Policy Gradient for Continuous-Time Robust Markov Decision Processes

Anonymous authors
Paper under double-blind review

Abstract

The framework of robust Markov decision processes (RMDPs) allows the design of reinforcement learning agents that satisfy performance guarantees under worst-case transition dynamics. Traditional RMDPs consider discrete-time dynamics and recently, sample-efficient policy gradient algorithms have been considered in this context. This paper investigates policy gradient algorithms within a continuous-time RMDP framework. Policy gradients and adversarial gradients are derived using pathwise and adjoint-based formulas for stochastic and ordinary differential equations. We propose double-loop optimisers to obtain linear convergence in the oracle-based setting and an $\tilde{\mathcal{O}}(\frac{1}{\epsilon^2})$ sample complexity in the sample-based setting in an analysis which also derives novel tools for the framework of undiscounted total cost MDPs. Additionally, we propose mean-field optimisers as distributional optimisers with an $\tilde{\mathcal{O}}(\frac{1}{K})$ oracle-based convergence rate and an $\tilde{\mathcal{O}}(\frac{N^2}{\epsilon})$ sample complexity under N-particle approximation. The effectiveness of continuous-time policy gradient algorithms is confirmed for both optimisers on continuous-time RMDPs with neural ordinary differential equation dynamics.

1 Introduction

Reinforcement learning agents require data trajectories of states, actions, and rewards to optimise the long-term cumulative reward. Such data trajectories are often obtained from poorly specified transition models, and therefore when an agent transfers to the true transition model, its performance will be far from optimal. The framework of robust Markov decision processes (RMDPs; Wiesemann et al. (2013); Nilim & Ghaoui (2005)) provides a natural approach to deal with such problems, with relevance to sim2real and robust control applications.

RMDPs optimise the policy of an agent under worst-case transition dynamics, based on some pre-defined uncertainty set. While RMDPs are often studied in mostly theoretical settings, they have also found their way to policy gradient algorithms, which are highly scalable and performant. In particular, double-loop policy gradient algorithms (Wang et al., 2024; 2025; Bossens & Nitanda, 2025; Li et al., 2026b) perform repeated gradient ascent in the space of transition kernels to guarantee that the policy at each time step is evaluated on its worst-case transition kernel. These works have found strong theoretical and empirical support.

Many problems in various application domains, e.g. in control, physics, chemistry, video generation, etc., are modeled within continuous-time formalisms such as partial differential equations (PDEs) and continuous-time Markov chains, which contrasts with traditional MDP based frameworks. For instance, recent works investigate neural ordinary differential equations (ODEs) (Chen et al., 2018), neural stochastic differential equations (SDEs) (Tzen & Raginsky, 2019), diffusion models (Song et al., 2021), and gradient flow Santambrogio (2017). In RL, while these formalisms have been used to a varying degree for varying purposes (exploration, policy definition, etc.), their use for policy optimisation algorithms in continuous-time systems is limited to relatively few works (Zhan et al., 2023; Zhao et al., 2023; Ainsworth et al., 2021; Munos, 2006). More so, the design of continuous-time policy gradient algorithms for RMDPs remains an unexplored challenge.

With the above in mind, our paper makes the following contributions:

- We introduce a novel framework that we will call *Continuous-time robust Markov decision processes*, which considers decision-making with worst-case robustness by extending the RMDP framework to the continuous-time setting.
- Pathwise and adjoint-based gradient computations for the policy and the adversary under deterministic and stochastic policies in a continuous-time RMDP problems, including dynamics based on ODE and SDE systems.
- A last-iterate convergence proof for two classes of policy-gradient algorithms to solve the continuous-time RMDP problem: i) a double-loop algorithm for gradient-based policy optimisation with linear convergence in the oracle-based setting and a sample complexity of $\tilde{\mathcal{O}}(\frac{1}{\epsilon^2})$ in the sample-based setting; and ii) a single-loop mean-field optimisation algorithm for distributional gradient-based policy optimisation with oracle-based rate of $\tilde{\mathcal{O}}(\frac{1}{K})$ and sample-based rate of $\tilde{\mathcal{O}}(\frac{N^2}{\varepsilon})$. The double-loop analysis in particular also introduces new tools for analysing rectangularity under indirect parametrisation, a performance difference for transition kernels within the total cost RMDP framework, and PL conditions for the policy and the adversary.
- Empirical experiments comparing the double-loop and mean-field optimisers with different instantiations of the policy and adversary optimisers. Experiments on LQR with neural ODE dynamics and neural network parametrised uncertainty sets show that robust discrete-time policy gradient algorithms have poor robust optimisation performance in continuous-time RMDPs while continuous-time policy gradient algorithms have a clear benefit in the setting.

2 Related work

2.1 Continuous-time formulations in RL

First we consider previous works that consider continuous-time formalisms in the non-robust RL and control. From a technical perspective we distinguish between HJB-based algorithms, pathwise algorithms, and adjoint-based algorithms for policy optimisation as the most directly related works.

HJB-based value-based algorithms approximate the Q-value from which the optimal control can be determined Kim et al. (2021); Jia & Zhou (2023). HJB-based actor-critic algorithms use a similar Q-value approximation to approximate the policy gradient Cohen et al. (2025). Our paper focuses on pure policy gradient algorithms without any function approximation, instead making use of an adjoint state and applying Pontryagin's maximum principle (Pontryagin et al., 1962).

One of our contributions is the formulation of pathwise policy gradients for continuous-time RMDPs. Munos (2006) previously derived the policy gradient for continuous-time (non-robust) undiscounted MDPs with deterministic dynamics based on ODEs. The work showed that likelihood ratio methods result in exponential blow-up of the variance and proposed a pathwise gradient that converges almost surely to the true gradient when the time step approaches zero. Our work takes place in the same setting, and in addition to extending the pathwise policy gradient estimator to time intervals, we also derive a pathwise adversarial gradient for continuous-time RMDPs.

Another contribution is the formulation of adjoint-based policy gradients for continuous-time RMDPs. Along the same lines, Zhan et al. (2023) leverage Pontryagin's maximum principle to connect continuous-time policy optimisation to minimizing the integral of the associated Hamiltonian of a neural ODE over a fixed time horizon. They use first-order optimisation methods and employ a state augmentation technique to convert the total cost into a state component, after which an adjoint method is employed to calculate the policy gradient. Within the discounted MDP setting, Jia & Zhou (2022) explores policy gradient and actor-critic in continuous time discounted MDPs, where the state dynamics are governed by an SDE. Similarly, Zhao et al. (2023) proposes continuous-time policy optimisation with discounted MDPs, showing performance-difference and local-approximation formulas, and extending these from basic policy gradient to PPO and TRPO. These approaches do not present convergence rates for the optimisers and do not deal with worst-case robustness. Most closely related to our work is Ainsworth et al. (2021), which proposes a continuous-time policy gradient

with the same total cost objective albeit without any proof. Our adjoint-based policy gradient is similar except to this work. In addition to providing its theoretical proof, we also derive an adversary policy gradient for robust MDPs. Two further differences are that we also include time-dependencies in the policy and the dynamics and that our gradient is not limited to the sample-and-hold model.

Other work employing the maximum principle in continuous-time includes Leeftink et al. (2025), where the authors assume imperfect knowledge of the underlying systems' dynamics and proceed to probabilistically minimize the mean Hamiltonian with respect to epistemic uncertainty. However, they do not frame the problem in a robust optimisation framework. The authors also propose a numerical method scalable to large-scale probabilistic dynamical models including ensemble neural ODEs.

Some works that tackle policy gradients in continuous-time for SDEs and stochastic control problems include Zhou et al. (2024); Zhou & Lu (2025). The former optimises SDEs (viewed as continuous-time generative models in this context) using policy gradient and a perturbation process argument. The latter analyses the gradient flow of the stochastic control as the continuous-time limit of policy gradient. Lastly, on a more general note, the foundational work Sirignano & Spiliopoulos (2017) proposes a computationally efficient stochastic gradient descent in continuous-time which has found its way to many of the above related works.

Beyond policy optimisation in environments with continuous-time dynamics, continuous-time formalisms also find their way to alternative directions in RL. For instance, continuous-time formalisms have found their way in the use of diffusion policies (e.g. Ren et al. (2025)) which can be used for improved stability and exploration. Another example is the use of gradient flow for RL, where they apply continuous-time methods for parameter updates (Kerimkulov et al., 2025; Zorba et al., 2025; Pfau et al., 2025; Müller et al., 2025). While gradient flow techniques are orthogonal to the continuous-time RMDP framework, we also demonstrate how mean-field gradient flow (Liu et al., 2025) can be applied to distributional optimisation within the framework.

Issues arising from discretisation and analogs between discrete-time and continuous-time formalisms are of wide interest. Formalisms such as continuous-time MDPs and semi-MDPs focus on MDPs with discrete state-action spaces where each transition takes a random amount of time (Bradtke & Duff, 1994). A discretised approach to the policy gradient was previously explored for SDEs (Zhou et al., 2024). The relation between the Q-function in discrete-time RL and the Hamiltonian has been discussed in earlier works (e.g. Mehta & Meyn (2009)). Our empirical study investigates the time discretisation sensitivity and the comparison of discrete-time vs continuous-time algorithms in the context of robust PDE dynamics; our regret analyses are based on the discrete-time framework of undiscounted total cost MDPs; and we show that the discrete-time stochastic and deterministic policy gradient Sutton & Barto (2012); Silver et al. (2014) converge to our policy gradient formulations as $\Delta t \to 0$ under a sample-and-hold model. Naturally, in practice, a much more coarse time step should be chosen, which is where the benefit of continuous-time algorithms comes in. Our study shows that robust discrete-time policy gradient algorithms have poor robust optimisation performance in continuous-time RMDPs and we empirically demonstrate the benefit of continuous-time policy gradient algorithms.

2.2 Robust reinforcement learning and control

Now we discuss related works in robust reinforcement learning, which include policy gradient algorithms for RMDPs and PDE-based techniques for robust control and RL.

Our first proposed optimiser is an instance of the class of *double-loop policy gradient algorithms* in RMDPs (Wang et al., 2024; 2025; Bossens & Nitanda, 2025; Li et al., 2026b), which solve the inner problem ϵ-optimally at each iteration. In the oracle-based last-iterate regret, we obtain a linear convergence rate, which matches recent results (Wang et al., 2024; Bossens & Nitanda, 2025). In the sample-based average regret, we obtain an $\tilde{\mathcal{O}}(\frac{1}{\epsilon^3})$ sample complexity, which matches the state-of-the-art (Bossens & Nitanda, 2025). In addition to our work considering continuous-time dynamics, there are a few further differences. Our formulation focuses on the undiscounted total cost MDP setting, which is a discretisation of our continuous-time RMDP framework which has equivalent policy gradients as $\Delta t \to 0$. We also do not consider constraints (as in robust constrained MDPs; Bossens & Nitanda (2025)). Our regret analysis also holds under a general policy parameterization whereas these previous works considered direct and softmax parameterizations only. Our work requires only

Lipschitz-continuity properties in the policy, dynamics, and cost-functions, from which Polyak-Lojasiewicz (PL) conditions are proven for the policy and applied within the total cost MDP framework. Recent work challenges subgradient dominance of the robust cost function for general RMDPs but points to the property holding with further assumptions (e.g. regularised objectives, uniqueness of the worst transition kernel) (Kitamura et al., 2026). Our results are based on separate gradient dominance conditions for the policy and adversary objectives under specific smoothness, differentiability, and Lipschitz assumptions, and a restriction on the transition kernel mismatch between adversarial updates. Together with the upper bound of the iterates' cost deviations to the robust cost, as guaranteed by the double-loop policy algorithm, this leads to a robust cost upper bound. For the adversary, we proposed a modified PL condition suitable for projected gradient descent. Our work is the first to propose a policy gradient solution for continuous-time RMDPs and we consider a range of policy gradients, including stochastic adjoint-based, deterministic adjoint-based, and pathwise. Rectangularity may be required for the PL condition so we also derive a rectangularity result under an indirect parameterization to support the analysis. It may under some conditions be necessary apply a correction for non-rectangular sets as in Li et al. (2026b), who show that under direct parameterization, a Frank-Wolfe algorithm can obtain convergence guarantees with an additional factor in the convergence rate.

Our regret analysis in terms of undiscounted total cost RMDPs is novel, to our knowledge, with only one recent work appearing in the robust MDP literature. This work (Su et al., 2025) explored entropic risk and entropic value-at-risk measures in this context, and concludes that the total reward framework may be preferable to discounted MDPs in a broad range of domains. Our work connects a continuous-time formalism with the discrete-time total cost RMDP. The framework of average-reward MDPs is explored more widely in the robust MDP literature (Sun, 2024; Wang et al., 2025) but is conceptually distinct in that it emphasises long-term asymptotic behaviour rather than transient behaviour. Nevertheless, to demonstrate the competitiveness of our algorithm, a linear oracle-based convergence was found for rectangular RMDPs in Sun (2024) and an $\mathcal{O}(\frac{1}{\epsilon^4})$ oracle-based iteration complexity was found for non-rectangular RMDPs in Wang et al. (2025).

Our second proposed optimiser is a distributional optimiser, which optimises a problem an objective formulated over a distribution of parameters. Specifically, we directly use the analysis of the Mean-field Langevin Stochastic Descent-Ascent (MFL-SDA) algorithm from the mean-field optimisation literature (Liu et al., 2025). In the context of GAN training, previous work has highlighted distributional optimisation as a favourable global optimiser with theoretical guarantees (Hsieh et al., 2019). Our work extends the approach to RL objectives, where distributional optimisation is not commonly explored; note that that this is distinct from distributional RL (e.g. Bellemare et al. (2017)), where the return distribution is modeled to improve value estimates, distributed RL (e.g. Espeholt et al. (2018)), where distinct actors are trained in parallel for accelerated training, or uncertain MDPs (e.g. Ahmed & Jaillet (2013)), where the objective is to optimise the policy over a fixed distribution over transition kernels. Although here are a few prior works which update the adversary based on single-particle Langevin dynamics in the context of non-rectangular uncertainty sets (Wang et al., 2025; Li et al., 2026b), to the best of our knowledge, the use of distributional optimisation, let alone mean-field optimisation, has not yet been explored in the context of RMDPs. In our theory, the primary benefit we obtain from the MFL-SDA optimiser over the double-loop implementation is that we obtain a guarantee for the *last-iterate* regret rather than the average regret.

While the extension of RMDPs to the continuous-time setting has not been considered, works on robust control in continuous time have proposed solutions related to ours. For instance, Linear Quadratic Regulator (LQR) problems are frequently studied by control theorists, often in a more specialized set of control strategies and algorithms (Vinodh Kumar & Jerome, 2013; Kalman, 1960). Similarly, recent work has also explored worst-case robustness in Linear Quadratic Gaussian problems (Taskesen et al., 2023). By contrast, our approach tackles a more general problem structure including neural network parameterized dynamics, of which LQR type problems are one instance (as shown in the experiments). Within the context of a robotics application, previous work has considered uncertainty-aware parameterized SDEs to fit observed dynamics, which allowed a soft-robust RL using traditional PPO on SDE samples (Djeumou et al., 2023). Our paper investigates continuous-time policy gradient algorithms and considers hard-robustness, obtaining the worst-case environment from the uncertainty set. Previous work has also considered risk-averse control with the

maximum principle but this has focused on alternative risk measures and outside the context of policy gradient algorithms (Bonalli & Bonnet, 2023).

3 Preliminaries

Our goal is to use policy gradient in the context of RMDPs with continuous-time dynamics. We briefly describe the setting, the main objective, and the key concepts and general assumptions underlying our algorithmic solutions and proof strategies.

3.1 General notations

A few general notations we use throughout the paper are: $\mathbb{P}(x)$ to denote the probability of an event x; $\mathbb{P}[\mathcal{Y}]$ to denote the space of probability distributions over a set $\mathcal{Y}$; ∂_y to denote the partial derivative w.r.t. a uni-dimensional variable y and ∇_y to denote the gradient w.r.t. a multi-dimensional variable y; and $\mathbf{E}[x]$ to denote the expectation of a random variable x. Further notations are specific to the RL setting explained below or are introduced in the rest of the main body.

3.2 Continuous-time setting

The setting includes a continuous state space $\mathcal{X} \subset \mathbb{R}^{d_x}$ and a continuous control space $\mathcal{U} \subset \mathbb{R}^{d_u}$ where the dynamics evolve according to a controlled differential equation of the form

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = f_\xi(t, x_t, u_t)\,, \tag{1}$$

where $\xi \in \mathbb{R}^{d_\xi}$ parametrizes the dynamics, which will be considered as the inner problem of the robust optimisation.

The agent at time $t \geq 0$ observes the dynamics as x_t and takes actions as the control $u_t \sim \pi_\theta(t, x_t)$, where π_θ is a stochastic policy in $\Pi \coloneqq \{\pi_\theta : [0, T] \times \mathbb{R}^d \to \mathbb{P}[\mathcal{U}]\}$. When the policy is deterministic, we denote the policy as $\mu_\theta(t, x)$ for a state $x \in \mathcal{X}$. Given a particular dynamics model f_ξ, the agent seeks to optimise the expected cumulative cost of the trajectory according to

$$\min_\theta J(\theta, \xi) \coloneqq \mathbf{E}_{\pi_\theta}\left[\int_0^T r(\tau, x_\tau, u_\tau)\,\mathrm{d}\tau + R(x_T)\middle| d_0, f_\xi\right]\,, \tag{2}$$

where d_0 is the starting distribution, $r : [0, T] \times \mathcal{X} \times \mathcal{U} \to \mathbb{R}^+$ is the running cost, and $R : \mathcal{X} \to \mathbb{R}^+$ is the terminal cost. For simplicity of exposition, we further assume that the cost function is non-negative.

To facilitate the computation of the policy gradient and the expected cumulative cost, we make use of two related value functions. First, the action-value, which is the cost-to-go from a particular time and state-action pair, is formulated for continuous-time algorithms as

$$Q(t, x, u) = \mathbf{E}_{\pi_\theta}\left[\int_t^T r(\tau, x_\tau, u_\tau)\,\mathrm{d}\tau + R(x_T)\middle| x_t = x, u_t = u, f_\xi\right]\,. \tag{3}$$

Similarly, the value can be formulated for a particular state and time as

$$V(t, x) = \min_u \mathbf{E}\left[Q(t, x, u)\middle| f_\xi, \pi_\theta\right]\,. \tag{4}$$

3.3 Continuous-time RMDP formalism

Designing a formalism for continuous-time RMDP (CT-RMDPs), we are interested in obtaining the worst-case transition dynamics chosen by nature. The general objective matches that of the traditional RMDP,

$$\min_\theta \left\{\Phi(\theta) \coloneqq \sup_{\xi \in \Xi} J(\theta, \xi)\right\}\,, \tag{5}$$

where obtaining the worst-case dynamics is also referred to as the "inner problem" of the RMDP.

In the CT-RMDP setting, we consider underlying parametrised dynamics of the type (1) with

$$f_\xi(t, x, u) = \bar{f}(t, x, u) + g_\xi(t, x, u)\,, \tag{6}$$

where $\bar{f}$ is the nominal dynamics, g_ξ is a Lipschitz-continuous function, and the parameters ξ are restricted in the compact space Ξ.

3.4 Maximum principle

The maximum principle (MP) is a general framework using the calculus of variations that presents optimality conditions of a continuous, controlled trajectory subject to a cost functional J (Pontryagin et al., 1962). The MP is employed for systems with an open-loop control (i.e. $u(t)$ rather than $u(t, x_t)$). Alternatively, the Hamilton-Jacobi-Bellman partial differential equation (HJB PDE) provides optimality conditions for a *closed-loop* policy/control using a dynamic programming formalism (Bellman, 1957). These two approaches are related, as we will see in the following.

