SCALABLE CONTINUOUS-TIME REINFORCEMENT LEARNING FOR HIGH-DIMENSIONAL CONTROL VIA SOFT PERSISTENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Lots of real-world physical systems evolve continuously over time and require control across diverse timescales. Standard reinforcement learning typically represents continuous control as instantaneous state-to-action decisions, leaving the temporal structure of actions implicit. Continuous-time approaches introduce persistence through adaptive decision timing. However, committing the entire action vector delays feedback-based revisions, which become restrictive in high-dimensional systems with heterogeneous actuator requirements. We analyze the trade-off between commitment loss and value-estimation error amplification, and show how heterogeneous temporal requirements can restrict shared commitment as action dimensionality increases. Our derivation shows that bounded action revision can extend the feasible control horizon beyond hard holding. Motivated by the analysis, we propose Soft Persistence Regularization (SOPER), which learns action dynamics by optimizing task return under a conditional-drift budget. Discretizing this objective yields a practical regularizer that integrates directly into standard actor-critic learning, encouraging temporal coherence while allowing action components to evolve at different rates. Experiments on diverse high-dimensional continuous control benchmarks show sample efficiency advantages of SOPER over discrete-time and continuous-time RL baselines. It can support distinct temporal behavior across actuators in different task and regularization strength. We also demonstrate the success of SOPER in a real humanoid robot experiment. The results suggest soft persistence as a scalable way to incorporating continuous-time action dynamics into efficient learning of control.

1 INTRODUCTION

Biological and robotic systems evolve continuously, their effective control requires coordinating actions across diverse, task-dependent timescales. Some commands can remain nearly unchanged over extended periods, whereas others require rapid adjustments in response to feedback. In biological motor control, task-level feedback explains the temporal recruitment of muscle synergies during postural correction (Safavynia & Ting, 2012). Humanoid controllers similarly coordinate locomotion with concurrent upper-body behavior while adapting movement timing to task demands (Xue et al., 2025). These observations motivate learning how individual action components should evolve across task-dependent timescales to support more efficient high-dimensional control learning.

Reinforcement learning (RL) has achieved substantial progress in high-dimensional robot and musculoskeletal control (Seo et al., 2025; Wei et al., 2026). However, standard formulations represent control as repeated state-to-action decisions at fixed intervals (Haarnoja et al., 2018). Fine discretization provides frequent feedback opportunities but increases the number of decisions over a fixed physical horizon, potentially complicating coherent exploration and amplifying errors at comparable per-decision estimation accuracy (Metelli et al., 2020). While temporal smoothness regularizers encourage consistency between policy predictions, this consistency does not directly characterize the temporal evolution of executed actions (Mysore et al., 2021).

Continuous-time RL formulates control through continuously evolving state dynamics and rewards accumulated over physical time, providing a framework for reasoning about the temporal organiza-

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.

tion of behavior (Doya, 2000). Recent works explicitly learn temporal structure by jointly selecting control commands and their execution durations (Karimi et al., 2023; Treven et al., 2024; Iten et al., 2026). These approaches adapt command persistence to the state and task, supporting temporal coordination. However, action revisions remain tied to shared decision times, which can be restrictive under heterogeneous temporal requirements across action components.

In this paper, we analyze the benefits and limitations of action persistence in continuous time. We show how drift-constrained action dynamics can extend the feasible control horizon under high-dimensional continuous control with heterogeneous temporal requirements. Motivated by this analysis, we propose Soft Persistence Regularization (SOPER), which learns the temporal evolution of control commands by optimizing task return under an action-drift budget. SOPER encourages temporal coherence while allowing action components to evolve at different rates. Discretizing the continuous-time objective yields a practical regularizer that integrates directly into standard actor-critic learning. Experiments on high-dimensional benchmarks demonstrate improved sample efficiency of SOPER over recent discrete-time and continuous-time RL baselines across diverse task objectives, action dimensionalities, and actuator semantics. SOPER efficiently learns to control a 700-actuator musculoskeletal system, and successfully controls a humanoid robot in real world, highlighting its scalability and practicality of learning high-dimensional temporal action dynamics.

Our contributions: (1) We analyze the persistence-estimation trade-off and show how constrained action dynamics can overcome shared-commitment limitations under heterogeneous temporal requirements. (2) We propose SOPER to learn heterogeneous action dynamics under a drift budget and provide a practical implementation compatible with standard actor-critic learning. (3) The new method has sample efficiency advantages over discrete-time and continuous-time RL baselines across diverse high-dimensional control tasks and actuator semantics, demonstrating success in 700-actuator musculoskeletal control and real-world humanoid control.

2 RELATED WORK

High-dimensional continuous control. High-dimensional robotic and musculoskeletal control has motivated approaches that exploit structure in dynamics, actuator coordination, and exploration. Model-based methods perform trajectory optimization using learned latent dynamics (Hansen et al., 2024). Learning can be simplified through reusable behavioral priors (Caggiano et al., 2023) and synergistic representations (Berg et al., 2024; He et al., 2024). Exploration strategies further exploit self-organizing behavior (Schumacher et al., 2023), latent perturbations (Chiappa et al., 2023), and value-guided probability flows (Wei et al., 2026). Recent FlashSAC improves the scalability and stability of off-policy learning through residual networks and normalization (Kim et al., 2026). Despite these advances, many methods still learn through frequent discrete decisions, which can compound the effects of value-estimation errors and hinder efficient learning in high-dimensional systems.

Continuous-time RL and temporal policy learning. Continuous-time RL formulates learning through dynamics and returns defined over physical time (Yildiz et al., 2021; Jia & Zhou, 2022; 2023; Hua et al., 2026). Foundational work derives value and policy updates from Hamilton-Jacobi-Bellman equations (Doya, 2000), while subsequent research addresses the sensitivity of value learning to time discretization (Tallec et al., 2019). Control under bounded action rates has also been studied through continuous-time value learning and value-based action selection (Kim et al., 2021). Another line of work learns temporal structure through action persistence or temporally extended policies. Recent study such as TaCoS jointly select control commands and their execution durations (Karimi et al., 2023; Treven et al., 2024; Iten et al., 2026). These methods introduce persistence by committing to an action or trajectory for a shared duration, which can be restrictive under heterogeneous temporal requirements across actuators.

Temporal regularization and smooth exploration. Policy regularization encourages smooth control by penalizing temporal and spatial differences between policy outputs (Mysore et al., 2021; Kobayashi, 2022). Architectural methods incorporate learnable filtering dynamics into policy networks (Wang et al., 2025). Structured exploration also promotes smoother behavior through periodically resampled state-dependent noise (Raffin et al., 2022). Recent approaches regularize changes in action gradients (Lee et al., 2024). Algorithm ASAP aligns actions with predictions from preceding states and penalizes second-order temporal differences (Kwak & Hwang, 2026). These studies show that regulating action variability can improve control performance and robustness. However, policy

smoothness alone does not specify how executed commands should evolve across task-dependent timescales.

3 PRELIMINARIES

Continuous-time high-dimensional control. We formulate the control of high-dimensional embodied systems as an infinite-horizon continuous-time Markov decision process $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \boldsymbol{f}, r, \rho, \mu_0)$, with state $\boldsymbol{x}_t \in \mathcal{X} \subseteq \mathbb{R}^{|\mathcal{X}|}$ and action $\boldsymbol{a}_t \in \mathcal{A} \subseteq \mathbb{R}^{|\mathcal{A}|}$. The state evolves according to deterministic dynamics

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{a}_t)\, dt, \qquad \boldsymbol{x}_0 \sim \mu_0, \tag{1}$$

where $\boldsymbol{f} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{|\mathcal{X}|}$ specifies the instantaneous rate of state change under the applied action. The initial-state distribution is $\mu_0 \in \mathcal{P}(\mathcal{X})$, where $\mathcal{P}(\mathcal{X})$ denotes the probability measures on $\mathcal{X}$. Let $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ be a measurable reward rate satisfying $|r(\boldsymbol{x}, \boldsymbol{a})| \leq r_{\max} < \infty$. Our objective is to learn a controller $\pi \in \Pi$ that maximizes the expected discounted return

$$J(\pi) = \mathbb{E}^{\pi}\left[\int_0^{\infty} e^{-\rho t} r(\boldsymbol{x}_t, \boldsymbol{a}_t)\, dt\right], \qquad \rho > 0, \tag{2}$$

where ρ is the continuous-time discount rate. For a fixed initial state, the value of π is

$$V^{\pi}(\boldsymbol{x}) = \mathbb{E}^{\pi}\left[\int_0^{\infty} e^{-\rho t} r(\boldsymbol{x}_t, \boldsymbol{a}_t)\, dt \,\middle|\, \boldsymbol{x}_0 = \boldsymbol{x}\right], \tag{3}$$

and the optimal value is $V^{\star}(\boldsymbol{x}) = \sup_{\pi \in \Pi} V^{\pi}(\boldsymbol{x})$. High-dimensional control poses challenges for exploration and value learning, where effective behavior depends on coordinating many actuators with different temporal requirements. For example, human motion control requires combining locomotor patterns with task-specific muscle activation timings (Ivanenko et al., 2005). The controller should learn the temporal evolution of control commands.

Temporal policy learning via persistence. Conventional fixed-interval policy operates on a prescribed decision schedule. Recent continuous-time RL methods explores the temporal behavior by jointly selecting the control command and its execution duration. This allows the decision schedule to adapt to the current state (Treven et al., 2024). For an action-holding parameterization, starting from $t_0 = 0$, the controller selects

$$(\boldsymbol{a}_j, \tau_j) \sim \pi(\cdot \mid \boldsymbol{x}_{t_j}), \qquad \boldsymbol{a}_{t_j+s} = \boldsymbol{a}_j \quad (0 \leq s < \tau_j), \qquad t_{j+1} = t_j + \tau_j. \tag{4}$$

More generally, the controller can select parameters of an action trajectory together with its duration (Karimi et al., 2023). Here, temporal structure is established through *hard persistence*: the selected command or trajectory is not revised using new observations before t_{j+1}. The execution duration is shared across the action vector, although individual components of a prescribed trajectory may have different temporal profiles.

4 UNDERSTANDING PERSISTENCE IN CONTINUOUS-TIME

Longer action persistence reduces feedback opportunities, but can also reduce the accumulation of action-selection errors over a discounted physical horizon. This trade-off between control loss and estimation-error propagation has been studied for action repetition in discrete-time batch RL (Metelli et al., 2020). In this section, we derive a continuous-time performance bound that separates these two effects.

