfers to the extent that its uncertainty set is built from transitions the nominal dynamics actually produce. All five robust-RL baselines are consistent with this prediction. Although PR-MDP never trains on delayed transitions, it reaches $0.185\,[0.164, 0.207]$ against Vanilla's $0.075\,[0.059, 0.094]$, with $P(\succ \text{Vanilla}) = 0.898\,[0.847, 0.942]$ — its set differs from ours only in the mixture index: which action, rather than for how long. The two whose sets are not drawn from the nominal dynamics do not transfer, neither improving on Vanilla: SA-PPO (0.113, $P = 0.469\,[0.393, 0.542]$) and RNAC-IPM ($0.079$, $P = 0.420\,[0.339, 0.501]$). The two domain-randomization baselines form a controlled pair: they share the average-case objective and differ only in the axes they randomize, yet DR-seen† shows a small but significant effect (0.115, $P = 0.652\,[0.568, 0.736]$) and DR-unseen none ($0.097$, $P = 0.406\,[0.327, 0.484]$). Physics randomization thus does not reliably confer temporal robustness: what benefit it gives depends on the axes randomized, since a physical perturbation need not yield transitions the nominal dynamics actually produce.

4.3 Robustness to Dynamics Perturbations

We now test the practical consequence of Thm. 3: Training against an adversarial clock confers *zero-shot* robustness to physical dynamics perturbations the agent never sees during training. We evaluate on two perturbation benchmarks: *action noise* ($\alpha \in \{0.2, 0.3, 0.5\}$), where the executed action is replaced with a uniformly random action with probability α and *gravity scaling* ($g \in \{1.25, 1.5, 2.0\}$) across HalfCheetah, Hopper, and Walker2d. Per-cell numbers across all environments and methods are reported in Appx. C.1. We omit InvertedPendulum: under gravity scaling performance saturates at the 1000-step cap, while

Table 2: Aggregate dynamics robustness over 18 cells: HalfCheetah, Hopper and Walker2d, each under three action-noise and three gravity levels. Metrics and CIs as defined in Appendix C.7. **Bold** is the best per IQM column among the zero-shot methods; † DR-seen is in-distribution for all 18 cells — an upper reference, excluded from the bolding.

Method	Aggregate IQM (18)	Action-noise IQM (9)	Gravity IQM (9)	P(DTR two-head $\succ$)
Vanilla PPO	0.202 [0.183, 0.221]	0.289 [0.265, 0.312]	0.098 [0.080, 0.119]	0.773 [0.717, 0.824]
Fixed delay	0.179 [0.155, 0.206]	0.254 [0.233, 0.275]	0.084 [0.069, 0.104]	0.828 [0.778, 0.874]
Uniform delay	0.284 [0.261, 0.308]	0.364 [0.343, 0.386]	0.159 [0.137, 0.184]	0.623 [0.546, 0.698]
PR-MDP	**0.443 [0.411, 0.480]**	**0.515 [0.493, 0.543]**	**0.268 [0.224, 0.323]**	0.314 [0.250, 0.380]
SA-PPO	0.233 [0.220, 0.247]	0.284 [0.268, 0.301]	0.149 [0.135, 0.165]	0.752 [0.687, 0.812]
RNAC-IPM	0.213 [0.195, 0.232]	0.294 [0.269, 0.319]	0.098 [0.082, 0.117]	0.740 [0.682, 0.795]
DR-unseen	0.342 [0.316, 0.370]	0.410 [0.383, 0.438]	0.236 [0.207, 0.271]	0.469 [0.397, 0.542]
DR-seen†	0.473 [0.422, 0.526]	0.565 [0.518, 0.611]	0.344 [0.284, 0.415]	0.307 [0.241, 0.373]
DTR (single-head)	0.331 [0.306, 0.357]	0.419 [0.395, 0.441]	0.205 [0.180, 0.231]	0.481 [0.400, 0.563]
DTR (two-head)	0.343 [0.321, 0.366]	0.431 [0.415, 0.445]	0.203 [0.181, 0.227]	—

under action noise no method survives past step 135, so neither family separates the methods. The two DTR variants are statistically indistinguishable on this axis, as they are on the temporal one ($0.331\,[0.306, 0.357]$ and $0.343\,[0.321, 0.366]$, $P = 0.481\,[0.400, 0.563]$); we quote the two-head variant throughout.

Aggregating per-seed scores over all 18 cells, DTR attains an IQM of $0.343\,[0.321, 0.366]$ against Vanilla PPO's $0.202\,[0.183, 0.221]$ — a 70% relative improvement with non-overlapping intervals and $P(\text{DTR} \succ \text{Vanilla}) = 0.773\,[0.717, 0.824]$ — confirming the dynamics-generalization benefits of training against an adversarial clock. Furthermore, it places third of the ten methods, and second among those evaluated zero-shot, ahead of the other temporal regimes (Fixed delay 0.179, Uniform delay 0.284) and of the robust-RL baselines SA-PPO (0.233) and RNAC-IPM (0.213), and in a statistical tie with DR-unseen (0.342, $P = 0.469\,[0.397, 0.542]$) — physics domain randomization over held-out axes — despite receiving no simulator-side physics perturbation during training. PR-MDP is higher still (0.443). We read the pair as mirror images rather than as a ranking: each inherits robustness along the axis it never trains on — DTR observes no physics perturbation, PR-MDP no delay — and each is stronger along the axis it does train on, DTR by 0.723 to 0.185 in Table 1 and PR-MDP by 0.443 to 0.343 here. Highest of all is DR-seen† (0.473), which randomizes over the very axes it is evaluated on: an upper reference, not a zero-shot comparison.

5 Conclusion

We introduced DTR-SMDPs to bridge the gap between robust RL and temporal uncertainty, modeling interaction intervals as state-dependent adversarial delays instead of average-case noise. We proved that achieving temporal robustness is intrinsically NP-Hard, but that a rectangularity assumption renders the problem tractable, and under this assumption proposed a provably convergent policy iteration scheme and characterized the optimal worst-case temporal adversaries, proving their sparsity. We further established a mathematical equivalence between DTR-RL and transition-kernel robust MDPs, showing that resilience to altered timing yields resilience to altered physical dynamics and providing a practical route to robust RL without complex simulation engineering. Finally, we developed a practical algorithm to solve DTR-SMDPs, empirically validating both directions of this duality.

The primary limitation of our framework stems from the foundational SMDP assumption, which essentially dictates that observations always arrive "fresh" at decision time. This simplifies general delayed RL environments, where delays can result in stale observations, thereby restricting the broader scope of temporal perturbations our current framework can capture. In future work, we aim to tackle these more general cases by extending our robust adversarial approach to handle non-fresh observations. Specifically, we hypothesize that the formal mathematical connection between generalization over time and generalization over transition dynamics can be successfully established within these harder, non-SMDP temporal domains.

AI use statement

In this work, we used generative AI tools to assist in the writing of proofs, to propose and refine hypotheses, to obtain feedback on research methodology and experimental design, to implement methods, and to interpret results. We have not used generative AI tools for any other task with required disclosure: they were not used to develop the theoretical models or conceptual frameworks of this work, to formulate its mathematical claims, to provide critical ingredients for proving those claims, to generate synthetic datasets, to clean or reformat data, to support qualitative or thematic data analysis, or for translation. Additionally, we used generative AI tools to draft and edit parts of the manuscript and improve its readability, to write and debug experiment and analysis code, to produce figures and tables from experimental logs, and to search for and summarize related literature.

We have reviewed all AI-assisted work. The definitions, theorem statements, and proof strategies in this paper are the authors' own; generative AI was used to write up arguments the authors had already established, and every proof in this paper was verified line by line by two authors. All AI-assisted code underwent manual review by the authors to confirm that the implementation matches the method as described in the paper. Interpretations of experimental results proposed with AI assistance were checked by the authors against the underlying logs before inclusion. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

This work involves no human subjects and no personal or sensitive data. All experiments run in simulated MuJoCo control tasks (Todorov et al., 2012), and the third-party code and pretrained checkpoints we build on are credited where they are used (Appx. C.7).

Reproducibility statement

Theorem 1 concerns general uncertainty sets; all subsequent theoretical results assume the s-rectangular sets introduced in Sec. 3.2. Complete proofs are in the appendix: Theorem 1 in Appx. B.1, the policy-iteration results of Sec. 3.2 in Appx. B.2, Corollaries 1 and 2 in Appx. B.3, and Theorem 3, with its extensions to action chunking and to the discounted criterion, in Appx. B.4. Code that trains and evaluates every method is available at `https://anonymous.4open.science/r/dtr-code-release-6E74/`; its README maps each method and experiment to the script and settings that produced it. Algorithm 2 (Appx. C.6) gives the practical algorithm in pseudocode. Appx. C.7 specifies implementation, training, evaluation and score aggregation, Appx. C.8 lists the hyperparameters of all ten methods with the search set of each tuned value, and Appx. C.9 reports the compute used.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f514cec81cb148559cf475e7426eed5e-Paper.pdf.

Yonatan Ashlag, Uri Koren, Mirco Mutti, Esther Derman, Pierre-Luc Bacon, and Shie Mannor. State entropy regularization for robust reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=rtG7n93Ru8.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 1726–1734. AAAI Press, 2017.

J. Andrew Bagnell, Andrew Y. Ng, and Jeff G. Schneider. Solving uncertain Markov decision processes. Technical Report CMU-RI-TR-01-25, Robotics Institute, Carnegie Mellon University, August 2001. URL https://api.semanticscholar.org/CorpusID:6406768.

André Biedenkapp, Raghu Rajan, Frank Hutter, and Marius Lindauer. Temporl: Learning when to act. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 914–924. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/biedenkapp21a.html.

Nico Bohlinger, Grzegorz Czechmanowski, Maciej Krupka, Piotr Kicki, Krzysztof Walas, Jan Peters, and Davide Tateo. One policy to run them all: an end-to-end learning approach to multi-embodiment locomotion. *Conference on Robot Learning*, 2024.

Yann Bouteiller, Simon Ramstedt, Giovanni Beltrame, Christopher Pal, and Jonathan Binas. Reinforcement learning with random delays. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=QFYnKlBJYR.

Esther Derman, Gal Dalal, and Shie Mannor. Acting in delayed environments with non-stationary Markov policies. In *International Conference on Learning Representations*, 2021.

Benjamin Eysenbach and Sergey Levine. Maximum entropy RL (provably) solves some robust RL problems. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PtSAD3caaA2.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020. URL https://arxiv.org/abs/2004.07219.

Uri Gadot, Esther Derman, Navdeep Kumar, Maxence Elfatihi, Kfir Levy, and Shie Mannor. Solving non-rectangular reward-robust MDPs via frequency regularization. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i19.30101. URL https://doi.org/10.1609/aaai.v38i19.30101.

Kaizhe Hu, Ray Chen Zheng, Yang Gao, and Huazhe Xu. Decision transformer under random frame dropping. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NmZXv4467ai.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. CleanRL: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274): 1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

Chieko Sarah Imai, Minghao Zhang, Yuchen Zhang, Marcin Kierebiński, Ruihan Yang, Yuzhe Qin, and Xiaolong Wang. Vision-guided quadrupedal locomotion in the wild with multi-modal delay randomization. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5556–5563, 2022. doi: 10.1109/IROS47612.2022.9981072.

Navdeep Kumar, Adarsh Gupta, Maxence Mohamed ELFATIHI, Giorgia Ramponi, Kfir Yehuda Levy, and Shie Mannor. Non-rectangular robust MDPs with normed uncertainty sets. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Xx0cJGXU7n.

Qiyang Li, Seohong Park, and Sergey Levine. Decoupled Q-Chunking, 2025a. URL https://arxiv.org/abs/2512.10926.

Qiyang Li, Zhiyuan Zhou, and Sergey Levine. Reinforcement learning with action chunking, 2025b. URL https://arxiv.org/abs/2507.07969.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8. IEEE Press, 2018. doi: 10.1109/ICRA.2018.8460528. URL https://doi.org/10.1109/ICRA.2018.8460528.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2817–2826. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/pinto17a.html.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994. ISBN 0471619779.

Allen Z. Ren, Justin Lidard, Lars Lien Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mEpqHvbD2h.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Younggyo Seo, Carmelo Sferrazza, Juyue Chen, Guanya Shi, Rocky Duan, and Pieter Abbeel. Learning sim-to-real humanoid locomotion in 15 minutes, 2025. URL https://arxiv.org/abs/2512.01996.

Sahil Sharma, Aravind S. Lakshminarayanan, and Balaraman Ravindran. Learning to repeat: Fine grained action repetition for deep reinforcement learning. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1GOWV5eg.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112 (1):181–211, 1999. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(99)00052-1. URL https://www.sciencedirect.com/science/article/pii/S0004370299000521.

Corentin Tallec, Léonard Blier, and Yann Ollivier. Making deep q-learning methods robust to time discretization. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6096–6104. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/tallec19a.html.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6215–6224. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/tessler19a.html.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

David Valensi, Esther Derman, Shie Mannor, and Gal Dalal. Tree search-based policy optimization under stochastic execution delay. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RaqZX9LSGA.

Yuhui Wang, Hao He, and Xiaoyang Tan. Robust reinforcement learning in POMDPs with incomplete and noisy observations, 2019. URL https://arxiv.org/abs/1902.05795.

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust Markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, February 2013. ISSN 1526-5471. doi: 10.1287/moor.1120.0566. URL http://dx.doi.org/10.1287/moor.1120.0566.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21024–21037. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f0eb6568ea114ba6e293f903c34d7488-Paper.pdf.

Ruida Zhou, Tao Liu, Min Cheng, Dileep Kalathil, Panganamala Kumar, and Chao Tian. Natural actor-critic for robust reinforcement learning with function approximation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=wxkBdtDbmH.

A Related Works

Delayed and Asynchronous Reinforcement Learning. The challenge of temporal delays in RL is typically categorized into execution delays (Valensi et al., 2024) and observation delays (Bouteiller et al., 2021), but these settings typically assume a fixed interaction protocol. Our work addresses intermittent missing observations—often termed frame skipping (Wang et al., 2019). While prior approaches generally model these interruptions as an unstructured, independent and identically distributed (i.i.d.) noise process (Hu et al., 2023), they fail to capture the structural uncertainty inherent in real-world systems. In contrast, our Decision-Time-Robust SMDP framework explicitly handles state-dependent observation drops and optimizes against a family of possible masking processes via a min-max objective, shifting the paradigm from merely tolerating average-case stochastic noise to providing rigorous safety guarantees against systematic, worst-case temporal shifts. A separate line targets the discretization itself: Tallec et al. (2019) show that Q-learning degenerates as the discretization of a single step shrinks, and derive an algorithm that performs consistently across discretizations. There the discretization is fixed and known within a run; in DTR it is the base step that is fixed, while the number of such steps between consecutive decisions varies within an episode, depends on the state, and is chosen adversarially.

Temporal Abstractions and Hierarchical RL. Our formulation formally captures the asynchronous nature of real-world interactions by building upon Semi-Markov Decision Processes (SMDPs), and can thus be related to the options framework (Sutton et al., 1999; Bacon et al., 2017). However, whereas standard option learning aims to discover optimal termination conditions, we treat the termination time (i.e., the arrival of the next observation) as adversarial. Furthermore, while options are typically closed-loop, our agent must commit to open-loop sequences during blind intervals. Consequently, our work complements recent advances in action chunking (Li et al., 2025a;b), possibly offering additional theoretical insights to the method's generalization success. We diverge from prior work by modeling the duration of these chunks not as a fixed hyperparameter or a learned termination condition, but as an adversarial variable governed by a strict uncertainty budget. The closest such prior work learns the duration of action repetition (Sharma et al., 2017; Biedenkapp et al., 2021): the agent picks an action and how long to hold it, then acts on a fresh observation, which is the action-repetition SMDP we build on. There the agent chooses the duration to maximize return; in DTR an adversary chooses it to minimize return under a mean constraint, turning a single-agent SMDP into a zero-sum game.