Consider the controlled ODE dynamics (1), subject to the cost functional (2). We construct the corresponding Hamiltonian:

$$H(t, x_t, u_t, p_t) \coloneqq r(t, x_t, u_t) + p_t^\mathsf{T} f_\xi(t, x_t, u_t) \tag{7}$$

where $p_t \in \mathbb{R}^{d_x}$ is the time-varying costate/adjoint term. In essence, the Hamiltonian is a Lagrangian that considers the ODE dynamics as a constraint while the p_t are interpreted as the Lagrange multipliers. The MP states that the optimal state x_t^*, optimal control u_t^*, and corresponding costate p_t^* must minimize the Hamiltonian: $H(t, x_t^*, u_t^*, p_t^*) \leq H(t, x_t, u_t, p_t)$ for all t, x_t, u_t, p_t. The trajectory of the costate p_t is determined by the partial derivative of the Hamiltonian w.r.t. the state variables and is solved backwards in time:

$$-\dot{p}_t = \partial_x H(t, x_t, u_t, p_t), \quad p_T = \nabla_x R(x_T). \tag{8}$$

Note that the state evolution is prescribed by

$$\dot{x}_t = \partial_p H(t, x_t, u_t, p_t), \quad x_0 = x \sim d_0 \tag{9}$$

which reduces to (1). On the other hand, if we have a value function $V(t, x_t)$ that denotes the minimum cost-to-go from time t, the HJB equation describes its evolution:

$$-\partial_t V(t, x_t) = \min_u \{r(t, x_t, u_t) + \nabla_x V(t, x_t)^\mathsf{T} f_\xi(t, x_t, u_t)\}. \tag{10}$$

The costate p_t is physically and mathematically identified with the spatial gradient of the value function: $p_t = \nabla_x V(t, x_t)$. Additionally, in the RL setting, it is known that the Q-function (4) is closely related to the Hamiltonian (7), in that the optimal Hamiltonian is equivalent to the optimal Q function obtained from Q-learning (Mehta & Meyn, 2009). The MP can be used to learn global feedback policies (Ainsworth et al., 2021), as we will see.

3.5 General assumptions

The following general assumptions will be made throughout our analyses. These are standard and are physically realisable through bounded features/weights (when applicable). We assume:

1. We require $f_\xi(t, x, u)$, $r(t, x, u)$, and $R(x)$ to be smooth in their arguments in order to apply the pathwise derivation for the pathwise gradient formula below.
2. The functions f_ξ, r, R, g_ξ are Lipschitz (in all their arguments) on a compact (invariant) set.
3. The derivatives are bounded $\|r_u\| \leq C_{ru},\ \|r_x\| \leq C_{rx},\ \ \|R_x\| \leq C_{Rx}$.
4. $\|\nabla_u f\| \leq C_{fu}$,$\|f_x\| \leq C_{fx}$.

5. $\left\|\nabla_x(g_\xi(t,x,u))\right\| \leq C_{g\xi}$.
6. $\left\|\nabla_\theta \pi_\theta(t,x_t)\right\| \leq C_{\pi\theta}$ for stochastic policy π_θ and $\left\|\nabla_\theta \mu_\theta(t,x_t)\right\| \leq C_{\mu\theta}$ for deterministic policy μ_θ
7. $\nabla_\theta \pi_\theta$ is $L_{\pi\theta}$-Lipschitz in θ; analogously, $\nabla_\theta \mu_\theta$ is $L_{\mu\theta}$-Lipschitz in θ.
8. π_θ is L_π-,$L_{\pi x}$- and $L_{\pi t}$-Lipschitz in θ, x, and t; analogously, μ_θ is L_μ-, $L_{\mu x}$-, and $L_{\mu t}$-Lipschitz continuous. It follows that, for instance, $\left\|\pi_{\theta'}(t,x') - \pi_\theta(t,x)\right\| \leq L_{\pi x}\left\|x' - x\right\| + L_\pi\left\|\theta' - \theta\right\|$. Additionally, we assume the policies $\nabla_x \pi_\theta(t,x)$ and $\nabla_x \mu_\theta(t,x)$ are uniformly bounded.
9. Likewise, the Jacobians of r and f are Lipschitz in x and in u.

These smoothness, Lipschitz and boundedness assumptions are fairly standard (and not restrictive) in order to obtain guarantees for smoothness of the policy gradient (see Lemma 15), and understand the dependencies of the regret on various facets of the system. Moreover, they are required to guarantee the existence and uniqueness of the solutions of the ODE system (from the Picard-Lindelöf theorem).

4 Policy gradient for the CT-RMDP problem

To solve the robust optimisation problem (5), we alternatingly optimise θ and ξ in double- or single-loop manner based on policy gradient. At each iteration $k \in [K]$, the policy optimiser performs gradient descent based on $\nabla_\theta J(\theta,\xi)$ with fixed inner problem parameters ξ, while the adversary optimiser performs gradient *ascent*, where gradients are based on $\nabla_\xi J(\theta,\xi)$ with fixed policy parameters θ. As a crucial part of our algorithm, we perform a forward-backward algorithm based on the Maximum Principle (Li et al., 2018; Ainsworth et al., 2021; Jin et al., 2020).

4.1 Pathwise policy gradient

A first policy gradient we derive extends the pathwise gradients of (Munos, 2006) for ODEs. While Munos (2006) considers the estimator for a single, terminal cost, we extend the reasoning here to the total cost (2) and we assume the availability of $\nabla_x f_\xi(t,x_t,u_t)$ rather than approximating it, which is not a strong assumption in the robust MDP case where the model f_ξ is typically available.

Let $(z_t := \nabla_\theta x_t)_{t\geq 0}$ be the process that describes how the state changes with respect to the policy parameters, with $z_t \in \mathbb{R}^{d_x \times d_\theta}$. Using the policy sensitivity process given by

$$\frac{\mathrm{d}z_t}{\mathrm{d}t} = \nabla_\theta f_\xi(t,x_t,u_t) = \nabla_u f_\xi(t,x_t,u_t) \cdot \big(\nabla_x \mu_\theta(t,x_t) z_t + \nabla_\theta \mu_\theta(t,x_t)\big) + \nabla_x f_\xi(t,x_t,u_t) z_t\,, \tag{11}$$

a pathwise policy gradient is derived below.

Theorem 1 (Pathwise policy gradient) *Let $\mu_\theta : [0,T] \times \mathcal{X} \to \mathcal{U}$ be a deterministic policy parametrised by $\theta \in \Theta$ and let $f_\xi : [0,T] \times \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ be the ODE dynamics parametrised by $\xi \in \Xi$ according to* (1)*. The pathwise policy gradient for μ_θ is given by*

$$\nabla_\theta J(\theta,\xi) = \mathbf{E}\left[\int_0^T \big(\nabla_u r(t,x_t,u_t) \cdot (\nabla_x \mu_\theta(t,x_t) \cdot z_t + \nabla_\theta \mu_\theta(t,x_t)) + \nabla_x r(t,x_t,u_t) \cdot z_t\big)\,\mathrm{d}t + \nabla_x R(x_T) \cdot z_T \middle| d_0, \mu_\theta, f_\xi\right]\,, \tag{12}$$

where z_t is the policy sensitivity process (11)*.*

Proof: For an instantaneous cost at the terminal state, the gradient is given by $\nabla_x R(x_T) \cdot z_T$. Additionally differentiating the cost in terms of the total derivative and integrating over the time interval $[0,T)$, we obtain

$$\nabla_\theta J(\theta,\xi) = \mathbf{E}\left[\int_0^T \big(\nabla_u r(t,x_t,u_t) \cdot (\nabla_x \mu_\theta(t,x_t) \cdot z_t + \nabla_\theta \mu_\theta(t,x_t)) + \nabla_x r(t,x_t,u_t) \cdot z_t\big)\,\mathrm{d}t + \nabla_x R(x_T) \cdot z_T\right].$$

□

For the adversary ξ, we similarly define the adversary sensitivity process $(y_t \coloneqq \nabla_\xi x_t)_{t\geq 0}$, where $y_t \in \mathbb{R}^{d_x \times d_\xi}$:

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = \nabla_x f_\xi \cdot y_t + \nabla_\xi f_\xi(t, x_t, u_t) + \nabla_u f_\xi(t, x_t, u_t) \cdot \nabla_x \mu_\theta(t, x_t) \cdot y_t. \tag{13}$$

This leads to the pathwise adversarial policy gradient below.

Theorem 2 (Pathwise adversarial policy gradient) *Let $\mu_\theta : [0, T] \times \mathcal{X} \to \mathcal{U}$ be a deterministic policy parametrised by $\theta \in \Theta$ and let $f_\xi : [0, T] \times \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ be the ODE dynamics parametrised by $\xi \in \Xi$ according to* (1)*.*

$$\nabla_\xi J(\theta, \xi) = \mathbf{E}\left[\int_0^T (\nabla_x r(t, x_t, u_t) \cdot y_t + \nabla_u r(t, x_t, u_t) \nabla_x \mu_\theta(t, x_t) \cdot y_t \,\mathrm{d}t + \nabla_x R(x_T) \cdot y_T \middle| d_0, \mu_\theta, f_\xi\right], \tag{14}$$

where $(y_t \coloneqq \nabla_\xi x_t)_{t\geq 0}$ is the adversary sensitivity process (13)*.*

Proof: The gradient is derived analogously to Theorem 1 but now with respect to ξ and with the adversary sensitivity process. □

To compute the gradients $\nabla_u r(t, x_t, u_t)$ and $\nabla_x r(t, x_t, u_t)$, the pathwise gradients above require access to the reward model $r(\cdot)$, and similarly $R(\cdot)$. They therefore come under the classification of model-based algorithms. The below adjoint-gradient methods do not require such model access.

4.2 Adjoint-based policy gradient

In this section, we derive policy gradients based on the maximum principle, which can learn global feedback policies, using a forward-backward approach. For ODE dynamics, we follow Li et al. (2018), which uses the method of successive approximations (MSA). The algorithm assumes deterministic system and parametrisation. Similar to Section 4.2 of Li et al. (2018), a time discretisation based on the Euler method is applied and the maximization is based on a single step of gradient descent. The algorithm is summarised in Algorithm 1. We fix the parameters of the other player during this iterative process.

Algorithm 1 The Discrete-time MSA algorithm.

1: **Input:** discretisation Δt.
2: **for** $j = 1, \ldots, N_{\text{it}}$ **do**
3: **Forward:** obtain the state process $(x_t)_{t=0,\Delta t,\ldots,T}$ starting from $x_0 = x$
4: **Backward:** obtain the costate process $(p_t)_{t=T,T-\Delta t,\ldots,0}$ starting from $p_T = -\nabla R_x(x_T)$
5: **Maximize Hamiltonian:** $\theta_j = \arg\max_{\theta\in\Theta} H(t)$ for all $t = 0, \Delta t, \cdots, T$
6: **end for**

In essence, we first compute forward rollouts and the adjoint (backward) rollouts (alternating between holding the policy parameters fixed and the adversary parameters fixed). We use those values to compute the following policy gradients below.

Remark 1 *The (robust) HJB equation of this finite-horizon problem is given by*

$$\partial_t V(x_t, t) + \inf_u \sup_{\xi_t} \left[r(t, x_t, u) + \nabla_x V(t, x_t)^\top f_\xi(t, x_t, u_t)\right] = 0, \quad V(T, x_T) = R(x_T)\,. \tag{15}$$

Solving this PDE is an alternate way to obtain the value function and the optimum value, but it is prohibitive in high dimensions and we merely present it for completeness.

To find an approximate solution to this problem, we implement gradient descent-ascent schemes where one of both parameters is updated based on the obtained trajectories. The Hamiltonian is then computed in backward manner from $p_T = V(T, x_T)$ to $p_0 = V(0, x_0)$.

The policy gradient We present our formulations for the robust policy and adversary gradients of the objective (2) in the cases of stochastic and deterministic policies.

First, we derive the adjoint policy gradient in the case of a stochastic policy. The *sample-and-hold* model describes an action $u_{t_n} \sim \pi_\theta(t_n, x_{t_n})$ being sampled and held for the time interval $[t_n, t_{n+1})$ (or $[t_n, t_{n-1})$ in the backwards case).

The following is an adjoint deterministic policy gradient formula.

Theorem 3 (Adjoint-based deterministic policy gradient) *Let $\mu_\theta : [0,T] \times \mathcal{X} \to \mathcal{U}$ be a deterministic policy parametrised by $\theta \in \Theta$ and let $f_\xi : [0,T] \times \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ be the ODE dynamics parametrised by $\xi \in \Xi$ according to* (1)*. The adjoint-based deterministic policy gradient is given by*

$$\nabla_\theta J(\theta, \xi) = \mathbf{E}\left[\int_0^T \left(\nabla_\theta \mu_\theta(t, x_t) + \nabla_x \mu_\theta(t, x_t) z_t\right)^\mathsf{T} \nabla_u H(t, x_t, u_t, p_t)\,\mathrm{d}t \middle| d_0, \mu_\theta, f_\xi\right], \tag{16}$$

where $H(t, x_t, u_t, p; \xi) = r(t, x_t, u_t) + p^\mathsf{T} f_\xi(t, x_t, u_t)$ and

$$-\dot{p}_t = \nabla_x r(t, x_t, u_t) + \nabla_x f_\xi(t, x_t, u_t)^\mathsf{T} p_t, \quad p_T = \nabla_x R(x_T), \;\; u_t = \mu_\theta(t, x_t).$$

Under the sample-and-hold model, as $\Delta t \to 0$, this converges to the formula for the discrete-time deterministic policy gradient (Silver et al., 2014), i.e.

$$\nabla_\theta J_h(\theta, \xi) = \mathbf{E}\left[\sum_{n=0}^{N-1} \nabla_\theta \mu_\theta(t, x_t)^\mathsf{T} \nabla_u Q_h(t_n, x_{t_n}, u_n)\right],$$

where Q_h is the cost-to-go from the sample-and-hold system.

Proof: We first analyse the first variation of x_t, u_t, and J w.r.t. θ and then we eliminate δx using an adjoint.

Variation w.r.t. θ : Take a perturbation $\theta^\epsilon = \theta + \epsilon\delta\theta$, which induces perturbed trajectories x_t^ϵ and controls $u_t^\epsilon = \mu_{\theta^\epsilon}(t, x_t^\epsilon)$. Define

$$\delta x_t \coloneqq \frac{\mathrm{d}}{\mathrm{d}\epsilon} x_t^\epsilon \bigg|_{\epsilon=0}, \quad \delta u_t \coloneqq \frac{\mathrm{d}}{\mathrm{d}\epsilon} u_t^\epsilon \bigg|_{\epsilon=0}$$

Differentiate the feedback control (linearisation):

$$u_t^\epsilon = \mu_{\theta^\epsilon}(t, x_t^\epsilon), \quad \delta u_t = \nabla_\theta \mu_\theta(t, x_t)\delta\theta + \nabla_x \mu(t, x_t)\delta x_t.$$

Likewise, we linearise the dynamics to get $\delta\dot{x}_t = \nabla_x f_\xi(t, x_t, u_t)\delta x_t + \nabla_u f_\xi(t, x_t, u_t)\delta u_t$. Differentiating the cost J yields

$$\delta J \coloneqq \frac{\mathrm{d}}{\mathrm{d}\epsilon} J(\theta^\epsilon, \xi)\bigg|_{\epsilon=0} = \mathbf{E}\left[\int_0^T \left(\nabla_x r(t, x_t, u_t)\delta x_t + \nabla_u r(t, x_t, u_t)\delta u_t\right)\mathrm{d}t + \nabla_x R(x_T)\delta x_T\right]$$

Costate to eliminate δx: Define the costate as $p_t \in \mathbb{R}^{d_x}$ by the following adjoint ODE:

$$-\dot{p}_t = \nabla_x r(t, x_t, u_t) + \nabla_x f_\xi(t, x_t, u_t)^\mathsf{T} p_t, \quad p_T = \nabla_x R(x_T)$$

Using $\delta x_0 = 0$ (the initial state does not depend on θ),

$$p_T^\mathsf{T} \delta x_T = \int_0^T \left(\dot{p}_t \delta x_t + p_t^\mathsf{T} \delta\dot{x}_t\right)\mathrm{d}t$$

Substituting for $\dot{p}_t$ and $\delta\dot{x}_t$, we obtain

$$p_T^\mathsf{T}\delta x_T = \int_0^T \left(\left(-\nabla_x r(t,x_t,u_t) - \nabla_x f_\xi(t,x_t,u_t)^\mathsf{T} p_t\right)^\mathsf{T} \delta x_t + p_t^\mathsf{T}\left(\nabla_x f_\xi(t,x_t,u_t)\delta x_t + \nabla_u f_\xi(t,x_t,u_t)\delta u_t\right)\right) \mathrm{d}t$$

which simplifies to

$$p_T^\mathsf{T}\delta x_T + \int_0^T \nabla_x r(t,x_t,u_t)^\mathsf{T}\delta x_t \,\mathrm{d}t = \int_0^T p_t^\mathsf{T}\nabla_u f_\xi(t,x_t,u_t)\delta u_t \,\mathrm{d}t$$

This leads to

$$\delta J = \mathbf{E}\left[\int_0^t \left(\underbrace{\nabla_u r(t,x_t,u_t) + \nabla_u f_\xi(t,x_t,u_t)^\mathsf{T} p_t}_{=:\nabla_u H(t,x_t,u_t,p_t;\xi)}\right)^\mathsf{T} \delta u_t \,\mathrm{d}t\right]$$

Now, we substitute in the linearisation of δu_t to obtain

$$\delta J = \mathbf{E}\left[\int_0^T \left(\left(\nabla_\theta \mu_\theta(t,x_t) + \nabla_x \mu(t,x_t) z_t\right)^\mathsf{T} \nabla_u H(t,x_t,u_t,p_t;\xi)\right) \mathrm{d}t\right] \cdot \delta\theta$$

noting that $\delta x_t = z_t \delta\theta$. Since this holds for all directions $\delta\theta$, the policy gradient is

$$\nabla_\theta J(\theta,\xi) = \mathbf{E}\left[\int_0^T \left(\nabla_\theta \mu_\theta(t,x_t) + \nabla_x \mu(t,x_t) z_t\right)^\mathsf{T} \nabla_u H(t,x_t,u_t,p_t;\xi) \,\mathrm{d}t\right]$$

Equivalence to discrete-time deterministic policy gradient Under the sample-and-hold model, the gradient w.r.t. x drops from the policy, and with $h \equiv \Delta t \to 0$, we have

$$\begin{aligned}
\nabla_\theta J(\theta,\xi) &= \mathbf{E}\left[\int_0^T \left(\nabla_\theta \mu_\theta(t,x_t)\right)^\mathsf{T} \nabla_u H(t,x_t,u_t,p_t;\xi) \,\mathrm{d}t\right] \\
&= \mathbf{E}\left[\int_0^T \left(\nabla_\theta \mu_\theta(t,x_t)\right)^\mathsf{T} \nabla_u \left(r(t,x_t,u_t) + f_\xi(t,x_t,u_t)^\mathsf{T} p_t + \partial_t V(t,x_t)\right) \mathrm{d}t\right] \\
&= \mathbf{E}\left[\int_0^T \left(\nabla_\theta \mu_\theta(t,x_t)\right)^\mathsf{T} \nabla_u Q(t,x_t,u_t) \,\mathrm{d}t\right] \\
&\approx \mathbf{E}\left[\sum_{n=0}^{N-1} h \left(\nabla_\theta \mu_\theta(t_n,x_{t_n})\right)^\mathsf{T} \nabla_u Q(t_n,x_{t_n},u_{t_n})\right] \\
&= \mathbf{E}\left[\sum_{n=0}^{N-1} \left(\nabla_\theta \mu_\theta(t_n,x_{t_n})\right)^\mathsf{T} \nabla_u Q_h(t_n,x_{t_n},u_{t_n})\right].
\end{aligned}$$

□

Under the deterministic policy μ_θ, we obtain the following adversarial gradient.