We first formalize the commitment loss incurred by holding an action before resuming optimal feedback. For a fixed duration $\tau > 0$, consider policies that hold each selected action for duration τ, then observe the current state and select the next action. Let $\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}}$ denote the trajectory starting from $\boldsymbol{x}$ under the constant action $\boldsymbol{a}$. The corresponding Bellman operator is

$$(\mathcal{T}_{\tau} V)(\boldsymbol{x}) = \sup_{\boldsymbol{a} \in \mathcal{A}} \left[\int_0^{\tau} e^{-\rho t} r(\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}}, \boldsymbol{a})\, dt + e^{-\rho \tau} V(\boldsymbol{x}_{\tau}^{\boldsymbol{x},\boldsymbol{a}})\right]. \tag{5}$$

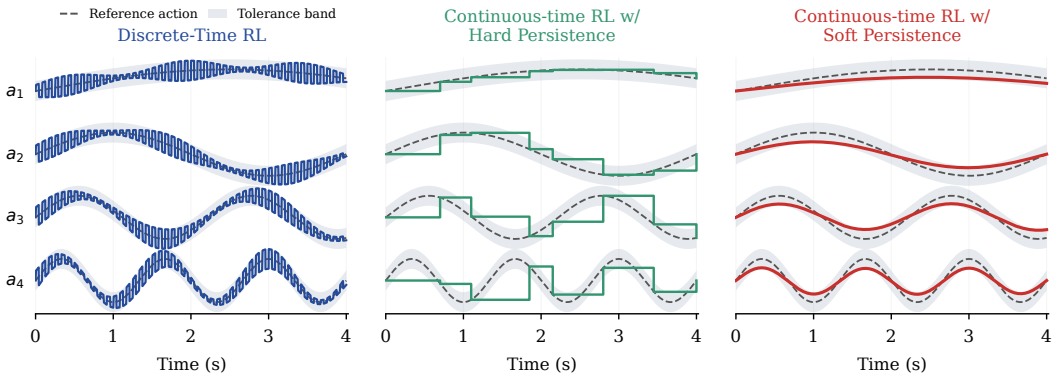

Figure 1: **Command tracking with different temporal demands.** Dashed curves denote references and shaded bands indicate tolerances. Discrete-time RL tracks references through frequent updates without explicitly modeling temporal structure. Hard persistence holds all components over shared intervals, limiting tracking of faster references. Soft persistence allows temporal coherent action evolution at different rates while maintaining tracking.

This operator is an $e^{-\rho\tau}$-contraction in the supremum norm. Its unique fixed point, $V_\tau^\star$, is the optimal value within the holding policy class. The associated action-value function is

$$Q_\tau^\star(\boldsymbol{x},\boldsymbol{a}) = \int_0^\tau e^{-\rho t} r(\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}},\boldsymbol{a})\,dt + e^{-\rho\tau} V_\tau^\star(\boldsymbol{x}_\tau^{\boldsymbol{x},\boldsymbol{a}}), \qquad V_\tau^\star(\boldsymbol{x}) = \sup_{\boldsymbol{a}} Q_\tau^\star(\boldsymbol{x},\boldsymbol{a}). \tag{6}$$

Thus, $Q_\tau^\star$ evaluates a finite action-holding interval followed by optimal continuation within the same policy class. We define the single-interval commitment loss as

$$D(\boldsymbol{x},\tau) = V^\star(\boldsymbol{x}) - (\mathcal{T}_\tau V^\star)(\boldsymbol{x}) \geq 0, \qquad \overline{D}(\tau) = \sup_{\boldsymbol{x}} D(\boldsymbol{x},\tau). \tag{7}$$

The continuation value here is the unrestricted optimum $V^\star$. Hence, $D(\boldsymbol{x},\tau)$ isolates the loss of holding an action during the first while permitting optimal feedback thereafter. Let $\mathcal{I} \subseteq (0,\infty)$ denote the holding durations under consideration. We define the task-gap rate as

$$\ell(\boldsymbol{x},\boldsymbol{a}) = \rho V^\star(\boldsymbol{x}) - r(\boldsymbol{x},\boldsymbol{a}) - \nabla V^\star(\boldsymbol{x})^\top \boldsymbol{f}(\boldsymbol{x},\boldsymbol{a}). \tag{8}$$

Using the following conditions, we compare commitment loss and estimation-error amplification across durations.

Assumption 1 (State-domain invariance and classical optimality)**.** *The state domain is forward invariant under admissible controls, and $V^\star \in C^1$. The classical optimality condition $\min_{\boldsymbol{a}\in\mathcal{A}} \ell(\boldsymbol{x},\boldsymbol{a}) = 0$ holds for every $\boldsymbol{x}$, with the minimum attained.*

Assumption 2 (Bounded dynamics and task-gap regularity)**.** *There exist finite constants $F, L_\ell \geq 0$, such that $\|\boldsymbol{f}(\boldsymbol{x},\boldsymbol{a})\| \leq F$, and $\ell(\cdot,\boldsymbol{a})$ is L_ℓ-Lipschitz uniformly over actions.*

Assumption 3 (Estimation accuracy)**.** *There exists a common $\eta \geq 0$, such that the estimated action-value functions satisfy $\|\widehat{Q}_\tau - Q_\tau^\star\|_\infty \leq \eta, \tau \in \mathcal{I}$.*

Proposition 1 (Persistence-estimation trade-off)**.** *Under Assumptions 1, 2, and Assumption 3, let $\widehat{\pi}_\tau$ be a measurable greedy policy with respect to $\widehat{Q}_\tau$, executing each selected action for duration τ. Then, for every $\tau \in \mathcal{I}$, with $K = FL_\ell$,*

$$\|V^\star - V^{\widehat{\pi}_\tau}\|_\infty \leq \frac{\overline{D}(\tau) + 2\eta}{1 - e^{-\rho\tau}} \leq \underbrace{\frac{K\tau}{2\rho}}_{\text{commitment}} + \underbrace{\frac{2\eta}{\rho\tau} + 2\eta}_{\text{estimation}}. \tag{9}$$

*For fixed K and η, the commitment contribution is nondecreasing and the estimation contribution is nonincreasing in τ. (See **Appendix A.1** for the proof.)*

Proposition 1 shows that *longer persistence can reduce the amplification of value-estimation errors within a prescribed commitment-loss tolerance*. This robustness to imperfect value estimates

is desirable for high-dimensional policy learning, where accurate value estimation can be challenging. Exploiting this benefit, however, requires sufficiently long holding durations with acceptable commitment loss. Under shared commitment, a single duration must accommodate the temporal requirements of different actuators. We next illustrate how increasing action dimension can restrict this duration by a simple case analysis.

Case analysis: shared commitment in command tracking. Consider a $|\mathcal{A}|$-dimensional command-tracking task with actions $\boldsymbol{a} \in [-A, A]^{|\mathcal{A}|}$, phase dynamics $\dot{\varphi}_i = \omega_i > 0$, and references $q_i(\boldsymbol{\varphi}) = A \sin \varphi_i$. The reward rate is $r(\boldsymbol{\varphi}, \boldsymbol{a}) = \min\left(\epsilon - \|\boldsymbol{a} - \boldsymbol{q}(\boldsymbol{\varphi})\|_\infty, 0\right)$ with $0 < \epsilon < A$. Starting from $\boldsymbol{\varphi}(0) = \mathbf{0}$, hold the initial action $\boldsymbol{a}(t) \equiv \mathbf{0}$. The longest persistent duration with zero loss is $\tau_{\mathrm{hold},|\mathcal{A}|} = \min_{1 \le i \le |\mathcal{A}|} \frac{\arcsin(\epsilon/A)}{\omega_i} = \frac{\arcsin(\epsilon/A)}{\max_{i \le |\mathcal{A}|} \omega_i}$. (See **Appendix A.2** for the proof.)

In this case, the longest persistent duration is non-increasing as more references are added, and decreases whenever a new reference has a higher frequency than all existing ones. Figure 1 illustrates the imitation, where shared commitment produces larger tracking errors for the faster references. This comparison motivates exploring heterogeneous temporal structure instead of hard persistence.

5 Soft Persistence Regularization

In this section, we connect hard persistence with instantaneous action trajectory by formulating action dynamics in continuous time. We show the admissible persistence time can be extended with relaxed dynamics drift constraint. Building on this analysis, we introduce Soft Persistence Regularization (SoPer), a continuous-time RL method that learns heterogeneous action dynamics in high-dimensional continuous control.

5.1 From persistence to controlled action dynamics

Continuous-time control can represent actions as evolving variables whose rates of change are governed by feedback (Kim et al., 2021):

$$\dot{\boldsymbol{x}}_t = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{a}_t), \qquad \dot{\boldsymbol{a}}_t = \boldsymbol{u}_t(\boldsymbol{x}_t, \boldsymbol{a}_t), \qquad \|\boldsymbol{u}_t\|_2^2 \le B, \tag{10}$$

where $\boldsymbol{u} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{|\mathcal{A}|}$ is the action drift conditioned on the current state and action. The budget B bounds the total squared action-change rate. Setting $B = 0$ recovers hard persistence, while increasing B allows progressively faster action revision. In the limit $B \to \infty$, the drift restriction disappears, permitting unrestricted next-action selection.

We consider continuous admissible drifts $\boldsymbol{u} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{|\mathcal{A}|}$ and assume the induced dynamics are well posed. For an initial pair $(\boldsymbol{x}, \boldsymbol{a})$ with $\ell(\boldsymbol{x}, \boldsymbol{a}) \le \delta$, let $(\boldsymbol{x}_t^{\boldsymbol{u}}, \boldsymbol{a}_t^{\boldsymbol{u}})$ denote the trajectory starting from $(\boldsymbol{x}_0^{\boldsymbol{u}}, \boldsymbol{a}_0^{\boldsymbol{u}}) = (\boldsymbol{x}, \boldsymbol{a})$. We define the feasible evolution horizon as

$$\tau_{\max}(\boldsymbol{x}, \boldsymbol{a}; B, \delta) = \sup\Big\{T \ge 0 : \exists\, \boldsymbol{u},\ \sup_{t \in [0,T]} \|\boldsymbol{u}(\boldsymbol{x}_t^{\boldsymbol{u}}, \boldsymbol{a}_t^{\boldsymbol{u}})\|_2^2 \le B,\ \sup_{t \in [0,T]} \ell(\boldsymbol{x}_t^{\boldsymbol{u}}, \boldsymbol{a}_t^{\boldsymbol{u}}) \le \delta\Big\}. \tag{11}$$

This quantity characterizes how long the action dynamics can maintain the task-gap tolerance under the drift budget. At $B = 0$, it recovers the corresponding horizon under hard persistence. Below, we show that the feasible horizon is nondecreasing in the drift budget.

Proposition 2 (Relaxing drift enlarges the feasible horizon)**.** *For a fixed initial pair $(\boldsymbol{x}, \boldsymbol{a})$ with $\ell(\boldsymbol{x}, \boldsymbol{a}) \le \delta$,*

$$0 \le B_1 \le B_2 \quad \Longrightarrow \quad \tau_{\max}(\boldsymbol{x}, \boldsymbol{a}; B_1, \delta) \le \tau_{\max}(\boldsymbol{x}, \boldsymbol{a}; B_2, \delta). \tag{12}$$

*If an admissible drift generates a trajectory from $(\boldsymbol{x}, \boldsymbol{a})$ satisfying $\ell(\boldsymbol{x}_t, \boldsymbol{a}_t) \le \delta$ and $\|\dot{\boldsymbol{a}}_t\|_2^2 \le B_\star < \infty$ for all $t \ge 0$, then $\tau_{\max}(\boldsymbol{x}, \boldsymbol{a}; B, \delta) = \infty$ for every $B \ge B_\star$. (See **Appendix A.3** for the proof.)*

The result follows from inclusion of the feasible trajectory classes as B increases. At $B = 0$, the constraint forces $\boldsymbol{u}_t = \mathbf{0}$ almost everywhere. Holding therefore appears as the zero-drift case of a broader family of feedback-controlled action dynamics.