Robust Reinforcement Learning. Robust RL has extensively explored methods to robustify agents against uncertainty in reward functions and transition dynamics (Gadot et al., 2024; Pinto et al., 2017). To the best of our knowledge, our work introduces the novel concept of Decision-Time Robustness. Crucially, we establish a formal equivalence between Decision-Time-Robust RL and Transition Kernel Robust MDPs. This proves that resilience to adversarial timing is mathematically equivalent to resilience against physical dynamics shifts. This insight addresses the conservatism problem inherent in standard robust RL by naturally restricting perturbations to dynamically reachable states. It also offers a computationally cheap, "simulation-free" alternative to standard practices: rather than engineering complex physical perturbations in a simulator, agents can achieve meaningful dynamics generalization simply by training against an adversarial clock.

B Proofs

B.1 Proof of Theorem 1

Before proving the theorem, we rigorously represent the temporal uncertainty sets and define the concept of relative error.

Uncertainty Set Representation We consider uncertainty sets of the mappings $\tau : \mathcal{S} \to \Delta_{[D]}$ that can be represented as affine transformations of a base parameter space. Specifically, let $\Xi \subset \mathbb{R}^q$ be a bounded polyhedron representing the global adversarial parameters. We define Ξ as the intersection of closed halfspaces:

$$\Xi := \{\xi \in \mathbb{R}^q : C\xi \leq b\} \tag{1}$$

where $C \in \mathbb{R}^{L\times q}$ and $b \in \mathbb{R}^L$ encode the L linear constraints restricting the adversary's choices. The computational size of the input Ξ is defined by the encoding length of these L linear constraints. For every state $s \in \mathcal{S}$, the transition time distribution $\tau_s^\xi \in \Delta_{[D]}$ is determined by an affine mapping from the global parameter vector ξ:

$$\tau_s^\xi := k_s + K_s\xi$$

where $k_s \in \mathbb{R}^D$ and $K_s \in \mathbb{R}^{D\times q}$. To mathematically guarantee that τ_s^ξ is always a valid probability measure (i.e., non-negative and summing to 1 for all $s \in \mathcal{S}$), we assume these standard simplex constraints are strictly subsumed within the L linear inequalities defining Ξ. The overall decision-time uncertainty set $\mathcal{U}$ is then formally defined as the image of Ξ under these state-dependent affine mappings:

$$\mathcal{U} := \left\{\tau \in (\Delta_{[D]})^{\mathcal{S}} : \exists \xi \in \Xi \text{ such that } \tau_s = k_s + K_s\xi \quad \forall s \in \mathcal{S}\right\} \tag{2}$$

Note that this parametric representation naturally accommodates the structural restrictions introduced later in this paper. The rectangular uncertainty set $\mathcal{U}_D = \bigtimes_{s\in\mathcal{S}} \mathcal{U}_s$ is recovered when the constraints defining Ξ factorize across states (e.g., K_s matrices are block-diagonal, decoupling the adversarial parameters ξ_s). Furthermore, the fixed-mean constrained set $\mathcal{U}_{c,D}$ is simply a special case where the expected duration condition ($\mathbb{E}_{d\sim\tau_s}[d] = c$) is encoded as an additional linear equality constraint within Ξ.

Relative Error To analyze the hardness of approximation, we adopt a standard piecewise definition of relative error. We say that an approximation $\tilde{J}$ to the true worst-case expected return J^* has a relative error of β if:

$$\begin{cases} \frac{|\tilde{J}-J^*|}{\min\{|\tilde{J}|,|J^*|\}} \leq \beta & \text{if } \min\{|\tilde{J}|, |J^*|\} > 0, \\ |\tilde{J} - J^*| \leq \beta & \text{otherwise.} \end{cases}$$

This definition ensures the error metric remains mathematically well-defined even when the optimal or approximate return is zero.

We are now ready to prove the theorem, which we restate more formally below.

State	$P(\tau = 1)$	$P(\tau = 2)$	π
s_j	ξ_j	$1 - \xi_j$	a_1
s_j^1	ξ_j	$1 - \xi_j$	a_2
s_j^L	ξ_j	$1 - \xi_j$	a_1
s_j^R	1	0	a_1
s_j^m	1	0	a_1

Table 3: Specification of the delay distribution τ^ξ and policy π for the states in the j-th sub-MDP used in proof of Theorem 1

Theorem 4. *For any relative error constant β, the problem of approximating the robust expected return of a given policy over a temporal uncertainty set $\mathcal{U}$ of the form equation 2 up to a relative error β is NP-hard.*

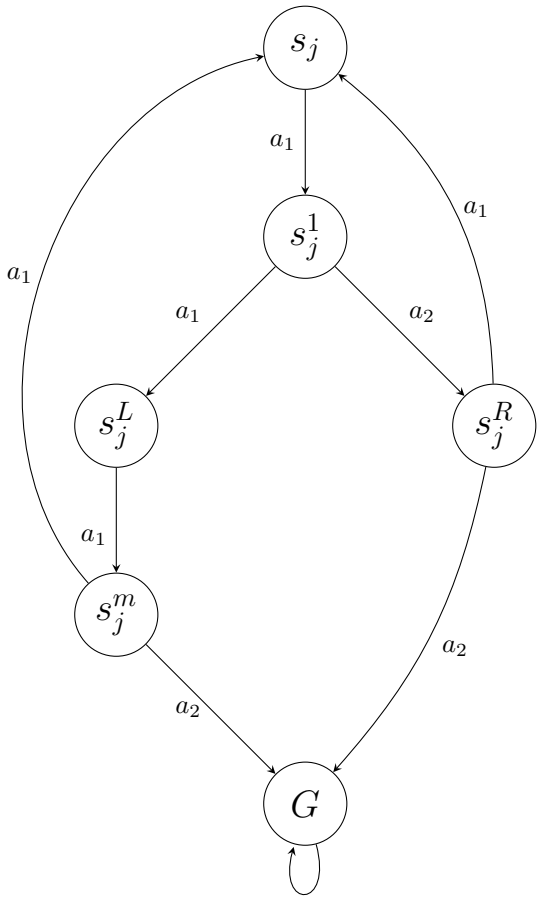

Figure 2: Sub-MDP structure used in proof of Theorem 1. Transitions are deterministic and labels on edges represent actions.

Proof. Our complexity results rely on a polynomial-time reduction from the well-known NP-hard 0/1 Integer Programming (IP) problem, inspired by the general proof architecture introduced by Wiesemann et al. (2013). An instance of the IP problem asks whether there exists a binary vector $x \in \{0,1\}^n$ satisfying $Fx \leq g$, where $F \in \mathbb{Z}^{m \times n}$ and $g \in \mathbb{Z}^m$. We utilize a key algebraic property regarding such polytopes (see Lemma 2.5 in Wiesemann et al. (2013)): one can always safely round a fractional solution $x \in ([0,\epsilon] \cup [1-\epsilon,1])^n$ to a valid binary solution $y \in \{0,1\}^n$ provided that $\epsilon < \min_i\{(\sum_j |F_{ij}|)^{-1}\}$. Consequently, finding a point within the polytope that is ϵ-close to the boolean hypercube under the infinity norm is mathematically equivalent to proving the IP instance is feasible.

For a given IP instance, we construct a decision-time robust SMDP whose size scales polynomially. The state space consists of n disjoint copies of the sub-MDP depicted in Figure 2, all feeding into a shared absorbing state G. The agent starts uniformly at random in one of the n root states $\{s_j\}_{j=1}^n$ and selects between actions $\{a_1, a_2\}$. All transition rewards in the base MDP are exactly 0, except for those entering G, which grant a reward of $R > \frac{\beta(1+\beta)n}{\gamma^2\epsilon^2}$, where $\gamma \in (0,1)$ is the discount factor and ϵ is the rounding threshold defined above. We evaluate the policy π alongside the affine delay mappings τ^ξ specified in Table 3. The adversary's global parameters are restricted to the IP polytope $\Xi = \{\xi \in [0,1]^n \mid F\xi \leq g\}$. By construction, Ξ and the resulting temporal uncertainty set $\mathcal{U}$ conform to the parametric structures defined in equation 1 and equation 2.The fundamental mechanism of this reduction lies in the relationship between the IP feasibility and the true worst-case expected return, J^*. Because our sub-MDPs penalize fractional parameter choices, the global expected return evaluates to a strict lower bound: $J^* \geq \frac{1}{n}\sum_{j=1}^n \xi_j(1-\xi_j)\gamma^2 R$. We can categorize the IP instance into two exhaustive cases based on this return:

- Feasible IP: If the IP admits a binary solution, the adversary can select $\xi^* \in \{0,1\}^n$. For this specific choice, the fractional penalty term vanishes, yielding a return of exactly zero. Since rewards are non-negative, this implies $J^* = 0$.

- Infeasible IP: If no binary solution exists, the IP is infeasible. Furthermore, by the rounding lemma, no valid $\xi \in \Xi$ can even reside within the ϵ-margins. Every ξ must contain at least one coordinate strictly bounded away from the boundaries (i.e., $\xi_k \in (\epsilon, 1-\epsilon)$). This forces a strictly positive return that we can universally lower-bound: $J^* > \frac{1}{n}\epsilon^2\gamma^2 R > \beta(1+\beta)$.

Leveraging this massive gap in the true return J^*, we can demonstrate that any algorithm capable of outputting a β-approximation $\tilde{J}$ acts as a perfect decision oracle for the IP problem. Specifically, we prove that the IP is feasible if and only if $|\tilde{J}| \leq \beta$.

If the IP is feasible, $J^* = 0$. To satisfy the relative error guarantee, the approximation must fall within the absolute error bound $|\tilde{J} - J^*| \leq \beta$, which naturally requires $|\tilde{J}| \leq \beta$.

Conversely, suppose the IP is infeasible (meaning $J^* > \beta(1+\beta)$), but the algorithm falsely returns $|\tilde{J}| \leq \beta$. Evaluating the relative error yields:

$$\frac{|\tilde{J} - J^*|}{\min\{|\tilde{J}|, |J^*|\}} \geq \frac{J^* - |\tilde{J}|}{|\tilde{J}|} \geq \frac{J^* - \beta}{\beta} > \frac{\beta(1+\beta) - \beta}{\beta} = \beta$$

This strictly violates the algorithm's β-approximation guarantee. Therefore, an infeasible IP necessitates that $|\tilde{J}| > \beta$.

Because checking the simple threshold $|\tilde{J}| \leq \beta$ perfectly dictates the feasibility of the underlying integer program, approximating the decision-time robust SMDP to any relative error β is NP-hard. □

B.2 Proofs for Section 3.2

B.2.1 Proof of Proposition 1

Proof. Let $V, U \in \mathbb{R}^{\mathcal{S}}$ be two bounded vectors. For any state $s \in \mathcal{S}$,

$$\begin{aligned}
|TV(s) - TU(s)| &= \Bigg| \max_{\pi \in \Pi^{\mathrm{AC}}} \min_{\tau_s \in \mathcal{U}_{c,D,s}} \sum_{d=1}^{D} \tau_s(d) [T^\pi V](s,d) \\
&\qquad - \max_{\pi \in \Pi^{\mathrm{AC}}} \min_{\tau_s \in \mathcal{U}_{c,D,s}} \sum_{d=1}^{D} \tau_s(d) [T^\pi U](s,d) \Bigg| \\
&\leq \max_{\pi \in \Pi^{\mathrm{AC}}} \max_{\tau_s \in \mathcal{U}_{c,D,s}} \left| \sum_{d=1}^{D} \tau_s(d)\, \gamma^d \sum_{s' \in \mathcal{S}} \bar{P}^{\pi}_{(d)}(s'|s) \big(V(s') - U(s')\big) \right| \\
&\leq \max_{\pi} \max_{\tau_s} \sum_{d=1}^{D} \tau_s(d)\, \gamma^d \, \|V - U\|_\infty \\
&\leq \max_{d \in \{1,\ldots,D\}} \gamma^d \, \|V - U\|_\infty \;=\; \gamma \, \|V - U\|_\infty .
\end{aligned}$$

The first inequality applies the standard bounds $|\max_x f(x) - \max_x g(x)| \leq \max_x |f(x) - g(x)|$ then $|\min_x f(x) - \min_x g(x)| \leq \max_x |f(x) - g(x)|$, and uses that the immediate-reward term $\bar{R}^\pi(s,d)$ cancels between $[T^\pi V]$ and $[T^\pi U]$. The second uses that $\bar{P}^{\pi}_{(d)}(\cdot \mid s)$ is a probability measure over $\mathcal{S}$. The third uses $\sum_d \tau_s(d) = 1$ together with $\gamma^d \leq \gamma^1 = \gamma$ for all $d \geq 1$. Since this holds for every $s \in \mathcal{S}$, T is a γ-contraction in supremum norm, and Banach's fixed-point theorem yields a unique fixed point V^*. □

B.2.2 Proof of Theorem 2

Proof. We establish $\|V_k - V^*\|_\infty \leq \gamma \|V_{k-1} - V^*\|_\infty$ in four steps.

Step 1: Robust evaluation operator. For any (action-chunk) policy $\pi \in \Pi^{\mathrm{AC}}$, recall from Sec. 3.2:

$$\begin{aligned}
B^\pi V(s) &:= \min_{\tau_s \in \mathcal{U}_{c,D,s}} \sum_{d=1}^{D} \tau_s(d)\, [T^\pi V](s,d) \\
&= \min_{\tau_s \in \mathcal{U}_{c,D,s}} \sum_{d=1}^{D} \tau_s(d) \left(\bar{R}^\pi(s,d) + \gamma^d \textstyle\sum_{s' \in \mathcal{S}} \bar{P}^{\pi}_{(d)}(s'|s) V(s') \right), \quad s \in \mathcal{S}.
\end{aligned}$$

By the same argument as in the proof of Prop. 1 (with the outer max removed), B^π is also a γ-contraction, i.e., given $V, W \in \mathbb{R}^{\mathcal{S}}$:

$$|B^\pi V(s) - B^\pi W(s)| \leq \max_{\tau_s \in \mathcal{U}_{c,D,s}} \sum_{d=1}^{D} \tau_s(d)\, \gamma^d \, \|V - W\|_\infty \leq \gamma \, \|V - W\|_\infty .$$

We denote its unique fixed point by V^π. Both B^π and T are moreover monotone: if $V \geq W$ pointwise then $[T^\pi V](s,d) \geq [T^\pi W](s,d)$ for every $s \in \mathcal{S}$ and $d \in [D]$, because $\gamma^d \bar{P}^{\pi}_{(d)}(\cdot|s)$ has nonnegative entries; averaging against any τ_s, minimizing over $\mathcal{U}_{c,D,s}$ and maximizing over π all preserve the inequality.

Step 2: Key properties of DTR-PI. From Algorithm 1:

- $V_k = B^{\pi^k} V_k$ (policy evaluation yields the fixed point of B^{π^k});
- π^{k+1} is greedy with respect to V_k, hence $B^{\pi^{k+1}} V_k = T V_k$;
- $V^* = T V^*$ (the optimal value is the fixed point of T).