Theorem 4 (Adjoint-based deterministic adversarial policy gradient) *Let $f_\xi : [0,T] \times \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ be the ODE dynamics parametrised by $\xi \in \Xi$ according to* (1)*. Then the adjoint-based adversarial policy gradient is*

$$\nabla_\xi J(\theta,\xi) = \mathbf{E}\left[\int_0^T p_t^\mathsf{T} \nabla_\xi f_\xi(t,x_t,u_t) \,\mathrm{d}t \middle| d_0, f_\xi\right] \tag{17}$$

where the action u_t is based on a stochastic or deterministic policy, and $p_t = \nabla_x V(t,x_t)$ is the costate.

Proof: We demonstrate that this expression arises from a variational argument, similar to the one above for the deterministic policy gradient. First, we fix a sample path of u_t and x_0 (we take expectations at the end).

Linearisation of the state w.r.t. ξ Let $\xi^\epsilon = \xi + \epsilon\delta\xi$, and let x_t^ϵ be the corresponding state. Define the variation

$$\delta x_t \coloneqq \frac{\mathrm{d}}{\mathrm{d}\epsilon} x_t^\epsilon \bigg|_{\epsilon=0}.$$

Differentiate the ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta x_t = \nabla_x f_\xi(t, x_t, u_t)\delta x_t + \nabla_\xi f_\xi(t, x_t, u_t)\delta\xi, \quad \delta x_0 = 0.$$

Differentiating the cost J along ξ^ϵ,

$$\delta J \coloneqq \frac{\mathrm{d}}{\mathrm{d}\epsilon}\bigg|_{\epsilon=0} \left(\int_0^T r(t, x_t^\epsilon, u_t)\,\mathrm{d}t + R(x_T^\epsilon) \right) = \int_0^T \nabla_x r(t, x_t, u_t)^\mathsf{T} \delta x_t\,\mathrm{d}t + \nabla_x R(x_T)\delta x_T.$$

Introducing the adjoint to eliminate δx : Using the same adjoint state p_t defined by

$$-\dot{p}_t = \nabla_x r(t, x_t, u_t) + \nabla_x f_\xi(t, x_t, u_t)^\mathsf{T} p_t, \quad p_T = \nabla_x R(x_T)$$

and the same steps as in the deterministic case (when computing the variation w.r.t. θ), we obtain

$$\int_0^T \nabla_x r(t, x_t, u_t)\delta x_t\,\mathrm{d}t + p_T^\mathsf{T}\delta x_T = \int_0^T p_t^\mathsf{T} \nabla_\xi f_\xi(t, x_t, u_t)\delta\xi\,\mathrm{d}t.$$

Substitute into δJ Since $p_T = \nabla_x R(x_T)$, we can replace terms in δJ to obtain

$$\delta J = \int_0^T p_t^\mathsf{T} \nabla_\xi f_\xi(t, x_t, u_t)\delta\xi\,\mathrm{d}t.$$

As this holds for all directions $\delta\xi$, we conclude:

$$\nabla_\xi J(\theta, \xi) = \int_0^T p_t^\mathsf{T} \nabla_\xi f_\xi(t, x_t, u_t)\,\mathrm{d}t.$$

Finally, taking expectation(s) over the randomness (w.r.t. the initial state distribution d_0 and the policy) gives the final form. Once we have $\nabla_\xi J(\theta, \xi)$, performing the projected ascent step is standard. □

Theorem 5 (Stochastic adjoint-based gradient) *Let $\pi_\theta : [0, T] \times \mathcal{X} \to \mathbb{P}[\mathcal{U}]$ be a stochastic policy parametrised by $\theta \in \Theta$ and let $f_\xi : [0, T] \times \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ be the ODE dynamics parametrised by $\xi \in \Xi$ according to* (1)*. Then the adjoint-based policy gradient is given by*

$$\nabla_\theta J(\theta, \xi) = \mathbf{E}\Big[\int_0^T \big[\nabla_\theta \log(\pi_\theta(u_t|t, x_t)) + \nabla_x \log(\pi(u_t|t, x_t))z_t\big]^\mathsf{T} Q(t_n, x_{t_n}, u_n)\,\mathrm{d}t \Big| d_0, f_\xi \Big], \tag{18}$$

where $Q(t_n, x_{t_n}, u_n) = r(t_n, x_{t_n}, u_n) + \nabla_x V(n, x_{t_n})^\mathsf{T} f_\xi(n, x_{t_n}, u_n) + \partial_t V(t_n, x_{t_n})$. Under the sample-and-hold model, as $\Delta t \to 0$, this converges to the formula for the discrete-time policy gradient (Lemma 8; (Sutton & Barto, 2012)), i.e.

$$\nabla_\theta J_h(\theta, \xi) = \mathbf{E}\left[\sum_{n=0}^{N-1} \nabla_\theta \log \pi_\theta(u_n|t_n, x_{t_n}) Q_h(t_n, x_{t_n}, u_n) \right],$$

where Q_h is the cost-to-go from the sample-and-hold system.

Proof: For sample-and-hold actions $\{u_n\}$, a discrete policy gradient identity is given according to Lemma 8 by

$$\nabla_\theta J_h(\theta,\xi) = \mathbf{E}\left[\sum_{n=0}^{N-1} \nabla_\theta \log \pi_\theta(u_n|t_n,x_{t_n}) Q_h(t_n,x_{t_n},u_n)\right],$$

where $Q_h(t_n,x_{t_n},u_n) \coloneqq \mathbf{E}[G(\tau)|t_n,x_n,u_n]$ is the cost-to-go from the sample-and-hold system.

Using the above equation, and noting that $\nabla_x \pi(u_t|t,x_t)z_t$, one can take the limit $h \to 0$ and thereby $N \to \infty$ to obtain an equivalent form in relation to the continuous-time action-value $Q(t_n,x_{t_n},u_n) = r(t_n,x_{t_n},u_n) + \nabla_x V(n,x_{t_n})^\mathsf{T} f_\xi(n,x_{t_n},u_n) + \partial_t V(t_n,x_{t_n})$ according to

$$\nabla_\theta J(\theta,\xi) = \mathbf{E}\left[\sum_{n=0}^{N-1} \nabla_\theta \log \pi_\theta(u_n|t_n,x_{t_n}) Q(t_n,x_{t_n},u_n) h\right].$$

The term $\nabla_x \log(\pi(u_t|t,x_t))z_t$ follows from the chain rule based on the sensitivity of the state x_t with respect to the parameter θ. Analogous to Theorem 3, this term vanishes under the sample-and-hold model since the policy is constant. □

4.3 Extension to SDEs

Previously, we had considered an ODE-based robust policy optimisation with a static adversary $\xi \in \Xi$. In the following, we deal with the case where the underlying dynamics are represented by a stochastic differential equation (SDE) instead. More specifically, we consider the problem of robust continuous-time RL under *controlled diffusions*. The main considerations we make are using a forward-backward stochastic differential equation (FBSDE) via the stochastic maximum principle to compute gradients. To this end, we present a unified min-max gradient framework where both $\nabla_\theta J(\theta,\xi)$ and $\nabla_\xi J(\theta,\xi)$ are computed with forward simulation and a backward adjoint SDE, avoiding high-dimensional PDE solvers. To this end, we consider SDEs of the form

$$\mathrm{d}X_t = b_\xi(t,X_t,U_t) + \sigma_\xi(t,X_t,U_t)\,\mathrm{d}W_t\,, \tag{19}$$

where $X_t \in \mathbb{R}^{d_x}$ as before, the drift $b_\xi(t,X_t,U_t) \in \mathbb{R}^{d_x}$, and the diffusion $\sigma_\xi(t,X_t,U_t) \in \mathbb{R}^{d_x \times d_m}$ where d_m is the dimension of $\mathrm{d}W_t$. The corresponding stochastic HJB equation is

$$\partial_t V(t,X_t) + \min_u \{\mathcal{A}V(t,X_t) + r(t,X_t,U_t)\} = 0\,, \tag{20}$$

where $\mathcal{A}$ is the infinitesimal generator of the process (19), defined to be

$$\mathcal{A}V(t,X_t) \coloneqq \nabla_x V(t,X_t)^\mathsf{T} b_\xi(t,X_t,U_t) + \frac{1}{2}(\sigma_\xi(t,X_t,U_t)\sigma_\xi(t,X_t,U_t)^\mathsf{T}\nabla^2 V(t,X_t)). \tag{21}$$

We include the above for completeness, as it would be useful for (i) Verification: if $V(t,X_t)$ solves the HJB and $u^*(t) = \pi_{\theta^*}(t,X_t)$ minimizes the Hamiltonian, then it is optimal, and (ii) Duality viewpoints between the stochastic maximum principle and PDE theory. Solving the HJB is intractable beyond small d_x in general and as such, we will employ the stochastic maximum principle for implementable gradients.

4.3.1 Stochastic Maximum Principle

For the stochastic system (19) above, with the same cost function (2), we can construct a Hamiltonian and an adjoint backward SDE (BSDE) describing the costate process.

In this setting, the Hamiltonian is given by

$$H(t,X_t,U_t,P_t,Q_t;\xi) \coloneqq r(t,X_t,U_t) + P_T^\mathsf{T} b_\xi(t,X_t,U_t) + \mathrm{Tr}(Q_t^\mathsf{T}\sigma_\xi(t,X_t,U_t))\,, \tag{22}$$

where there are now two costates (P_t,Q_t) compared to just p_t in the deterministic dynamics setting. The costate P_t has the same interpretation as p_t – i.e. equivalent to $\nabla_x V(t,X_t) \in \mathbb{R}^{d_x}$ – while the additional costate

$$Q_t = \frac{1}{2}\nabla_x^2 V(t,X_t)\sigma_\xi(t,X_t) \in \mathbb{R}^{d_x \times d_m} \tag{23}$$

is interpreted as how the curvature of the value couples with the noise and its effect on the expected cost.[1]

As computing Q_t is generally intractable, we assume oracle access for the remainder of this work. This enables us to understand the evolution of the gradients. The adjoint BSDE is the following system:

$$\begin{aligned} \mathrm{d}P_t &= -\nabla_x H(t, X_t, U_t, P_t, Q_t; \xi)\,\mathrm{d}t + Q_t\,\mathrm{d}W_t \\ P_T &= \nabla_x R(X_T)\,. \end{aligned}$$

Note that $\nabla_x H(t, X_t, U_t, P_t, Q_t; \xi) = \nabla_x r(t, X_t, U_t) + P_t^\mathsf{T} \nabla_x b_\xi(t, X_t, U_t) + Q_t^\mathsf{T} \nabla_x \sigma_\xi(t, X_t, U_t)$, and similarly $\nabla_u H(t, X_t, U_t, P_t, Q_t; \xi) = \nabla_u r(t, X_t, U_t) + P_t^\mathsf{T} \nabla_u b_\xi(t, X_t, U_t) + Q_t^\mathsf{T} \nabla_u \sigma_\xi(t, X_t, U_t)$. These will be useful for the construction of the policy gradients below.

4.3.2 Policy gradients for the SDE case

Analogously to the deterministic setup, we also construct a (now stochastic) process for the sensitivity of the state X_t with respect to the policy parameter θ, denoted by $Z_t \coloneqq \nabla_\theta X_t$. Differentiating w.r.t. time to get the parameter sensitivity process, we obtain a linear SDE

$$\mathrm{d}Z_t = (\nabla_x b_\xi(t, X_t, U_t) \cdot Z_t + \nabla_u b_\xi(t, X_t, U_t) \cdot \nabla_\theta U_t)\,\mathrm{d}t + (\nabla_x \sigma_\xi(t, X_t, U_t) \cdot Z_t + \nabla_u \sigma_\xi(t, X_t, U_t) \cdot \nabla_\theta U_t)\,\mathrm{d}W_t.$$

where we note that $\nabla_\theta U_t \equiv \nabla_\theta \mu_\theta(t, X_t) + \nabla_x \mu_\theta(t, X_t) Z_t$.

We have via Ito's lemma that

$$\mathrm{d}V(t, X_t) = \left(\partial_t V(t, X_t) + (\nabla_x V(t, X_t))^\mathsf{T} b_\xi(t, X_t) + \frac{1}{2}\nabla_x^2 V(t, X_t) D(t, X_t)\right)\mathrm{d}t + \nabla_x V(t, X_t)\sigma_\xi(t, X_t)\,\mathrm{d}W_t\,,$$

where $b_\xi(t, X_t) = b_\xi(t, X_t, \mu_\theta(t, X_t))$ and $D(t, X_t) = \sigma_\xi(t, X_t)^\mathsf{T}\sigma_\xi(t, X_t)$ for $\sigma_\xi(t, X_t) = \sigma_\xi(t, X_t, \pi_\theta(t, X_t))$. Taking the expectation, the diffusion term cancels since $\mathbf{E}[\mathrm{d}W_t] = \mathbf{0}$. This can then be used to derive a policy gradient for the SDE.

Theorem 6 (Deterministic policy gradient for SDE dynamics) *Let the transition dynamics be the SDE* (19) *and fix $\xi \in \Xi$. Then the deterministic policy gradient is given by*

$$\begin{aligned} \nabla_\theta J(\theta, \xi) &= \mathbf{E}\Big[\int_0^T \left[\nabla_\theta \mu_\theta(t, X_t) + \nabla_x \mu_\theta(t, X_t) Z_t\right]^\mathsf{T} \nabla_u \left(r(t, X_t, U_t) + \mathcal{A}V(t, X_t)\right)\mathrm{d}t \Big| d_0, f_\xi\Big] \\ &= \mathbf{E}\Big[\int_0^T \left[\nabla_\theta \mu_\theta(t, X_t) + \nabla_x \mu_\theta(t, X_t) Z_t\right]^\mathsf{T} \\ &\qquad \nabla_u \left(r(t, X_t, U_t) + P_t^\mathsf{T} b_\xi(t, X_t) + Q_t^\mathsf{T} \sigma_\xi(t, X_t)\right)\mathrm{d}t \Big| d_0, f_\xi\Big]\,. \end{aligned} \tag{24}$$

Proof: This proof is exactly analogous to that of the ODE case, where the generator/Hamiltonian now has an extra term $Q_t \sigma_\xi(t, X_t)$ to account for the effect of the diffusion. □

Now we derive the adjoint-based adversary gradient, where ξ parametrizes the drift and the diffusion. If ξ only parametrizes the drift b_ξ, it is easy to drop the trace term. To ascend in ξ, one only needs a forward simulation of (X_t, U_t), then solve the adjoint BSDE for (P_t, Q_t), and subsequently integrate the expression below.

Theorem 7 (Adjoint adversary gradient for SDE) *If ξ parametrizes both the drift and diffusion as in Equation* (19) *and fixing θ, the adjoint adversary gradient is given by*

$$\begin{aligned} \nabla_\xi J(\theta, \xi) &= \mathbf{E}_{u_t \sim \pi(t, x_t)}\Big[\int_0^T \left[\nabla_\xi f_\xi(t, x_t, u_t)\right]^\mathsf{T} \\ &\qquad \left(r(t, x_t, u_t) + (\nabla_x V(t, x_t))^\mathsf{T} b_\xi(t, x_t) + \frac{1}{2}\nabla_x^2 V(t, x_t) D(t, x_t)\right)\mathrm{d}t \Big| d_0, f_\xi\Big] \\ &= \mathbf{E}\left[\int_0^T P_t^\mathsf{T} \nabla_\xi b_\xi(t, X_t, U_t) + \mathrm{Tr}(Q_t^\mathsf{T} \nabla_\xi \sigma_\xi(t, X_t, U_t))\,\mathrm{d}t\right]\,, \end{aligned} \tag{25}$$

[1]This second costate Q_t is not equivalent to the discrete-time Q_h-function in earlier sections.

where $\xi \in \mathbb{R}^{d_\xi}$, $\nabla_\xi \sigma_\xi \in \mathbb{R}^{d_x \times d_m \times d_\xi}$.

Proof: The proof for the above adversary gradient in the SDE dynamics setting follows similarly to that of the adversary gradient in the ODE case, with one notable change: the use of Itô's lemma to compute differentials. Additionally, for this proof, we abuse notation to denote $b_\xi(t, X_t)$ and $\sigma_\xi(t, X_t)$ for the closed-loop coefficients i.e. $b_\xi(t, X_t, \pi_\theta(t, X_t))$ and $\sigma_\xi(t, X_t, \pi_\theta(t, X_t))$, respectively. We fix θ and differentiate with respect to ξ.

Perturb the adversary parameter : Let $\xi^\epsilon = \xi + \epsilon\delta\xi$ where $\delta\xi \in \mathbb{R}^{d_\xi}$ is an arbitrary direction. Let X_t^ϵ be the trajectory obtain from using ξ^ϵ, i.e. X_t^ϵ solves

$$\mathrm{d}X_t^\epsilon = b_{\xi^\epsilon}(t, X_t^\epsilon)\,\mathrm{d}t + \sigma_{\xi^\epsilon}(t, X_t^\epsilon), \quad X_0^\epsilon = X_0.$$

Define the first variation

$$Y_t \coloneqq \frac{\mathrm{d}}{\mathrm{d}\epsilon} X_t^\epsilon \bigg|_{\epsilon=0}.$$

Linearising the original SDE, we get that Y_t solves the linear variational SDE

$$\mathrm{d}Y_t = \big(\nabla_x b_\xi(t, X_t)Y_t + \nabla_\xi b_\xi(t, X_t)\delta\xi\big)\,\mathrm{d}t + \big(\nabla_x \sigma(t, X_t)Y_t + \nabla_\xi \sigma_\xi(t, X_t)\delta\xi\big)\,\mathrm{d}W_t, \quad Y_0 = 0.$$

Linearising the objective Set

$$J(\xi) = \mathbf{E}\left[\int_0^T r(t, X_t)\,\mathrm{d}t + R(X_T)\right].$$

Taking the directional derivative in $\delta\xi$.

$$DJ(\xi)[\delta\xi] = \frac{\mathrm{d}}{\mathrm{d}\epsilon} J(\xi^\epsilon)\bigg|_{\epsilon=0} = \mathbf{E}\left[\int_0^T \nabla_x r(t, X_t)^\mathsf{T} Y_t\,\mathrm{d}t + \nabla_x R(X_T)^\mathsf{T} Y_T\right].$$

Similarly to the ODE case, now we introduce the adjoint BSDE to eliminate $\delta\xi$.

The adjoint BSDE Let the costates (P_t, Q_t) satisfy

$$\mathrm{d}P_t = -\left(\nabla_x r(t, X_t) + \nabla_x b_\xi(t, X_t)^\mathsf{T} P_t + \sum_{j=1}^m \underbrace{\nabla_x \sigma_\xi^{(j)}(t, X_t)}_{\in \mathbb{R}^{d\times d}} \underbrace{Q_t^{(j)}}_{\in \mathbb{R}^d}\right)\mathrm{d}t + Q_t\,\mathrm{d}W_t$$

where the superscript (j) denotes the j-th column.

Itô's rule for $P^\mathsf{T} Y_t$: We compute $\mathrm{d}(P_t^\mathsf{T} Y_t)$ using Itô's product rule: $\mathrm{d}(P_t^\mathsf{T} Y_t) = P_t^\mathsf{T}\,\mathrm{d}Y_t + Y_t^\mathsf{T}\,\mathrm{d}P_t + \mathrm{d}\langle P, Y\rangle_t$ where the last term is the quadratic covariation. We substitute in the SDE for $\mathrm{d}Y_t$ in the first term, and the SDE for $\mathrm{d}P_t$ in the second term.