Revisiting the tracking example. For the preceding tracking task, the scaled trajectory $a_i(t) = (1 - \epsilon/A) q_i(\boldsymbol{\varphi}(t))$ remains within tolerance and has squared action-change rate bounded by $B_{\mathrm{suff}} =$

$(A-\epsilon)^2 \sum_{i=1}^{|\mathcal{A}|} \omega_i^2$. Consequently, $\tau_{\max}(\mathbf{0}, \mathbf{0}; B, 0) = \infty$ for $B \geq B_{\text{suff}}$. (See **Appendix A.4** for the proof.)

As shown in Figure 1, a finite drift budget can permit consistent tracking, extending the finite horizon imposed by hard persistence. This construction illustrates how limited action revision can serve as a *soft persistence* to extend acceptable heterogeneous temporal behavior beyond the holding horizon.

5.2 Learning with soft persistence

The preceding analysis motivates us to propose Soft Persistent Regularization (SoPer), which learns drift constrained action dynamics to optimize task performance while exploring the high-dimensional temporal evolution of control.

Continuous-time objective. We parameterize the controller $\pi_{\boldsymbol{\theta}} = (\pi_{\boldsymbol{\theta}}^{\text{init}}, \boldsymbol{u}_{\boldsymbol{\theta}})$ by an initial-action policy $\pi_{\boldsymbol{\theta}}^{\text{init}} : \mathcal{X} \to \mathcal{P}(\mathcal{A})$ and an admissible action drift $\boldsymbol{u}_{\boldsymbol{\theta}} : \mathcal{Z} \to \mathbb{R}^{|\mathcal{A}|}$, where $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$ and $\boldsymbol{z}_t = (\boldsymbol{x}_t, \boldsymbol{a}_t)$. The resulting action process satisfies

$$\boldsymbol{a}_0 \sim \pi_{\boldsymbol{\theta}}^{\text{init}}(\cdot \mid \boldsymbol{x}_0), \qquad \dot{\boldsymbol{a}}_t = \boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{z}_t), \qquad t > 0. \tag{13}$$

The initial policy selects the starting action from the initial physical state, while the drift governs subsequent action revision through state-action feedback. SoPer learns the action dynamics by maximizing the expected return under a finite drift budget:

$$\max_{\boldsymbol{\theta}} \; J(\pi_{\boldsymbol{\theta}}) \qquad \text{subject to} \qquad \rho\, \mathbb{E}^{\pi_{\boldsymbol{\theta}}}\left[\int_0^\infty e^{-\rho t} \|\boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{z}_t)\|_2^2 \, dt\right] \leq B. \tag{14}$$

Here B bounds the discounted average squared action-change rate. This average constraint relaxes the pointwise budget in Eq. (10), allowing the controller to allocate action revision across states and action components. We can connects the all-time feasible process in Proposition 2 to the performance of the constrained optimum.

Corollary 3 (Performance of the drift-constrained optimum)**.** *Under Assumption 1, suppose the policy class contains a controller $\bar{\pi}$ satisfying the budget in Eq.* (14) *and $\ell(\boldsymbol{x}_t, \boldsymbol{a}_t) \leq \delta$ almost surely for all $t \geq 0$. If $\pi_B^\star$ is a global maximizer of Eq.* (14)*, then $0 \leq \mathbb{E}_{\boldsymbol{x}_0 \sim \mu_0}[V^\star(\boldsymbol{x}_0)] - J(\pi_B^\star) \leq \frac{\delta}{\rho}$. (See **Appendix A.5** for the proof.)*

Thus, a finite drift budget can preserve task performance when the policy class contains a sufficiently good feasible action process.

5.3 Practical implementation

While directly optimizing the continuous-time objective is challenging, we develop a practical implementation of SoPer by discretizing the action dynamics. Let $h > 0$ be the control interval and $\gamma = e^{-\rho h}$. For a continuing control step with current physical state $\boldsymbol{x}$ and incoming action $\boldsymbol{a}$, a first-order action update gives the deterministic action and the corresponding squared displacement:

$$\bar{\boldsymbol{a}}_{\boldsymbol{\theta}}(\boldsymbol{z}) = \boldsymbol{a} + h\, \boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{z}), \qquad \boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{a}), \qquad c_{\boldsymbol{\theta}}(\boldsymbol{z}) := \|\bar{\boldsymbol{a}}_{\boldsymbol{\theta}}(\boldsymbol{z}) - \boldsymbol{a}\|_2^2 = h^2 \|\boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{z})\|_2^2. \tag{15}$$

Consequently, a drift-rate budget B corresponds to a displacement budget $b = h^2 B$. Under above discretization, we can get easily integrate SoPer into standard reinforcement learning pipeline.

Discretized SoPer objective. Let r_k denote the discounted reward accumulated over $[kh, (k+1)h]$, with discount measured from kh. For $k \geq 1$, the displacement cost $c_{\boldsymbol{\theta}}(\boldsymbol{z}_k)$ corresponds to the action update over $[(k-1)h, kh]$. Approximating the drift as constant within each interval and using $\rho \int_0^h e^{-\rho s}\, ds = 1 - \gamma$ yields

$$\max_{\boldsymbol{\theta}} \; \mathbb{E}^{\pi_{\boldsymbol{\theta}}}\left[\sum_{k=0}^{\infty} \gamma^k r_k\right] \quad \text{subject to} \quad (1-\gamma)\, \mathbb{E}^{\pi_{\boldsymbol{\theta}}}\left[\sum_{k=1}^{\infty} \gamma^{k-1} c_{\boldsymbol{\theta}}(\boldsymbol{z}_k)\right] \leq b, \quad b = h^2 B. \tag{16}$$

Action-space parameterization. At a fixed control interval, we directly parameterize the next-action distribution $\pi_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{z}_k)$. Our implementation uses a diagonal squashed-Gaussian actor with

mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ and standard deviation $\boldsymbol{\sigma}_{\boldsymbol{\theta}}$ networks. Using the deterministic action prediction $\bar{\boldsymbol{a}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k) = \tanh \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}_k)$, we compute the drift cost as $c_{\boldsymbol{\theta}}(\boldsymbol{z}_k) = \|\bar{\boldsymbol{a}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k) - \boldsymbol{a}_{k-1}\|_2^2$.

Integration with actor-critic learning. As illustrated in Algorithm 1, we optimize the discretized objective using a primal-dual method. The Lagrangian of Eq. (16) is

$$\mathcal{L}_h(\boldsymbol{\theta}, \lambda) = \mathbb{E}^{\pi_{\boldsymbol{\theta}}}\left[\sum_{k=1}^{\infty} \gamma^k \big(r_k - \lambda c_{\boldsymbol{\theta}}(\boldsymbol{z}_k)\big)\right] + \frac{\lambda \gamma b}{1-\gamma}, \qquad \lambda \geq 0, \tag{17}$$

where we rescale the nonnegative multiplier to serve as the per-step penalty coefficient. For fixed λ, the final term is independent of the policy parameters. We parameterize the regularized action-value function with a critic network $Q_{\boldsymbol{\phi}} : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$. For a fixed policy π and multiplier λ, the corresponding value functions satisfy

$$Q_\lambda^\pi(\boldsymbol{z}, \boldsymbol{a}) = \mathbb{E}[r + \gamma V_\lambda^\pi(\boldsymbol{z}') \mid \boldsymbol{z}, \boldsymbol{a}], \quad V_\lambda^\pi(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{z})}\left[Q_\lambda^\pi(\boldsymbol{z}, \boldsymbol{a})\right] - \lambda c_\pi(\boldsymbol{z}). \tag{18}$$

Thus, Q_λ^π includes future displacement penalties, while the current penalty enters policy evaluation explicitly. The critic $Q_{\boldsymbol{\phi}}$ approximates $Q_\lambda^{\pi_{\boldsymbol{\theta}}}$ using bootstrapped targets from Eq. (18). For an off-policy implementation, the critic and actor minimize the following loss given samples from replay buffer $\mathcal{D}$:

$$\mathcal{L}_Q(\boldsymbol{\phi}) = \frac{1}{2}\,\mathbb{E}_{(\boldsymbol{z}, \boldsymbol{a}, r, \boldsymbol{z}') \sim \mathcal{D}}\left[\left(Q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{a}) - r - \gamma V_\lambda^\pi(\boldsymbol{z}')\right)^2\right], \tag{19}$$

$$\mathcal{L}_\pi(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{z} \sim \mathcal{D}, \boldsymbol{a} \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{z})}\left[-Q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{a}) + \lambda c_{\boldsymbol{\theta}}(\boldsymbol{z})\right], \tag{20}$$

The current displacement cost is differentiated directly through the actor, and the multiplier is updated from the estimated budget residual in Eq. (16):

$$\mathcal{L}_\lambda(\lambda) = \mathbb{E}_{\boldsymbol{z} \sim \mathcal{D}}\left[\lambda\left(b - c_{\boldsymbol{\theta}}(\boldsymbol{z})/\gamma\right)\right], \qquad \lambda \geq 0. \tag{21}$$

We specify the drift budget through a target per-dimensional root-mean-square displacement ε in normalized action space, with $b = |\mathcal{A}|\,\varepsilon^2$. This gives the budget a consistent per-coordinate interpretation across embodiments with different action dimensionalities.

Compatibility with RL backbones. SOPER computes its displacement regularizer from the policy's deterministic action output and the previously executed command, with gradients propagated directly through the actor. It can be incorporated into standard actor-critic backbones by augmenting their actor objectives and value targets. This formulation facilitates integration with complementary advances in policy architectures (Lee et al., 2025; Celik et al., 2025) and training regularization (Bhatt et al., 2024; Palenicek et al., 2026). The same objective can also be adapted to on-policy actor-critic methods (Schulman et al., 2017). By expressing persistence directly in action space, SOPER provides a general regularization mechanism applicable across different actuator semantics and learning backbones.

6 EXPERIMENTS

In this section, we conduct a comprehensive evaluation of SOPER in high-dimensional continuous control. We first compare its learning efficiency with recent baselines on whole-body control tasks spanning different objectives, action dimensionalities, and actuator semantics. We then analyze the temporal action dynamics learned by SOPER, and demontrate its application in real-world humanoid control. Our code and video results are in the **project page**.

Implementation and baselines. We compare SOPER with FlashSAC (Kim et al., 2026), which is a representative discrete-time RL baseline, and demonstrates strong performance and training efficiency across diverse high-dimensional control tasks. We also select TaCoS (Treven et al., 2024), and ASAP (Kwak & Hwang, 2026) as recent advances in continuous-time RL and temporal policy regularization. We implement all methods using FlashSAC as the actor-critic backbone, aligning network configurations and gradient-update budgets for a fair comparison. With actions normalized to $[-1, 1]^d$, we set the per-dimensional RMS displacement budget to $\varepsilon = 0.3$ for position-controlled tasks and $\varepsilon = 0.5$ for other actuator types (See **Appendix C**).