Step 3: Monotonicity. By the greedy improvement principle,

$$TV_k(s) = \max_{\pi \in \Pi^{\mathrm{AC}}} \min_{\tau_s} \sum_{d=1}^{D} \tau_s(d)[T^{\pi} V_k](s,d)$$
$$\geq \min_{\tau_s} \sum_{d=1}^{D} \tau_s(d)[T^{\pi^k} V_k](s,d) = B^{\pi^k} V_k(s) = V_k(s).$$

By Step 2, $B^{\pi^{k+1}} V_k = TV_k \geq V_k$. Applying the monotone operator $B^{\pi^{k+1}}$ repeatedly therefore gives $V_k \leq B^{\pi^{k+1}} V_k \leq (B^{\pi^{k+1}})^2 V_k \leq \cdots$, and by Step 1 the right-hand side converges to the unique fixed point $V^{\pi^{k+1}} = V_{k+1}$, so that $V_k \leq V_{k+1}$. The same argument with T in place of $B^{\pi^{k+1}}$, whose iterates converge to V^*, gives $V_k \leq V^*$. Hence

$$V_0 \leq V_1 \leq \cdots \leq V_k \leq \cdots \leq V^*.$$

Step 4: Contraction bound. Using $V^* = TV^*$ and $B^{\pi^k} V_{k-1} = TV_{k-1}$ (greedy construction at iteration k), together with $V_k = B^{\pi^k} V_k$, we decompose

$$V^*(s) - V_k(s) = \big(TV^*(s) - TV_{k-1}(s)\big) + \big(B^{\pi^k} V_{k-1}(s) - B^{\pi^k} V_k(s)\big).$$

For the first term, the γ-contraction of T gives

$$TV^*(s) - TV_{k-1}(s) \leq \|TV^* - TV_{k-1}\|_\infty \leq \gamma \|V^* - V_{k-1}\|_\infty.$$

For the second term, the monotonicity of B^{π^k} (Step 1) combined with $V_k \geq V_{k-1}$ yields

$$B^{\pi^k} V_{k-1}(s) - B^{\pi^k} V_k(s) \leq 0.$$

Combining both bounds with $V_k \leq V^*$ (Step 3) gives $0 \leq V^*(s) - V_k(s) \leq \gamma \|V^* - V_{k-1}\|_\infty$ for every $s \in \mathcal{S}$. Taking the supremum over $s \in \mathcal{S}$:

$$\|V_k - V^*\|_\infty \leq \gamma \|V_{k-1} - V^*\|_\infty.$$

□

The fixed point is the DTR-SMDP optimum. Proposition 1 and Theorem 2 are statements about the fixed point V^* of T; we close the loop with the max-min problem of Sec. 3.2. Write $V^{\pi,\tau}(s)$ for the discounted SMDP return of π from $s_0 = s$ under a fixed transition-time process $\tau \in \mathcal{U}_{c,D}$, so that $J(\pi,\tau) = \mathbb{E}_{s_0}[V^{\pi,\tau}(s_0)]$, and let $B^{\pi,\tau}V(s) := \sum_{d=1}^{D} \tau_s(d)\,[T^{\pi}V](s,d)$. The argument of Proposition 1 makes $B^{\pi,\tau}$ a γ-contraction, and $V^{\pi,\tau}$ is its fixed point, since decomposing the return at the first decision time gives exactly $V^{\pi,\tau} = B^{\pi,\tau}V^{\pi,\tau}$. Because $\mathcal{U}_{c,D}$ is s-rectangular, the minimization defining B^{π} decouples across states, so $B^{\pi}V = \min_{\tau \in \mathcal{U}_{c,D}} B^{\pi,\tau}V$ pointwise, the minimum being attained as the objective is linear in τ_s and $\mathcal{U}_{c,D,s}$ is compact.

Consequently $V^{\pi} = \min_{\tau \in \mathcal{U}_{c,D}} V^{\pi,\tau}$ pointwise: on the one hand $B^{\pi}V \leq B^{\pi,\tau}V$ for every τ and V, so monotonicity and Banach's theorem give $V^{\pi} \leq V^{\pi,\tau}$; on the other hand, if $\tau^\star$ attains the minimum at V^{π}, then $B^{\pi,\tau^\star}V^{\pi} = B^{\pi}V^{\pi} = V^{\pi}$, so $V^{\pi} = V^{\pi,\tau^\star}$ by uniqueness of the fixed point. Finally $V^* = \max_{\pi} V^{\pi}$ pointwise: for every π, $V^* = TV^* \geq B^{\pi}V^*$, so iterating the monotone B^{π} yields $V^* \geq (B^{\pi})^n V^* \to V^{\pi}$, while if π^* attains the maximum in TV^* then $B^{\pi^*}V^* = TV^* = V^*$, whence $V^* = V^{\pi^*}$. Since a single $\tau^\star$ minimizes at every state simultaneously, $\max_{\pi} \min_{\tau \in \mathcal{U}_{c,D}} J(\pi,\tau) = \mathbb{E}_{s_0}[V^*(s_0)]$, attained by any policy greedy with respect to V^* — so DTR-PI converges to the optimal value of the DTR-SMDP, and its greedy policies to an optimal one.

B.2.3 Proof of Proposition 2

Proof. Throughout, $\gamma \in (0,1)$, $D \geq 2$ and $c \in [1,D]$, so that $\mathcal{U}_{c,D,s} \neq \emptyset$, and δ_j denotes the Dirac mass at j. Step 1 computes the worst-case effective discount in closed form, Step 2 converts it into a contraction modulus for T, and Step 3 propagates it through the policy-iteration argument of Theorem 2.

Step 1: Worst-case effective discount. Define

$$\tilde{\gamma}_{c,D} \;:=\; \max_{\tau_s \in \mathcal{U}_{c,D,s}} \mathbb{E}_{d\sim\tau_s}\big[\gamma^d\big]\,.$$

Upper bound. The map $f : d \mapsto \gamma^d$ satisfies $f''(d) = (\ln\gamma)^2\gamma^d > 0$, hence is strictly convex, and therefore lies below its chord on $[1, D]$:

$$\gamma^d \;\le\; \ell(d) \;:=\; \gamma + \frac{\gamma^D - \gamma}{D-1}\,(d-1), \qquad d \in [1, D],$$

where ℓ is the affine interpolant of f through $(1, \gamma)$ and (D, γ^D). Every $\tau_s \in \mathcal{U}_{c,D,s}$ is supported in $\{1, \dots, D\} \subset [1, D]$, so taking expectations and using that ℓ is affine together with the mean constraint $\mathbb{E}_{d\sim\tau_s}[d] = c$,

$$\mathbb{E}_{d\sim\tau_s}\big[\gamma^d\big] \;\le\; \mathbb{E}_{d\sim\tau_s}[\ell(d)] \;=\; \ell(\mathbb{E}_{d\sim\tau_s}[d]) \;=\; \ell(c) \;=\; \frac{(D-c)\,\gamma + (c-1)\,\gamma^D}{D-1}.$$

Attainment. Let $\tau_s^\star := \frac{D-c}{D-1}\,\delta_1 + \frac{c-1}{D-1}\,\delta_D$. Its weights are nonnegative since $1 \le c \le D$, and they sum to $\frac{(D-c)+(c-1)}{D-1} = 1$, so $\tau_s^\star \in \Delta_{[D]}$. It satisfies the mean constraint,

$$\mathbb{E}_{d\sim\tau_s^\star}[d] \;=\; \frac{(D-c)\cdot 1 + (c-1)\,D}{D-1} \;=\; \frac{c\,(D-1)}{D-1} \;=\; c,$$

so $\tau_s^\star \in \mathcal{U}_{c,D,s}$, and it attains the upper bound,

$$\mathbb{E}_{d\sim\tau_s^\star}\big[\gamma^d\big] \;=\; \frac{D-c}{D-1}\,\gamma + \frac{c-1}{D-1}\,\gamma^D \;=\; \ell(c).$$

Hence $\tilde{\gamma}_{c,D} = \ell(c)$. Note that $\tau_s^\star$ is supported on the two extreme delays $\{1, D\}$, consistent with the vertex structure established in Corollary 2.

Range. $\tilde{\gamma}_{c,D} = \ell(c)$ is a convex combination of γ and γ^D, and $\gamma^D \le \gamma$, so $\tilde{\gamma}_{c,D} \le \gamma$; the inequality is strict whenever $c > 1$, since the weight $\frac{c-1}{D-1}$ placed on $\gamma^D < \gamma$ is then positive. Conversely, Jensen's inequality applied to the convex f under $\tau_s^\star$ gives $\mathbb{E}_{d\sim\tau_s^\star}[\gamma^d] \ge \gamma^{\mathbb{E}_{\tau_s^\star}[d]} = \gamma^c$, so $\gamma^c \le \tilde{\gamma}_{c,D} \le \gamma$.

Step 2: T is a $\tilde{\gamma}_{c,D}$-contraction. Let $V, U \in \mathbb{R}^{\mathcal{S}}$ be bounded and fix $s \in \mathcal{S}$. The first two inequalities in the proof of Proposition 1 apply verbatim: the immediate reward $\bar{R}^\pi(s, d)$ cancels between $[T^\pi V]$ and $[T^\pi U]$, and $\bar{P}^\pi_{(d)}(\cdot \mid s)$ is a probability measure, so

$$|TV(s) - TU(s)| \;\le\; \max_{\pi\in\Pi^{\mathrm{AC}}} \max_{\tau_s\in\mathcal{U}_{c,D,s}} \sum_{d=1}^{D} \tau_s(d)\,\gamma^d\,\|V - U\|_\infty.$$

Rather than relaxing $\gamma^d \le \gamma$ as in Proposition 1, we evaluate the inner maximum exactly: it equals $\tilde{\gamma}_{c,D}\,\|V - U\|_\infty$ by Step 1. The resulting bound does not depend on π or on s, so

$$\|TV - TU\|_\infty \;\le\; \tilde{\gamma}_{c,D}\,\|V - U\|_\infty,$$

and deleting the outer maximum gives the same modulus for B^π, for every action-chunk policy π.

Step 3: Convergence rate of DTR-PI. We reuse the decomposition of Step 4 in the proof of Theorem 2, which is unchanged:

$$V^*(s) - V_k(s) = \big(TV^*(s) - TV_{k-1}(s)\big) + \big(B^{\pi^k}V_{k-1}(s) - B^{\pi^k}V_k(s)\big).$$

For the first term, the $\tilde{\gamma}_{c,D}$-contraction of Step 2 gives

$$TV^*(s) - TV_{k-1}(s) \;\le\; \|TV^* - TV_{k-1}\|_\infty \;\le\; \tilde{\gamma}_{c,D}\,\|V^* - V_{k-1}\|_\infty.$$

The second term is non-positive, since B^{π^k} is monotone (Step 1 of the proof of Theorem 2) and $V_{k-1} \le V_k$ (Step 3 there). With $V_k \le V^*$ from the same step, taking the supremum over $s \in \mathcal{S}$,

$$\|V_k - V^*\|_\infty \;\le\; \tilde{\gamma}_{c,D}\,\|V_{k-1} - V^*\|_\infty. \qquad \square$$

B.3 Proofs for Section 3.3

Proof. Let $L : [D] \to \mathbb{R}$ be any objective function. Consider the adversary's optimization problem for a state s, subject to a mean delay constraint c:

$$\min_{\tau_s \in \Delta_{[D]}} \sum_{d=1}^{D} \tau_s(d) L(d) \quad \text{s.t.} \quad \sum_{d=1}^{D} d\tau_s(d) = c \tag{3}$$

The optimization problem above is a Linear Program (LP) where the decision variables are the probabilities $\tau_s(1), \ldots, \tau_s(D)$. The feasible region is defined by the intersection of the probability simplex and the mean constraint hyperplane. Specifically, we have two linear equality constraints:

$$\sum_{d=1}^{D} \tau_s(d) = 1 \quad \text{(Probability Simplex)} \tag{4}$$

$$\sum_{d=1}^{D} d \cdot \tau_s(d) = c \quad \text{(Mean Constraint)} \tag{5}$$

The feasible set is non-empty whenever $c \in [1, D]$, as it contains the Bernoulli mixture of $d = 1$ and $d = D$ with mean c, and it is compact, being the intersection of the simplex with a hyperplane; since the objective is linear, an optimal solution exists. According to the Fundamental Theorem of Linear Programming, if an optimal solution exists for a bounded LP in standard form, there exists a basic feasible solution (BFS) that is optimal. A basic feasible solution is characterized by having at most m non-zero variables, where m is the number of linearly independent constraints.

In our setting, there are exactly $m = 2$ constraints. Therefore, there exists an optimal adversary τ^* with at most 2 non-zero entries. Geometrically, this corresponds to the adversary selecting either a single $d = c$ (if c is an integer and optimal) or mixing between two distinct transition times $d_1 < c < d_2$ to satisfy the mean constraint exactly. □

Corollary 1 is the unconstrained case. The same argument applies verbatim to $\mathcal{U}_D$, where $\mathcal{U}_s = \Delta_{[D]}$: dropping the mean constraint leaves $m = 1$, so an optimal basic feasible solution has a single non-zero entry, i.e. τ_s^* is the point mass at some $d^* \in \arg\min_{d \in [D]} L(d)$.

B.4 Proof of Theorem 3

We prove the equivalence by constructing, for every adversarial transition-time process $\tau \in \mathcal{U}_{c,D}$, an effective MDP whose average return under any policy π matches the SMDP average return under τ, and showing that the resulting set of effective MDPs is exactly the uncertainty set $\mathcal{U}$ of Theorem 3.

Notation. Throughout the proofs below, for a chunk $\mathbf{a} = (a^{(1)}, \ldots, a^{(D)}) \in \mathcal{A}^D$ we write

$$\bar{R}(s, \mathbf{a}, d) := \mathbb{E}\big[\textstyle\sum_{t=0}^{d-1} \gamma^t R(s_t, a_t) \;\big|\; s_0 = s, \mathbf{a}\big] = \sum_{k=1}^{d} \gamma^{k-1} [P_{a^{(1)}} \cdots P_{a^{(k-1)}} R_{a^{(k)}}](s),$$
$$P^{(d)}(s'|s, \mathbf{a}) := \mathbb{P}(s_d = s' \mid s_0 = s, \mathbf{a}) = P_{a^{(1)}} P_{a^{(2)}} \cdots P_{a^{(d)}}(s'|s), \quad \forall s' \in \mathcal{S},$$

for the reward accumulated and the state reached while the first d actions of $\mathbf{a}$ are executed. We take $\gamma = 1$ in the average-reward sections, where $\bar{R}$ enters the renewal-reward identity as the undiscounted reward of one decision cycle, and $\gamma \in (0, 1)$ in the discounted case treated at the end of this appendix. Under action repetition $\mathbf{a} = (a, \ldots, a)$, so that $P^{(d)}(\cdot|s, \mathbf{a}) = (P_a)^d(\cdot|s)$, and we then write $\bar{R}(s, a, d)$.

Effective MDP induced by τ. Fix any $\tau \in \mathcal{U}_{c,D}$. Under action repetition, when the agent is at state s and selects action a, the duration $d \sim \tau_s$ determines that a is repeated for d base-MDP timesteps. Marginalizing the base-MDP dynamics over τ_s, we define the

effective per-decision transition kernel and reward:

$$\tilde{P}^{\tau}(s'|s,a) := \sum_{d=1}^{D} \tau_s(d)(P_a)^d(s'|s), \tag{6}$$

$$\tilde{R}^{\tau}_{\text{raw}}(s,a) := \sum_{d=1}^{D} \tau_s(d)\bar{R}(s,a,d). \tag{7}$$

Equation 6 is a valid transition kernel (a convex combination of stochastic matrices), so $(\tilde{P}^{\tau}, \tilde{R}^{\tau}_{\text{raw}})$ defines a standard MDP $\tilde{\mathcal{M}}^{\tau}$ on $(\mathcal{S}, \mathcal{A})$.