The first term is

$$P_t^\mathsf{T}\,\mathrm{d}Y_t = P_t^\mathsf{T}(\nabla_x b_\xi(t, X_t)Y_t + \nabla_\xi b_\xi(t, X_t)\delta\xi)\,\mathrm{d}t + P_t^\mathsf{T}\left(\nabla_x \sigma_\xi(t, X_t)Y_t + \nabla_\xi \sigma_\xi(t, X_t)\delta\xi\right)\mathrm{d}W_t.$$

The second term is

$$Y_t^\mathsf{T}\,\mathrm{d}P_t = -Y_t^\mathsf{T}\left(\nabla_x r(t, X_t) + (\nabla_x b_\xi(t, X_t))^\mathsf{T} P_t + \sum_{j=1}^m \nabla_x \sigma_\xi^{(j)}(t, X_t)^\mathsf{T} Q_t^{(j)}\right)\mathrm{d}t + Y_t^\mathsf{T} Q_t\,\mathrm{d}W_t.$$

For the quadratic covariation term, only the diffusion parts contribute:

$$\mathrm{d}\langle P, Y\rangle_t = \sum_{j=1}^{m} Q_t^{(j)\mathsf{T}} \left(\nabla_x \sigma_\xi^{(j)}(t, X_t) Y_t + \nabla_\xi \sigma_\xi^{(j)}\right) \mathrm{d}t.$$

The remaining drift, after some terms cancel, is

$$\mathrm{d}(P_t^\mathsf{T} Y_t) = \left(-\nabla_x r(t, X_t)^\mathsf{T} Y_t + P_t^\mathsf{T} \nabla_\xi b_\xi(t, X_t)\delta\xi + \sum_{j=1}^{m} \nabla_x \sigma_\xi^{(j)}(t, X_t)^\mathsf{T} Q_t^{(j)}\right) \mathrm{d}t + \mathrm{d}M_t \tag{26}$$

where $\mathrm{d}M_t$ is a martingale.

Integrate and take expectations : Integrating Equation (26) from 0 to T,

$$P_T^\mathsf{T} Y_T - P_0^\mathsf{T} Y_0 = \int_0^T \left(-\nabla_x r(t, X_t)^\mathsf{T} Y_t + P_t^\mathsf{T} \nabla_\xi b_\xi(t, X_t)\delta\xi + \sum_{j=1}^{m} \nabla_x \sigma_\xi^{(j)}(t, X_t)^\mathsf{T} Q_t^{(j)}\right) \mathrm{d}t + M_T.$$

Since $Y_0 = 0$, and noting that $p_T = \nabla_x R(X_T)$ and $\mathbf{E}[M_T] = 0$, we take expectations to get

$$\mathbf{E}[\nabla_x R(X_T)^\mathsf{T} Y_T] = \mathbf{E}\int_0^T \left(-\nabla_x r(t, X_t)^\mathsf{T} Y_t + P_t^\mathsf{T} \nabla_\xi b_\xi(t, X_t)\delta\xi + \sum_{j=1}^{m} \nabla_x \sigma_\xi^{(j)}(t, X_t)^\mathsf{T} Q_t^{(j)}\right) \mathrm{d}t$$

Substituting this into the expression for $DJ(\xi)[\delta\xi]$ to get

$$DJ(\xi)[\delta\xi] = \mathbf{E}\int_0^T \left(P_t^\mathsf{T} \nabla_\xi b_\xi(t, X_t)\delta\xi + \sum_{j=1}^{m} \nabla_x \sigma_\xi^{(j)}(t, X_t)^\mathsf{T} Q_t^{(j)}\right) \mathrm{d}t$$

As this holds for every perturbation $\delta\xi$, we can conclude the adjoint formula above. □

4.4 Discretisation analysis

In this section, we are interested in deriving the Euler-based discretisation error of the policy gradient $\nabla_\theta J(\theta, \xi)$ with a time step size of $\Delta t \equiv h$.

Lemma 1 (One-step state discretisation error) *Let the exact trajectory be denoted by $\frac{d}{dt}\bar{x}(t) = f_\xi(t, \bar{x}(t), \bar{u}_n)$ for $t \in [t_n, t_{n+1})$, $\bar{x}(t_0) = x_0$ where $\bar{u}_n \sim \pi_\theta(\cdot|t_n, \bar{x}(t_n))$ for a general stochastic policy π_θ. Moreover, let f_ξ be Lipschitz in x with constant L_x, in u with constant L_u, and in time t with constant L_t. Additionally, let the drift be bounded uniformly in t, x, and u, i.e. $\|f_\xi(t, x, u)\| \leq M$, and let π_θ be Lipschitz in the state under the W_1 (1-Wasserstein distance) metric:*

$$W_1(\pi_\theta(\cdot|t, x), \pi_\theta(\cdot|t, y)) \leq L_\pi \|x - y\|$$

for all t, x, y. Then it follows that an Euler-based discretisation with time step h has an error upper bounded by

$$\mathbf{E}\|\bar{x}(t_n) - x_n\| \leq \underbrace{\frac{e^{L_{\mathrm{cl}}(T-t_0)} - 1}{2L_{\mathrm{cl}}}(L_x M + L_t)}_{C_x(T)} h = \mathcal{O}(h)$$

Proof: The proof is given in Appendix A.3. □

Using the above, it is possible to show that the policy and adversary gradients also have a limited discretisation error as shown in the corollary below.

Corollary 1 (discretisation error of policy and adversary gradients) *Under the preconditions of Lemma 1, and the standard Lipschitz assumptions in our general assumptions, the policy and adversary gradients have a discretisation error bounded by $\mathcal{O}(h)$.*

Proof: The policy and adversary gradients compute an expression of the form

$$\nabla J(\theta, \xi) = \mathbf{E}\left[\int_0^T g(t, x_t, u_t)\,\mathrm{d}t + G(x_T)\right],$$

for some functionals $g : \mathbb{R} \times \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, $G : \mathcal{X} \to \mathbb{R}$.

Adding and subtracting $g(t_n, \bar{x}(t_n), \bar{u}_n)$ shows that the integral discretisation error comes from the quadrature error and the grid error:

$$\int_{t_n}^{t_{n+1}} g(t, \bar{x}(t), \bar{u}_n)\,\mathrm{d}t - hg(t_n, x_n, u_n) = \underbrace{\int_{t_n}^{t_{n+1}} g(t, \bar{x}(t), \bar{u}_n) - g(t_n, \bar{x}(t_n), \bar{u}_n)\,\mathrm{d}t}_{\text{quadrature error}} \tag{27}$$

$$+ \underbrace{h(g(t_n, \bar{x}(t_n), \bar{u}_n) - g(t_n, x_n, u_n))}_{\text{grid error}}. \tag{28}$$

Following the preconditions, g is Lipschitz in state and time with constants $L_{g,x}$ and $L_{g,t}$. Therefore, the quadrature error gives

$$\left|\int_{t_n}^{t_{n+1}} g(t, \bar{x}(t), \bar{u}_n) - g(t_n, \bar{x}(t_n), \bar{u}_n)\right| \leq \frac{1}{2}(L_{g,x}M + L_{g,t})h^2.$$

The grid error can be obtained based on a similar coupling argument as in Equation (73),

$$\mathbf{E}[\text{grid error}] \leq (L_{g,x} + L_{g,u}L_\pi)\mathbf{E}\left\|\bar{x}(t_n) - x_n\right\|.$$

For the final bound, we sum the above over n, using $N = T/h$ and use the state discretisation error (77) to get the first-order error

$$|J - J_h| \leq h\left[\frac{T}{2}(L_{g,x}M + L_{g,t}) + C_x(T)(T(L_{g,x} + L_{g,u}L_\pi) + L_G)\right] = \mathcal{O}(h).$$

□

We can analogously bound the backward process' discretisation error by following a similar procedure as above (i.e. the discretisation for p_t), and then adding them. This leads to an overall first-order discretisation error $\mathcal{O}(h)$.

Note that the overall order is limited by the **state discretisation error** (i.e. improvement can occur when one uses a higher-order integrator, such as Runge-Kutta 2 or 4).

5 Regret analysis

Having defined various policy gradient and adversary gradients for CT-RMDPs, we now turn to a regret analysis for two selected algorithms compatible with CT-RMDPs. Both algorithms alternate policy and adversary gradients to provide a robust optimisation to approximate solve (5). To ensure the adversary's parameters remain within the uncertainty set at iteration $k \in [K]$, the adversary is updated via projected gradient ascent, i.e.

$$\xi_k \leftarrow \mathrm{proj}_\Xi\left(\xi_{k-1} + \eta\nabla_\xi J(\xi_{k-1})\right). \tag{29}$$

The first analysis is based on a double-loop algorithm, where the inner problem is assumed to be solved ϵ_k-optimally at each iteration $k \in [K]$. The double-loop analysis leads to a linear convergence rate in the oracle-based setting and an $\tilde{\mathcal{O}}(\frac{1}{\varepsilon^2})$ sample complexity in the sample-based setting, where ε is the specified error tolerance. The second analysis is based on a mean-field optimisation algorithm, which is based on a distributional min-max optimisation. It leads to an $\tilde{\mathcal{O}}(\frac{1}{K})$ convergence rate in the oracle-based setting and an $\tilde{\mathcal{O}}(\frac{N^2}{\varepsilon})$ sample complexity in the sample-based setting. While this result may in some cases correspond to a weaker rate, it allows a parallel particle based optimisation and allows distributional optimisation.

5.1 Rectangularity

Rectangularity is one of the main conditions for efficient solutions to RMDPs. Traditional RMDP analysis is based on directly parametrised uncertainty sets, where the notion of (x, u)-rectangularity is given by Definition 1.

Definition 1 (Rectangularity) *An uncertainty set is (x, u)-rectangular if it can be represented as a Cartesian product of independent sets for each state and action, i.e. $\mathcal{P} = \bigotimes_{(x,u)\in\mathcal{X}\times\mathcal{U}} \mathcal{P}_{x,u}$, where $\mathcal{P}_{x,u} \subseteq \mathbb{P}[\mathcal{X}]$.*

To illustrate the meaning, consider a distance measure D. A typical example is the uncertainty set defined by a distance metric according to $\mathcal{P} = \otimes_{(x,u)\in\mathcal{X}\times\mathcal{U}}\{\mathbf{P} \in \mathbb{P}[\mathcal{X}]^{\mathcal{X}\times\mathcal{U}} : D(\mathbf{P}(\cdot|x, u), \bar{\mathbf{P}}(\cdot|x, u)) \leq \psi(x, u)\}$, where $\psi(x, u)$ is the budget for a particular (x, u). A non-rectangular set example is $\mathcal{P} = \{\mathbf{P} \in \mathbb{P}[\mathcal{X}]^{\mathcal{X}\times\mathcal{U}} : D(p, \bar{\mathbf{P}}) \leq \psi\}$ for some fixed budget ψ.

Below we demonstrate that the inner problem optimisation problem, which typically assumes the rectangularity property, can be feasible for indirect parametrisations. In general though, uncertainty sets may be non-rectangular, with the efficiency of the adversarial optimiser being related to the distance from the rectangular approximation. Due to the additional time dependency and the need to construct more statistically efficient (i.e. more narrow, less conservative) indirectly parametrised uncertainty sets, it is still of interest to solve non-rectangular uncertainty sets. In such cases, the efficiency of the uncertainty set is traded off with a worse regret bound. To modify our regret analysis to account for non-rectangular uncertainty sets, an additional factor similar to that introduced in Li et al. (2026b) may need to be introduced, which depends on the non-rectangularity degree (i.e. the distance to the closest rectangular uncertainty set).

Discretised setting We first analyse rectangularity in the setting of discrete state and time, which aligns with the total cost MDP analysis (Section 5.2), in which we distinguish between transient states and recurrent states and maintain the generality of RL without requiring a special-case control theory setting. In our parametrisation, the system dynamics are time-, state-, and action-dependent. When the dynamics are discretised with time step Δt, we can write any SDE dynamics model (and therefore also any ODE dynamics model as a special case with zero variance) as some transition kernel

$$\mathbf{P}_\xi(\cdot|t_n, x_n, u_n) = Z\mathcal{N}(x_n + \Delta t b_\xi(t_n, x_n, u_n), \Delta t \sigma_\xi(t_n, x_n, u_n)),$$

where $Z > 0$ is a normalisation constant.

If the time-dependency is omitted, one obtains the Markov transition kernel of a traditional MDP

$$\mathbf{P}_\xi(\cdot|x_n, u_n) = Z\mathcal{N}(x_n + \Delta t b_\xi(x_n, u_n), \Delta t \sigma_\xi(x_n, u_n)). \tag{30}$$

Similarly, our adjoint-based gradient formulations demonstrate equivalence with the discrete policy and adversarial policy gradient as in Theorem 5.

With the above discretisation argument, it is possible to cast the problem in terms of a traditional (x, u)-rectangular parametrisation with the uncertainty set being comprised of the Cartesian product of all uncertainty sets with state-action dependent budgets. We identify a set of indirectly parametrised Gaussian transition kernels that are consistent with an SDE with linear dynamics and which satisfy the rectangularity property as in Definition 1. For simplicity, we investigate the case where ξ parametrizes only the drift although the variance may still be state-action dependent. Note that in our continuous-time dynamics parametrisation (6), the deviation from the nominal is given by $g_\xi(x, u)$. For several classes of functions, e.g. linear functions and many classes of MLPs, setting $\xi = \mathbf{0}$ will result in $g_\xi(x, u) = \mathbf{0}$. Formalizing the resulting transition dynamics in discrete-time, the problem is equivalent to shifting the mean of the normal distribution. If we project the state-actions to a high-dimensional feature space, it is possible to achieve state-action rectangularity as shown below.

Lemma 2 (A rectangular indirect parametrisation for linear SDEs) *Let $\mathcal{Z} = \{(x_n, u_n)\}_{n=1}^N \subset \mathcal{X} \times \mathcal{U}$ be a grid discretising the state-action space. Moreover, let $\phi : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^d$ be a fixed feature-map, where $d \geq N$, such that $\mathcal{B} = \{\phi(x, u)\}_{(x,u)\in\mathcal{X}\times\mathcal{U}} \subset \mathbb{R}^d$ are linearly independent. Then an uncertainty set $\mathcal{P}_\Xi$ of*

Gaussian transition kernels can be derived from the drift parametrisation

$$b_\xi(x,u) = \phi(x,u)^\mathsf{T}\xi\,, \xi \in \Xi \subset \mathbb{R}^{d\times d_x}\,, \tag{31}$$

for a linear SDE such that it is a well-defined (x,u)-rectangular uncertainty set with respect to Z. If further Ξ is a dense set with drifts spanning the sets $B_{x,u} = \{b_\xi(x,u) \in \mathbb{R}^{d_x} : \xi \in \Xi\} = \{b \in \mathbb{R}^{d_x} : \|\Delta t b\| \leq \rho(x,u)\}$ for all $(x,u) \in \mathcal{Z}$ and $\rho(x,u) \in \mathbb{R}$, then this parametrisation satisfies $\mathcal{P}_\Xi = \mathcal{P}$ for some L_1 uncertainty set $\mathcal{P}$.

Proof: The proof is given in Appendix A.4. □

Continuous setting While the above discretisation argument facilitates the analysis in Section 5.2, the mean-field optimisation analysis in Section 5.3 is more general in the sense that it can be applied to the full CT-RMDP with continuous states and continuous-time dynamics. We expect that rectangularity conditions may be required to meet some of the preconditions of the analysis, so we present here a generalisation to continuous states and continuous-time.

To deal with continuous states in RMDPs, recent work (Li et al., 2026a) has designed an indirectly parametrised x-rectangular uncertainty set for continuous state spaces. The uncertainty set is based on a mixture of parametrised transition kernels and uses the ϕ-divergence as a distance measure:

$$\mathcal{P}_x = \left\{ \sum_i w_i \mathbf{P}_{\xi_i}(\cdot|x,\cdot) : D_\phi(w,\bar{w}) \leq \psi(x)\,, \sum_i w_i = 1 \right\}\,, \tag{32}$$

where $\{\xi_i\}_i$ is a set of random samples or a grid over Ξ, D_ϕ is a ϕ-divergence, N is the number of components of the mixture, and $\bar{w}$ is the nominal weight vector. The exact parametrisation of $\mathbf{P}_{\xi_i}$ may vary.

Applying (32) to the standard SDE setting, the weights are set as one-hot vectors. Under this uncertainty set, one form is the linear SDE, which gives transition kernels of the form $\mathbf{P}_{\xi_i}(\cdot|x,u) = \mathcal{N}(x + \Delta t b_{\xi_i}(x,u), \Delta t \sigma_{\xi_i}(x,u))$, i.e. a Gaussian density for state $x \in \mathcal{X}$ and action $u \in \mathcal{U}$. The nominal model will be a mixture model, i.e. a Gaussian mixture for linear SDEs, while the models in the uncertainty set are Gaussians (with only one component).

To obtain the state dynamics in continuous-time, note that the time-homogeneous Fokker-Planck equation gives the evolution of the state density p_ξ over time,

$$\frac{\mathrm{d}}{\mathrm{d}t} p_\xi(x_t) = -\nabla_x \left(b_\xi(x_t,u_t) p_\xi(x_t) \right) + \nabla_x^2 \left(\frac{1}{2} \sigma_\xi(x_t,u_t) p_\xi(x_t) \right)\,. \tag{33}$$

This leads to an alternative, continuous-time characterisation of the uncertainty set for linear SDEs as

$$\mathcal{P}_x = \left\{ \sum_i w_i \frac{\mathrm{d}}{\mathrm{d}t} p_{\xi_i}(x_t) : D_\phi(w,\bar{w}) \leq \psi(x)\,, \sum_i w_i = 1\,, w_i \in \{0,1\} \right\}\,, \tag{34}$$

where it is emphasised that w is a one-hot vector but the nominal weight vector $\bar{w}$ is any vector with sum 1 such that the nominal model is a mixture SDE. The downside of the formulation 34 is that the nominal model itself is typically not in the uncertainty set.

For mixture SDEs, the restriction to one-hot vectors can be removed,

$$\mathcal{P}_x = \left\{ \sum_i w_i \frac{\mathrm{d}}{\mathrm{d}t} p_{\xi_i}(x_t) : D_\phi(w,\bar{w}) \leq \psi(x)\,, \sum_i w_i = 1 \right\}\,. \tag{35}$$

However, this is no longer a standard SDE, and so the policy and adversary gradient formulations of Section 4.3 do not apply, and this is beyond the scope of the paper.

5.2 Double-loop analysis

The first analysis uses a double-loop robust policy gradient algorithm (Algorithm 2). The algorithm performs basic gradient descent in the policy space and projected gradient ascent in the space of transition dynamics. In this setting, we first prove the Polyak-Łojasiewicz (PL) assumption (see Assumption 1). We then show the linear oracle-based convergence rate and the $\tilde{\mathcal{O}}(\frac{1}{\epsilon^3})$ sample complexity. We provide our result for a general parametrisation.