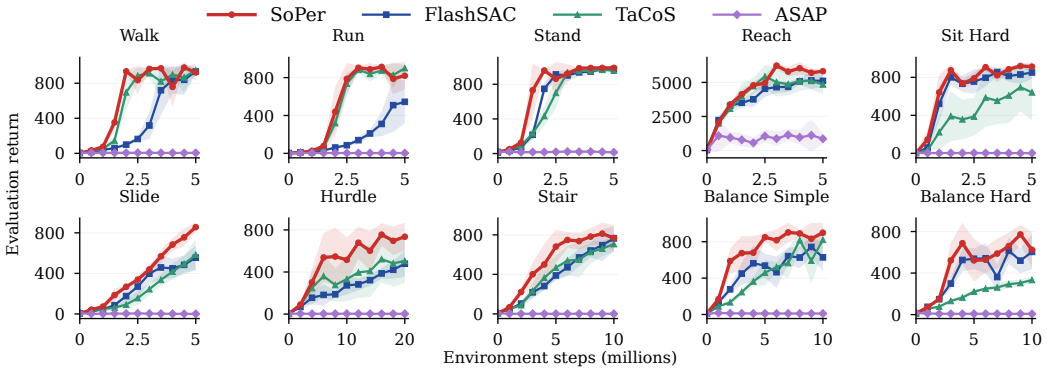

Figure 2: Evaluation returns on HumanoidBench H1-hand tasks. Lines show the mean and shaded regions show one standard deviation across 5 seeds.

6.1 High-dimensional control Performances

Task-driven control. We evaluate task-driven learning on HumanoidBench (Sferrazza et al., 2024) using the humanoid equipped with two dexterous hands with 61 actuators. The selected tasks span locomotion, postural control, and whole-body coordination. As shown in Figure 2, TaCoS demonstrate efficient learning on rhythmic locomotion tasks such as Walk and Run compared to FlashSAC, but performs less effectively on demanding postural and balance tasks such as sit-hard and balance-hard. This pattern is consistent with our analysis: persistence can benefit learning when commands remain useful over extended intervals, whereas rapid corrections required by some action components can restrict the admissible shared holding duration. SoPer achieves consistent sample-efficiency gains across these task types compared to baselines, supporting the flexibility of soft persistence under heterogeneous control requirements.

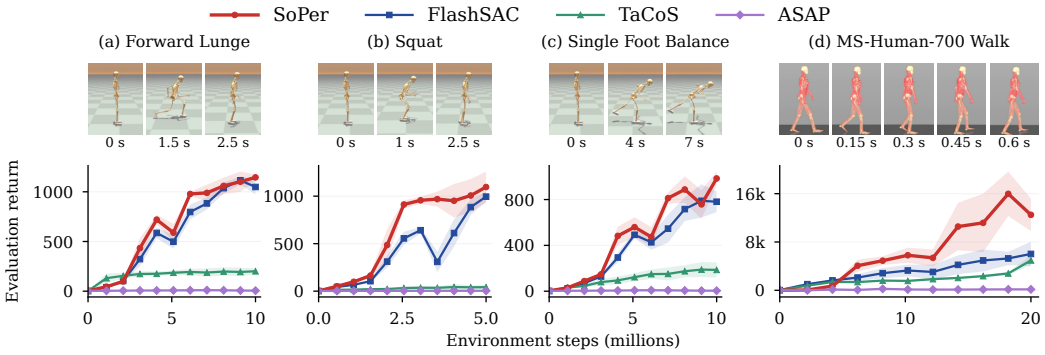

Figure 3: Evaluation performance and representative motion sequences on (a) Forward Lunge, (b) Squat, and (c) Single Foot Balance in LocoMuJoCo, and (d) MS-Human-700 Walk in MS-Gym. Lines show the mean and shaded regions show one standard deviation across 5 seeds. Frames above each plot show motions controlled by SoPer.

Imitation-based control. Imitation specifies reference behavior in addition to the task objective, placing explicit requirements on motion progression or postural accuracy. We evaluate these requirements through dynamic motion imitation and static posture maintenance across different actuator semantics.

We evaluate the imitation control of forward lunge, squatting and single-foot balancing using the torque-actuated skeletal model in LocoMuJoCo [1], which has 27 actuators. Forward Lunge and squatting require coordinated tracking of changing references, whereas single-foot balancing requires maintaining a prescribed posture. As shown in Figure 5(a)-(c), SoPer reaches strong imi-

[1] https://github.com/robfiras/loco-mujoco

tation returns earlier than all baselines on all three tasks, successfully complete the motion. These results support the sample-efficiency benefits of soft persistence for both dynamic tracking and postural stabilization.

We evaluate walking imitation with the full-body MS-Human-700 musculoskeletal model[2]. This task requires coordinating a 700-dimensional muscle-excitation action space. As shown in Figure 5(d), SoPer substantially outperforms all baselines, enabling cyclic whole-body walking. The results demonstrate that our method remains efficient under extremely high-dimensional action space, and generalize to different actuator semantics.

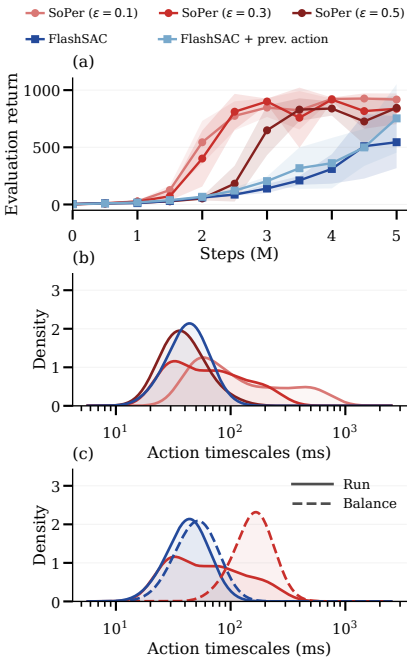

Figure 4: Algorithm analysis. (a) HumanoidBench Run performances, with mean and one standard deviation across 5 seeds. (b) Action timescales distribution in Run task. (c) Action timescales distribution across different task. Densities retain the common log-space KDE bandwidth of the source analysis.

6.2 Algorithm Analysis

We conduct ablation study and action dynamics analysis using HumanoidBench tasks.

Observation augmentation and drift budget. We ablate the augmented state and drift budget of SoPer. Figure 4(a) shows that Adding the previous action to FlashSAC does not reproduce the efficiency of SoPer. Across reasonable range of ε, SoPer achieves broadly comparable learning performance.

Measuring action timescales. We record all actuator commands during evaluatoon at control interval $h = 20\,\mathrm{ms}$. For each actuator i and episode, we compute the corresponding action timescale as

$$\tau_{\Delta,i} = h \frac{\mathrm{std}_k(a_{i,k})}{\sqrt{\mathrm{mean}_k[(a_{i,k+1} - a_{i,k})^2]}}. \tag{22}$$

Larger values indicate slower action changes relative to the signal's overall variation. In Figure 4(b), FlashSAC concentrates action within a narrow range of short timescales. SoPer exhibits broader distributions extending to longer timescales. Different drift budget also induce different temporal behavior to run at the same speed. Figure 4(c) shows that SoPer enables learning of task-depend temporal behaviors, exhibiting a broad timescale distribution on Run task but longer, more concentrated timescales on Balance task. This shift is consistent with the diverse temporal demands of cyclic locomotion and the more slowly varying adjustments associated with maintaining balance.

Real-world deployment. We integrate SoPer with PPO for on-policy training in simulation and deploy the resulting policy on an adult-sized humanoid robot. The robot successfully performs upper-body dance motions while maintaining balance, demonstrating the applicability of SoPer to on-policy learning and real-world high-dimensional control (see **Appendix E**).

7 Conclusion

In this paper, we investigate how heterogeneous temporal requirements can restrict shared commitment and how bounded action revision can extend the feasible control horizon. Building on this perspective, we introduced Soft Persistence Regularization (SoPer), which encourages heterogeneous action dynamics through a drift-budget formulation compatible with standard actor-critic learning. Experiments demonstrate improved sample efficiency over recent discrete-time and continuous-time RL baselines across diverse tasks and actuator semantics, including 700-actuator musculoskeletal control and whole-body humanoid robot control. These findings support soft persistence as a scalable and practical temporal inductive bias for learning high-dimensional embodied control.

[2] https://mujoco.readthedocs.io/en/stable/models.html#biomechanical

AI USE STATEMENT

In this work, we used generative AI tools for assist in the writing of proofs and method implementation. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, interpret results. Generate synthetic datasets, clean and reformat dataset, support qualitative and thematic data analysis are not applicable to this work. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness. 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 follows the ICLR Code of Ethics. We have considered the potential ethical impacts of this work. We report assumptions and limitations and strive to promote beneficial and responsible use of this work.

REPRODUCIBILITY STATEMENT

We provide an anonymous codebase, full hyperparameters, and exact evaluation protocols to enable full reproduction of our experimental results.

REFERENCES

Cameron Berg, Vittorio Caggiano, and Vikash Kumar. Sar: Generalization of physiological agility and dexterity via synergistic action representation. *Autonomous Robots*, 48(8):28, 2024.

Aditya Bhatt, Daniel Palenicek, Boris Belousov, Max Argus, Artemij Amiranashvili, Thomas Brox, and Jan Peters. Crossq: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PczQtTsTIX.

Vittorio Caggiano, Sudeep Dasari, and Vikash Kumar. MyoDex: A generalizable prior for dexterous manipulation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 3327–3346. PMLR, 2023. URL https://proceedings.mlr.press/v202/caggiano23a.html.

Onur Celik, Zechu Li, Denis Blessing, Ge Li, Daniel Palenicek, Jan Peters, Georgia Chalvatzaki, and Gerhard Neumann. DIME: Diffusion-based maximum entropy reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Aw6dBR7Vxj.

Alberto Silvio Chiappa, Alessandro Marin Vargas, Ann Zixiang Huang, and Alexander Mathis. Latent exploration for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 56508–56530, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/b0ca717599b7ba84d5e4f4c8b1ef6657-Abstract-Conference.html.

Kenji Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1): 219–245, 2000. doi: 10.1162/089976600300015961.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Kaibo He, Chenhui Zuo, Chengtian Ma, and Yanan Sui. DynSyn: Dynamical synergistic representation for efficient learning and control in overactuated embodied systems. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 18115–18132. PMLR, 2024. URL https://proceedings.mlr.press/v235/he24o.html.

Chengxiu Hua, Jiawen Gu, and Yushun Tang. Continuous q-score matching: Diffusion guided reinforcement learning for continuous-time control. *Advances in Neural Information Processing Systems*, 38:12573–12602, 2026.

Klemens Iten, Lenart Treven, Florian Dorfler, and Andreas Krause. Sample-efficient and scalable exploration in continuous-time rl. In *International Conference on Learning Representations*, volume 2026, pp. 73669–73696, 2026.