Matching the average return. The induced process $\{s_k\}_{k\geq 0}$, sampled at decision times, is a Markov chain on $\mathcal{S}$ governed by $\tilde{P}^{\tau}$. By the standard renewal-reward identity for SMDPs (see, e.g., Puterman, 1994, Ch. 11), the average reward of policy π in the SMDP under transition-time process τ is

$$J_{\text{avg}}(\pi, P, R, \tau) \;=\; \frac{\sum_{s,a} \tilde{d}^{\pi}(s,a)\, \tilde{R}^{\tau}_{\text{raw}}(s,a)}{\sum_{s,a} \tilde{d}^{\pi}(s,a)\, \mathbb{E}_{d\sim\tau_s}[d]}, \tag{8}$$

where $\tilde{d}^{\pi}$ is the stationary state-action distribution of the embedded chain under $\tilde{P}^{\tau}$ and π, which we assume unichain for every $\tau \in \mathcal{U}_{c,D}$. The fixed-mean constraint $\mathbb{E}_{d\sim\tau_s}[d] = c$ for all s collapses the denominator to c, giving

$$J_{\text{avg}}(\pi, P, R, \tau) \;=\; \big\langle \tilde{d}^{\pi},\, \tfrac{1}{c}\tilde{R}^{\tau}_{\text{raw}} \big\rangle \;=\; \big\langle \tilde{d}^{\pi},\, \tilde{R}^{\tau} \big\rangle, \tag{9}$$

where $\tilde{R}^{\tau} := \tilde{R}^{\tau}_{\text{raw}}/c$. The right-hand side of equation 9 is exactly the average return of policy π in the standard MDP $(\tilde{P}^{\tau}, \tilde{R}^{\tau})$. Hence, for every π and every $\tau \in \mathcal{U}_{c,D}$,

$$J^{\text{SMDP}}_{\text{avg}}(\pi, \tau) \;=\; J^{\text{MDP}}_{\text{avg}}(\pi, \tilde{P}^{\tau}, \tilde{R}^{\tau}). \tag{10}$$

Equivalence of uncertainty sets. Let $\Phi_s : \tau_s \mapsto \big(\tilde{P}^{\tau}(\cdot|s,\cdot),\, \tilde{R}^{\tau}(s,\cdot)\big)$ denote the per-state marginalization map defined by equation 6–equation 7 together with the $1/c$ rescaling, and let $\Phi := (\Phi_s)_{s\in\mathcal{S}}$. In the notation of Theorem 3, $\Phi_s(\tau_s) = (\tilde{P}_{\tau_s}, \tilde{R}_{\tau_s})$, so $\mathcal{U}_s$ is by definition the image of $\mathcal{U}_{c,D,s}$ under Φ_s, and

$$\mathcal{U} \;=\; \Phi(\mathcal{U}_{c,D}).$$

Note that s-rectangularity is preserved: since $\mathcal{U}_{c,D}$ factorizes across states (τ_s is chosen independently for each s), and Φ acts on each τ_s independently to produce $(\tilde{P}(\cdot|s,\cdot), \tilde{R}(s,\cdot))$, the image $\mathcal{U}$ factorizes across states as well.

Conclusion. Combining equation 10 with $\mathcal{U} = \Phi(\mathcal{U}_{c,D})$, for every policy π we have

$$\min_{\tau\in\mathcal{U}_{c,D}} J^{\text{SMDP}}_{\text{avg}}(\pi, \tau) \;=\; \min_{(\tilde{P},\tilde{R})\in\mathcal{U}} J^{\text{MDP}}_{\text{avg}}(\pi, \tilde{P}, \tilde{R}) \;=\; J^{\mathcal{U}}_{\text{avg}}(\pi).$$

Taking the maximum over π on both sides, the two max-min problems share the same value and the same set of optimizers. In particular, if π^* is optimal for the DTR-SMDP under the average reward criterion, then $\pi^* \in \arg\max_{\pi} J^{\mathcal{U}}_{\text{avg}}(\pi)$. □

B.4.1 Equivalence under action chunking

We now extend the equivalence to the action chunking setting, where at each decision time the agent commits to a sequence $\mathbf{a} = (a^{(1)}, \ldots, a^{(D)}) \in \mathcal{A}^D$. We present two complementary forms of the equivalence: a clean version in the augmented action space (Theorem 5), and a back-projection to the original action space that exposes a physically interpretable, policy-aware uncertainty set (Theorem 6).

Equivalence in the augmented action space. Under action chunking, the role of the per-decision action in the action-repetition proof is played by the chunk $\mathbf{a} \in \mathcal{A}^D$. We write $\bar{\pi} : \mathcal{S} \to \Delta_{\mathcal{A}^D}$ for a policy on the augmented space, reserving π for policies on $\mathcal{A}$. Marginalizing the base-MDP dynamics over τ_s, the effective per-decision kernel and reward become:

$$\tilde{P}^{\tau}(s'|s,\mathbf{a}) := \sum_{d=1}^{D} \tau_s(d)\, P^{(d)}(s'|s,\mathbf{a}), \tag{11}$$

$$\tilde{R}^{\tau}(s,\mathbf{a}) := \frac{1}{c}\sum_{d=1}^{D} \tau_s(d)\, \bar{R}(s,\mathbf{a},d), \tag{12}$$

where $P^{(d)}$ and $\bar{R}$ are as defined at the start of App. B.4. Mirroring Theorem 3, define the s-rectangular uncertainty set on the augmented MDP $(\mathcal{S}, \mathcal{A}^D)$:

$$\mathcal{U}_s^{\text{aug}} := \Big\{ \big(\tilde{P}^{\tau}(\cdot|s,\cdot),\, \tilde{R}^{\tau}(s,\cdot)\big) \;\Big|\; \tau_s \in \mathcal{U}_{c,D,s} \Big\}, \qquad \mathcal{U}^{\text{aug}} := \bigtimes_{s\in\mathcal{S}} \mathcal{U}_s^{\text{aug}},$$

with $\tilde{P}^{\tau}$ and $\tilde{R}^{\tau}$ as in equation 11–equation 12.

Theorem 5 (Chunking equivalence, augmented space)**.** *Let $\bar{\pi}^*$ be the optimal policy for the DTR-SMDP under action chunking and the average reward criterion. Then $\bar{\pi}^*$ is optimal for the Robust MDP on $(\mathcal{S}, \mathcal{A}^D)$ defined by $\mathcal{U}^{\text{aug}}$:*

$$\bar{\pi}^* \in \arg\max_{\bar{\pi}} \min_{(\tilde{P},\tilde{R})\in\mathcal{U}^{\text{aug}}} J_{\text{avg}}^{\text{MDP}}(\bar{\pi}, \tilde{P}, \tilde{R}).$$

Proof. The proof is identical to that of Theorem 3 in Sec. B.4, with the per-decision action a replaced throughout by the chunk $\mathbf{a}$ and $(P_a)^d$ replaced by $P^{(d)}(\cdot|s,\mathbf{a})$. The renewal-reward identity, the fixed-mean simplification of the denominator, and the image-of-Φ argument all carry over unchanged. □

Back-projection to the original action space. Theorem 5 establishes the equivalence cleanly, but the robust MDP it constructs lives on the augmented action space $\mathcal{A}^D$, which is exponentially larger than $\mathcal{A}$ and mixes the agent's executed action with its hypothetical future commitments. We now show that, by marginalizing $\bar{\pi}^*$'s tail, the equivalence projects cleanly back to a robust MDP on the original $(\mathcal{S}, \mathcal{A})$.

For any adversary $\tau \in \mathcal{U}_{c,D}$, define the effective kernel and reward on $(\mathcal{S}, \mathcal{A})$ induced by $\bar{\pi}^*$:

$$\tilde{P}^{\tau,\bar{\pi}^*}(s'|s,a) := \sum_{d=1}^{D}\sum_{\mathbf{a}_{2:D}} \tau_s(d)\, \bar{\pi}^*(\mathbf{a}_{2:D}|s,a)\, P^{(d)}(s'|s,a,\mathbf{a}_{2:D}), \tag{13}$$

$$\tilde{R}^{\tau,\bar{\pi}^*}(s,a) := \frac{1}{c}\sum_{d=1}^{D}\sum_{\mathbf{a}_{2:D}} \tau_s(d)\, \bar{\pi}^*(\mathbf{a}_{2:D}|s,a)\, \bar{R}(s,a,\mathbf{a}_{2:D},d). \tag{14}$$

Here we have written the chunk as $(a, \mathbf{a}_{2:D})$, separating the first action a from the tail $\mathbf{a}_{2:D} \in \mathcal{A}^{D-1}$, and we use $\bar{\pi}^*(\mathbf{a}_{2:D}|s,a)$ to denote the conditional distribution of the tail given that the first action is a. Again mirroring Theorem 3, let

$$\mathcal{U}_s^{\bar{\pi}^*} := \Big\{ \big(\tilde{P}^{\tau,\bar{\pi}^*}(\cdot|s,\cdot),\, \tilde{R}^{\tau,\bar{\pi}^*}(s,\cdot)\big) \;\Big|\; \tau_s \in \mathcal{U}_{c,D,s} \Big\}, \qquad \mathcal{U}^{\bar{\pi}^*} := \bigtimes_{s\in\mathcal{S}} \mathcal{U}_s^{\bar{\pi}^*}.$$

Theorem 6 (Chunking equivalence, projected)**.** *Let $\bar{\pi}^*$ be the optimal policy for the DTR-SMDP under action chunking, and let $\pi^*(a|s) := \sum_{\mathbf{a}_{2:D}} \bar{\pi}^*(a, \mathbf{a}_{2:D}|s)$ be its marginal on the first action. Then π^* is optimal for the Robust MDP on $(\mathcal{S}, \mathcal{A})$ defined by $\mathcal{U}^{\bar{\pi}^*}$:*

$$\pi^* \in \arg\max_{\pi} \min_{(\tilde{P},\tilde{R})\in\mathcal{U}^{\bar{\pi}^*}} J_{\text{avg}}^{\text{MDP}}(\pi, \tilde{P}, \tilde{R}).$$

Proof. We proceed in three steps. First, we show that for any adversary τ, the SMDP return of $\bar{\pi}^*$ matches the robust-MDP return of π^* under the projected dynamics. Second, we extend this matching to a class of augmented policies that share $\bar{\pi}^*$'s tail. Third, we conclude optimality by contradiction.

Step 1: Return matching for $\bar{\pi}^*$. Fix any $\tau \in \mathcal{U}_{c,D}$. By the renewal-reward identity for SMDPs (cf. the proof of Theorem 3), the average return of $\bar{\pi}^*$ in the DTR-SMDP under τ is

$$J_{\text{avg}}^{\text{SMDP}}(\bar{\pi}^*, \tau) = \frac{\sum_s \tilde{d}_\tau^{\bar{\pi}^*}(s) \sum_{\mathbf{a}} \bar{\pi}^*(\mathbf{a}|s) \sum_d \tau_s(d) \bar{R}(s, \mathbf{a}, d)}{c}, \tag{15}$$

where $\tilde{d}_\tau^{\bar{\pi}^*}$ is the stationary distribution over decision-time states induced by $\bar{\pi}^*$ and τ. Splitting the chunk as $\mathbf{a} = (a, \mathbf{a}_{2:D})$ and using $\bar{\pi}^*(\mathbf{a}|s) = \pi^*(a|s)\, \bar{\pi}^*(\mathbf{a}_{2:D}|s, a)$, the numerator becomes

$$\sum_s \tilde{d}_\tau^{\bar{\pi}^*}(s) \sum_a \pi^*(a|s) \underbrace{\Big(\sum_{\mathbf{a}_{2:D}} \bar{\pi}^*(\mathbf{a}_{2:D}|s, a) \sum_d \tau_s(d) \bar{R}(s, a, \mathbf{a}_{2:D}, d) \Big)}_{= c\, \tilde{R}^{\tau, \bar{\pi}^*}(s,a)},$$

so equation 15 simplifies to $\langle \tilde{d}_\tau^{\bar{\pi}^*} \otimes \pi^*,\ \tilde{R}^{\tau, \bar{\pi}^*} \rangle$. The same chunk-splitting applied to the SMDP transition kernel between decision states yields exactly $\tilde{P}^{\tau, \bar{\pi}^*}$ from equation 13, so $\tilde{d}_\tau^{\bar{\pi}^*}$ is also the stationary distribution of the chain on $\mathcal{S}$ induced by π^* and $\tilde{P}^{\tau, \bar{\pi}^*}$. Therefore

$$J_{\text{avg}}^{\text{SMDP}}(\bar{\pi}^*, \tau) = J_{\text{avg}}^{\text{MDP}}(\pi^*, \tilde{P}^{\tau, \bar{\pi}^*}, \tilde{R}^{\tau, \bar{\pi}^*}). \tag{16}$$

Step 2: Return matching for tail-aligned policies. For any policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ on the original action space, define the augmented policy

$$\bar{\pi}(a, \mathbf{a}_{2:D}|s) \ :=\ \pi(a|s)\, \bar{\pi}^*(\mathbf{a}_{2:D}|s, a).$$

That is, $\bar{\pi}$ uses π for the first action and inherits $\bar{\pi}^*$'s tail. Repeating the argument of Step 1 with $\bar{\pi}$ in place of $\bar{\pi}^*$, the tail factor $\bar{\pi}^*(\mathbf{a}_{2:D}|s, a)$ that appears in the inner sum is unchanged — it is the *same* tail used to define $\tilde{P}^{\tau, \bar{\pi}^*}$ and $\tilde{R}^{\tau, \bar{\pi}^*}$. Hence for every $\tau \in \mathcal{U}_{c,D}$ and every π,

$$J_{\text{avg}}^{\text{SMDP}}(\bar{\pi}, \tau) = J_{\text{avg}}^{\text{MDP}}(\pi, \tilde{P}^{\tau, \bar{\pi}^*}, \tilde{R}^{\tau, \bar{\pi}^*}). \tag{17}$$

Note that the right-hand side ranges over the entire uncertainty set $\mathcal{U}^{\bar{\pi}^*}$ as τ ranges over $\mathcal{U}_{c,D}$, by construction.

Step 3: Optimality of π^* by contradiction. Suppose, toward contradiction, that π^* is not optimal for the robust MDP defined by $\mathcal{U}^{\bar{\pi}^*}$. Then there exists a policy π' on $\mathcal{S} \to \Delta_{\mathcal{A}}$ such that

$$\min_{(\tilde{P}, \tilde{R}) \in \mathcal{U}^{\bar{\pi}^*}} J_{\text{avg}}^{\text{MDP}}(\pi', \tilde{P}, \tilde{R}) \ >\ \min_{(\tilde{P}, \tilde{R}) \in \mathcal{U}^{\bar{\pi}^*}} J_{\text{avg}}^{\text{MDP}}(\pi^*, \tilde{P}, \tilde{R}). \tag{18}$$

Construct the augmented policy $\bar{\pi}'(a, \mathbf{a}_{2:D}|s) := \pi'(a|s)\, \bar{\pi}^*(\mathbf{a}_{2:D}|s, a)$. By Step 2 applied to both π' and π^*, taking the minimum over τ on each side of equation 17 gives

$$\min_{\tau \in \mathcal{U}_{c,D}} J_{\text{avg}}^{\text{SMDP}}(\bar{\pi}', \tau) = \min_{(\tilde{P}, \tilde{R}) \in \mathcal{U}^{\bar{\pi}^*}} J_{\text{avg}}^{\text{MDP}}(\pi', \tilde{P}, \tilde{R}),$$
$$\min_{\tau \in \mathcal{U}_{c,D}} J_{\text{avg}}^{\text{SMDP}}(\bar{\pi}^*, \tau) = \min_{(\tilde{P}, \tilde{R}) \in \mathcal{U}^{\bar{\pi}^*}} J_{\text{avg}}^{\text{MDP}}(\pi^*, \tilde{P}, \tilde{R}).$$

Combined with equation 18, this yields

$$\min_{\tau \in \mathcal{U}_{c,D}} J_{\text{avg}}^{\text{SMDP}}(\bar{\pi}', \tau) \ >\ \min_{\tau \in \mathcal{U}_{c,D}} J_{\text{avg}}^{\text{SMDP}}(\bar{\pi}^*, \tau),$$

contradicting the optimality of $\bar{\pi}^*$ in the DTR-SMDP. Therefore π^* is optimal for the robust MDP defined by $\mathcal{U}^{\bar{\pi}^*}$. □

Why this projected form is interesting. The uncertainty set $\mathcal{U}^{\bar{\pi}^*}$ has a clean physical interpretation: each kernel $\tilde{P}^{\tau,\bar{\pi}^*}(s'|s,a)$ answers the question *"if I take action a now, and then proceed optimally in open loop for an adversarially chosen number of steps, where will I end up?"* The agent is robust against precisely the family of dynamics perturbations one would expect to arise from temporally extended commitments under uncertain timing. Unlike standard L_p-ball uncertainty sets, this set is automatically restricted to the manifold of states reachable under the agent's own optimal behavior, addressing the conservatism problem identified in Kumar et al. (2025).