Previously, prior works have established PL conditions with respect to the robust policy gradient, in the context of directly parametrised discounted x-rectangular RMDPs (Kumar & Levy, 2023), average cost (x, u)-rectangular RMDPs (Sun, 2024), and discounted non-rectangular RMDPs (Li et al., 2026b). We provide results for a general stochastic policy parametrisation under PL condition assumptions within an undiscounted total cost RMDP framework. Alternative proofs under the softmax and direct parametrisations can be found in the double-loop mirror descent ascent scheme considered in prior works (Wang et al., 2024; Bossens & Nitanda, 2025). The analysis is based on the discrete-time policy gradient of our algorithm, in which case each continuous-time MDP in the uncertainty set is equivalent to an undiscounted total cost MDP for $\Delta t \to 0$; future work can consider deriving the convergence rates for the large time step case, where the continuous-time policy gradient will behave differently compared to the discrete-time policy gradient.

In the oracle-based setting, the linear convergence can be obtained for the last-iterate regret of transition mirror ascent in discounted MDPs (Wang et al., 2024; Bossens & Nitanda, 2025) and the last-iterate robust regret (Wang et al., 2024). While these techniques rely on direct parametrisations and require properties such as rectangularity and robust PL condition, below we focus on projected gradient ascent under general parametrisation on the undiscounted total cost MDP. This analysis relies on different PL assumptions (which may or may not partially overlap with rectangularity properties), our general Lipschitz and smoothness assumptions, and a geometrically shrinking mismatch of transition kernels.

Algorithm 2 Double-loop robust policy gradient for RMDPs.

Inputs: initial parameters θ_0,ξ_0, learning rate schedules η_1, η_2
Outputs: near-optimal parameters θ_K,ξ_K
for $k = 1, \ldots, K$ **do**
  Sample trajectories using $\pi_{\theta_{k-1}}$ and $f_{\xi_{k-1}}$
  Estimate $\nabla_\theta J(\theta_{k-1}, \xi_{k-1})$
  $\theta_k \leftarrow \theta_{k-1} - \eta_1(k-1)\nabla_\theta J(\theta_{k-1}, \xi_{k-1})$
  $t \leftarrow 0$
  while $J(\theta_k, \xi^t_{k-1}) + \epsilon_k < \max_{\xi \in \Xi} J(\theta_k, \xi)$ **do**
    Sample trajectories using π_{θ_k} and $f_{\xi^t_{k-1}}$
    Estimate $\nabla_\xi J(\theta_k, \xi^t_{k-1})$
    $\xi^{t+1}_{k-1} \leftarrow \text{proj}_\Xi \left(\xi^t_{k-1} + \eta_2(k-1, t-1)\nabla_\xi J(\theta_k, \xi^t_{k-1})\right)$
    $t \leftarrow t + 1$.
  end while
  $\xi_k \leftarrow \xi^t_{k-1}$.
end for

Assumption 1 (Polyak-Łojasiewicz (PL) condition) *Let $J : \Theta \to \mathbb{R}^+$ be a cost function and let $\mu > 0$, with optimal parameters θ^*. Then J satisfies the PL condition if*

$$\frac{1}{2}\left\|\nabla_\theta J(\theta)\right\|^2 \geq \mu(J(\theta) - J(\theta^*)) \,. \tag{36}$$

for all $\theta \in \Theta$.

We will make the assumption for both the policy optimisation and the adversarial optimisation, and we provide a lemma for both to support the assumption.

Regret analysis of the policy iterates Before deriving the regret of the policy with respect to Φ, we first obtain an expression for the regret based on the observed iterates $J(\theta_k, \xi_k)$ for $k = 1, \dots, K$. We start with a regret analysis of the policy iterates.

We formulate the analysis within the framework of undiscounted total cost MDPs, which can be seen as a discretised form of our continuous-time formulation. The transient vistation measure is a key concept within this framework which facilitates our analysis. Rather than working with the measure on all states, it is convenient to restrict the measure to the non-zero entries by only considering the set of transient states.

Definition 2 (Transient visitation measures) *For an undiscounted total cost MDP $\mathcal{M} = (s_0, \mathbf{P}, r, \mathcal{X}, \mathcal{U})$, where s_0 is the starting state, p is the transition kernel, r is the cost function, and $\mathcal{X}$ and $\mathcal{U}$ are the state and action spaces, define the* ***transient visitation measure*** *(Lee & Ryu, 2026) as $\tilde{d}^{\pi}_{s_0,\mathbf{P}}(s) = \sum_{t=0}^{\infty} \mathbf{P}(s_t = s | s_0, \mathcal{T}^{\pi,\mathbf{P}}) = e_{s_0}^{\intercal}(\mathbb{I} - \mathcal{T}^{\pi,\mathbf{P}})^{-1} e_s$, where $\mathcal{T}^{\pi,\mathbf{P}}$ is the transient matrix*

$$\mathcal{T}^{\pi,\mathbf{P}}(s, s') = \begin{cases} P^{\pi,\mathbf{P}}(s, s') & \text{if } s, s' \in \mathcal{X} \text{ are transient} \\ 0 & \text{otherwise}, \end{cases}$$

based on the induced transition probability $P^{\pi,\mathbf{P}}$ under policy π and transition dynamic $\mathbf{P}$, with e_j denoting the j'th unit vector. Similarly, we define the ***restricted transient visitation measure*** *$d^{\pi}_{s_0,\mathbf{P}}(s)$ as the transient visitation measure when restricted to the set of transient states.*

We first derive the PL condition for a stochastic policy on a fixed transition kernel, a result which can later be used to compute the worst-case PL coefficient. Note that this derives a squared norm based PL coefficient as opposed to the common non-squared norm based PL coefficient in the RMDP literature. The requirement of strictly positive costs-to-go for suboptimal policies is a mild condition that matches many control problems and the requirement of a minimal visitation probability is similar to the sufficient exploration requirement in policy gradient algorithms (Mei et al., 2020).

Lemma 3 (PL condition for the policy updates under direct and general parametrisations) *Let $\mathcal{M} = (d_0, \mathbf{P}, r, \mathcal{X}^+, \mathcal{U})$ be an undiscounted total cost MDP and let $J : \Theta \to \mathbb{R}^+$ be the total cost function with recurrent states $\mathcal{X}_{rec}$ and transient states $\mathcal{X} = \mathcal{X}^+ \setminus \mathcal{X}_{rec}$. Let $M = \max_{\pi,\pi'} \left\| \frac{d^{\pi}_{d_0,\mathbf{P}}}{d^{\pi'}_{d_0,\mathbf{P}}} \right\|_{\infty}$ and $0 < D_{min} \leq \min_{\pi,x} d^{\pi}_{d_0,\mathbf{P}}(x)$, where π is in the policy class parametrised by Θ and $x \in \mathcal{X}$. Denote the discretised action-value and state-value as $Q_h(x, u) \geq V_{min} > 0$ and $V_h(x) \geq V_{min}$ for $x \in \mathcal{X}$ and $u \in \mathcal{U}$. Then the PL condition is satisfied with coefficient $\mu = c^2/2$, where*
a) *under the direct parametrisation,*

$$c = (M d_\theta)^{-1/2} (D_{min} V_{min})^{1/2}\,; \quad \text{and}$$

b) *under a general parametrisation with $\left\| \mathbf{E}[Q_h(x, u) \nabla_\theta \log(\pi_\theta(u|x))] \right\|_{\infty} \geq \mathbf{E}[Q_h(x, u)]$,*

$$c = M^{-1/2} (D_{min} V_{min})^{1/2}\,.$$

Proof: The proof is given in Appendix A.5. □

The PL condition leads to the following descent property.

Lemma 4 (PL based descent property) *Let $(\theta_k, \xi_k)_{k=1}^{K}$ be a sequence of iterates. Let $J(\theta_k, \xi_k)$ be $l(\xi_k)$-smooth as a function of θ and let $J(\cdot, \xi_k)$ satisfy the PL condition (Assumption 1) with coefficient $\mu(\xi_k)$ for all $k = 0, \dots, K-1$. If the learning rate is $\eta_1(k-1) \leq 1/L$ for all $k = 1, \dots, K$, where $L = \sup_{\xi \in \Xi} l(\xi)$, then*

$$J(\theta_k, \xi_{k-1}) - J(\theta^*_{k-1}, \xi_{k-1}) \leq (1 - \eta_1(k-1)\mu_k)(J(\theta_{k-1}, \xi_{k-1}) - J(\theta^*_{k-1}, \xi_{k-1})) \tag{37}$$

where θ^*_{k-1} represents the optimal policy parameters under transition dynamics ξ_{k-1}.

Proof: This follows from the definition of $l(\xi_k)$-smoothness and the standard descent lemma (Lemma 9). For any $\Delta\theta$,

$$J(\theta_{k-1}+\Delta\theta,\xi_{k-1}) \leq J(\theta_{k-1},\xi_{k-1}) + \langle \nabla_\theta J(\theta_{k-1},\xi_{k-1}), \Delta\theta\rangle + \frac{L}{2}\|\Delta\theta\|^2$$

Set $\Delta\theta = -\eta_1(k-1)\nabla_\theta J(\theta_{k-1},\xi_{k-1})$. Then,

$$\begin{aligned} J(\underbrace{\theta_{k-1}-\eta_1(k-1)\nabla_\theta J(\theta_{k-1},\xi_{k-1})}_{\theta_k},\xi_k) &\leq J(\theta_{k-1},\xi_{k-1}) - \eta_1(k-1)\left(1-\frac{L}{2}\eta_1(k-1)\right)\left\|\nabla_\theta J(\theta_{k-1},\xi_{k-1})\right\|^2 \\ &= J(\theta_{k-1},\xi_{k-1}) - \frac{\eta_1(k-1)}{2}\left\|\nabla_\theta J(\theta_{k-1},\xi_{k-1})\right\|^2 \end{aligned}$$

From the PL condition (Assumption 1), we have that

$$\frac{1}{2}\left\|\nabla_\theta J(\theta_k,\xi_{k-1})\right\|^2 \geq \mu(\xi_{k-1})(J(\theta_{k-1},\xi_{k-1}) - J(\theta^*_{k-1},\xi_{k-1}))\,. \tag{38}$$

Subtracting the optimal cost from both sides and substituting the expression for the gradient norm, we get a linear rate to the optimum

$$J(\theta_k,\xi_{k-1}) - J(\theta^*_{k-1},\xi_{k-1}) \leq (1-\eta_1(k-1)\mu(\xi_{k-1}))\left(J(\theta_{k-1},\xi_{k-1}) - J(\theta^*_{k-1},\xi_{k-1})\right)\,. \tag{39}$$

□

Now the linear convergence of the policy iterates can be shown using the above descent lemma and a further restriction on the transition kernel mismatch between adversarial updates.

Lemma 5 (Regret of policy iterates) *Let $\{\xi_k\}_{k=1}^K$ be the sequence of transition dynamics parameters obtained from the algorithm, let $\epsilon_0 > 0$, let $\mu > 0$ lower bound the PL coefficient $\mu(\xi)$ for $J(\theta,\xi)$ for all $\xi \in \Xi$ as a function of θ, let $\eta_1(k) \geq \eta_{min} > 0$ for all $k \geq 0$, and let $M_k \geq \left\|\frac{d_{d_0}^{\pi,\mathbf{P}_{\xi_k}}}{d_{d_0}^{\pi,\mathbf{P}_{\xi_{k-1}}}}\right\|_\infty$ and let $\bar{M}$ upper bound the geometric mean of the sequence $\{M_k\}_{k=1}^{K-1}$. Moreover, let θ^*_{k-1} represent the optimal policy parameters under ξ_{k-1}. Then under the preconditions of Lemma 4, we have that*

$$J(\theta_K,\xi_{K-1}) - J(\theta^*_{K-1},\xi_{K-1}) \leq c_{pol}^K J(\theta_0,\xi_0) \tag{40}$$

where $c_{pol} = \bar{M}c$ where $c = (1-\eta_{min}\mu) < \frac{1}{\bar{M}}$.

Proof: Due to the descent property (Lemma 4) and the non-negativity of costs, the suboptimality is bounded for $k \geq 2$ by

$$\begin{aligned} J(\theta_k,\xi_{k-1}) - J(\theta^*_{k-1},\xi_{k-1}) &\leq (1-\eta_1(k-1)\mu(\xi_{k-1}))(J(\theta_{k-1},\xi_{k-1}) - J(\theta^*_{k-1},\xi_{k-1})) \\ &\leq (1-\eta_1(k-1)\mu(\xi_{k-1}))M_{k-1}\left(J(\theta_{k-1},\xi_{k-2}) - J(\theta^*_{k-1},\xi_{k-2}))\right)\,, \end{aligned} \tag{41}$$

where M_{k-1} is the multiplicative mismatch introduced by changing the transition dynamics. The second inequality follows from the properties of the transient visitation measure, i.e. for any $\theta \in \Theta$ we have

$$\begin{aligned} J(\theta,\xi_{k-1}) &= \sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_{\xi_{k-2}}}(x)\frac{d^{\pi_{\theta^*}}_{d_0,\mathbf{P}_{\xi_{k-1}}}(x)}{d^{\pi_\theta}_{d_0,\mathbf{P}_{\xi_{k-2}}}(x)}\langle r(x,\cdot),\pi_\theta(\cdot|x)\rangle \\ &\leq \sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_{\xi_{k-2}}}(x)\left\|\frac{d_{d_0}^{\pi,\mathbf{P}_{\xi_{k-1}}}}{d_{d_0}^{\pi,\mathbf{P}_{\xi_{k-2}}}}\right\|_\infty \langle r(x,\cdot),\pi_\theta(\cdot|x)\rangle \\ &\leq M_{k-1}J(\theta,\xi_{k-2})\,. \end{aligned} \tag{42}$$

Solving recursion (41) while noting the definition of c_{pol} and the non-negative cost function, we have

$$J(\theta_k, \xi_{k-1}) - J(\theta^*_{k-1}, \xi_{k-1}) \leq c^{k-1} \left(\Pi_{j=2}^{k} M_{j-1}\right) \left(J(\theta_1, \xi_0) - J(\theta^*_0, \xi_0)\right) \leq c_{\text{pol}}^{k} \left(\Pi_{j=2}^{k} M_{j-1}\right) J(\theta_1, \xi_0)\,.$$

For $k = 1$, note that

$$J(\theta_1, \xi_0) - J(\theta^*_0, \xi_0) \leq (1 - \eta_1(0)\mu(\xi_0))(J(\theta_0, \xi_0) - J(\theta^*_0, \xi_0)) \leq c_{\text{pol}}(J(\theta_0, \xi_0) - J(\theta^*_0, \xi_0))\,.$$

So we conclude from the product of mismatches and non-negativity that for any $K \geq 2$,

$$\begin{aligned} J(\theta_K, \xi_{K-1}) - J(\theta^*_{K-1}, \xi_{K-1}) &\leq c^K \left(\Pi_{j=2}^{K} M_{j-1}\right) (J(\theta_0, \xi_0) - J(\theta^*_0, \xi_0)) \\ &\leq c^K \left(\Pi_{j=2}^{K} M_{j-1}\right) J(\theta_0, \xi_0) \\ &\leq c_{\text{pol}}^K J(\theta_0, \xi_0)\,, \end{aligned}$$

where the last step is based on the geometric mean upper bound, i.e. $c^K \Pi_{j=2}^{K} M_{j-1} \leq c^K \bar{M}^{K-1} \leq c^K \bar{M}^K = c_{\text{pol}}^K$. □

Regret analysis of the adversary iterates Due to applying projected gradient ascent, the PL condition is slightly modified. Related to the LPL condition (Balashov et al., 2019), we derive a modified PL condition for projected gradient ascent.

Assumption 2 (Modified PL condition) *Let $J : \Xi \to \mathbb{R}$ be a utility function with optimal parameters ξ^*. Then J satisfies the Modified PL condition if*

$$\frac{1}{2}\left\|h_\eta(\xi)\right\|^2 \geq \mu(J(\xi^*) - J(\xi))\,. \tag{43}$$

for all $\xi \in \Xi$ for some $\mu \in \mathbb{R}$, where $h_\eta(\xi) = \frac{1}{\eta}\left(proj_\Xi\left(\xi + \eta \nabla_\theta J(\xi)\right) - \xi\right)$ is the projected gradient mapping.

The below lemma proves the condition for a directly parametrised uncertainty set under an additional condition that the projected gradient should always be a non-zero fraction of the unprojected gradient. This can be achieved using a soft projection to a point within the interior that is close to the boundary rather than exactly at the boundary, e.g. using barrier functions or by applying conservative policy updates near the boundary. Note that a similar technique could be applied to obtain a projected policy gradient. Similarly, the traditional PL condition is enough for particular uncertainty set definitions as a limited uncertainty set can be obtained under unbounded parameters as well (see also Remark 3).

Lemma 6 (Modified PL condition for directly parametrised adversary) *Let $J(\theta) : \Xi \to \mathbb{R}$ be a utility function with optimal parameters $\xi^* = p^*$ for an undiscounted total cost RMDP $\mathcal{M} = (s_0, \mathcal{P}, r, \mathcal{X}^+, \mathcal{U})$ with directly parametrised uncertainty set $\mathcal{P} = \Xi$ with recurrent states $\mathcal{X}_{rec}$ and transient states $\mathcal{X} = \mathcal{X}^+ \setminus \mathcal{X}_{rec}$. Let $M = \max_{\pi,\pi'} \left\| \frac{d^{\pi}_{d_0,\mathbf{P}}}{d^{\pi'}_{d_0,\mathbf{P}}} \right\|_\infty$ and $0 < D_{min} \leq \min_\pi d^{\pi}_{d_0,\mathbf{P}}(x')$ for all $x \in \mathcal{X}$, where π is in the policy class parametrised by Θ. Denote the state-value as $V_h(x) \geq V_{min} > 0$ for $x \in \mathcal{X} \setminus \mathcal{X}_{rec}$ and note that $\mathcal{X}_{rec}$ are absorbing states with $V_h(x) = 0$ for $x \in \mathcal{X}_{rec}$. Moreover, let*

$$\left\|h_\eta(\xi)\right\| \geq \beta \left\|\nabla_\xi J(\theta, \xi)\right\|$$

for $\beta \in (0, 1]$. Then the modified PL condition holds with parameter $\mu = c^2/2$, where $c = (M d_\xi)^{-1/2} (D_{min} V_{min})^{1/2} \beta$.

Proof: The proof is given in in Appendix A.6. □

Based on the modified PL condition, the below lemma formulates the ascent property for adversary gradient ascent.

Lemma 7 (Modified PL based ascent property) *Let $(\xi_k^{t'})_{t'=1}^{t_k}$ be a sequence of iterates from the inner problem, applying projected gradient ascent in a compact uncertainty set Ξ with ℓ_q distance projection for some $q \geq 1$. Let $J(\theta_k, \xi)$ be $l(\theta_k)$-smooth as a function of ξ and let $J(\theta_{k-1}, \xi_{k-1})$ satisfy the Modified PL condition (Assumption 2) with coefficient $\mu_k = \mu(\theta_{k-1})$ for all $k = 1, \ldots, K$. If the learning rate is $\eta_2(k-1, t-1) \leq \frac{1}{L}$ for all $k \in [K]$, $t \in [t_k]$, where $L = \sup_{\theta \in \Theta} l(\theta)$, then*

$$J(\theta_{k-1}, \xi_{k-1}^*) - J(\theta_{k-1}, \xi_{k-1}^t) \leq (1 - \eta_2(k-1, t-1)\mu_k)(J(\theta_{k-1}, \xi_{k-1}^*) - J(\theta_{k-1}, \xi_{k-1}^{t-1})) \tag{44}$$

where ξ_{k-1}^* represents the optimal transition dynamics for policy θ_{k-1}.

Proof: The proof is analogous to that of Lemma 4 but makes use of the projected gradient mapping. The full derivation is provided in Appendix A.7. □

The above leads to a linear convergence for the adversary, allowing to obtaini an ϵ_k-optimal transition kernel at each macro-iteration $k \in [K]$ within a limited number of updates t_k.