Yuri P Ivanenko, Germana Cappellini, Nadia Dominici, Richard E Poppele, and Francesco Lacquaniti. Coordination of locomotion with voluntary movements in humans. *The Journal of neuroscience*, 25(31):7238–7253, 2005.

Yanwei Jia and Xun Yu Zhou. Policy gradient and actor-critic learning in continuous time and space: Theory and algorithms. *Journal of Machine Learning Research*, 23(275):1–50, 2022.

Yanwei Jia and Xun Yu Zhou. q-learning in continuous time. *Journal of Machine Learning Research*, 24(161):1–61, 2023.

Amirmohammad Karimi, Jun Jin, Jun Luo, A. Rupam Mahmood, Martin Jagersand, and Samuele Tosatto. Dynamic decision frequency with continuous options. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7545–7552. IEEE, 2023. URL https://arxiv.org/abs/2212.04407.

Donghu Kim, Youngdo Lee, Minho Park, Kinam Kim, I Made Aswin Nahendra, Takuma Seno, Sehee Min, Daniel Palenicek, Florian Vogt, Danica Kragic, Jan Peters, Jaegul Choo, and Hojoon Lee. FlashSAC: Fast and stable off-policy reinforcement learning for high-dimensional robot control. *arXiv preprint arXiv:2604.04539*, 2026. URL https://arxiv.org/abs/2604.04539.

Jeongho Kim, Jaeuk Shin, and Insoon Yang. Hamilton-jacobi deep q-learning for deterministic continuous-time systems with lipschitz continuous controls. *Journal of Machine Learning Research*, 22(206):1–34, 2021.

Taisuke Kobayashi. L2c2: Locally lipschitz continuous constraint towards stable and smooth reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4032–4039. IEEE, 2022.

Kyoleen Kwak and Hyoseok Hwang. Enhancing control policy smoothness by aligning actions with predictions from preceding states. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22707–22715, 2026. doi: 10.1609/aaai.v40i27.39432. URL https://ojs.aaai.org/index.php/AAAI/article/view/39432.

Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. *arXiv preprint arXiv:2502.15280*, 2025.

I Lee, Hoang-Giang Cao, Cong-Tinh Dao, Yu-Cheng Chen, and I-Chen Wu. Gradient-based regularization for action smoothness in robotic control with reinforcement learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 603–610. IEEE, 2024. doi: 10.1109/IROS58592.2024.10801464. URL https://arxiv.org/abs/2407.04315.

Alberto Maria Metelli, Flavio Mazzolini, Lorenzo Bisi, Luca Sabbioni, and Marcello Restelli. Control frequency adaptation via action persistence in batch reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6862–6873. PMLR, 2020. URL https://proceedings.mlr.press/v119/metelli20a.html.

Siddharth Mysore, Bassel Mabsout, Renato Mancuso, and Kate Saenko. Regularizing action policies for smooth control with reinforcement learning. In *IEEE International Conference on Robotics and Automation*, 2021.

Daniel Palenicek, Florian Vogt, Joe Watson, Ingmar Posner, and Jan Peters. Xqc: Well-conditioned optimization accelerates deep reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 148719–148748, 2026.

Antonin Raffin, Jens Kober, and Freek Stulp. Smooth exploration for robotic reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1634–1644. PMLR, 2022. URL https://proceedings.mlr.press/v164/raffin22a.html.

Seyed A. Safavynia and Lena H. Ting. Task-level feedback can explain temporal recruitment of spatially fixed muscle synergies throughout postural perturbations. *Journal of Neurophysiology*, 107(1):159–177, 2012. doi: 10.1152/jn.00653.2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Pierre Schumacher, Daniel Häufle, Dieter Büchler, Syn Schmitt, and Georg Martius. DEP-RL: Embodied exploration for reinforcement learning in overactuated and musculoskeletal systems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=C-xa_D3oTj6.

Younggyo Seo, Carmelo Sferrazza, Haoran Geng, Michal Nauman, Zhao-Heng Yin, and Pieter Abbeel. Fasttd3: Simple, fast, and capable reinforcement learning for humanoid control. *arXiv preprint arXiv:2505.22642*, 2025.

Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv preprint arXiv:2403.10506*, 2024.

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

Lenart Treven, Bhavya Sukhija, Yarden As, Florian Dörfler, and Andreas Krause. When to sense and control? a time-adaptive approach for continuous-time RL. *Advances in Neural Information Processing Systems*, 37:63654–63685, 2024. URL https://arxiv.org/abs/2406.01163.

Yinuo Wang, Wenxuan Wang, Xujie Song, Tong Liu, Yuming Yin, Liangfa Chen, Likun Wang, Jingliang Duan, and Shengbo Li. ODE-based smoothing neural network for reinforcement learning tasks. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/6e0274fdb05f21654421db43cd8de042-Abstract-Conference.html.

Yunyue Wei, Chenhui Zuo, and Yanan Sui. Scalable exploration for high-dimensional continuous control via value-guided flow. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kIYNtxE13h.

Yufei Xue, Wentao Dong, Minghuan Liu, Weinan Zhang, and Jiangmiao Pang. A unified and general humanoid whole-body controller for versatile locomotion. In *Robotics: Science and Systems*, 2025.

Cagatay Yildiz, Markus Heinonen, and Harri Lähdesmäki. Continuous-time model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 12009–12018. PMLR, 2021.

A THEORETICAL PROOFS

Throughout this appendix, trajectories follow the deterministic state dynamics in Eq. (1). All admissible trajectories and policy evaluations are assumed to be well defined and measurable. Supremum norms range over the state domain and, for action-value functions, additionally over the action space. The bounded-reward assumption implies $\|V^\star\|_\infty \le r_{\max}/\rho$.

A.1 PERSISTENCE-ESTIMATION TRADE-OFF

Proposition 1. Under Assumptions 1, 2, and 3, let $\widehat{\pi}_\tau$ be a measurable greedy policy with respect to $\widehat{Q}_\tau$, executing each selected action for duration τ. Then, for every $\tau \in \mathcal{I}$, with $K = FL_\ell$,

$$\|V^\star - V^{\widehat{\pi}_\tau}\|_\infty \le \frac{\overline{D}(\tau) + 2\eta}{1 - e^{-\rho\tau}} \le \underbrace{\frac{K\tau}{2\rho}}_{\text{commitment}} + \underbrace{\frac{2\eta}{\rho\tau} + 2\eta}_{\text{estimation}}.$$

For fixed K and η, the commitment contribution in the final bound is nondecreasing in τ, while the estimation contribution is nonincreasing.

Proof. We first bound the single-interval commitment loss and then combine it with the loss induced by approximate action values.

Step 1: Single-interval commitment loss. Assumption 1 implies $\ell(\boldsymbol{x}, \boldsymbol{a}) \ge 0$ for every state-action pair. For each initial state $\boldsymbol{x}$, there exists an action $\boldsymbol{a}^\star(\boldsymbol{x})$ satisfying

$$\ell\big(\boldsymbol{x}, \boldsymbol{a}^\star(\boldsymbol{x})\big) = 0.$$

Fix a constant action $\boldsymbol{a}$. Since $V^\star \in C^1$, the chain rule along its state trajectory gives, almost everywhere,

$$\begin{aligned}\frac{d}{dt}\left[e^{-\rho t} V^\star(\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}})\right] \\ = -e^{-\rho t}\left[r(\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}}, \boldsymbol{a}) + \ell(\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}}, \boldsymbol{a})\right].\end{aligned}$$

Integrating over $[0, \tau]$ and rearranging yields

$$\begin{aligned}V^\star(\boldsymbol{x}) - \int_0^\tau e^{-\rho t} r(\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}}, \boldsymbol{a})\, dt - e^{-\rho\tau} V^\star(\boldsymbol{x}_\tau^{\boldsymbol{x},\boldsymbol{a}}) \\ = \int_0^\tau e^{-\rho t} \ell(\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}}, \boldsymbol{a})\, dt.\end{aligned} \tag{23}$$

Taking the infimum over constant actions gives

$$D(\boldsymbol{x}, \tau) = \inf_{\boldsymbol{a} \in \mathcal{A}} \int_0^\tau e^{-\rho t} \ell(\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}}, \boldsymbol{a})\, dt. \tag{24}$$

In particular, $D(\boldsymbol{x}, \tau) \ge 0$.

Now hold the initially optimal action $\boldsymbol{a}^\star(\boldsymbol{x})$. Forward invariance and the bounded-dynamics assumption imply

$$\left\| \boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}^\star(\boldsymbol{x})} - \boldsymbol{x} \right\| \le \int_0^t \left\| \boldsymbol{f}\big(\boldsymbol{x}_s^{\boldsymbol{x},\boldsymbol{a}^\star(\boldsymbol{x})}, \boldsymbol{a}^\star(\boldsymbol{x})\big) \right\| ds \le Ft.$$

Using the uniform Lipschitz condition on the task-gap rate,

$$\begin{aligned}0 \le \ell\big(\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}^\star(\boldsymbol{x})}, \boldsymbol{a}^\star(\boldsymbol{x})\big) \\ \le L_\ell \left\| \boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}^\star(\boldsymbol{x})} - \boldsymbol{x} \right\| \le Kt.\end{aligned}$$

Evaluating the infimum in Eq. (24) at this action and taking the supremum over initial states gives

$$\begin{aligned}\overline{D}(\tau) \le K \int_0^\tau t e^{-\rho t}\, dt \\ = \frac{K}{\rho^2}\left[1 - (1 + \rho\tau)e^{-\rho\tau}\right] \le \frac{K\tau^2}{2}.\end{aligned} \tag{25}$$

We retain the discounted integral when deriving the final bound.

Step 2: Performance loss from restricting the policy class. Fix $\tau \in \mathcal{I}$ and write $\gamma_\tau = e^{-\rho\tau}$. Define the interval reward

$$r_\tau(\boldsymbol{x}, \boldsymbol{a}) = \int_0^\tau e^{-\rho t} r(\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}}, \boldsymbol{a})\, dt.$$

For a policy π that selects an action at each holding decision, its Bellman operator is

$$(\mathcal{T}_\tau^\pi V)(\boldsymbol{x}) = \int_{\mathcal{A}} [r_\tau(\boldsymbol{x}, \boldsymbol{a}) + \gamma_\tau V(\boldsymbol{x}_\tau^{\boldsymbol{x},\boldsymbol{a}})]\, \pi(d\boldsymbol{a} \mid \boldsymbol{x}).$$

Because $|r_\tau| \le r_{\max}(1-\gamma_\tau)/\rho$, both $\mathcal{T}_\tau$ and $\mathcal{T}_\tau^\pi$ map bounded functions to bounded functions. For any bounded functions V and W,

$$\|\mathcal{T}_\tau V - \mathcal{T}_\tau W\|_\infty \le \gamma_\tau \|V - W\|_\infty,$$
$$\|\mathcal{T}_\tau^\pi V - \mathcal{T}_\tau^\pi W\|_\infty \le \gamma_\tau \|V - W\|_\infty.$$

The first inequality follows from $|\sup f - \sup g| \le \sup |f - g|$; the second follows by integrating against a probability measure. Their fixed points are $V_\tau^\star$ and V^π, respectively.