The dependence of $\mathcal{U}^{\bar{\pi}^*}$ on $\bar{\pi}^*$ deserves comment: it might appear circular, since the uncertainty set depends on the very policy we are trying to find. We emphasize that this dependence is not a deficiency but rather encodes the structure of the problem, in the spirit of policy-induced regularization (Eysenbach & Levine, 2022). More importantly, the dependence has no operational consequence at deployment: the agent only ever executes the first-action marginal π^*, and never needs to know $\bar{\pi}^*$ to act. The augmented policy $\bar{\pi}^*$ is a training-time scaffold that defines the uncertainty set against which π^* is robust; it is not required for inference.

This suggests a particularly clean deployment scenario: an agent is *trained* in the DTR-SMDP setting with action chunking against an adversarial clock, but *deployed* in a standard, synchronous environment ($\tau \equiv 1$). At deployment, the agent simply executes π^* at every timestep. By Theorem 6, this deployed policy is automatically robust to the dynamics uncertainty set $\mathcal{U}^{\bar{\pi}^*}$ — which, as discussed above, captures physically reachable perturbations rather than abstract norm-ball distortions. This realizes the "train against a clock, deploy against physics" program advertised in Sec. 3.4, and does so without ever requiring the deployed agent to operate in the augmented action space.

B.4.2 Equivalence in the discounted case

We now turn to the discounted return criterion. The structure of the equivalence carries over, but the discount factor introduces a new coupling between the adversary's choice of timing and the effective horizon of the problem. We first explain why the constraint $\mathbb{E}_\tau[d] = c$ used in the average-reward analysis no longer suffices, then introduce a discount-adapted constraint that restores the equivalence, and finally restate the three results (Theorem 3, Theorem 5, Theorem 6) under the new assumption.

The obstacle: a τ-dependent effective discount. Consider the optimal Bellman operator for the DTR-SMDP under the discounted return criterion (action chunking; the action repetition case is a specialization):

$$(TV)(s) = \max_{\bar{\pi}} \min_{\tau_s \in \mathcal{U}_{c,D,s}} \sum_{\mathbf{a}} \bar{\pi}(\mathbf{a}|s) \sum_{d=1}^{D} \tau_s(d) \Big[\bar{R}(s,\mathbf{a},d) + \gamma^d \sum_{s'} P^{(d)}(s'|s,\mathbf{a}) V(s') \Big]. \quad (19)$$

For the average-reward equivalence, marginalizing over τ produced a clean robust MDP with a fixed effective kernel and reward. The same attempt here yields:

$$(TV)(s) = \max_{\bar{\pi}} \min_{\tau_s \in \mathcal{U}_{c,D,s}} \sum_{\mathbf{a}} \bar{\pi}(\mathbf{a}|s) \Big[\tilde{R}^\tau(s,\mathbf{a}) + \big(\textstyle\sum_d \tau_s(d)\gamma^d \big) \sum_{s'} \hat{P}^\tau(s'|s,\mathbf{a}) V(s') \Big], \quad (20)$$

where $\tilde{R}^\tau(s,\mathbf{a}) := \sum_d \tau_s(d) \bar{R}(s,\mathbf{a},d)$ and $\hat{P}^\tau(s'|s,\mathbf{a}) := \frac{\sum_d \tau_s(d)\gamma^d P^{(d)}(s'|s,\mathbf{a})}{\sum_d \tau_s(d)\gamma^d}$ is a discount-weighted normalization of the multi-step kernel.

The issue is the parenthesized factor $\sum_d \tau_s(d)\gamma^d$, the *effective discount* induced by τ_s. Under the constraint $\mathbb{E}_{d\sim\tau_s}[d] = c$, this quantity is not fixed: adversaries with the same mean delay c can produce different effective discounts. As a result, equation 20 does not match the form of a standard robust MDP, which assumes a fixed discount factor γ. While one could in principle define a generalized robust MDP in which the discount factor is itself part of the uncertainty set, such an object is not standard in the robust-RL literature, and an equivalence to it would weaken the bridge between DTR-SMDPs and the existing body of work we wish to connect to.

Discount-adapted constraint. We resolve this by replacing the mean-delay constraint with a constraint on the effective discount itself. For a fixed *effective horizon* $c \in [1, D]$, define

$$\mathcal{U}^{\gamma}_{c,D} := \Big\{\tau \in \mathcal{U}_D \ : \ \mathbb{E}_{d\sim\tau_s}[\gamma^d] = \gamma^c \quad \forall s \in \mathcal{S}\Big\}.$$

This constraint pins down the effective discount that any feasible adversary can induce: every $\tau \in \mathcal{U}^{\gamma}_{c,D}$ produces exactly the same effective discount $\tilde{\gamma} := \gamma^c$, matching what one would obtain if the timing were deterministically equal to c. Beyond restoring the equivalence, this constraint is arguably the natural one for the discounted setting: the discounted return is fundamentally a statement about effective horizon, not raw elapsed time, and constraining $\mathbb{E}_\tau[\gamma^d]$ controls the quantity that actually appears in the value function. We retain the notation c for this effective horizon to mirror the average-reward case, and note that deterministic timing $\tau \equiv c$ is feasible under the discount-adapted constraint whenever c is an integer, just as it is under the mean-delay constraint — though in neither case is it the only feasible adversary.

Restated results. Under the discount-adapted constraint $\mathcal{U}^{\gamma}_{c,D}$, the parenthesized factor in equation 20 is constant across the uncertainty set, and the marginalization argument used for the average-reward case goes through with minor modifications. We restate the three equivalence results below.

For any $\tau \in \mathcal{U}^{\gamma}_{c,D}$, define the effective discounted kernel and reward:

$$\tilde{P}^{\tau}(s'|s,\mathbf{a}) := \frac{1}{\gamma^c}\sum_{d=1}^{D} \tau_s(d)\,\gamma^d\,P^{(d)}(s'|s,\mathbf{a}), \tag{21}$$

$$\tilde{R}^{\tau}(s,\mathbf{a}) := \sum_{d=1}^{D} \tau_s(d)\,\bar{R}(s,\mathbf{a},d). \tag{22}$$

The kernel $\tilde{P}^{\tau}$ is a valid stochastic matrix because the γ^c normalization is exactly the row-sum of the unnormalized weights (since $\sum_{s'} P^{(d)}(s'|s,\mathbf{a}) = 1$ and $\sum_d \tau_s(d)\gamma^d = \gamma^c$); on $\mathcal{U}^{\gamma}_{c,D}$ it therefore coincides with the normalized kernel $\hat{P}^{\tau}$ introduced below equation 20. The reward $\tilde{R}^{\tau}$ inherits no $1/c$ rescaling because, unlike the average-reward case, there is no SMDP renewal-reward identity to absorb.

Mirroring Theorem 3, let

$$\mathcal{U}^{\gamma,\mathrm{aug}}_s := \Big\{\big(\tilde{P}^{\tau}(\cdot|s,\cdot),\ \tilde{R}^{\tau}(s,\cdot)\big) \ \Big|\ \tau_s \in \mathcal{U}^{\gamma}_{c,D,s}\Big\}, \qquad \mathcal{U}^{\gamma,\mathrm{aug}} := \bigtimes_{s\in\mathcal{S}} \mathcal{U}^{\gamma,\mathrm{aug}}_s,$$

where $\mathcal{U}^{\gamma}_{c,D,s} := \{\tau_s \in \Delta_{[D]} : \mathbb{E}_{d\sim\tau_s}[\gamma^d] = \gamma^c\}$. This set lives on the augmented space $(\mathcal{S}, \mathcal{A}^D)$; the action-repetition sets $\mathcal{U}^{\gamma}_s$ and $\mathcal{U}^{\gamma} = \bigtimes_{s\in\mathcal{S}} \mathcal{U}^{\gamma}_s$ on $(\mathcal{S},\mathcal{A})$ are obtained by specializing $\mathbf{a} = (a,\dots,a)$, so that $P^{(d)}(\cdot|s,\mathbf{a}) = (P_a)^d(\cdot|s)$. We write $\mathcal{P}^{\gamma}_s$ and $\mathcal{R}^{\gamma}_s$ for the marginal projections of $\mathcal{U}^{\gamma}_s$. For the projected (back-to-$\mathcal{A}$) version, define $\mathcal{U}^{\gamma,\bar{\pi}^*}$ analogously, again as coupled pairs, by marginalizing equation 21–equation 22 against $\bar{\pi}^*$'s tail.

Theorem 7 (Discounted equivalence; action repetition)**.** *Under action repetition with the constraint $\tau \in \mathcal{U}^{\gamma}_{c,D}$, the optimal policy π^* for the DTR-SMDP under the discounted return J_γ is also optimal for the Robust MDP on $(\mathcal{S},\mathcal{A})$ defined by $\mathcal{U}^{\gamma}$ and discount $\tilde{\gamma} = \gamma^c$.*

Theorem 8 (Discounted equivalence; chunking, augmented space)**.** *Under action chunking with the constraint $\tau \in \mathcal{U}^{\gamma}_{c,D}$, the optimal policy $\bar{\pi}^*$ for the DTR-SMDP under J_γ is also optimal for the Robust MDP on $(\mathcal{S},\mathcal{A}^D)$ defined by $\mathcal{U}^{\gamma,\mathrm{aug}}$ and discount $\tilde{\gamma}$.*

Theorem 9 (Discounted equivalence; chunking, projected)**.** *Under action chunking with the constraint $\tau \in \mathcal{U}^{\gamma}_{c,D}$, the first-action marginal π^* of the optimal augmented policy $\bar{\pi}^*$ is optimal for the Robust MDP on $(\mathcal{S},\mathcal{A})$ defined by $\mathcal{U}^{\gamma,\bar{\pi}^*}$ and discount $\tilde{\gamma}$.*

Proof sketch. The proofs follow exactly those of Theorems 3, 5, and 6 respectively, with three modifications. (i) In place of the SMDP renewal-reward identity used in the average-reward

case, the discounted return matches via direct expansion of equation 19: plugging definitions equation 21–equation 22 into the standard discounted Bellman equation with discount $\tilde{\gamma} = \gamma^c$ recovers equation 19. (ii) The fixed-mean simplification $\mathbb{E}_{d\sim\tau_s}[d] = c$ is replaced by $\mathbb{E}_{d\sim\tau_s}[\gamma^d] = \gamma^c$, which holds by the constraint defining $\mathcal{U}^{\gamma}_{c,D}$. (iii) The rectangularity-preservation argument is unchanged, since $\mathcal{U}^{\gamma}_{c,D}$ factorizes across states by the same state-wise definition. □

Loss of interpretability. While the discounted equivalence holds under the discount-adapted constraint, we note that the resulting uncertainty set $\mathcal{P}^{\gamma}_s$ is somewhat less interpretable than its average-reward counterpart. The effective kernel equation 21 weights multi-step transitions by γ^d before normalization, so transitions over short delays are systematically up-weighted relative to long ones. The robust agent is asked to survive a transition model that not only blurs over reachable states (as in the average-reward case) but also biases that blurring toward short-horizon trajectories. This bias is mathematically natural — it is exactly the bias a discounted-return agent already applies to its own value estimates — but it makes the uncertainty set harder to reason about as a stand-alone object, which is why the average-reward analysis serves as the cleaner exposition of the temporal–dynamics duality.

C Experiments

The code for our experiments is available at `https://anonymous.4open.science/r/dtr-code-release-6E74/`.

C.1 Per-Environment and Per-Cell Results

Table 4 gives the per-environment nominal and worst-case returns from which the aggregates of Table 1 are computed. Tables 5 and 6 give the per-cell returns from which the aggregates of Table 2 are computed. Table 7 reports the same two perturbation families on InvertedPendulum, which is excluded from that aggregate.

Table 4: Per-environment temporal robustness: nominal and worst-case episodic return over 20 seeds. Nominal is the undelayed clock $\tau \equiv 1$. Worst-case is the per-seed minimum over the three mean-matched schedules of Section 4.2 — constant ($\tau \equiv 2$), uniform ($\tau \sim \mathrm{Unif}\{1,2,3\}$) and boundary ($d \in \{1,4\}$ with $\mathbb{E}[d] = 2$) — taken before averaging across seeds. Entries are the mean ± the half-width of a 95% percentile bootstrap over 20 seeds, 50k replications. InvertedPendulum worst-case values are given to one decimal as they lie near zero. † DR-seen is in-distribution for the dynamics perturbations of Section 4.3 and is an upper reference rather than a zero-shot comparison.

Method	HalfCheetah		Hopper		Walker2d		InvPendulum	
	Nom.	Wst.	Nom.	Wst.	Nom.	Wst.	Nom.	Wst.
Vanilla PPO	1696±286	129±40	2145±212	201±56	3109±345	405±80	975±16	7.7±0.4
Fixed delay	500±202	165±88	2452±315	372±71	**3726±400**	979±124	**1000±0**	15.7±0.9
Uniform delay	1344±200	587±80	2999±236	883±55	3722±341	1533±176	**1000±0**	37.2±2.2
PR-MDP	1550±267	266±53	2291±275	452±88	1755±285	740±74	874±115	17.3±1.1
SA-PPO	1477±274	82±53	1833±128	303±33	2463±167	537±78	741±31	5.7±0.3
RNAC-IPM	**1923±376**	118±73	2354±216	224±51	2763±303	451±84	869±37	4.7±0.3
DR-unseen	1677±236	121±48	1527±213	252±82	2144±246	472±74	944±20	4.7±0.4
DR-seen†	1157±235	92±86	623±246	291±48	993±278	603±60	998±4	8.1±0.7
DTR (single-head)	1229±102	644±44	2902±265	1656±170	2889±461	1921±207	**1000±0**	**741.5±34.3**
DTR (two-head)	1175±118	**669±47**	**3109±237**	**1720±133**	3274±408	**2005±237**	**1000±0**	701.3±44.5

Table 5: Per-cell episodic return under action noise: the executed action is replaced by a uniformly random action with probability α. Entries are the mean ± the half-width of a 95% percentile bootstrap over 20 seeds, 50k replications. **Bold** is the best per column among the zero-shot methods; underline is the runner-up. † DR-seen is in-distribution for these cells, making it an upper reference rather than a zero-shot comparison; it is excluded from the bolding.