Theorem 8 (Regret upper bound of adversary) *Let θ_k parametrize the policy at macro-iteration k before the adversary update, and let ξ_{k-1}^0 parametrize the transition dynamics at the start of the inner iterations. Moreover, let $c_{adv} = \max_{k,t \geq 0}(1 - \eta_2(k,t)\mu(\theta_k)) \in (0,1)$, where $\mu(\theta_k)$ is the modified PL coefficient for $J(\theta_k, \xi)$ as a function of ξ, and $\eta_2(k,t)$ is the step size of the adversary at macro-iteration k and micro-iteration t. Further let $J_{max} = \max_{\theta \in \Theta, \xi \in \Xi} J(\theta, \xi)$. Moreover, let ξ_{k-1}^* represent the optimal adversary parameters for macro-iteration k. Let the learning rate be $\eta_2(k,t) \leq 1/L$ for all $k, t \geq 0$, where $L = \sup_{\theta \in \Theta} l(\theta)$ is the maximal smoothness when J is parametrised with fixed θ. After t_k micro-iterations of updating the inner parameters, the regret is bounded by*

$$J(\theta_k, \xi_{k-1}^*) - J(\theta_k, \xi_{k-1}^t) \leq c_{adv}^{t_k}(J(\theta_k, \xi_{k-1}^*) - J(\theta_k, \xi_{k-1}^0)) \,. \tag{45}$$

Moreover, a setting of $t_k = \Theta\left(\log_{\frac{1}{c_{adv}}}(\frac{J_{max}}{\varepsilon})\right)$ is sufficient to obtain an ε-optimal solution.

Proof: From Lemma 7, we have that

$$J(\theta_k, \xi_{k-1}^*) - J(\theta_k, \xi_{k-1}^t) \leq (1 - \eta_2(k)\mu_k)(J(\theta_k, \xi_{k-1}^*) - J(\theta_k, \xi_{k-1}^{t-1})) \,.$$

Iteratively applying the above for all $t = 1, \ldots, t_k$ leads to

$$J(\theta_{k-1}, \xi_{k-1}^*) - J(\theta_{k-1}, \xi_{k-1}^t) \leq c_{\text{adv}}^t(J(\theta_k, \xi_{k-1}^*) - J(\theta_k, \xi_{k-1}^0)) \leq c_{\text{adv}}^t J_{\max} \,.$$

For $t_k = \Theta\left(\log_{\frac{1}{c_{\text{adv}}}}(\frac{J_{\max}}{\varepsilon})\right)$, we then have $c_{\text{adv}}^{t_k} J_{\max} = \mathcal{O}(\varepsilon)$. □

Oracle-based regret analysis The above theorems do not yet give an expression for the robust objective, Φ, in the RMDPs. The theorem below provides a linear convergence for the last-iterate regret upper bound with respect to Φ under a further condition on the learning rate.

Theorem 9 (Oracle-based last-iterate regret of double-loop algorithm) *Let $\epsilon_k \leq c_{pol}\epsilon_{k-1}$ be the adversary error tolerance. Following the definitions in Lemma 5, the double-loop algorithm obtains a linear convergence in the robust objective according to*

$$\Phi(\theta_K) - \Phi(\theta^*) \leq c_{pol}^K(J(\theta_0, \xi_0) + \mathcal{O}(\epsilon_0)) \,. \tag{46}$$

For an $\mathcal{O}(\varepsilon)$-optimal policy, at most $K = \mathcal{O}(\log(\frac{1}{\varepsilon}))$ iterations and $K_2 = \mathcal{O}\left(\log^2(\frac{1}{\varepsilon})\right)$ gradient evaluations are required.

Proof: The robust regret is upper bounded by

$$\begin{aligned}
\Phi(\theta_K) - \Phi(\theta^*) &\leq J(\theta_K, \xi_K) + \epsilon_K - J(\theta^*, \xi_K) && \text{(Lemma 10)}\\
&\leq J(\theta_K, \xi_K) + \epsilon_K - J(\theta^*_{K-1}, \xi_K) && (\theta^*_K \text{ is minimizer for } \xi_K)\\
&\leq M\left(J(\theta_K, \xi_{K-1}) + \epsilon_K - J(\theta^*_{K-1}, \xi_{K-1}\right) && \text{(multiplicative bound (42))}\\
&\leq M c_{\text{pol}}^K \left(J(\theta_0, \xi_0) + \epsilon_0\right),
\end{aligned}$$

where the last step follows from Lemma 5, the definition of ϵ_k, and non-negativity of the cost. Defining $\bar{t} = \max_{k\in[K]} t_k$, the number of gradient evaluations per inner optimisation loop is given by $\bar{t} = \Theta\left(\log_{\frac{1}{c_{\text{pol}}}}(\frac{J(\theta_0,\xi_0))}{\varepsilon})\right)$. Denoting $x = J(\theta_0, \xi_0)) + \epsilon_0$, $K = \mathcal{O}\left(\log_{\frac{1}{c_{\text{pol}}}}(\frac{x}{\varepsilon})\right)$ iterations are sufficient for an ε-optimal policy. With one policy step and at most $\bar{t}$ adversary steps, this leads to a total of $K_2 \leq K(1+\bar{t}) = \mathcal{O}\left(\log_{\frac{1}{c_{\text{pol}}}}(\frac{x}{\varepsilon}) \times (1 + \log_{\frac{1}{c_{\text{adv}}}}(\frac{J_{\max}}{\varepsilon}))\right)$ gradient evaluations. Dropping multiplicative constants leads to $K = \mathcal{O}(\log(\frac{1}{\varepsilon}))$ iterations and $K_2 = \mathcal{O}\left(\log^2(\frac{1}{\varepsilon})\right)$ gradient evaluations. □

Sample-based regret analysis To account for the inaccuracy of the policy gradient estimates, the below analysis provides the average regret upper bound based on a sample-based analysis.

Theorem 10 (Sample-based last-iterate regret of double-loop algorithm) *Let $J : \Theta \times \Xi \to \mathbb{R}^+$ be an L-smooth function in Θ and in Ξ for all $\xi \in \Xi$ and all $\theta \in \Theta$. Let $\varepsilon > 0$ and define the adversary error tolerance as $\epsilon_K = \mathcal{O}(\varepsilon)$ while M, K, ϵ_k, and c_{pol} are defined as in Lemma 5 and Theorem 8. Let the number of macro-iterations further satisfy $K = \Theta(\log_{1/c_{pol}}(\frac{1}{\varepsilon}))$. Let the initial learning rates satisfy $\eta_1(0) \leq 1/L$ and $\eta_2(k-1, 0) \leq \min\{1/L, \epsilon_0\}$ for all $k \geq 1$. Define learning rate schedules such that $\eta_1(k) \leq \gamma\eta_1(k-1) \geq \eta_{min} > 0$ and $\eta_2(k-1, t) \leq \gamma\eta_2(k-1, t-1)$ for some given $\gamma \in (0, 1)$ for all $k \geq 1$ and all $t \geq 1$. Further, let $J_{max} = \max_{\theta\in\Theta,\xi\in\Xi} J(\theta, \xi)$ and set the number of inner loop iterations $t_k = \Theta\left(\log_{\frac{1}{c_{adv}}}(\frac{1}{\epsilon_k})\right)$ and $\bar{t} = \max_{k\in[K]} t_k$. With these settings, the average regret with respect to the robust objective (5) is given by*

$$\Phi(\theta_k) - \Phi(\theta^*) \leq c_{pol}^K M \left(J_{max} + \tilde{\mathcal{O}}(\epsilon_0)\right) + M_\theta \sqrt{d_\theta}\mathcal{O}(\varepsilon) \tag{47}$$

for some constant $M, M_\theta > 0$, and the sample complexity is bounded by $\tilde{\mathcal{O}}(\frac{1}{\varepsilon^2})$.

Proof: Note that in the sample-based setting, an error occurs at each iteration $k \in [K]$ in the single step optimisation of the policy update and the t_k steps of gradient ascent for the inner problem. To account for the error in the inner problem, define projected gradient mappings as in Lemma 16,

$$\hat{h}(k, t) = \frac{1}{\eta_2(k-1, t-1)} \left(\text{proj}_\Xi \left(\xi_{k-1}^{t-1} + \eta_2(k-1, t-1)\hat{\nabla} J(\theta_{k-1}, \xi_{k-1}^{t-1})\right) - \xi_{k-1}^{t-1}\right)$$

and

$$h(k, t) = \frac{1}{\eta_2(k-1, t-1)} \left(\text{proj}_\Xi \left(\xi_{k-1}^{t-1} + \eta_2(k-1, t-1)\nabla J(\theta_{k-1}, \xi_{k-1}^{t-1})\right) - \xi_{k-1}^{t-1}\right).$$

Let $n_k = \tilde{\mathcal{O}}\left(\frac{1}{\epsilon_k^2}\right)$ be the number of samples for the adversary gradient approximation at macro-iteration k. Then via Lemma 16, the projected gradient is approximated ϵ_k-precisely such that $\left\|\hat{h}(k, t') - h(k, t')\right\| \leq \sqrt{d_\xi}\left\|\hat{h}(k, t') - h(k, t')\right\|_\infty \leq \sqrt{d_\xi}\epsilon_k$ for all $t' \in [t_k]$.

Now denote $X(k,t) := \left\|\xi_{k-1}^{t} - \xi_{k-1}^{t,\star}\right\|$. The deviation of iterates compared to iterates under exact gradients is given by

$$\begin{aligned}
X(k,t) &= \left\|\sum_{t=1}^{t_k} \eta_2(k-1,t-1)\left(\hat{h}(k,t) - h(k,t) + h(k,t) - h'(k,t)\right)\right\| \\
&\leq \frac{\eta_2(k-1,0)}{1-\gamma}\left(\max_{t'}\left\|\hat{h}(k,t') - h(k,t')\right\| + \max_{t'}\left\|h(k,t'') - h'(k,t''))\right\|\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(geometric learning rate and triangle inequality)} \\
&\leq \frac{\epsilon_0}{1-\gamma}\left(\sqrt{d_\xi}\epsilon_k + 2B_\xi\right). \qquad\qquad\qquad\qquad \text{(constant settings and relation between norms)}
\end{aligned}$$

Note that $X(k,0) = 0$ such that $X(k,t) = \mathcal{O}(\epsilon_0)$ for all $t \geq 1$. Due to the setting of $t_k = \Theta\left(\log(1/\epsilon_k)\right)$, the above error and Theorem 8 imply an $\tilde{\mathcal{O}}(\epsilon_k + \epsilon_0) = \tilde{\mathcal{O}}(\epsilon_0)$ last-iterate regret solution to the inner problem at each macro-iteration.

Denote $(\tilde{\theta}_k)_{k=1}^K$ as the iterates under exact gradient descent. Following similar arguments as in Theorem 9 and setting $\Delta_k := J(\theta_k, \xi_k) - J(\tilde{\theta}_k, \xi_k)$, the regret under inexact policy gradient descent with iterates $(\theta_k)_{k=1}^K$ is given by

$$\begin{aligned}
\Phi(\theta_K) - \Phi(\theta^*) &\leq \left(J(\tilde{\theta}_K, \xi_K) + \tilde{\mathcal{O}}(\epsilon_0) - J(\theta^*, \xi_K)\right) + \Delta_k \qquad\qquad \text{(Lemma 10)} \\
&\leq c_{\text{pol}}^K M\left(J(\tilde{\theta}_0, \xi_0) + \tilde{\mathcal{O}}(\epsilon_0))\right) + \Delta_k \qquad\qquad \text{(using Lemma 5 as in Theorem 9)} \\
&\leq c_{\text{pol}}^K M\left(J_{\max} + \tilde{\mathcal{O}}(\epsilon_0)\right) + \Delta_k\,,
\end{aligned}$$

where the last step follows from the setting of $K = \Theta(1/\varepsilon)$, the upper bound on the cost, and the geometric series.

Now denote $Y(k) := \left\|\theta_k - \tilde{\theta}_k\right\|$ as the deviation of the iterates from those under exact gradient descent $(\tilde{\theta}_k)_{k=1}^K$. Then

$$\begin{aligned}
Y(k) &= \left\|\sum_{k=1}^{K} \eta_1(k-1)\left(\hat{\nabla}J(\theta_{k-1}, \xi_k) - \nabla J(\tilde{\theta}_{k-1}), \xi_k)\right)\right\| \\
&= \left\|\sum_{k=1}^{K} \eta_1(k-1)\left(\hat{\nabla}J(\theta_{k-1}, \xi_k) - \nabla J(\theta_{k-1}), \xi_k) + \nabla J(\theta_{k-1}), \xi_k) - \nabla J(\tilde{\theta}_{k-1}, \xi_k)\right)\right\| \\
&\leq \frac{\eta_1(0)}{(1-\gamma)}\left\|\mathcal{O}(\varepsilon) + LY(k-1)\right\|\,.
\end{aligned}$$

To resolve the recursion, note that $Y(0) = 0$ such that $Y(k) = \mathcal{O}(\varepsilon)$ for all $k \geq 1$. Applying M_θ-Lipschitz continuity of J, we have for all $k \in [K]$ that

$$\Delta_k \leq M_\theta\left\|\theta_k - \theta_k'\right\| \leq M_\theta\sqrt{d_\theta}\left\|\theta_k - \theta_k'\right\|_\infty \leq M_\theta\sqrt{d_\theta}\mathcal{O}(\varepsilon)\,.$$

With $K = \Theta(\log(\frac{1}{\varepsilon}))$, the sample-based regret is given by

$$\begin{aligned}
\Phi(\theta_K) - \Phi(\theta^*) &\leq c_{\text{pol}}^K M\left(J_{\max} + \tilde{\mathcal{O}}(\epsilon_0)\right) + M_\theta\sqrt{d_\theta}\mathcal{O}(\varepsilon) \\
&= \mathcal{O}(\varepsilon)\,,
\end{aligned}$$

Now set $n = \tilde{\mathcal{O}}(\frac{1}{\varepsilon^2})$ as the number of trajectories required to estimate the policy gradient as in Lemma 16. Note that similarly, due to the definition of n_k, we have that $n_k \leq n_K = \tilde{\mathcal{O}}(\frac{1}{\varepsilon^2})$. Therefore, the total number

of required samples, n_{total}, is given by

$$n_{\text{total}} = \sum_{k=1}^{K} n + n_k t_k \leq K\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\right)(1+\bar{t}) = \tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\right)\left(1+\Theta\left(\log\left(\frac{1}{\varepsilon}\right)\right)\right) = \tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\right).$$

□

5.3 Mean-field optimisation

Now we consider mean-field optimisation, which is based on a distributional optimisation formulation. Such a formulation can more readily The second regret analysis is based on mean-field Langevin stochastic descent-ascent (MFL-SDA) (Liu et al., 2025), an algorithm for distributional min-max optimisation. We formulate the approach for RMDPs as summarised in Algorithm 3. The assumptions are inherited from (Liu et al., 2025). Rather than the PL condition, its distributional counterpart, the log-Sobolev inequality is assumed.

Definition 3 (Log-Sobolev inequality (LSI)) *A distribution ν' satisfies LSI with parameter $\alpha > 0$ if for all $\nu \ll \nu'$ (i.e. absolutely continuous),*

$$\mathcal{H}(\nu||\nu') \leq \frac{1}{2\alpha}\mathcal{I}(\nu|\nu'), \tag{48}$$

where $\mathcal{I}(\nu|\nu') = \mathbf{E}_\nu\left[\left\|\nabla \log \frac{\mathrm{d}\nu}{\mathrm{d}\nu'}\right\|^2\right]$ is the relative Fisher information.

5.3.1 Regret analysis

MFL-SDA solves the min-max optimisation problem

$$\min_{\nu\in\mathbb{P}[\Theta]} \max_{\rho\in\mathbb{P}[\Xi]} E(\nu,\rho) = G(\nu,\rho) + \tau\left(\mathcal{H}(\nu|\nu_{\text{ref}}) - \mathcal{H}(\rho|\rho_{\text{ref}})\right), \tag{49}$$

where $G(\nu,\rho) = \mathbf{E}_{\theta\sim\nu,\xi\sim\rho}\left[J(\theta,\xi)\right]$ and the ref subscripts indicate Gaussian reference distributions. $\mathbb{P}[\Theta]$ denotes the set of distributions over Θ, the policy parameter space, and $\mathbb{P}[\Xi]$ is defined similarly for the space of transition kernels. The energy functional is strongly convex in ν and strongly concave in ρ, leading to the existence of a unique Nash equilibrium. This leads to Wasserstein gradient flow equations given by

$$\mathrm{d}\theta_t = -\nabla\frac{\delta G[\nu_t,\rho_t](\theta_t)}{\delta\nu}\,\mathrm{d}t - \tau\theta_t\,\mathrm{d}t + \sqrt{2\tau}\,\mathrm{d}B_t^1 \tag{50}$$

$$\mathrm{d}\xi_t = \eta\nabla\frac{\delta G[\nu_t,\rho_t](\xi_t)}{\delta\rho}\,\mathrm{d}t - \tau\xi_t\,\mathrm{d}t + \sqrt{2\tau}\,\mathrm{d}B_t^2, \tag{51}$$

where $\frac{\delta G[\nu_t,\rho_t](\theta_t)}{\delta\nu}$ and $\frac{\delta G[\nu_t,\rho_t](\xi_t)}{\delta\rho}$ represent the first variations of G with respect to ν and ρ, and B_t^i for $i = 1,2$ represent separate Brownian terms initialised at $B_0^i = 0$. The terms $\tau\theta_t$ and $\tau\xi_t$ show up due to the KL divergence being equal to the entropy plus second order moment for standard-normal reference distributions. The time-scale ratio $\eta > 0$ can be set following Lu (2023) to ensure linear convergence. discretising the flow in time, MFL-SDA updates are given by the lines 4–5 of Algorithm 3.

The specific LSI assumption over the policy and adversary distributions is defined through Gibbs distributions representing the inner maximizers.

Assumption 3 *For any $\nu \in \mathbb{P}[\Theta]$ and any $\rho \in \mathbb{P}[\Xi]$, the Gibbs distributions*

$$P_\nu[\rho](\xi) \propto \exp\left(\tau^{-1}\frac{\delta G[\nu,P_\nu[\rho]](\xi)}{\delta\rho} - \frac{1}{2}\|\xi\|^2\right)$$

$$P_\rho[\nu](\theta) \propto \exp\left(-\tau^{-1}\frac{\delta G[P_\rho[\nu],\rho](\theta)}{\delta\nu} - \frac{1}{2}\|\theta\|^2\right)$$

both satisfy LSI (Definition 3) with parameter $\alpha > 0$.

The relation between the PL condition and LSI has been discussed in previous works. In particular, one of the interpretations is as a PL condition over the functional $\mathcal{H}(\nu|\nu') = \int \log(\frac{\mathrm{d}\nu}{\mathrm{d}\nu'})\,\mathrm{d}\nu$ in the Wasserstein geometry (Otto & Villani, 2000). Applying this reasoning with $\nu' = P_\nu[\rho]$,

$$\mathcal{H}(\nu|\nu') \leq \frac{1}{2\alpha}\mathcal{I}(\nu|\nu') = \frac{1}{2\alpha}\int \left\|\nabla \log \frac{\mathrm{d}\nu}{\mathrm{d}\nu'}\right\|^2 \mathrm{d}\nu = \frac{1}{2\alpha}\left\|\nabla \mathcal{H}(\nu|\nu')\right\|^2 . \tag{52}$$

Analogously $\mathcal{H}(\rho|\rho') \leq \frac{1}{2\alpha}\left\|\mathcal{H}(\rho|\rho')\right\|^2$ for $\rho' = P_\rho[\nu]$. Such a PL condition enables linear convergence to the optimal Gibbs distributions.