Using the fixed-point identity for $V_\tau^\star$,

$$\begin{aligned}\|V^\star - V_\tau^\star\|_\infty &\le \|V^\star - \mathcal{T}_\tau V^\star\|_\infty + \|\mathcal{T}_\tau V^\star - \mathcal{T}_\tau V_\tau^\star\|_\infty \\ &\le \overline{D}(\tau) + \gamma_\tau \|V^\star - V_\tau^\star\|_\infty.\end{aligned}$$

Therefore,

$$\|V^\star - V_\tau^\star\|_\infty \le \frac{\overline{D}(\tau)}{1-\gamma_\tau}. \tag{26}$$

Step 3: Performance loss from approximate action values. For a fixed state $\boldsymbol{x}$, let $\widehat{\boldsymbol{a}}$ be any action maximizing $\widehat{Q}_\tau(\boldsymbol{x}, \cdot)$. For every $\boldsymbol{a} \in \mathcal{A}$,

$$\begin{aligned}Q_\tau^\star(\boldsymbol{x}, \boldsymbol{a}) &\le \widehat{Q}_\tau(\boldsymbol{x}, \boldsymbol{a}) + \eta \\ &\le \widehat{Q}_\tau(\boldsymbol{x}, \widehat{\boldsymbol{a}}) + \eta \\ &\le Q_\tau^\star(\boldsymbol{x}, \widehat{\boldsymbol{a}}) + 2\eta.\end{aligned}$$

Taking the supremum over $\boldsymbol{a}$ yields

$$0 \le V_\tau^\star(\boldsymbol{x}) - Q_\tau^\star(\boldsymbol{x}, \widehat{\boldsymbol{a}}) \le 2\eta.$$

The greedy policy selects maximizing actions with probability one. Averaging over its possible choices therefore gives

$$0 \le V_\tau^\star(\boldsymbol{x}) - (\mathcal{T}_\tau^{\widehat{\pi}_\tau} V_\tau^\star)(\boldsymbol{x}) \le 2\eta.$$

By the policy-operator contraction and the fixed-point identity for $V^{\widehat{\pi}_\tau}$,

$$\begin{aligned}\|V_\tau^\star - V^{\widehat{\pi}_\tau}\|_\infty &\le \|V_\tau^\star - \mathcal{T}_\tau^{\widehat{\pi}_\tau} V_\tau^\star\|_\infty \\ &\quad + \|\mathcal{T}_\tau^{\widehat{\pi}_\tau} V_\tau^\star - \mathcal{T}_\tau^{\widehat{\pi}_\tau} V^{\widehat{\pi}_\tau}\|_\infty \\ &\le 2\eta + \gamma_\tau \|V_\tau^\star - V^{\widehat{\pi}_\tau}\|_\infty.\end{aligned}$$

Consequently,

$$\|V_\tau^\star - V^{\widehat{\pi}_\tau}\|_\infty \le \frac{2\eta}{1-\gamma_\tau}. \tag{27}$$

Combining Eqs. (26) and (27) through the triangle inequality proves

$$\|V^\star - V^{\widehat{\pi}_\tau}\|_\infty \le \frac{\overline{D}(\tau) + 2\eta}{1 - e^{-\rho\tau}}.$$

Step 4: Explicit dependence on the holding duration. Let

$$I_0(\tau) = \int_0^\tau e^{-\rho t}\, dt = \frac{1 - e^{-\rho\tau}}{\rho}, \qquad I_1(\tau) = \int_0^\tau t e^{-\rho t}\, dt.$$

Pairing t with $\tau - t$ gives

$$I_1(\tau) - \frac{\tau}{2} I_0(\tau) = \int_0^{\tau/2} \left(t - \frac{\tau}{2}\right) \left[e^{-\rho t} - e^{-\rho(\tau - t)}\right] dt \leq 0.$$

Indeed, on $[0, \tau/2]$, the first factor is nonpositive and the second is nonnegative. Together with Eq. (25), this implies

$$\frac{\overline{D}(\tau)}{1 - e^{-\rho\tau}} \leq \frac{K I_1(\tau)}{\rho I_0(\tau)} \leq \frac{K\tau}{2\rho}.$$

For the estimation contribution, the inequality $e^s - 1 \geq s$ for $s > 0$ gives

$$\frac{1}{1 - e^{-\rho\tau}} = 1 + \frac{1}{e^{\rho\tau} - 1} \leq 1 + \frac{1}{\rho\tau}.$$

Hence,

$$\frac{\overline{D}(\tau) + 2\eta}{1 - e^{-\rho\tau}} \leq \frac{K\tau}{2\rho} + \frac{2\eta}{\rho\tau} + 2\eta.$$

This holds for every $\tau > 0$, without a short-duration approximation.

Finally,

$$\frac{d}{d\tau} \frac{K\tau}{2\rho} = \frac{K}{2\rho} \geq 0, \qquad \frac{d}{d\tau} \left(\frac{2\eta}{\rho\tau} + 2\eta\right) = -\frac{2\eta}{\rho\tau^2} \leq 0.$$

These monotonicity statements concern the two upper-bound contributions under a common estimation-error envelope. They do not require unbiased estimates or imply monotonicity of the actual performance loss. □

A.2 Shared commitment in command tracking

Case analysis. Consider actions $\boldsymbol{a} \in [-A, A]^{|\mathcal{A}|}$, phase dynamics $\dot{\varphi}_i = \omega_i > 0$, references $q_i(\boldsymbol{\varphi}) = A \sin \varphi_i$, and reward

$$r(\boldsymbol{\varphi}, \boldsymbol{a}) = \min\left\{\epsilon - \|\boldsymbol{a} - \boldsymbol{q}(\boldsymbol{\varphi})\|_\infty, 0\right\}, \qquad 0 < \epsilon < A.$$

Starting from $\boldsymbol{\varphi}(0) = \mathbf{0}$, hold the initially matching action $\boldsymbol{a}(t) \equiv \mathbf{0}$. The longest holding duration with zero loss is

$$\tau_{\mathrm{hold},|\mathcal{A}|} = \min_{1 \leq i \leq |\mathcal{A}|} \frac{\arcsin(\epsilon/A)}{\omega_i} = \frac{\arcsin(\epsilon/A)}{\max_{1 \leq i \leq |\mathcal{A}|} \omega_i}.$$

With the existing frequencies, amplitude, and tolerance fixed, this duration is nonincreasing as components are added. It decreases strictly when a new component has a higher frequency than every existing component.

Proof. The phase dynamics imply $\varphi_i(t) = \omega_i t$. Under the held action, the tracking error is

$$\|\boldsymbol{a}(t) - \boldsymbol{q}(\boldsymbol{\varphi}(t))\|_\infty = A \max_{1 \leq i \leq |\mathcal{A}|} |\sin(\omega_i t)|.$$

Because the reward is continuous and nonpositive, zero accumulated loss over an interval is equivalent to remaining within tolerance throughout that interval.

Define

$$\vartheta = \arcsin(\epsilon/A) \in (0, \pi/2), \qquad \omega_{\max,|\mathcal{A}|} = \max_{1 \leq i \leq |\mathcal{A}|} \omega_i, \qquad \tau_* = \frac{\vartheta}{\omega_{\max,|\mathcal{A}|}}.$$

For every $t \in [0, \tau_*]$ and every component i,

$$0 \leq \omega_i t \leq \omega_{\max,|\mathcal{A}|} t \leq \vartheta.$$

Since $\sin$ is increasing on $[0, \pi/2]$,

$$A|\sin(\omega_i t)| \leq A \sin\vartheta = \epsilon.$$

Thus, all components remain within tolerance on $[0, \tau_*]$, establishing $\tau_{\mathrm{hold},|\mathcal{A}|} \geq \tau_*$.

Conversely, choose $i^\star$ such that $\omega_{i^\star} = \omega_{\max,|\mathcal{A}|}$. For any $T > \tau_*$, the interval

$$\left(\tau_*, \min\left\{T, \frac{\pi}{2\omega_{\max,|\mathcal{A}|}}\right\}\right)$$

is nonempty. At every time t in this interval,

$$A|\sin(\omega_{i^\star} t)| = A\sin(\omega_{\max,|\mathcal{A}|} t) > A\sin\vartheta = \epsilon.$$

The reward is therefore strictly negative on an interval of positive length. Consequently, no $T > \tau_*$ is a zero-loss holding duration, and

$$\tau_{\mathrm{hold},|\mathcal{A}|} = \tau_* = \frac{\arcsin(\epsilon/A)}{\omega_{\max,|\mathcal{A}|}}.$$

When a new component with frequency ω_{d+1} is added,

$$\tau_{\mathrm{hold},|\mathcal{A}|+1} = \frac{\arcsin(\epsilon/A)}{\max\{\omega_{\max,|\mathcal{A}|}, \omega_{|\mathcal{A}|+1}\}} \le \tau_{\mathrm{hold},|\mathcal{A}|}.$$

Since the numerator is strictly positive, the inequality is strict precisely when $\omega_{|\mathcal{A}|+1} > \omega_{\max,|\mathcal{A}|}$. □

A.3 Relaxing drift enlarges the feasible horizon

Proposition 2. For a fixed initial pair $(\boldsymbol{x}, \boldsymbol{a})$ with $\ell(\boldsymbol{x}, \boldsymbol{a}) \le \delta$,

$$0 \le B_1 \le B_2 \quad \Longrightarrow \quad \tau_{\max}(\boldsymbol{x}, \boldsymbol{a}; B_1, \delta) \le \tau_{\max}(\boldsymbol{x}, \boldsymbol{a}; B_2, \delta).$$

If an admissible drift generates a trajectory from $(\boldsymbol{x}, \boldsymbol{a})$ satisfying

$$\ell(\boldsymbol{x}_t, \boldsymbol{a}_t) \le \delta, \qquad \|\dot{\boldsymbol{a}}_t\|_2^2 \le B_\star < \infty, \qquad \forall t \ge 0,$$

then $\tau_{\max}(\boldsymbol{x}, \boldsymbol{a}; B, \delta) = \infty$ for every $B \ge B_\star$.

Proof. Fix the initial pair and tolerance. Let $\mathcal{H}_B$ denote the set of feasible horizons in Eq. (11), so that

$$\tau_{\max}(\boldsymbol{x}, \boldsymbol{a}; B, \delta) = \sup \mathcal{H}_B.$$

If $T \in \mathcal{H}_{B_1}$, there exists a continuous admissible drift $\boldsymbol{u}$ whose trajectory satisfies

$$\sup_{t\in[0,T]} \|\boldsymbol{u}(\boldsymbol{x}_t^{\boldsymbol{u}}, \boldsymbol{a}_t^{\boldsymbol{u}})\|_2^2 \le B_1, \qquad \sup_{t\in[0,T]} \ell(\boldsymbol{x}_t^{\boldsymbol{u}}, \boldsymbol{a}_t^{\boldsymbol{u}}) \le \delta.$$

For $B_2 \ge B_1$, the same drift and trajectory satisfy these conditions under budget B_2. Therefore, $\mathcal{H}_{B_1} \subseteq \mathcal{H}_{B_2}$. Taking suprema proves the claimed monotonicity.