	HalfCheetah			Hopper			Walker2d		
Method	α=0.2	α=0.3	α=0.5	α=0.2	α=0.3	α=0.5	α=0.2	α=0.3	α=0.5
Vanilla PPO	759±46	477±37	104±32	515±32	367±36	221±21	488±59	384±27	254±18
Fixed delay	281±113	183±85	11±38	563±20	423±17	235±16	548±42	406±18	281±16
Uniform delay	792±85	573±69	212±39	645±44	449±28	254±24	598±48	419±23	284±19
PR-MDP	**929±133**	**651±84**	250±46	**1058±90**	**651±39**	**316±15**	**872±97**	**593±51**	**322±18**
SA-PPO	684±71	431±63	84±43	576±29	432±16	248±11	540±26	402±25	220±20
RNAC-IPM	741±64	468±61	111±42	553±33	379±22	241±19	524±38	409±26	251±22
DR-unseen	772±88	505±74	124±49	765±67	525±37	298±28	687±63	489±26	301±30
DR-seen†	641±103	492±74	237±61	999±105	845±59	458±33	804±102	610±63	375±19
DTR (single-head)	791±34	603±24	**276±20**	724±40	492±18	266±22	619±67	467±41	291±19
DTR (two-head)	748±31	575±26	266±19	777±39	501±21	294±24	609±54	457±31	290±15

Table 6: Per-cell episodic return when the model's gravity is scaled by g. Entries are the mean ± the half-width of a 95% percentile bootstrap over 20 seeds, 50k replications. **Bold** is the best per column among the zero-shot methods; underline is the runner-up. † DR-seen is in-distribution for these cells, making it an upper reference rather than a zero-shot comparison; it is excluded from the bolding.

	HalfCheetah			Hopper			Walker2d		
Method	g=1.25	g=1.5	g=2	g=1.25	g=1.5	g=2	g=1.25	g=1.5	g=2
Vanilla PPO	1558±293	1216±298	162±263	881±193	510±59	337±14	917±233	401±35	296±21
Fixed delay	497±231	375±236	143±244	938±153	487±26	319±7	1208±286	456±28	330±21
Uniform delay	1442±211	1316±188	559±308	1390±347	538±56	333±11	1302±301	520±95	329±21
PR-MDP	1438±316	1193±357	370±337	**2212±336**	**1563±393**	**428±32**	**1957±383**	**1237±318**	373±35
SA-PPO	1424±321	1241±289	540±296	1004±114	553±48	339±7	1095±148	487±54	304±24
RNAC-IPM	**1824±379**	**1497±322**	−1±194	820±148	481±54	330±8	937±180	461±36	320±16
DR-unseen	1639±270	1469±294	617±339	1351±203	960±226	373±27	1736±326	842±148	376±30
DR-seen†	1469±273	1772±316	1696±311	1628±319	1563±393	850±238	1304±334	1270±374	837±303
DTR (single-head)	1325±95	1294±101	**730±276**	1712±315	628±53	333±7	1543±434	621±89	**383±30**
DTR (two-head)	1312±145	1280±158	622±279	1996±334	656±87	338±6	1564±405	557±51	361±18

Table 7: Per-cell episodic return on InvertedPendulum under action noise (the executed action is replaced by a uniformly random action with probability α) and gravity scaling (g). Entries are the mean ± the half-width of a 95% percentile bootstrap over 20 seeds, 50k replications. Return is the number of steps survived, capped at 1000. These cells are excluded from the aggregate of Table 2: under gravity scaling the methods that solve the task tie at the cap, while under action noise no method survives past step 135, so neither family ranks the methods. **Bold** is the best per column among the zero-shot methods, underline the runner-up; ties at the cap are all marked. † DR-seen randomizes gravity and action noise during training, so it is in-distribution for every column here and is excluded from the bolding.

	Action noise			Gravity scale		
Method	α=0.2	α=0.3	α=0.5	g=1.25	g=1.5	g=2
Vanilla PPO	62±5	34±3	15±1	944±27	941±38	833±75
Fixed delay	121±12	50±4	19±1	**1000±0**	**1000±0**	**1000±0**
Uniform delay	94±6	37±3	17±1	**1000±0**	**1000±0**	**1000±0**
PR-MDP	**135±12**	**60±4**	**23±1**	871±116	864±119	865±119
SA-PPO	48±2	24±1	11±1	761±27	787±37	702±30
RNAC-IPM	43±3	23±2	9±1	935±22	910±35	827±47
DR-unseen	45±4	24±2	10±1	939±19	952±29	869±45
DR-seen†	87±5	42±4	18±2	998±4	1000±0	998±4
DTR (single-head)	60±4	34±3	15±1	**1000±0**	**1000±0**	935±82
DTR (two-head)	65±5	34±2	15±1	**1000±0**	**1000±0**	**1000±0**

Table 8: **Per-state support of the trained adversary.** Percentage of visited states assigned to each of the three per-state distributions admitted by Corollary 2 at $c = 2$, $D = 4$, at a 90% mass threshold; Σ is their total. Entries are means over 20 seeds; $\pm$ is a 95% confidence interval over seeds, and the bracket on the aggregate rows is a stratified bootstrap that resamples seeds within each environment. The aggregate weights each environment equally.

		% of visited states (threshold 0.90)			
Environment	Adversary	δ_2	Bern$(1,3)$	Bern$(1,4)$	Σ
InvertedPendulum	two-head	0.1	0.0	98.5	98.6 $\pm$1.6
InvertedPendulum	single-head	0.0	0.0	99.9	99.9 $\pm$0.1
Hopper	two-head	3.2	6.2	78.0	87.4 $\pm$1.9
Hopper	single-head	1.8	0.0	86.8	88.6 $\pm$4.6
HalfCheetah	two-head	8.1	8.4	69.8	86.4 $\pm$2.0
HalfCheetah	single-head	10.4	0.1	73.2	83.7 $\pm$2.3
Walker2d	two-head	4.0	7.1	67.4	78.5 $\pm$3.4
Walker2d	single-head	4.5	0.0	73.9	78.4 $\pm$5.8
All (equal weight)	two-head	3.9	5.4	78.4	**87.7** [86.5, 88.8]
All (equal weight)	single-head	4.2	0.0	83.5	**87.7** [85.7, 89.4]

C.2 Adversary Behavior Analysis

We examine whether the learned adversary recovers the structure predicted by Corollary 2, which states that an optimal per-state distribution has support of size at most two: either deterministic at $d = c$, or a Bernoulli on (d_1, d_2) with $d_1 < c < d_2$. At the configuration used throughout our experiments ($c = 2$, $D = 4$) the mean constraint $\mathbb{E}[\tau_s] = c$ closes this family completely. A distribution on $\{1, \ldots, 4\}$ with mean exactly 2 and support of size at most two must be one of

$$\delta_2, \qquad \text{Bernoulli}(1,3) \text{ with weights } (\tfrac{1}{2}, \tfrac{1}{2}), \qquad \text{Bernoulli}(1,4) \text{ with weights } (\tfrac{2}{3}, \tfrac{1}{3}), \quad (23)$$

Corollary 2 therefore makes a prediction with no free parameters, and one that is checkable state by state.

Protocol. We roll out each trained agent together with its own trained adversary and record the adversary's full categorical distribution over delays at every decision point. We evaluate 20 seeds on each of four environments for both adversary architectures, 50 episodes per run. Each visited state is then assigned to one of the three distributions in equation 23 if that distribution's support carries at least 90% of the state's probability mass, and to a residual bucket otherwise. The four buckets are exhaustive and mutually exclusive. We report concentration rather than exact support because a softmax head cannot represent an exactly sparse distribution.

Results. Table 8 reports the taxonomy. Averaged over environments, 87.7% of visited states fall on one of the three admissible supports, for both the two-head architecture (95% CI $[86.5, 88.8]$) and the single-head architecture ($[85.7, 89.4]$). The dominant mode is the *extremal* straddle Bernoulli$(1,4)$, at 78.4% and 83.5% of states respectively: when the adversary randomizes, it almost always places its budget on the two temporal extremes rather than on the interior pair $(1,3)$. The single-head figure is the stronger of the two: its adversary is an unconstrained categorical over $\{1, \ldots, 4\}$ with the mean enforced only by a soft penalty, so nothing in its parameterization favors a sparse solution. We read this as convergence to the structure Corollary 2 predicts, obtained despite two features of the training setup that push against it: a softmax head, which cannot place exactly zero mass on a delay, and an entropy bonus on the adversary's objective that explicitly rewards spreading mass across delays.

Table 9: Support sweep at fixed $c = 2$, HalfCheetah. Episodic return, mean ± the half-width of a 95% percentile bootstrap over 20 seeds, 50k replications. Nominal is the undelayed clock $\tau \equiv 1$, which has mean 1 and is therefore not a member of $\mathcal{U}_{2,D}$; it is left unmarked, as the quantity of interest in that column is the absence of a trend rather than a winner. Worst case at test support D_{test} is the per-seed minimum over the three mean-2 schedules supported on $[1, D_{\text{test}}]$ — constant ($\tau \equiv 2$), uniform ($\tau \sim \text{Unif}\{1,2,3\}$) and boundary ($d \in \{1, D_{\text{test}}\}$ with $\mathbb{E}[d] = 2$) — taken before averaging across seeds. † marks the diagonal, where the training support equals the test support and the configuration is evaluated at the operating point it was trained for. **Bold** is the best per worst-case column, underline the runner-up.

	Nominal	Worst case at test support D_{test}			
Training config	$\tau \equiv 1$	3	4	6	8
$D = 3$, two-head	1320±193	831±75†	653±75	392±35	325±19
$D = 3$, single	1209±176	**846±103**†	654±86	395±43	311±16
$D = 4$, two-head	1133±103	697±31	**659±38**†	428±36	350±27
$D = 4$, single	1227±101	674±39	648±38†	461±43	372±24
$D = 6$, two-head	1303±140	595±31	581±30	**504±39**†	426±22
$D = 6$, single	1152±47	535±24	539±20	469±20†	403±19
$D = 8$, two-head	1190±44	558±30	531±24	461±18	423±12†
$D = 8$, single	1212±58	540±24	528±20	477±18	**439±17**†
Uniform$\{1,2,3\}$ ($c = 2$)	1319±197	830±88	605±98	357±51	312±30

C.3 Sensitivity to the temporal uncertainty set

The pair (c, D) is the one thing a practitioner must supply to instantiate $\mathcal{U}_{c,D}$, and Sec. 4.1 fixes it at $c = 2$, $D = 4$ throughout. This raises two distinct questions, which we separate here: whether the conclusions of Sec. 4.2 depend on that particular choice, and what it costs to specify the set wrongly — to train against an uncertainty set that does not match the clock encountered at deployment.

Setup. We sweep both parameters on HalfCheetah, with every run using the single-file CleanRL PPO backbone, hyperparameters and 10^6-decision budget of Appx. C.7 and 20 seeds per configuration: the support $D \in \{3, 4, 6, 8\}$ at fixed $c = 2$, and the budget $c \in \{1.5, 2, 3\}$ at fixed $D = 4$, for both adversary architectures. We begin the support sweep at $D = 3$ because $D = 2$ is degenerate at $c = 2$: the only distribution on $\{1, 2\}$ with mean 2 is the point mass at 2, which recovers the Fixed-delay baseline exactly, so $D = 3$ is the smallest support that leaves the adversary any freedom. For each configuration we also train the matched Uniform-delay baseline $\tau \sim \text{Unif}\{1, \ldots, 2c-1\}$ — the same mean-delay budget with random rather than adversarial delays. It is the strongest baseline of Sec. 4.2 and is the comparison of interest throughout.

Evaluation. Every checkpoint is evaluated zero-shot on the nominal clock $\tau \equiv 1$ and on the mean-matched schedules of Sec. 4.2 — constant ($\tau \equiv c$), uniform, and the boundary schedule on $\{1, D_{\text{test}}\}$ predicted by Corollary 2 — instantiated at every test support $D_{\text{test}} \in \{3, 4, 6, 8\}$ and every test budget $c_{\text{test}} \in \{1.5, 2, 3\}$. As in Sec. 4.2, the worst case at a given $(c_{\text{test}}, D_{\text{test}})$ is the per-seed minimum over the mean-matched schedules supported on $[1, D_{\text{test}}]$, taken before averaging across seeds; each condition is scored over 20 evaluation episodes and the unit of analysis is the seed. Entries are means over seeds rather than interquartile means, with 95% percentile bootstrap intervals over the 20 seeds, 50k replications.

Varying the support D at fixed $c = 2$. Reading down each worst-case column of Table 9, the configuration trained at that test support is the best of the eight, and it beats the matched Uniform baseline in every column. What changes is the margin: +2%, +9%, +41% and +41% at $D_{\text{test}} = 3, 4, 6, 8$ respectively, of which only the first two have overlapping intervals. The likely reason is that Unif$\{1, 2, 3\}$ is nearly in-distribution for the $D_{\text{test}} = 3$ schedules, so randomization is close to sufficient there, while the gap between the baseline's delay distribution and the worst-case schedule widens with the support. Enlarging the support is not paid for at the nominal clock: nominal return across $D \in \{3, 4, 6, 8\}$ spans 1133–1320 over both architectures, mean 1218, with no monotone trend, against 1319 for the matched Uniform baseline and 1696 for Vanilla PPO (Table 4). As an internal consistency

Table 10: Budget sweep at fixed $D = 4$, HalfCheetah. Episodic return, mean ± the half-width of a 95% percentile bootstrap over 20 seeds, 50k replications; nominal left unmarked as in Table 9. Worst case at test budget c_{test} is the per-seed minimum over the mean-c_{test} schedules supported on $[1, 4]$: {Unif$\{1, 2\}$, boundary} at $c_{\text{test}} = 1.5$, no integer constant schedule existing there; {constant, Unif$\{1, 2, 3\}$, boundary} at $c_{\text{test}} = 2$; and {constant, boundary} at $c_{\text{test}} = 3$, the baseline's Unif$\{1, \ldots, 5\}$ being excluded as its support exceeds $D = 4$. † marks each row's own operating point; the comparison of interest is a DTR row against the Uniform row carrying the same †. **Bold** is the best per worst-case column, underline the runner-up.

	Nominal	Worst case at test budget c_{test}		
Training config	$\tau \equiv 1$	1.5	2	3
$c = 1.5$, two-head	1278±86	803±41†	555±38	264±39
$c = 1.5$, single	1291±53	832±26†	566±26	279±29
$c = 2$, two-head	1133±103	812±49	659±38†	473±63
$c = 2$, single	1227±101	849±57	648±38†	456±45
$c = 3$, two-head	932±189	**1086±121**	**1074±89**	**1257±133**†
$c = 3$, single	806±146	958±91	908±105	980±132†
Uniform$\{1, 2\}$ ($c = 1.5$)	1546±230	812±56†	484±52	104±47
Uniform$\{1, 2, 3\}$ ($c = 2$)	1319±197	835±108	605±98†	320±82
Uniform$\{1, \ldots, 5\}$ ($c = 3$)	902±129	905±122	879±121	938±165†

check, the $D_{\text{test}} = 4$ column measures the same quantity as the HalfCheetah worst-case column of Table 4, here re-evaluated from an independent set of runs at that configuration; the two agree within their intervals, 648 and 659 against 644 and 669 for the two DTR variants and 605 against 587 for Uniform.

Misspecification, in contrast, does carry a cost, and it is the form over-conservatism takes in this framework. At $D_{\text{test}} = 3$ the agents trained at $D = 8$ retain 540–558 against the 831–846 of the agents trained at $D = 3$ — a loss of about a third — and fall below the Uniform baseline's 830. An agent that has prepared for delays as long as 8 has given up worst-case return on a clock that never exceeds 3. Knowledge of D is therefore worth having, and the support is best set to the longest interruption actually anticipated rather than to a conservative upper bound.