In addition to the LSI assumption, MFL-SDA requires the three below assumptions.

Assumption 4 (Initial condition) *The initial distribution has finite energy, i.e. $E(\nu_0, \rho_0) < \infty$, and the initial third order moments satisfy $\mathbf{E}_{\theta\sim\nu_0}\left[\|\theta\|_2^4\right], \mathbf{E}_{\xi\sim\rho_0}\left[\|\xi\|_2^4\right] < \infty$.*

Assumption 5 (Regularity of G) *The functional $G(\nu, \rho)$ is convex in ν and concave in ρ. The first variations have bounded higher-order derivatives, up to the fourth power, i.e. $\left\|\nabla^i \frac{\delta G}{\delta \nu}\right\|_F, \left\|\nabla^i \frac{\delta G}{\delta \rho}\right\|_F \leq M_i$ where $\|\cdot\|_F$ indicates the Frobenius norm. The cross second-order variations are bounded, i.e. $\left\|\frac{\delta^2 G}{\delta\nu\delta\rho}[\nu,\rho](\xi)\right\|_\infty \leq C_0$. Moreover, the Hessian and Jacobian of its second variations are bounded, i.e. $\left\|\nabla_\theta \nabla_\theta^\mathsf{T} \frac{\delta^2 G}{\delta\nu\delta\rho}\right\|, \left\|\nabla_\theta \nabla_\theta^\mathsf{T} \frac{\delta^2 G}{\delta\nu\delta\rho}\right\| \leq C_1$ and $\left\|\nabla_\theta \frac{\delta^2 G}{\delta\nu\delta\rho}\right\|, \left\|\nabla_\rho \frac{\delta^2 G}{\delta\nu\delta\rho}\right\| \leq C_2$.*

Note that the convexity in ν is not restrictive since for any $\gamma \in (0, 1)$

$$\begin{aligned} G(\gamma\nu + (1-\gamma)\nu', \rho) &= \int_\Xi \int_\Theta J(\theta, \xi)\,(\gamma\nu + (1-\gamma)\nu)(\theta)\rho(\xi) \\ &= \int_\Xi \int_\Theta J(\theta,\xi)\gamma\nu(\theta)\rho(\xi) + \int_\Xi \int_\Theta J(\theta)(1-\gamma)(\theta,\xi)\rho(\xi) = \gamma G(\nu,\rho) + (1-\gamma)G(\nu',\rho)\,. \end{aligned}$$

For concavity in ρ, the analogous equality follows.

Assumption 6 (Bounded moments) *Define $g_k \coloneqq \nabla \frac{\delta}{\delta\nu} G[\nu_k, \rho_k](\theta_k)$ and $h_k \coloneqq \frac{\delta J}{\delta\rho}[\nu_{k+1}, \rho_k](\xi_k)$. Then there exists a scalar $\zeta \geq 0$ such that $\mathbf{E}\left[\|g_k - \hat{g}_k\|^4 |\nu_k, \rho_k\right], \mathbf{E}\left[\left\|h_k - \hat{h}_k\right\|^4 |\nu_{k+1}, \rho_k\right] \leq \zeta$.*

Following the uniform-in-time propagation of chaos property of MFL dynamics (Chen et al., 2025), the empirical distribution of an N-particle system will closely match that of the true distribution, with an approximation error of at most $\mathcal{O}(1/N)$, for all iterations of the algorithm. Therefore, we consider a practical implementation with N particles to implement the algorithm.

Due to G being bilinear in its arguments, the exact Wasserstein gradients are given by

$$\nabla \frac{\delta G}{\delta \nu}[\nu,\rho](\theta) = \mathbf{E}_{\xi\sim\rho}[\nabla_\theta J(\theta,\xi)], \tag{53}$$

$$\nabla \frac{\delta G}{\delta \rho}[\nu,\rho](\xi) = \mathbf{E}_{\theta\sim\nu}[\nabla_\xi J(\theta,\xi)]\,. \tag{54}$$

We then consider finite-particle approximations similar to SGD-MFLD (Suzuki et al., 2023),

$$\nabla \frac{\delta G}{\delta \nu}[\nu,\rho](\theta^i) \approx \frac{1}{N}\sum_{j=1}^N \nabla_\theta J(\theta^i, \xi^j), \tag{55}$$

$$\nabla \frac{\delta G}{\delta \rho}[\nu,\rho](\xi^j) \approx \frac{1}{N}\sum_{i=1}^N \nabla_\xi J(\theta^i, \xi^j)\,. \tag{56}$$

Algorithm 3 Mean-field Langevin Stochastic Descent-Ascent (MFL-SDA) for RMDPs.

Inputs: initial parameter distributions ν_0,ρ_0, initial N policy and adversary parameters θ_0, ξ_0, learning rates η_1, η_2, Gaussian noise $\{e_k^i\}_{k=1}^K$ with $e_k^i \in \mathbb{R}^{N\times d}$ for $i = 1, 2$.
Outputs: near-optimal parameter distributions ν_K,ρ_K
for $k = 1, \dots, K$ **do**
  Sample trajectories using $\pi_{\theta_{k-1}^i}$ and $f_{\xi_{k-1}^i}$ for all $i = 1, \dots, N$.
  Estimate $\nabla \frac{\delta G}{\delta \nu}[\nu_{k-1}, \rho_{k-1}](\theta_{k-1})$ (e.g. (55)).
  Update $\theta_k \leftarrow \theta_{k-1} - \eta_1 \left(\hat{\nabla} \frac{\delta G}{\delta \nu}[\nu_{k-1}, \rho_{k-1}](\theta_{k-1}) + \tau\theta_{k-1}\right) + \sqrt{2\eta_1\tau} e_k^1$
  Sample trajectories using $\pi_{\theta_{k-1}^i}$ and $f_{\xi_{k-1}^i}$ for all $i = 1, \dots, N$.
  Estimate $\nabla \frac{\delta G}{\delta \rho}[\nu_k, \rho_{k-1}](\xi_{k-1})$ (e.g. (56)).
  Update $\xi_k \leftarrow \xi_{k-1} + \eta_2 \left(\hat{\nabla} \frac{\delta G}{\delta \rho}[\nu_k, \rho_{k-1}](\xi_{k-1}) - \tau\xi_{k-1}\right) + \sqrt{2\eta_2\tau} e_k^2$
end for

MFL-DA assumes the availability of exact gradients while MFL-SDA works under unbiased estimates of the true gradient. Both lead to a regret bound composed of a geometric decay term and a bias term, and differ only in that the bias term of now depends on higher-order moments, which are bounded by Assumption 6, and by the regret being formulated in terms of an expectation over the stochasticity of the gradient. Therefore, for brevity, we only treat the stochastic gradient case here.

The stochastic gradient analysis sets the number of iterations to $K = \mathcal{O}(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon}))$, where the $\varepsilon = \mathcal{O}(\eta_1)$ in the denominator appears due to the time discretisation of Equations (50) and (51) into Algorithm 3. While this number is slightly increased compared to the setting $K = \mathcal{O}(\frac{1}{\varepsilon})$ in the sample-based analysis double-loop algorithm in Section 5.2, this algorithm operates on the space of parameter distributions and may have practical benefits in global optimisation.

The proof relies on the Lyapunov function, which is defined for a fixed $\lambda > 0$ as

$$\begin{aligned}
\mathcal{L}(\nu, \rho) &= \mathcal{L}_1(\nu) + \lambda \mathcal{L}_2(\nu, \rho) \\
\mathcal{L}_1(\nu) &= \max_{\rho' \in \mathbb{P}[\Xi]} E(\nu, \rho') - \min_{\nu \in \mathbb{P}[\Theta]} \max_{\rho \in \mathbb{P}[\Xi]} E(\nu, \rho) \\
\mathcal{L}_2(\nu, \rho) &= \max_{\rho' \in \mathbb{P}[\Xi]} E(\nu, \rho') - E(\nu, \rho)\,,
\end{aligned} \tag{57}$$

and which vanishes to zero for distributions equal to the global optimum (ν^*, ρ^*) in the weak sense.

Theorem 11 (Last-iterate regret of MFL-SDA (Corollary 1 of Liu et al. (2025) restated)) *Let LSI with $\alpha > 0$ hold for the Gibbs distributions $P_\nu[\rho]$ and $P_\rho[\nu]$ as in Assumption 3. Let $\eta_1 \le \frac{1}{C_1}$, $\tau \le \frac{1}{2C_1^2}$, $\eta_2 \le \frac{1}{2\tau\alpha}$, and $\eta_1 = \min\{\lambda, 0.2, \frac{1}{\tau\alpha}\}\tau\alpha\eta_2$. Further, require that $\eta_1 = \mathcal{O}(\frac{\varepsilon\tau^3\alpha^3}{d})$. Then under Assumptions 4 to 6, MFL-SDA obtains a (distributional) last-iterate regret w.r.t. the saddle point of*

$$\tau\left(\mathcal{H}(\nu|\nu^*) - \mathcal{H}(\rho|\rho^*)\right) \le (1 - 2\eta_1\tau\alpha)^K \mathcal{L}(\nu_0, \rho_0) + \mathcal{O}(\varepsilon)\,. \tag{58}$$

Under the aforementioned N-particle approximation, we obtain an $\tilde{\mathcal{O}}(\frac{1}{\varepsilon})$ iteration complexity.

5.3.2 Regret analysis under particle approximation

The convergence guarantee of Liu et al. (2025) applies to the population-level MFL-SDA dynamics. In the implementation, the distributions (ν_k, ρ_k) are replaced by the aforementioned empirical measures (ν_k^N, ρ_k^N). For one-population MFLD minimization, Suzuki et al. (2023) established *uniform-in-time* particle error bounds of the order $\mathcal{O}(1/N)$ in suitable objective/squared metrics under LSI and regularity assumptions. However, their result does not directly apply to our present two-population descent-ascent dynamics. In particular, uniform-in-time propagation-of-chaos for minimax MFLD requires additional stability arguments/assumptions since the coupled saddle dynamics need not be contractive under the assumptions used for the one-population result.

Propagation of chaos refers to the phenomenon that, as the number of particles N increases, any fixed subset of the interacting particles becomes asymptotically independent and distributed according to the population mean-field law. That is, the empirical measure of a set of particles X^i, $i \in [N]$,

$$\mu^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{X^i},$$

converges to the mean-field distribution μ as $N \to \infty$. As shown by Chen et al. (2025) as the analogue for the single population case, this implies that the energy function E to the minimum over the product measure is close to the mean-field optimum, i.e.

$$\inf_{\mu^N} \frac{1}{N} E^N(\mu^N) - E(\mu^*) \leq \frac{C_\tau}{N}\,, \tag{59}$$

where $C_\tau > 0$ is a constant that depends on the regularisation constant τ.

We formulate a similar result as equation (59) in our min-max setting, which allows to make a *conditional* finite-particle sample-complexity statement. In particular, if the empirical two-population MFL-SDA approximation satisfies

$$\sup_k \mathbf{E}\left[\mathcal{L}(\nu_k^N, \rho_k^N) - \mathcal{L}(\nu_k, \rho_k)\right] \leq \frac{C_N}{N^\alpha} \tag{60}$$

for some $\alpha > 0$, then choosing $N = \Theta(\epsilon^{-1/\alpha})$ makes the particle error $\mathcal{O}(\epsilon)$. Since each MFL-SDA iteration uses N^2 policy-adversary pairs and the population iteration complexity is $K = \mathcal{O}(\epsilon^{-1}\log(1/\epsilon))$, the resulting trajectory complexity is $KN^2 = \tilde{\mathcal{O}}(\epsilon^{-1-2/\alpha})$. Particularly, an $\mathcal{O}(1/N)$ particle error would yield $\tilde{\mathcal{O}}(\epsilon^{-3})$ while an $\mathcal{O}(N^{-1/2})$ particle error would yield $\tilde{\mathcal{O}}(\epsilon^{-5})$.

For *finite-time* propagation-of-chaos, note that in existing results (e.g. Theorem 3 of Suzuki et al. (2023)), an additional constant of $C_K \sim e^{-CK}$ is introduced for some $C_K > 0$ depending on the learning rate η and the regularisation τ. Such results require a contractivity/dissipativity assumption to obtain a uniform-in-time result in $\mathcal{O}(1/N)$. Informally, the Langevin regularisations $-\tau\theta$ and $-\tau\xi$ must dominate the destabilizing coupling caused by J. In our setting, the τ terms do give some dissipativity, but the cross terms from $\nabla_\theta J(\theta,\xi)$ and $\nabla_\xi J(\theta,\xi)$ can destroy contraction.

In other words, for MFL-SDA, the dissipativity is a condition on the drift/vector field of the Langevin descent-ascent dynamics. Let

$$z = (\theta, \xi).$$

Then for the population MFL-SDA flow, ignoring measure-dependence for clarity, the drift for regularised descent-ascent is

$$b(z) = \begin{pmatrix} -\nabla_\theta J(\theta,\xi) - \tau\theta \\ \nabla_\xi J(\theta,\xi) - \tau\xi \end{pmatrix}.$$

Dissipativity states that for two points z, z',

$$\langle z - z', b(z) - b(z')\rangle \leq -m\|z - z'\|^2$$

for some $m > 0$. In the gradient flow dynamics, it implies that $\left\|z_k - z_k'\right\|^2 \leq e^{-2mk}\left\|z_0 - z_0'\right\|^2$ by Grönwall's inequality.

Expanding the RHS, we obtain the terms

$$-\langle \theta - \theta', \nabla_\theta J(\theta,\xi) - \nabla_\theta J(\theta',\xi')\rangle + \langle \xi - \xi', \nabla_\xi J(\theta,\xi) - \nabla_\xi J(\theta',\xi')\rangle - \tau\|\theta - \theta'\| - \tau\|\xi - \xi'\|^2.$$

The last two terms are always negative, whereas the first two can oscillate depending on J. This necessitates further investigation and a derivation of stability criterion for this setting.

The precise drift b of the ascent-descent Langevin dynamics would also have measure-dependence, i.e.

$$b_\theta(\theta;\rho) = -\mathbf{E}_{\xi\sim\rho}[\nabla_\theta J(\theta,\xi)] - \tau\theta$$
$$b_\xi(\xi;\nu) = \mathbf{E}_{\theta\sim\nu}[\nabla_\xi J(\theta,\xi)] - \tau\xi.$$

The stability condition for this more precise setting would need to be of the form

$$\langle z - z', b(z;\nu,\rho) - b(z';\nu',\rho')\rangle \leq -m\|z-z'\|^2 + C\left(W_2^2(\nu,\nu') + W_2^2(\rho,\rho')\right) \tag{61}$$

where the first term gives the contraction between the particles and the second term accounts for the fact that the two particles may see different empirical distributions.

Remark 2 *Note that Assumption 5 provides the convex-concave regularity required for the population-level MFL-SDA convergence theorem. However, this is distinct from the joint dissipativity/monotonicity condition needed to obtain uniform-in-time finite-particle propagation of chaos for the two-population particle system. The latter would require additional control of the coupled descent-ascent drift, for example strong monotonicity of the game vector field or dominance of the Langevin regularisation over the cross-interaction constants.*

5.3.3 A note on Mirror Mean-Field Langevin Dynamics

The above MFL-SDA algorithm typically solves an *unconstrained* mean-field Langevin descent-ascent problem in Euclidean ambient space, with feasibility handled by the energy function

$$E(\nu,\rho) = G(\nu,\rho) + \tau\left(\mathcal{H}(\nu|\nu_{\text{ref}}) - \mathcal{H}(\rho|\rho_{\text{ref}})\right)$$

where ν_{ref} and ρ_{ref} denote reference measures (taken to be standard Gaussians). Note that there is no projection operator in Algorithm 3. Given that our ascent measure ρ (parametrised by ξ) is defined to be in a compact set in our original RMDP problem (5), one may need to find an alternate way to incorporate the constraint if the MFL-SDA is to be used. A naive replacement by projection here would (under certain conditions) change the dynamics qualitatively. A hard projection introduces a non-smooth boundary operation, resulting in the inability to use the arguments/techniques in Liu et al. (2025). To this end, we can leverage the techniques in Gu & Kim (2025), which deal with mirror dynamics of mean-field Langevin dynamics (MFLD).

In this case, we can assume that the min-max problem (49) (at least, for the adversary ascent ρ) is performed over measures supported on a *convex* set $X \subset \mathbb{R}^d$. Thus, the optimisation problem reads

$$\min_{\nu\in\mathbb{P}[\Theta]}\max_{\rho\in\mathbb{P}[\Xi]} E(\nu,\rho) = G(\nu,\rho) + \tau\left(\mathcal{H}(\nu|\nu_{\text{ref}}) - \mathcal{H}(\rho|\rho_{\text{ref}})\right).$$

For the ascent, recall the first variation with respect to ρ:

$$\frac{\delta E}{\delta\rho}[\nu,\rho](\xi), \quad h_\rho(\nu,\rho;\xi) := \nabla_\xi \frac{\delta E}{\delta\rho}[\nu,\rho](\xi).$$

We assume the existence of an appropriate Legendre mirror map/barrier function for the constrained set; that is, there exists a $\psi : \mathbb{P}[\Xi] \to \mathbb{R}$. The primal variable at iteration k, $\xi_k = \nabla\psi^*(\tilde{\xi}_k)$ where $\tilde{\xi}$ is the dual variable.

Then the deterministic mirror mean-field ascent step for a constrained particle ξ_k^j at iteration k is

$$\tilde{\xi}_k^j = \nabla\psi(\xi_k^j),$$
$$\tilde{\xi}_{k+1/2}^j = \tilde{\xi}_k^j + \eta_2 h_\rho(\nu_k,\rho_k;\xi_k^j),$$
$$\xi_{k+1}^j = \nabla\psi^*(\xi_{k+1/2}^j).$$

Equivalently, in proximal/Bregman form, we have the particle update

$$\xi_{k+1}^j = \arg\max_{\xi\in\Xi}\left\{\langle h_\rho(\nu_k,\rho_k;\xi_k^j),\xi\rangle - \frac{1}{\eta_2}D_\psi(\xi,\xi_k^j)\right\}$$

where

$$D_\psi(\xi, \xi_k^j) = \psi(\xi) - \psi(\xi_k^j) - \langle \nabla\psi(z_k^j), \xi - \xi_k^j \rangle$$

is the Bregman divergence. Thus, we update in dual coordinates $\tilde{\xi}$ and map it back with $\nabla\psi^* : \mathbb{R}^{d_\xi} \to \Xi$, ensuring feasibility. For more details, we refer to Gu & Kim (2025).

Remark 3 *Certain parametrisations involving squashing functions at the output layer can be practically useful even when the parameter space Ξ is unconstrained. For instance, the parametrisation $g_\xi(t, x) = \tanh(W(t, x)^\mathsf{T} + B)$ with $\xi = (W, B) \in \Xi = \mathbb{R}^{d_x \times (1+d_x)} \times \mathbb{R}^{d_x}$ will obtain deviations from the nominal that are limited to the space $[-1, 1]^{d_x}$ despite the parameters being unbounded.*

6 Experiments

We now turn to an empirical comparison of the different policy gradient estimators for CT-RMDPs. We focus on the double-loop robust policy gradient algorithm (Algorithm 2). Similarly to earlier implementations of double-loop robust policy gradients (Wang et al., 2024; Bossens & Nitanda, 2025), our double-loop algorithms are implemented by using multiple updates per macro-iteration for the policy and the transition kernel, and the inner problem is solved using a projected gradient ascent.