For completeness, when $B = 0$, every feasible trajectory satisfies

$$\boldsymbol{u}(\boldsymbol{x}_t^{\boldsymbol{u}}, \boldsymbol{a}_t^{\boldsymbol{u}}) = \boldsymbol{0}, \qquad t \in [0, T].$$

Integrating the action dynamics gives $\boldsymbol{a}_t^{\boldsymbol{u}} = \boldsymbol{a}$. Conversely, the constant action is generated by the continuous drift $\boldsymbol{u} \equiv \boldsymbol{0}$. Hence,

$$\tau_{\max}(\boldsymbol{x}, \boldsymbol{a}; 0, \delta) = \sup\left\{T \ge 0 : \sup_{t\in[0,T]} \ell(\boldsymbol{x}_t^{\boldsymbol{x},\boldsymbol{a}}, \boldsymbol{a}) \le \delta\right\}.$$

This is the holding horizon of the prescribed initial action.

Finally, suppose one admissible drift satisfies the stated task-gap and drift bounds for all $t \ge 0$. For any $B \ge B_\star$ and any finite T, its restriction to $[0, T]$ witnesses that $T \in \mathcal{H}_B$. Every finite horizon is therefore feasible, yielding

$$\tau_{\max}(\boldsymbol{x}, \boldsymbol{a}; B, \delta) = \infty.$$

□

A.4 SOFT PERSISTENCE IN THE TRACKING EXAMPLE

Revisited case analysis. For the tracking task in Appendix A.2, starting from $(\boldsymbol{\varphi}(0), \boldsymbol{a}_0) = (\mathbf{0}, \mathbf{0})$, the scaled reference trajectory

$$a_i(t) = \left(1 - \frac{\epsilon}{A}\right) q_i(\boldsymbol{\varphi}(t)) = (A - \epsilon)\sin(\omega_i t)$$

remains within the tracking tolerance and satisfies

$$\|\dot{\boldsymbol{a}}_t\|_2^2 \le B_{\text{suff}}, \qquad B_{\text{suff}} = (A-\epsilon)^2 \sum_{i=1}^{|\mathcal{A}|} \omega_i^2.$$

It can be generated by a continuous admissible drift. Consequently,

$$\tau_{\max}(\mathbf{0}, \mathbf{0}; B, 0) = \infty, \qquad B \ge B_{\text{suff}}.$$

Proof. The reward is nonpositive, while unrestricted tracking $\boldsymbol{a}_t = \boldsymbol{q}(\boldsymbol{\varphi}(t))$ achieves zero reward at every time. Thus, $V^\star(\boldsymbol{\varphi}) = 0$ for every phase vector. The task-gap rate is therefore

$$\ell(\boldsymbol{\varphi}, \boldsymbol{a}) = -r(\boldsymbol{\varphi}, \boldsymbol{a}) = \left(\|\boldsymbol{a} - \boldsymbol{q}(\boldsymbol{\varphi})\|_\infty - \epsilon\right)_+, \tag{28}$$

where $(s)_+ = \max\{s, 0\}$.

We first construct a drift that generates the proposed trajectory and preserves the bounded action domain. Define the Lipschitz cutoff

$$\chi(s) = \min\left\{1, \frac{(A - |s|)_+}{\epsilon}\right\}.$$

It satisfies $\chi(s) = 1$ for $|s| \le A - \epsilon$, $\chi(\pm A) = 0$, and $0 \le \chi(s) \le 1$. Consider

$$u_i(\boldsymbol{\varphi}, \boldsymbol{a}) = \chi(a_i)(A - \epsilon)\omega_i \cos\varphi_i, \qquad i = 1, \dots, d. \tag{29}$$

Together with $\dot{\varphi}_i = \omega_i$, this defines a globally Lipschitz system with bounded derivatives. At each boundary $a_i = \pm A$, the corresponding action derivative vanishes. Uniqueness of solutions therefore preserves $\boldsymbol{a}_t \in [-A, A]^{|\mathcal{A}|}$.

For the proposed trajectory,

$$\varphi_i(t) = \omega_i t, \qquad a_i(t) = (A - \epsilon)\sin(\omega_i t),$$

we have $|a_i(t)| \le A - \epsilon$, so $\chi(a_i(t)) = 1$. Differentiating verifies

$$\dot{a}_i(t) = (A - \epsilon)\omega_i \cos(\omega_i t) = u_i(\boldsymbol{\varphi}(t), \boldsymbol{a}_t).$$

The initial conditions also hold, so this is the trajectory generated by the constructed drift.

Its tracking error satisfies

$$|a_i(t) - q_i(\boldsymbol{\varphi}(t))| = \epsilon|\sin(\omega_i t)| \le \epsilon.$$

Consequently, $\ell(\boldsymbol{\varphi}(t), \boldsymbol{a}_t) = 0$ for all $t \ge 0$. Moreover,

$$\begin{aligned} \|\dot{\boldsymbol{a}}_t\|_2^2 &= (A-\epsilon)^2 \sum_{i=1}^{|\mathcal{A}|} \omega_i^2 \cos^2(\omega_i t) \\ &\le (A-\epsilon)^2 \sum_{i=1}^{|\mathcal{A}|} \omega_i^2 = B_{\text{suff}}. \end{aligned} \tag{30}$$

For every $B \ge B_{\text{suff}}$, this single admissible drift satisfies the budget and zero task-gap condition for all time. Proposition 2 therefore gives

$$\tau_{\max}(\mathbf{0}, \mathbf{0}; B, 0) = \infty.$$

The budget B_{suff} is sufficient for this construction; the argument does not require it to be minimal among all admissible action paths. □

A.5 PERFORMANCE OF THE DRIFT-CONSTRAINED OPTIMUM

Corollary 3. Under Assumption 1, suppose the policy class contains a controller $\bar{\pi}$ satisfying the budget in Eq. (14) and

$$\ell(\boldsymbol{x}_t, \boldsymbol{a}_t) \le \delta \quad \text{almost surely for all } t \ge 0.$$

If $\pi_B^\star$ is a global maximizer of Eq. (14), then

$$0 \le \mathbb{E}_{\boldsymbol{x}_0 \sim \mu_0}[V^\star(\boldsymbol{x}_0)] - J(\pi_B^\star) \le \frac{\delta}{\rho}.$$

Proof. We first express the return loss of the feasible comparison controller $\bar{\pi}$ through its task-gap rate. Along almost every trajectory generated by $\bar{\pi}$, the chain rule gives, almost everywhere in time,

$$\frac{d}{dt}\left[e^{-\rho t} V^\star(\boldsymbol{x}_t)\right] = -e^{-\rho t}\left[r(\boldsymbol{x}_t, \boldsymbol{a}_t) + \ell(\boldsymbol{x}_t, \boldsymbol{a}_t)\right].$$

Integrating over $[0, T]$ yields

$$\begin{aligned} V^\star(\boldsymbol{x}_0) - \int_0^T e^{-\rho t} r(\boldsymbol{x}_t, \boldsymbol{a}_t)\, dt - e^{-\rho T} V^\star(\boldsymbol{x}_T) \\ = \int_0^T e^{-\rho t} \ell(\boldsymbol{x}_t, \boldsymbol{a}_t)\, dt. \end{aligned} \tag{31}$$

The classical optimality condition ensures $\ell \ge 0$. The assumed task-gap tolerance therefore implies

$$0 \le \int_0^T e^{-\rho t} \ell(\boldsymbol{x}_t, \boldsymbol{a}_t)\, dt \le \frac{\delta}{\rho}(1 - e^{-\rho T}).$$

Since the reward is bounded,

$$\left|e^{-\rho T} V^\star(\boldsymbol{x}_T)\right| \le e^{-\rho T} \frac{r_{\max}}{\rho} \longrightarrow 0.$$

The discounted reward integral converges absolutely. Taking $T \to \infty$ in Eq. (31) gives

$$\begin{aligned} V^\star(\boldsymbol{x}_0) - \int_0^\infty e^{-\rho t} r(\boldsymbol{x}_t, \boldsymbol{a}_t)\, dt \\ = \int_0^\infty e^{-\rho t} \ell(\boldsymbol{x}_t, \boldsymbol{a}_t)\, dt. \end{aligned} \tag{32}$$

Averaging over the initial-state distribution and controller randomization yields

$$\begin{aligned} 0 &\le \mathbb{E}_{\boldsymbol{x}_0 \sim \mu_0}[V^\star(\boldsymbol{x}_0)] - J(\bar{\pi}) \\ &= \mathbb{E}^{\bar{\pi}}\left[\int_0^\infty e^{-\rho t} \ell(\boldsymbol{x}_t, \boldsymbol{a}_t)\, dt\right] \le \frac{\delta}{\rho}. \end{aligned}$$

These expectations and limits are justified by bounded rewards, bounded $V^\star$, and the integrable bound $\delta e^{-\rho t}$ on the comparison controller's discounted task-gap rate.

Because $\bar{\pi}$ belongs to the optimization class and satisfies the drift budget, global optimality implies

$$J(\pi_B^\star) \ge J(\bar{\pi}).$$

The unrestricted value $V^\star$ upper-bounds the return of every admissible controller. Hence,

$$\begin{aligned} 0 &\le \mathbb{E}_{\boldsymbol{x}_0 \sim \mu_0}[V^\star(\boldsymbol{x}_0)] - J(\pi_B^\star) \\ &\le \mathbb{E}_{\boldsymbol{x}_0 \sim \mu_0}[V^\star(\boldsymbol{x}_0)] - J(\bar{\pi}) \le \frac{\delta}{\rho}. \end{aligned}$$

This proves the result. The optimizer satisfies the average drift budget by feasibility, but need not inherit the comparison controller's pointwise task-gap bound.

To connect this result with Proposition 2, an all-time comparison controller with $\|\dot{\boldsymbol{a}}_t\|_2^2 \le B_\star \le B$ almost surely automatically satisfies

$$\rho\, \mathbb{E}^{\bar{\pi}}\left[\int_0^\infty e^{-\rho t} \|\dot{\boldsymbol{a}}_t\|_2^2\, dt\right] \le \rho B_\star \int_0^\infty e^{-\rho t}\, dt = B_\star \le B.$$

Thus, an all-time feasible controller in the policy class provides the feasible comparator required by the corollary. □

B PSEUDOCODE OF SOPER

Algorithm 1 Soft Persistence Regularization

Require: Initialize actor parameters $\boldsymbol{\theta}$, critic parameters $\boldsymbol{\phi}$, multiplier $\lambda \geq 0$, budget b, discount γ, and learning rates $\eta_Q, \eta_\pi, \eta_\lambda$.
Initialize replay buffer $\mathcal{D}$ and backbone state.

for each training iteration **do**
  Collect data using $\pi_{\boldsymbol{\theta}}$ and append them to $\mathcal{D}$.
  $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \eta_Q \nabla_{\boldsymbol{\phi}} \mathcal{L}_Q(\boldsymbol{\phi})$. ▷ Eq. (19)
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta_\pi \nabla_{\boldsymbol{\theta}} \mathcal{L}_\pi(\boldsymbol{\theta})$. ▷ Eq. (20)
  $\lambda \leftarrow [\lambda - \eta_\lambda \nabla_\lambda \mathcal{L}_\lambda(\lambda)]_+$. ▷ Eq. (21)
end for

C IMPLEMENTATION DETAILS AND HYPERPARAMETERS

Shared implementation. We implement SOPER on top of the official FlashSAC codebase[3] and use the same backbone for all comparisons. For TaCoS[4] and ASAP[5], we refer to their official implementations and adapt their method-specific components to this backbone. Thus, these baselines are implementations within our common training framework. We align the actor and critic architectures, replay capacity, batch size, reward normalization, optimizer schedules, evaluation protocol, and environment-step budgets. Method-specific differences, including action duration and auxiliary prediction losses, are retained and inferred from original implementation. Code for reproducing the paper results are in the anonymous link: `https://anonymous.4open.science/r/FlashSAC_dev-0708/`.