Varying the budget c at fixed $D = 4$. Head-to-head against the matched Uniform baseline at its own operating point — the rows sharing a † in Table 10 — DTR wins or ties at all three budgets: 832 against 812 at $c = 1.5$ (+2%, intervals overlapping), 659 against 605 at $c = 2$ (+9%, intervals overlapping), and 1257 against 938 at $c = 3$ (+34%). The margin grows with the budget because the adversary's freedom does: the largest mass it may place at the boundary is $(c - 1)/(D - 1)$, which rises from 1/6 at $c = 1.5$ to 2/3 at $c = 3$, so the worst-case schedule pulls further from the uniform one and there is correspondingly more for adversarial training to find.

The budget differs from the support in one respect that matters for practice: it is not an operating point that has to be matched. The agents trained at $c = 3$ hold the best worst-case return in every column of Table 10, including the two budgets they never trained on, where a mismatched support costs about a third of the worst-case return in Table 9. Training against a slower clock than the one deployed transfers down; training against a wider support than the one deployed does not. Nominal return does fall with the budget ($1278 \to 1133 \to 932$ for the two-head variant), but this is the price of a slower control clock rather than of worst-case optimization: the randomized baseline at the same budget pays it as well ($1546 \to 1319 \to 902$), and at $c = 3$ DTR is nominally above it. The budget should accordingly be set to the timing one expects at deployment, and any method operating at the same mean decision interval pays comparably for it.

C.4 Evaluation against a learned adversary

The three evaluation schedules of Sec. 4.2 are fixed in advance and applied independently of the state, whereas a DTR-SMDP is defined by *state-dependent* timing. This appendix reports the complementary evaluation: for each frozen agent we train an adversary from

Table 11: **Per-environment return under a learned adversary.** Episodic return induced by a state-dependent adversary trained from scratch against each frozen agent ($D = 4$, $c = 2$, two-head architecture, 300K decisions, 20 evaluation episodes per agent). Entries are means over 20 seeds; brackets are 95% percentile bootstrap CIs over seeds, 50k replications. SA-PPO's negative HalfCheetah entry is a genuine value. InvertedPendulum values are given to one decimal as they lie near zero. † DR-seen is in-distribution for the dynamics perturbations of Sec. 4.3 and is an upper reference rather than a zero-shot comparison.

Method	HalfCheetah	Hopper	Walker2d	InvPendulum
Vanilla PPO	39 [−4, 80]	193 [130, 263]	479 [383, 587]	7.3 [6.8, 7.8]
Fixed delay	132 [44, 222]	356 [286, 429]	1169 [1002, 1345]	13.4 [12.3, 14.4]
Uniform delay	563 [478, 662]	977 [880, 1074]	1776 [1562, 1996]	36.0 [32.6, 39.5]
PR-MDP	210 [155, 273]	411 [293, 541]	793 [705, 889]	16.5 [14.8, 18.2]
SA-PPO	−24 [−72, 24]	300 [249, 353]	641 [556, 725]	5.0 [4.8, 5.2]
RNAC-IPM	20 [−46, 85]	160 [124, 199]	467 [389, 558]	4.2 [4.0, 4.4]
DR-unseen	17 [−33, 69]	247 [161, 347]	539 [461, 618]	3.9 [3.6, 4.1]
DR-seen†	8 [−80, 116]	305 [230, 409]	632 [543, 724]	6.9 [6.0, 7.9]
DTR (single-head)	596 [543, 659]	1887 [1658, 2134]	2328 [2040, 2616]	**813 [763, 862]**
DTR (two-head)	**603 [556, 653]**	**1919 [1770, 2070]**	**2441 [2063, 2827]**	749 [697, 801]

Table 12: **Aggregate temporal robustness under a learned adversary.** Computed on the per-seed scores of Table 11 pooled across the four environments. Scores are min–max normalized within each environment over the seed-level scores of all ten methods under the learned adversary, as each column of Table 1 is normalized on its own (Appx. C.7); compare rankings and probabilities of improvement with that table, not IQM levels. IQM and probability of improvement are computed with *rliable* (Agarwal et al., 2021); stratified bootstrap over 20 seeds, 50k replications. Brackets are 95% CIs. The two DTR variants lie within each other's intervals and both are marked.

Method	IQM	$P(\succ$ Vanilla$)$	$P($DTR two-head $\succ)$
Vanilla PPO	0.055 [0.041, 0.071]	—	1.000 [1.000, 1.000]
Fixed delay	0.132 [0.112, 0.153]	0.832 [0.770, 0.891]	0.972 [0.944, 0.992]
Uniform delay	0.329 [0.301, 0.357]	0.996 [0.988, 1.000]	0.853 [0.791, 0.911]
PR-MDP	0.127 [0.108, 0.149]	0.869 [0.809, 0.921]	0.990 [0.970, 1.000]
SA-PPO	0.086 [0.072, 0.099]	0.453 [0.380, 0.524]	1.000 [1.000, 1.000]
RNAC-IPM	0.044 [0.035, 0.056]	0.389 [0.307, 0.471]	1.000 [1.000, 1.000]
DR-unseen	0.067 [0.053, 0.083]	0.400 [0.321, 0.478]	1.000 [1.000, 1.000]
DR-seen†	0.073 [0.059, 0.089]	0.562 [0.473, 0.650]	0.988 [0.963, 1.000]
DTR (single-head)	**0.630 [0.598, 0.663]**	0.999 [0.996, 1.000]	0.497 [0.406, 0.587]
DTR (two-head)	**0.641 [0.608, 0.671]**	1.000 [1.000, 1.000]	—

scratch and report the return it induces. Every method is evaluated against an adversary drawn from the same admissible set $\mathcal{U}_{c,D}$, including the methods never trained with delays.

Protocol. For each frozen agent checkpoint we train a fresh adversary from scratch against that agent. Only the adversary and its critic are updated. The adversary uses the paper's $D = 4$, $c = 2$ and the two-head architecture, which satisfies the mean constraint by construction, so the admissible set is identical for every method. The adversary is trained by PPO with the hyperparameters of joint DTR training (Appx. C.8), except as follows. Because the agent is frozen, the adversary has its own critic and its own Adam optimizer rather than sharing the agent's. The critic is trained on the unnormalized negated macro-step reward $-R_t$ (Appx. C.6), and the learning rate is held at 3×10^{-4} rather than decayed. The entropy bonus is annealed to zero, so the final behavior reflects the optimizer rather than an exploration bonus. Reported returns are those induced by the final adversary, averaged over 20 episodes per (agent, adversary) pair before aggregation across seeds.

Adversary training budget. Each adversary is trained for 300K decisions, far below the 10^6 used for joint DTR training on the locomotion tasks. The smaller budget is appropriate because only a small categorical delay head is optimized, against a fixed opponent. We confirm it suffices: per-run terminal slopes over $[200\text{K}, 300\text{K}]$ are distributed around zero for every method, including the delay-trained DTR variants, and a sign test on whether return fell over the final third splits 52%/48%. The adversary has therefore converged for every victim, and the comparison below is not an artifact of stopping early on the harder ones.

Results. The picture of Sec. 4.2 is preserved. Both DTR variants remain far ahead of every baseline (IQM 0.641 and 0.630), taking the top two places on all four environments, against the strongest baseline, Uniform delay, at 0.329, with $P(\text{DTR two-head} \succ \text{Uniform}) = 0.853\,[0.791, 0.911]$. In raw return (Table 11 against the worst case of Table 4), DTR two-head exceeds Uniform by 96% on Hopper and 37% on Walker2d, against 95% and 31% under the fixed schedules; by 7% on HalfCheetah, against 14%; and on InvertedPendulum by a factor of 21, against 19. The reverse direction of the duality also survives: PR-MDP remains the only method never trained on delays that separates from Vanilla ($P = 0.869\,[0.809, 0.921]$), while SA-PPO, RNAC-IPM and DR-unseen stay at or below it. Relative to Table 1 the ordering changes only in four adjacent pairs — (DTR single-head, DTR two-head), (Fixed delay, PR-MDP), (SA-PPO, DR-seen) and (RNAC-IPM, Vanilla PPO) — in each case between methods whose intervals overlap under both evaluations. The one qualitative change is DR-seen†, whose small improvement over Vanilla under the fixed schedules does not survive here ($P = 0.562\,[0.473, 0.650]$): under a state-dependent adversary, no method that randomizes over physics inherits temporal robustness, and PR-MDP stands alone.

A search over dynamics perturbations. By Theorem 3, choosing a state-dependent delay distribution $\tau_s \in \mathcal{U}_{c,D}$ and choosing a coupled kernel–reward pair $(\tilde{P}_p, \tilde{R}_p) \in \mathcal{U}_s$ are the same choice. This evaluation is therefore not only a temporal test: it is an optimized search for the worst dynamics perturbation *inside* the induced set, evaluated at the hardest point of $\mathcal{U}_s$. The coupling is realized — the simulator accumulates the true reward over the held steps, so agents face the pair induced by a single τ_s and not an arbitrary element of $\mathcal{P}_s \times \mathcal{R}_s$. It complements Sec. 4.3, which deliberately evaluates outside the induced set: an in-set evaluation is, by the equivalence, a test of the very objective DTR optimizes, and so cannot on its own establish transfer to perturbation families the training never covers.

Where the learned adversary is weaker than the boundary schedule. The learned adversary does not always induce a lower return than the fixed boundary schedule of Sec. 4.2. Two things account for this. First, the boundary schedule is not an arbitrary stress test: it is the extremal Bern$(1, 4)$, the vertex that Corollary 2 predicts an optimal adversary to occupy, and the one the trained adversary does occupy at 78.4% and 83.5% of visited states (Table 8). It is therefore already near-optimal wherever the per-state optimum sits at that same vertex, leaving the learned adversary little to gain and its residual optimization error to lose. Second, the adversary optimizes discounted return, while the tables report undiscounted return as is standard.

C.5 Action Chunking with Diffusion Policies

We extend the analysis of Sec. 4.2 to action-chunking policies.

Setup. We use the DPPO pipeline (Ren et al., 2025): a diffusion policy pretrained on the D4RL (Fu et al., 2020) `medium` dataset of each environment, then fine-tuned online with PPO through the denoising chain. We take the released pretrained checkpoints, network architectures and default fine-tuning hyperparameters unchanged, so every method starts from an identical prior and differs only in its online fine-tuning; see Ren et al. (2025) for the architecture. Fine-tuning and evaluation run on the Gymnasium MuJoCo `-v4` tasks, not the D4RL environments in which the pretraining data were collected. Environments are Hopper, Walker2d and HalfCheetah, 5 seeds each, with $c = 2$ and a prediction horizon of $T_p = 4$, so that $D = T_p$ matches the main text. The policy emits a chunk of T_p actions, of which the first τ execute before the next observation and replan; since $\tau \le T_p$, no action is ever held stale, so action repetition never enters this comparison.

Methods. All methods predict the same horizon $T_p = 4$ and differ only in the commit length τ used during fine-tuning, which plays the role of the delay; Fixed-2, Uniform and the two DTR variants are exactly as in Sec. 4.2. We add **Fixed-4** ($\tau \equiv T_p$), which commits the entire predicted chunk — the way chunking policies are ordinarily deployed, and the control of practical interest here.

Table 13: Aggregate temporal robustness for diffusion action-chunking policies. Interquartile mean of per-cell min–max normalized return (1.0 = best single run in that environment and cell, 0 = worst) over 5 seeds × 3 environments, with 95% stratified bootstrap confidence intervals; the normalization is the one used in Appendix C.7. Nominal is the undelayed clock $\tau \equiv 1$; *worst case* is the per-seed minimum over the three mean-2 schedules; *learned adv.* is a state-dependent adversary trained against each frozen checkpoint under the same $\mathbb{E}[d] = 2$ constraint. Lower block: probability that DTR (two-head) outperforms each baseline. Per-environment returns are in Table 14.

Method	Nominal	Worst case	Learned adv.
Fixed-2	0.489 [0.34, 0.63]	0.138 [0.08, 0.22]	0.145 [0.09, 0.21]
Fixed-4	0.136 [0.09, 0.20]	0.729 [0.58, 0.79]	0.564 [0.41, 0.70]
Uniform	**0.899 [0.85, 0.95]**	0.686 [0.60, 0.75]	0.527 [0.46, 0.56]
DTR (single-head)	0.820 [0.73, 0.89]	0.931 [0.91, 0.95]	0.833 [0.77, 0.89]
DTR (two-head)	0.791 [0.70, 0.87]	**0.940 [0.92, 0.96]**	**0.887 [0.87, 0.90]**
Probability of improvement, DTR (two-head) over:			
Uniform	0.333	**1.000**	**1.000**
Fixed-4	0.960	0.947	0.907
Fixed-2	0.840	**1.000**	**1.000**

Table 14: Per-environment temporal robustness for diffusion action-chunking policies: nominal, worst-case and learned-adversary episodic return over 5 seeds. Nominal (Nom.) is the undelayed clock $\tau \equiv 1$. Worst case (Wst.) is the per-seed minimum over the three mean-matched schedules of Section 4.2 — constant ($\tau \equiv 2$), uniform ($\tau \sim \mathrm{Unif}\{1, 2, 3\}$) and boundary ($d \in \{1, 4\}$ with $\mathbb{E}[d] = 2$) — taken before averaging across seeds. Adv. is a state-dependent adversary trained from scratch against each frozen checkpoint under the same $\mathbb{E}[d] = 2$ constraint, scored on its final 30 iterations. Entries are the mean ± the half-width of a 95% Student-t interval over 5 seeds; best method per column in bold. Aggregates over these cells are in Table 13.

Method	Hopper			Walker2d			HalfCheetah		
	Nom.	Wst.	Adv.	Nom.	Wst.	Adv.	Nom.	Wst.	Adv.
Fixed-2	2252±1390	1225±422	924±295	1621±1465	961±375	690±220	6321±297	2708±588	2648±631
Fixed-4	1490±973	2340±1327	1572±1160	500±487	2265±996	1243±764	5432±177	5036±139	4853±144
Uniform	**2831±590**	2471±336	1543±334	4211±334	3061±671	1658±405	**7221±186**	4323±535	4253±382
DTR (single)	2626±958	3040±114	2053±111	**4431±95**	4008±132	2948±467	6634±264	**5336±178**	**5167±246**
DTR (two)	2700±954	**3111±89**	**2152±130**	4203±112	**4040±170**	**3187±154**	6586±90	5301±116	5151±35

Adversary and hyperparameters. The diffusion policy uses the DPPO defaults throughout. The timing adversary has the architecture of Sec. 4.1 and is trained by PPO on the same rollout, with advantage $-\hat{A}$ from the DPPO critic. It has its own Adam optimizer and is updated after each diffusion update, with clip 0.1, entropy coefficient 0.01, learning rate held at 3×10^{-4}, DPPO's 5 epochs and a minibatch of 4096, so that agent and adversary take a comparable number of minibatches per epoch (4 and 5 respectively). The single-head penalty is $\lambda = 10$.

Evaluation. Each final checkpoint is scored under the four timing schedules of Sec. 4.2 — nominal ($\tau \equiv 1$), constant ($\tau \equiv 2$), uniform ($\tau \sim \mathrm{Unif}\{1, 2, 3\}$) and boundary ($d \in \{1, 4\}$ with $\mathbb{E}[d] = 2$) — with the worst case taken as the per-seed minimum over the three mean-2 schedules; nominal is excluded from it, being a mean-1 schedule and therefore not a member of $\mathcal{U}_{2,4}$. We additionally train a learned adversary from scratch for 100 iterations against each frozen checkpoint under the same $\mathbb{E}[d] = 2$ constraint, with the two-head architecture, and report the victim return averaged over its final 30 iterations. Its advantages come from the frozen victim's DPPO critic, and its entropy coefficient is held at 0.01. Returns are min–max normalized as in Appx. C.7; we report interquartile means over 5 seeds × 3 environments with 95% stratified bootstrap confidence intervals and the probability of improvement.