Table 1: Shared backbone hyperparameters. One vector step collects a transition from each parallel environment.

Parameter	Value
Actor	2 FlashSAC residual blocks, width 128
Twin critics	2 FlashSAC residual blocks each, width 256
Distributional critic	101 atoms, support $[-5, 5]$
Replay capacity / minibatch size	10^6 / 2048
Discount / return horizon	0.99 per base step / one-step backup
Optimizer	Adam
Learning rate	Cosine decay from 3×10^{-4} to 1.5×10^{-4}
Critic updates	One per vector environment step after warm-up
Actor update interval	Every 2 critic updates
Critic target update coefficient	0.01
Initial entropy temperature	0.01, automatically learned
Target entropy parameter	Target standard deviation 0.15
Reward normalization	Enabled; return-scale parameter 5
Parallel training environments	16 (HumanoidBench); 128 (imitation tasks)
Evaluation	Deterministic policy; 50 episodes per evaluation

D IMITATION TASKS AND REWARDS

LocoMuJoCo. We use the torque-controlled `SkeletonTorque` environment from LocoMuJoCo[6] with reference tasks `onesteplong` (Forward Lunge), `squat` (Squat), and `balance` (Single Foot Balance). They require tracking a forward-lunge motion, a squatting motion, and a single-foot balancing motion, respectively. All three use trajectory-based initialization, the

[3] `https://github.com/Holiday-Robot/FlashSAC`
[4] `https://github.com/lasgroup/TaCoS`
[5] `https://github.com/AIRLABkhu/ASAP`
[6] `https://github.com/robfiras/loco-mujoco`

Table 2: Method-specific hyperparameters used in the main comparisons.

Method	Parameter	Value
SoPer	RMS budget ε	0.3 (HumanoidBench); 0.5 (imitation)
	Initial / maximum multiplier	0.01 / 100
	Multiplier learning rate	3×10^{-4} initially; actor schedule
	Previous-action / validity input	Enabled
	Drift penalty in critic target	Enabled
TaCoS	Interaction cost	0.1 per decision
	Hold duration	1–30 base steps
	Physical duration	0.02–0.60 s (HumanoidBench, MS-Gym); 0.01–0.30 s (LocoMuJoCo)
	Warm-up / replay availability	10,000 base steps / 1 completed transition
ASAP	Spatial / prediction / temporal weight	30 / 2 / 0.5
	Predictor target coefficient	0.01 per actor update
FlashSAC, SoPer, ASAP	Replay warm-up	10,000 transitions
	Exploration repetition parameters	$\zeta_\mu = 2$, $\zeta_{\max} = 16$
TaCoS	Exploration repetition cap	$\zeta_{\max} = 1$

Table 3: Training budgets, measured in base environment steps.

Suite	Tasks	Steps
HumanoidBench	Walk, Run, Stand, Reach, Sit Hard, Slide	5×10^6
	Stair, Balance Simple, Balance Hard	10^7
	Hurdle	2×10^7
LocoMuJoCo	Forward Lunge, Single Foot Balance	10^7
	Squat	5×10^6
MS-Gym	MS-Human-700 Walk	2×10^7

`GoalTrajMimic` goal, `RootPoseTrajTerminalStateHandler`, a 0.01 s control interval, and a maximum of 1000 steps.

The configured `MimicReward` is

$$r_{\text{Loco}} = \max\big\{0,\, 0.4e^{-10D_q} + 0.2e^{-2D_{\dot{q}}} + 0.5e^{-100D_p} + 0.3e^{-10D_o} + 0.1e^{-0.1D_\omega} + 0.1e^{-0.1D_v} - \min(0.01C_{\text{bounds}}, 1)\big\}. \tag{33}$$

The D terms compare simulation and reference: generalized position, generalized velocity, relative mimic-site position, relative orientation representation, and relative angular and linear velocity. They use mean squared errors, except that D_q adds the implementation's mean quaternion angular distance to the non-quaternion position MSE. C_{bounds} is the mean squared action-bound violation. Acceleration, torque, and action-rate penalties have zero coefficients. The tracking weights are not normalized; both velocity terms receive weight 0.1.

MS-Human-700 Walk. We use the full-body locomotion environment of MS-Gym[7], controlling 700 muscle actuators to follow the provided walking reference. The control interval is 0.02 s. Episodes initialize at a randomized reference phase and track up to five gait cycles, with earlier termination upon excessive pose error. Let q exclude the three pelvis-translation coordinates, p be pelvis position, and $\tilde{x}_k = x_k - p$ be the pelvis-relative position of a tracked body; starred quantities denote the reference. The native reward is

$$r_{\text{MS}} = -50\|q - q^*\|_2^2 - 50\sum_{k\in\mathcal{K}} \|\tilde{x}_k - \tilde{x}_k^*\|_2^2 - 50\|p - p^*\|_2^2 + \frac{0.1}{700}\sum_{i=1}^{700} f_i + 100\,\mathbb{1}_{\text{healthy}}. \tag{34}$$

[7] https://github.com/LNSGroup/msgym

Here f_i is the signed MuJoCo actuator force, as used by the native implementation; it is not a squared-excitation cost. The tracked bodies are the sternum, head/neck, bilateral toes, and bilateral proximal hand rows. The healthy condition requires mean absolute pose error no greater than 0.05. The pose term uses coordinate differences, including the remaining root-orientation coordinates.

E Real-World Deployment

Robot and simulation. We evaluate SoPer in-place upper-body imitation on a Kuavo 4 Pro Max humanoid. Training uses MuJoCo[8] with an S49-derived model containing a floating base, 26 controlled leg and arm joints, and two unactuated head joints. Each arm has seven joints. The model has no waist joint and a nominal mass of $59.4\,\mathrm{kg}$. Robot documentation is available through LeTools.[9]

Joint-position targets are tracked by torque-limited PD control,

$$\tau_j = \mathrm{clip}\big(K_{p,j}(q_j^{\mathrm{tar}} - q_j) - K_{d,j}\dot{q}_j, -\bar{\tau}_j, \bar{\tau}_j\big)\,. \tag{35}$$

The model includes passive damping and joint friction. Torque caps are set to $0.8\min(\tau_j^{\mathrm{S49\,URDF}}, \tau_j^{\mathrm{S46\,training}})$; gains, reflected inertia, and friction use S46-derived priors. These parameters are not identified on the deployed hardware.

Task and policy interface. The robot stands on a level plane and tracks an analytic, 14-joint arm-dance reference with an $8\,\mathrm{s}$ period and smooth entry and exit envelopes. Physics runs at $200\,\mathrm{Hz}$ and the policy at $50\,\mathrm{Hz}$; episodes last up to $20\,\mathrm{s}$. Training starts at a uniformly sampled reference phase with matching arm positions and velocities. Evaluation starts at phase zero and uses deterministic actions. Observation noise, dynamics randomization, external disturbances, and actuator delay are disabled.

The actor and critic share a 118-dimensional observation comprising body angular velocity, projected gravity, a four-dimensional standing command, positions relative to the default pose and velocities of the 26 controlled joints, the previous action, reference arm positions and velocities, and sine/cosine phase coordinates. Neither network receives privileged state information. The Gaussian policy outputs a 26-dimensional residual action, mapped to joint targets as

$$q_{t,j}^{\mathrm{tar}} = \mathrm{clip}_{\mathcal{Q}_j^{\mathrm{soft}}}\left(q_j^{\mathrm{default}} + \mathbf{1}_{j\in\mathcal{A}} q_{t,j}^{\mathrm{ref}} + s_j a_{t,j}\right), \quad s_j = \begin{cases} 0.25, & j \in \text{legs}, \\ 0.20, & j \in \text{arms}, \end{cases} \tag{36}$$

where $\mathcal{A}$ is the arm-joint set and $\mathcal{Q}_j^{\mathrm{soft}}$ spans 90% of the model's joint-position range. Raw actions are not clipped before this mapping. Targets are held between policy updates. Deployment requires matching joint order, signs, offsets, and low-level control timing and saturation.

Reward and training. The imitation reward is

$$r_{\mathrm{arm}} = 4\exp\left(-\frac{\|q_{\mathcal{A}} - q^{\mathrm{ref}}\|_2^2}{14\cdot 0.09}\right) + 0.25\exp\left(-\frac{\|\dot{q}_{\mathcal{A}} - \dot{q}^{\mathrm{ref}}\|_2^2}{14}\right). \tag{37}$$

Additional terms encourage standing height, upright posture, foot contact, and zero planar velocity, while penalizing slip, undesired contacts, abrupt actions, joint acceleration, power, torque, and joint-limit violations. Episodes terminate on excessive tilt, low base height, or sustained arm-tracking error.

We train the SoPer PPO variant with separate actor and critic MLPs of widths $(512, 256, 128)$, ELU activations, and observation normalization. The drift budget is set to $\varepsilon = 0.3$, consistent with HumanoidBench evaluations. The shaped reward with drift cost $r_t - \lambda c_t$ is used for value targets and advantages. The actor loss also includes the mean drift cost and divided by the pre-normalization advantage standard deviation.

Each PPO update collects 24 steps from 512 environments and uses five epochs and four mini-batches. The configured training budget is 10M steps, achieving mean joint angle error of 2.84°.

[8] https://mujoco.readthedocs.io/en/stable/

[9] https://www.letools.lejurobot.com/docs.html?type=learning#get_started/letools.md

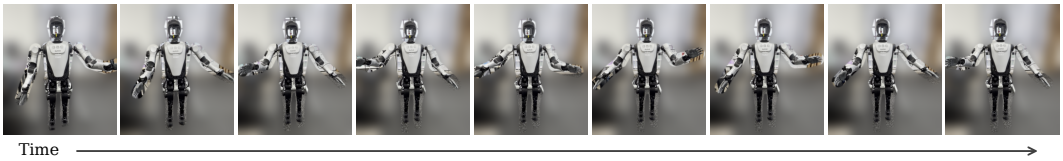

Figure 5: Sim-to-real deployment of SoPer.