Results. The conclusion of Sec. 4.2 transfers without qualification (Table 13 in aggregate, Table 14 per environment). Both DTR variants attain the best worst case — IQM 0.940 and 0.931, against 0.729 for Fixed-4, 0.686 for Uniform and 0.138 for Fixed-2 — and DTR (two-head) beats Uniform on every seed–environment pair. Against the learned adversary

the separation widens rather than narrowing: DTR (two-head) holds 0.887 where Fixed-4 falls to 0.564 and Uniform to 0.527. Fixed-2 and Fixed-4 trade places between the nominal clock and the worst case, while DTR is at or near the top of all three columns. The nominal clock is where DTR pays, and it pays modestly: 0.791 and 0.820 against 0.899 for Uniform, with overlapping intervals. Fixed-4 — committing the whole predicted chunk, the standard deployment — pays far more, reaching only 0.136.

C.6 Pseudocode for the adversarial clock

Algorithm 2 gives the training loop shared by both DTR variants. The adversary is a state-conditioned policy: at each decision time it maps s_t to a distribution $\tau^{\psi}_{s_t} \in \Delta_{[D]}$, and the agent acts on that same s_t. The two variants differ only in how the constraint $\mathbb{E}_{d\sim\tau_s}[d] = c$ is imposed — a soft penalty on the adversary's loss, or exactly by construction — which is the single **if** in each of the two blocks below. Writing $R_t = \sum_{i<d_t} \gamma^i r_{t+i}$ for the reward accumulated over a macro-step, advantages are the standard GAE with every γ replaced by γ^{d_t}, which is the only change the SMDP requires. For readability the macro-step is written as running the full d_t base steps; in the implementation an episode may terminate sooner, and the realized duration is used in its place.

Algorithm 2 DTR-PPO (single-head / two-head)

Input: max delay D, target mean c, adversary loss scale η, mean-penalty coefficient λ (single-head only), PPO hyperparameters Θ (agent, critic) and Θ^{adv} (adversary)
Initialize: agent π_ϕ, critic V_ω, adversary τ^ψ
while budget remains **do**
  Rollout: T macro-steps
  for $t = 0, \dots, T-1$ **do**
    $a_t \sim \pi_\phi(\cdot \mid s_t)$
    if two-head **then**
      $\tau^L, \tau^H \leftarrow$ softmax over $\{1, \dots, \lfloor c \rfloor\}$, $\{\lfloor c \rfloor{+}1, \dots, D\}$
      $\alpha_{s_t} \leftarrow (\mu^H - c)/(\mu^H - \mu^L)$ *// head means; $\mu^L \le c < \mu^H$ so $\alpha_{s_t} \in [0,1]$*
      $\tau^{\psi}_{s_t} \leftarrow \alpha_{s_t}\tau^L + (1-\alpha_{s_t})\tau^H$ *// $\mathbb{E}[d] = c$ exactly*
    else
      $\tau^{\psi}_{s_t} \leftarrow$ softmax over $\{1, \dots, D\}$ *// mean enforced in the loss*
    end if
    $d_t \sim \tau^{\psi}_{s_t}$
    $s_{t+1}, R_t, \text{done}_t \leftarrow \textsc{Step}(s_t, a_t, d_t)$ *// a_t held fixed for d_t base steps*
    store the transition with $\log \pi_\phi(a_t \mid s_t)$, $\log \tau^{\psi}_{s_t}(d_t)$, $V_\omega(s_t)$
  end for
  Advantages: GAE with SMDP-corrected discounting
  $\delta_t \leftarrow R_t + \gamma^{d_t} V_\omega(s_{t+1})(1 - \text{done}_t) - V_\omega(s_t)$
  $\hat{A}_t \leftarrow \delta_t + \gamma^{d_t}\lambda_{\text{GAE}}(1 - \text{done}_t)\hat{A}_{t+1}$ *// γ^{d_t}, not γ, in both lines*
  Update: joint minimax PPO
  for K epochs, each minibatch B **do**
    $\mathcal{L}_\pi \leftarrow \textsc{PPOLoss}(\pi_\phi; B, \hat{A}, \Theta)$, $\mathcal{L}_V \leftarrow \textsc{ValueLoss}(V_\omega; B, \Theta)$
    $\mathcal{L}_\tau \leftarrow \textsc{PPOLoss}(\tau^\psi; B, -\hat{A}, \Theta^{\text{adv}})$
    if single-head **then**
      $\mathcal{L}_\tau \leftarrow \mathcal{L}_\tau + \lambda(\mathbb{E}_{d\sim\tau^{\psi}_s}[d] - c)^2$ *// exact expectation, not sampled*
    end if
    gradient step on (ϕ, ω, ψ) minimizing $\mathcal{L}_\pi + \mathcal{L}_V + \eta\mathcal{L}_\tau$
  end for
end while
Return: π_ϕ

C.7 Training and evaluation details

Implementation. All ten methods of Sec. 4.1 are single-file modifications of CleanRL's `ppo_continuous_action.py` (Huang et al., 2022), run on the Gymnasium 0.29.1 MuJoCo

-v4 tasks. Vanilla PPO, Fixed and Uniform delay, both DTR variants, PR-MDP and the two DR baselines are our own implementations: the delay-based methods add an action-repetition wrapper, DTR adds the adversary of Algorithm 2, PR-MDP trains the protagonist and adversary of Tessler et al. (2019) jointly by PPO, and DR resamples its randomization parameters at every reset. SA-PPO and RNAC-IPM port their authors' public code onto the same backbone: the SGLD regularizer of Zhang et al. (2020) and the IPM-robust critic target of Zhou et al. (2023). The action-chunking study (Appx. C.5) builds instead on the public DPPO code and checkpoints (Ren et al., 2025).

Training. We train 20 seeds per (method, environment) pair for 10^6 agent decisions (2.5×10^5 on InvertedPendulum). A decision is one macro-step: a delay-based method executes $\approx c$ simulator steps per decision, but every method makes the same number of decisions and policy-gradient updates.

Evaluation. The final checkpoint of each run is evaluated zero-shot on the conditions of Sections 4.2 and 4.3, with actions sampled from the stochastic policy and observation-normalization statistics frozen at their training values. Each (seed, condition) pair is scored over 25 episodes. An episode's return is its undiscounted sum of simulator rewards within the standard 1000-simulator-step limit, whatever the clock. Appendices C.2–C.5 state their own protocols, and Appx. C.9 the compute used.

Aggregation. Tables 1 and 2 pool per-seed scores across environments. Each score is min–max normalized over the seed-level scores of all ten methods: within each environment and column in Table 1, and within each cell in Table 2. We report the interquartile mean (IQM) and the probability of improvement, computed with *rliable* (Agarwal et al., 2021). Brackets are 95% stratified bootstrap CIs over 50k replications. In Table 1 each environment has its own 20 trained agents, and seeds are resampled within each environment. In Table 2 the six cells of an environment are scored by the same 20 agents, so we resample agents within each environment, each keeping its scores on all six perturbations.

C.8 Hyperparameters

Shared PPO backbone. Every method uses the default hyperparameters of CleanRL's single-file `ppo_continuous_action.py` (Table 15), adopted unchanged and identical for all ten methods; we did not tune them for any method. Rollout length and training budget are counted in decisions (Appx. C.7).

Table 15: Shared PPO backbone: the CleanRL defaults, used unchanged by all ten methods.

Hyperparameter	Value	Hyperparameter	Value
Optimizer	Adam, $\epsilon_{\text{Adam}} = 10^{-5}$	Discount γ	0.99
Learning rate	3×10^{-4}, linear decay to 0	GAE λ_{GAE}	0.95
Rollout length	2048 decisions	Environments	1
Minibatches	32 of size 64	Update epochs	10
Clip coefficient	0.2 (policy and value)	Target KL	none
Entropy coefficient	0	Value coefficient	0.5
Max gradient norm	0.5	Advantage norm.	per minibatch
Actor and critic	separate, 64×64 tanh	Initialization	orthogonal
Policy	Gaussian, state-indep. std	Action clipping	to bounds
Observation norm.	running, clip ± 10	Reward norm.	running, clip ± 10

Method-specific hyperparameters. Table 16 lists every hyperparameter beyond the shared backbone. All searches used 5 seeds per setting, and each selected value is used unchanged on every environment, InvertedPendulum included. Their compute is reported in Appx. C.9.

Table 16: Method-specific hyperparameters. DTR's searches ran on HalfCheetah with the single-head variant, the baselines' on HalfCheetah, Hopper and Walker2d, at 5 seeds per setting. Rows without a search set were fixed in advance.

Method	Hyperparameter	Searched over	Selected
DTR (both)	max delay D	—	4
DTR (both)	target mean delay c	—	2
DTR (both)	adversary loss scale η	$\{0.1, 0.5, 1\}$	0.5
DTR (both)	adversary entropy coef.	$\{0, 0.003, 0.01, 0.03\}$	0.01
DTR single-head	mean-delay penalty λ	$\{0.2, 2, 20\}$	2
PR-MDP	mixing probability α	$\{0.05, 0.1, 0.2\}$	0.1
PR-MDP	adversary learning rate	$\{0.75, 1.5, 3\} \times 10^{-4}$	3×10^{-4}
SA-PPO	perturbation radius ϵ	$\{0.03, 0.045, 0.06, 0.075, 0.15, 0.3\}$	0.03
SA-PPO	regularization weight κ	—	1
RNAC-IPM	robustness radius δ	$\{10^{-5}, 10^{-4}, 10^{-3}\}$	10^{-4}
DR-seen	gravity scale	—	$\mathrm{Unif}(1, 2)$
DR-seen	action-noise probability	—	$\mathrm{Unif}(0, 0.5)$
DR-unseen	mass, friction scales	—	$\mathrm{Unif}(0.5, 1.5)$
Fixed delay	delay τ	—	2
Uniform delay	delay τ	—	$\mathrm{Unif}\{1, 2, 3\}$

DTR. The max delay D and target mean delay c are not tuned: they specify the uncertainty set $\mathcal{U}_{c,D}$, i.e. the robustness problem being posed, and are matched across the delay-based methods (Fixed $\tau \equiv c$, Uniform $\tau \sim \mathrm{Unif}\{1, \ldots, 2c-1\}$). Appx. C.3 varies both. The adversary is a separate network with the agent's architecture and a softmax output over delays, updated in the same gradient step as the agent and critic (Algorithm 2); its PPO hyperparameters Θ^{adv} equal the agent's Θ except for the entropy coefficient. The remaining knobs were each set once, on HalfCheetah with the single-head variant, and used unchanged for the two-head variant. The loss scale η is the largest value for which training remained stable. The entropy coefficient is the value under which the adversary was most effective during training, i.e. produced the largest reduction in the agent's training return; smaller values collapsed the delay distribution and larger ones drove it toward uniform. To tune λ, we trained the single-head variant with each candidate and selected the smallest value for which the empirical mean delay $\bar{c}$, averaged over the final 10 episodes, fell within 5% of the target $c = 2$. Figure 3 shows the selected λ on Walker2d. The two-head variant has no λ.

Robust-RL baselines. PR-MDP, SA-PPO and RNAC-IPM were tuned by the same rule: a sweep on HalfCheetah, Hopper and Walker2d, selecting the setting with the best average nominal return. For PR-MDP we searched the mixing probability α and the adversary's learning rate as a 3×3 grid; all its other hyperparameters, including a zero entropy coefficient, equal the agent's. For SA-PPO we used 10 SGLD steps and a single regularization weight $\kappa = 1$ on every environment, the value Zhang et al. (2020) use for Hopper; we annealed ϵ over the first half of training and swept it over a range bracketing their per-environment values. For RNAC-IPM the only method-specific knob is the radius δ: $\delta = 10^{-4}$ matched or improved on the published 10^{-5} on all three tasks, while 10^{-3} drove nominal return to a floor on Hopper and Walker2d. We use $\delta = 10^{-4}$, an order of magnitude above the published value and hence a strictly more conservative uncertainty set.

Domain randomization. The randomization ranges were fixed in advance, not tuned. DR-seen draws a gravity scale and an action-noise probability per episode, over ranges that contain the evaluation grid of Sec. 4.3; DR-unseen scales body mass and friction, each by a per-episode factor.

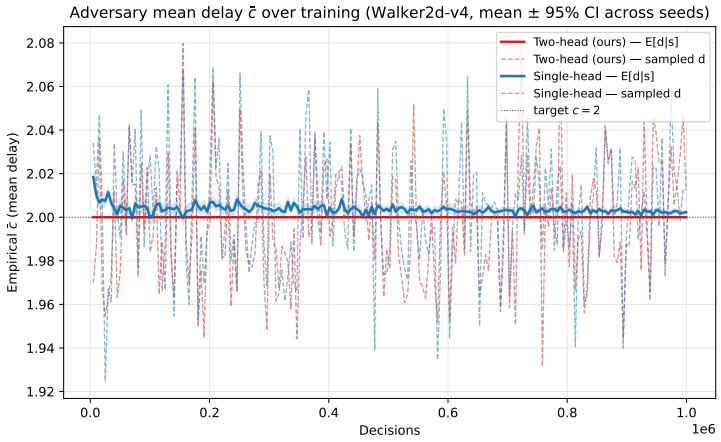

Figure 3: Adversary mean delay during training on Walker2d. Solid lines show $\bar{c}_t := \frac{1}{|B_t|}\sum_{s\in B_t}\mathbb{E}_{d\sim\tau_s}[d]$, where B_t is the last minibatch (64 rollout states) of the PPO update at decision t and τ is the adversary after that update; dashed lines average the delays actually sampled at those states during the rollout. The two-head's expected delay equals the target $c = 2$ throughout training, while the single-head, which enforces the constraint via a soft penalty, shows a small but persistent positive bias. All curves are means over 20 seeds; shading shows 95% CIs.

C.9 Compute Resources

All action-repetition experiments reported in the paper were trained and evaluated on CPU only, one run pinned to a single logical CPU, on a shared internal cluster node with 256 logical CPUs (AMD EPYC 7742). A single 10^6-decision run takes roughly half an hour for the methods without a learned adversary (Vanilla, Fixed, Uniform, DR, RNAC-IPM, SA-PPO), and roughly 1.5× that for the methods that train one (PR-MDP, DTR); every method performs the same number of decisions and policy gradient updates, so this is a wall-clock cost only. We note that the implementation is not optimized for efficiency, it is the stock single-environment CleanRL PPO loop and not run on a GPU, and these numbers could likely be improved substantially by a tuned implementation. Across all methods, seeds, environments and ablations, training plus evaluation (including the appendix evaluations: the learned worst-case adversary and the D/c ablation) came to roughly 1,600 core-hours. Hyperparameter sweeps added ≈210 core-hours and exploratory or superseded work a further ≈900.

The action-chunking study (Appx. C.5) ran on a single RTX 5090, six fine-tunes packed per card, with HalfCheetah roughly 1.8× the cost of Hopper and Walker2d. The diffusion fine-tune dominates, so the DTR adversary adds essentially nothing and all arms cost the same. The 75 runs took ≈215 GPU-hours and their evaluation ≈45; bring-up and exploratory runs added ≈50, for ≈310 GPU-hours in total. Offline pretraining is excluded — we reuse the released DPPO checkpoints.