6.1 Robust LQR Neural ODE domain

The first domain of interest is based on a classical linear quadratic control problem but with an added non-linear term to represent the uncertainty set. The nominal dynamics are given by

$$\bar{f}(t, x_t, u_t) = Ax_t + Bu_t\,, \tag{62}$$

for $A \in \mathbb{R}^{d_x \times d_x}$ and $B \in \mathbb{R}^{d_x \times d_u}$, and the running cost is given by the quadratic equation

$$r(t, x_t, u_t) = x_t^\mathsf{T} Q x_t + u_t^\mathsf{T} B u_t\,, \tag{63}$$

for $Q \in \mathbb{R}^{d_x \times d_x}$, and the terminal cost is $R(x_T) = x_T^\mathsf{T} Q x_T \Delta t$. The dynamics of the system under parametrisation ξ are given by the addition of the nominal dynamics and an uncertainty term depending on ξ as in (6). The uncertainty term is implemented as an MLP of the form

$$g_\xi(t, x_t, u_t) = 2(W_2 \tanh(W_1(t, x_t, u_t) + B_1) + B_2)\,, \tag{64}$$

where $\xi \in \Xi = \{(W_1, B_1, W_2, B_2) : \left\|(W, b)\right\|_\infty \leq \phi\}$ represents the weights and biases. All experiments are conducted with $\mathcal{X}, \mathcal{U} = [-5.0, 5.0]^2$ and $\phi = 1.0$. The ODE parameters, including A, B and the starting state, are uniquely determined for each independent run based on samples from Gaussian distributions.

The policy is an MLP that takes the time and state as input and outputs the action. The policy has 2 fully connected hidden layers of size 128, the first of which is followed by dropout regularisation (with drop probability 0.6) and ReLU activation, and the second of which is followed by ReLU activation. Its output layer depends on the type of algorithm. For the pathwise gradient and the deterministic Hamiltonian algorithms, the output layer is a fully connected layer which directly outputs the action through a sigmoid and normalisation. For the deterministic Hamiltonian gradient with exploration, the standard deviation for a Gaussian distribution is defined by an initial log standard deviation of -1.0 and geometric annealing is performed with factor 0.95. For the stochastic Hamiltonian gradient and discrete (REINFORCE) algorithms, the standard deviation is defined through a learnable parameter on the log scale, with range in $[-3.0, 0.0]$.

Effectiveness of the adversary Table 1 shows that both adversarial policy gradients are equally effective in maximizing the cost, with a higher cost compared to zero order optimisation when training for 100 iterations on a randomly initialized policy. As a validation, Table 2 also shows that the empirical maxima observed in the robustness test are well below the costs observed in the adversary test.

Table 1: Comparison of adversaries w.r.t. the mean and standard error of the total cost across 5 independent runs of adversarial optimisation on a randomly initialised policy. For each run, the score is normalised to $[0, 1]$ based on the the cost obtained by the pathwise adversary on that run, such that 1.0 indicates a score on par with the pathwise adversary.

Pathwise Adversary	1.000 ± 0.000
ZO Adversary	0.818 ± 0.041
Hamiltonian Adversary	1.000 ± 0.000

Table 2: Normalized performance (mean ± standard error) on adversarial test (top) and robustness test (bottom) for each discretisation time step (Δt), aggregated across 5 independent runs. The bold entries highlight the best two performing algorithms in the adversary test. The adversary test performs 100 iterations of pathwise gradient ascent against the deterministic evaluation policy obtained from the training; the robustness test runs 50 randomly sampled transition kernels and takes the mean and maximum.

(a) Adversary test.

	$\Delta t = 0.0005$	$\Delta t = 0.001$	$\Delta t = 0.005$	$\Delta t = 0.01$	$\Delta t = 0.05$
Discrete Non-robust	1.303 ± 0.506	1.303 ± 0.506	1.305 ± 0.506	1.308 ± 0.506	1.337 ± 0.506
Discrete Robust	0.974 ± 0.180	0.974 ± 0.180	0.977 ± 0.181	0.981 ± 0.182	1.013 ± 0.188
Stochastic Hamiltonian Non-robust	0.510 ± 0.157	0.510 ± 0.157	0.511 ± 0.157	0.513 ± 0.158	0.528 ± 0.159
Stochastic Hamiltonian Robust	**0.410** ± 0.130	**0.410** ± 0.130	**0.411** ± 0.130	**0.411** ± 0.130	**0.420** ± 0.132
Stochastic Hamiltonian Robust ZO Both	1.123 ± 0.016	1.124 ± 0.016	1.125 ± 0.016	1.127 ± 0.016	1.144 ± 0.017
Stochastic Hamiltonian Robust ZO Inner	0.572 ± 0.112	0.572 ± 0.112	0.574 ± 0.112	0.576 ± 0.112	0.592 ± 0.113
Deterministic Hamiltonian Non-robust	0.550 ± 0.107	0.550 ± 0.107	0.551 ± 0.107	0.553 ± 0.108	0.565 ± 0.110
Deterministic Hamiltonian Robust	0.518 ± 0.108	0.518 ± 0.108	0.520 ± 0.108	0.521 ± 0.108	0.533 ± 0.108
Deterministic Hamiltonian (explore) Non-robust	0.558 ± 0.106	0.558 ± 0.106	0.559 ± 0.107	0.561 ± 0.107	0.574 ± 0.109
Deterministic Hamiltonian (explore) Robust	0.522 ± 0.106	0.523 ± 0.106	0.524 ± 0.106	0.525 ± 0.106	0.535 ± 0.107
Pathwise Non-robust	0.550 ± 0.107	0.550 ± 0.107	0.551 ± 0.108	0.553 ± 0.108	0.565 ± 0.110
Pathwise Robust	**0.434** ± 0.117	**0.434** ± 0.117	**0.435** ± 0.117	**0.436** ± 0.118	**0.445** ± 0.118
Pathwise Robust ZO Both	1.574 ± 0.043	1.574 ± 0.043	1.576 ± 0.043	1.579 ± 0.043	1.603 ± 0.043
Pathwise Robust ZO Inner	0.521 ± 0.114	0.521 ± 0.114	0.522 ± 0.115	0.524 ± 0.115	0.535 ± 0.116

(b) Robustness test.

	$\Delta t = 0.0005$		$\Delta t = 0.001$		$\Delta t = 0.005$		$\Delta t = 0.01$		$\Delta t = 0.05$	
	Mean	Maximum	Mean	Maximum	Mean	Maximum	Mean	Maximum	Mean	Maximum
Discrete Non-robust	0.421 ± 0.222	1.207 ± 0.527	0.421 ± 0.222	1.207 ± 0.527	0.422 ± 0.222	1.207 ± 0.528	0.423 ± 0.222	1.208 ± 0.528	0.431 ± 0.223	1.210 ± 0.533
Discrete Robust	0.316 ± 0.077	0.882 ± 0.213	0.316 ± 0.077	0.882 ± 0.213	0.317 ± 0.077	0.884 ± 0.213	0.319 ± 0.078	0.887 ± 0.214	0.333 ± 0.082	0.907 ± 0.220
Stochastic Hamiltonian Non-robust	0.099 ± 0.016	0.358 ± 0.079	0.099 ± 0.016	0.358 ± 0.079	0.099 ± 0.016	0.358 ± 0.079	0.100 ± 0.016	0.359 ± 0.079	0.103 ± 0.016	0.360 ± 0.078
Stochastic Hamiltonian Robust	0.123 ± 0.017	0.423 ± 0.095	0.123 ± 0.017	0.423 ± 0.095	0.124 ± 0.017	0.421 ± 0.095	0.125 ± 0.017	0.420 ± 0.094	0.134 ± 0.016	0.460 ± 0.075
Stochastic Hamiltonian Robust ZO Both	0.322 ± 0.010	0.999 ± 0.036	0.322 ± 0.010	0.999 ± 0.036	0.323 ± 0.010	1.001 ± 0.036	0.323 ± 0.010	1.003 ± 0.036	0.329 ± 0.011	1.020 ± 0.035
Stochastic Hamiltonian Robust ZO Inner	0.128 ± 0.023	0.400 ± 0.101	0.128 ± 0.023	0.400 ± 0.101	0.129 ± 0.023	0.399 ± 0.101	0.129 ± 0.023	0.398 ± 0.100	0.136 ± 0.023	0.391 ± 0.093
Deterministic Hamiltonian Non-robust	0.094 ± 0.013	0.353 ± 0.072	0.094 ± 0.013	0.353 ± 0.072	0.094 ± 0.013	0.353 ± 0.072	0.095 ± 0.013	0.354 ± 0.072	0.098 ± 0.013	0.356 ± 0.074
Deterministic Hamiltonian Robust	0.116 ± 0.024	0.477 ± 0.112	0.116 ± 0.024	0.477 ± 0.112	0.116 ± 0.024	0.478 ± 0.113	0.117 ± 0.024	0.478 ± 0.113	0.121 ± 0.025	0.484 ± 0.114
Deterministic Hamiltonian (explore) Non-robust	0.097 ± 0.014	0.353 ± 0.075	0.097 ± 0.014	0.353 ± 0.075	0.097 ± 0.014	0.354 ± 0.075	0.098 ± 0.014	0.355 ± 0.075	0.101 ± 0.014	0.362 ± 0.076
Deterministic Hamiltonian (explore) Robust	0.115 ± 0.022	0.459 ± 0.110	0.115 ± 0.022	0.459 ± 0.110	0.115 ± 0.022	0.459 ± 0.110	0.115 ± 0.022	0.460 ± 0.111	0.119 ± 0.022	0.464 ± 0.111
Pathwise Non-robust	0.094 ± 0.013	0.353 ± 0.072	0.094 ± 0.013	0.353 ± 0.072	0.094 ± 0.013	0.353 ± 0.072	0.095 ± 0.013	0.354 ± 0.072	0.099 ± 0.013	0.356 ± 0.074
Pathwise Robust	0.120 ± 0.026	0.465 ± 0.112	0.120 ± 0.026	0.465 ± 0.112	0.120 ± 0.026	0.466 ± 0.112	0.121 ± 0.026	0.466 ± 0.112	0.124 ± 0.026	0.471 ± 0.114
Pathwise Robust ZO Both	0.559 ± 0.019	1.511 ± 0.034	0.559 ± 0.019	1.511 ± 0.034	0.560 ± 0.019	1.513 ± 0.034	0.561 ± 0.019	1.514 ± 0.034	0.570 ± 0.019	1.530 ± 0.032
Pathwise Robust ZO Inner	0.105 ± 0.019	0.434 ± 0.112	0.105 ± 0.019	0.434 ± 0.112	0.105 ± 0.019	0.434 ± 0.112	0.106 ± 0.019	0.435 ± 0.111	0.109 ± 0.019	0.437 ± 0.109

Effectiveness of non-robust and robust policy optimisation. The general test performance of policies can be observed in the adversary test in Table 2, which is based on the hyperparameter settings as shown in Table 4 of Appendix B.1. The pathwise and deterministic Hamiltonian policy gradient algorithms are effective in both non-robust and robust optimisation, and the best performing algorithm is the robust pathwise optimiser. The stochastic Hamiltonian policy gradient algorithm performs best overall. The traditional, discrete-time algorithm (REINFORCE) yields a comparatively low performance. The zero order optimisers generally perform lower, although if only the adversary is zero order, a reasonable test performance on par with non-robust optimisation can be obtained.

Discretisation effects Sample-and-hold training appears to be nearly equivalent to the training without it (i.e. considering factors involving e.g. $\nabla_x \mu_\theta$ to reflect policy changes between time steps). The best score was found in the pathwise robust sample-and-hold based training, perhaps due to it being slightly less complex to learn. Due to the similarity, and slightly improved performance, we only report the sample-and-hold training.

Large improvements are found when comparing continuous-time algorithms to the discrete-time algorithm, even for the nearly equivalent stochastic Hamiltonian gradient algorithm. The large effect is attributed to the continuous-time algorithms taking into account the dynamics in between two time steps to compute the policy gradient more accurately, i.e. based on an accurate approximation of the quadratic cost terms. These effects are likely underestimated because – in the absence of an adversarial gradient for discrete-time algorithms in deterministic dynamics – the inner problem was solved with our deterministic Hamiltonian adversary.

The trained policies themselves do not seem to depend on the Discretisation time step at test time. The larger time step has slightly higher performance due to setting the terminal reward proportional to the time step (consistent with discrete-time algorithms). Beyond this effect, there are only limited effects observed from the test-time time steps. The results shown in Table 2 are based on an adversary trained with the coarsest time step ($\Delta t = 0.05$). However, we also conducted preliminary experiments where the tests separately trained the adversaries for each time step; such adversarial tests were found to be unreliable, with the adversary performing lower than some empirical maxima, which can be attributed to the 100 iterations being insufficient to optimise the much more complex optimisation problems. We also confirm that with larger number of iterations inversely proportional to the time step, the performance still remains similar across time steps.

Table 3: Normalized adversarial test performance of the best policy under different MFL-SDA optimisers. For each Discretisation time step (Δt), the lowest cost is selected out of the $N = 25$ policies and it is aggregated across 5 independent runs (mean $\pm$ standard error). The bold entries highlight the best-performing algorithm for each Discretisation time step. The normalisation is consistent with Table 2, allowing direct comparison.

	$\Delta t = 0.0005$	$\Delta t = 0.001$	$\Delta t = 0.005$	$\Delta t = 0.01$	$\Delta t = 0.05$
Deterministic Hamiltonian Robust	**0.193** $\pm$ 0.034	**0.193** $\pm$ 0.034	**0.195** $\pm$ 0.034	**0.198** $\pm$ 0.034	0.224 $\pm$ 0.034
Pathwise Robust	0.205 $\pm$ 0.048	0.205 $\pm$ 0.048	0.206 $\pm$ 0.048	0.207 $\pm$ 0.048	**0.218** $\pm$ 0.047
Discrete Robust	0.404 $\pm$ 0.120	0.405 $\pm$ 0.120	0.405 $\pm$ 0.120	0.406 $\pm$ 0.119	0.414 $\pm$ 0.114

Distributional optimisation with MFL-SDA In addition to the above-mentioned double-loop algorithm experiments, we also present MFL-SDA based experiments. These experiments are conducted in the same environment with the same adversary test. We conduct experiments with $N = 25$ particles for the policies and the adversaries to form the particle-based gradient estimates as in Equations (55) and (56). To limit computational expense, we only evaluate Discrete Robust, Deterministic Hamiltonian Robust, and Pathwise Robust, and perform only 50 iterations. The hyperparameter settings are shown in Table 4 of Appendix B.1. The results in Table 3 show that the best policy in the set of particles is invariably better-performing than the obtained double-loop algorithm policies, showcasing a solid global optimisation property. Consistent with the previous results, the discrete algorithm is outperformed by the continuous-time algorithms. The deterministic Hamiltonian and pathwise algorithm are comparable in performance with the Hamiltonian algorithm performing slightly better for small time steps and the pathwise performing slightly better for

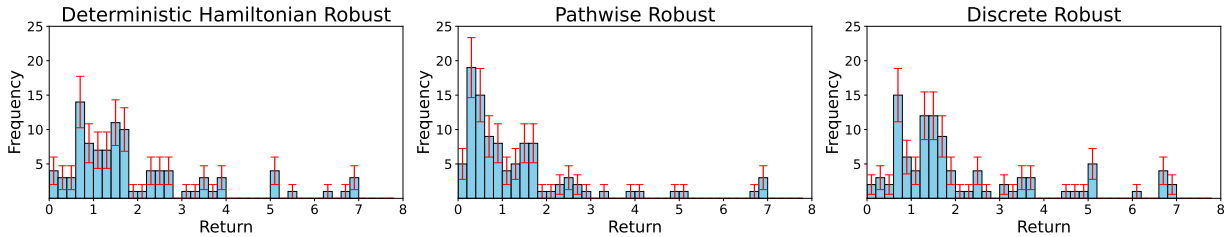

Figure 1: Histogram of the test performance of the policies obtained by the distributional optimisers. The y-axis indicates the total frequency across all policies for all seeds. The error bars indicate $\sqrt{n}$, where $n = \text{seeds} \times \text{N}$, in line with a Poisson distribution assumption. The Discretisation time step is $\Delta t = 0.0005$.

$\Delta t = 0.05$. The test performance varies widely (Figure 1), and we note that more narrow distributions can be obtained using a lower regularisation constant τ. Overall, the pathwise gradient obtains the best scores in few-shot adaptation and in the distributional sense.

7 Conclusion

This paper addresses the problem of continuous-time algorithms for RMDPs, focusing specifically on policy gradient algorithms. We first derive analytical gradient formulas for the policy and the adversary in the context of ODEs, random ODEs, and SDEs. We then derive double-loop optimization and mean-field optimization based regret bounds and investigate double-loop optimization empirically in a robust linear quadratic regulator domain. Through an empirical study, the effectiveness of stochastic and deterministic continuous-time policy gradient algorithms is confirmed, and a large improvement is found compared to discrete-time optimization and zero-order optimization. The empirical study also shows mean-field optimization as a powerful framework to achieve improved robust optimizers. Additional contributions are the tools derived to analyse rectangularity, gradient dominance, discretisation, and properties of double-loop algorithms under general policy parameterizations.

References

Asrar Ahmed and Patrick Jaillet. Regret based Robust Solutions for Uncertain Markov Decision Processes. In *Advances in Neural Information Processing (NeurIPS 2013)*, pp. 1–9, 2013.

Samuel Ainsworth, Kendall Lowrey, John Thickstun, Zaid Harchaoui, and Siddhartha Srinivasa. Faster Policy Learning with Continuous-Time Gradients. In *Proceedings of the Conference on Learning for Dynamics and Control (L4DC 2021)*, pp. 1054–1067, 2021.

Maxim Balashov, Boris Polyak, and Andrey Tremba. Gradient projection and conditional gradient methods for constrained nonconvex minimization. *arXiv preprint arXiv:1906.11580*, 2019.

Marc G Bellemare, Will Dabney, and Rémi Munos. A Distributional Perspective on Reinforcement Learning. In *International Conference on Machine Learning (ICML 2017)*, volume 70, 2017.

Richard Bellman. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957. ISBN 9780691079516.

Riccardo Bonalli and Benoît Bonnet. First-order pontryagin maximum principle for risk-averse stochastic optimal control problems. *SIAM Journal on Control and Optimization*, 61(3):1881–1909, 2023. doi: 10.1137/22M1489137.

David Mark Bossens and Atsushi Nitanda. Mirror descent policy optimisation for robust constrained markov decision processes. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. J2C Certification.

Steven Bradtke and Michael Duff. Reinforcement learning methods for continuous-time markov decision problems. In *Advances in Neural Information Processing Systems (NeurIPS 1994)*, 1994.

Fan Chen, Zhenjie Ren, and Songbo Wang. Uniform-in-time propagation of chaos for mean field langevin dynamics. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 61(4), November 2025. ISSN 0246-0203. doi: 10.1214/24-aihp1499.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS 2018)*, volume 31, 2018.

Samuel N. Cohen, Jackson Hebner, Deqing Jiang, and Justin Sirignano. Neural actor-critic methods for hamilton-jacobi-bellman pdes: Asymptotic analysis and numerical studies. *arXiv preprint arXiv:2507.06428*, pp. 1–41, 2025.

Franck Djeumou, Cyrus Neary, and Ufuk Topcu. How to Learn and Generalize From Three Minutes of Data: Physics-Constrained and Uncertainty-Aware Neural Stochastic Differential Equations. In *Conference on Robot Learning (CoRL 2023)*, 2023.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. In *International Conference on Machine Learning (ICML 2018)*, 2018.

Anming Gu and Juno Kim. Mirror mean-field langevin dynamics. *arXiv preprint arXiv:2505.02621*, 2025.

Ya-Ping Hsieh, Chen Liu, and Volkan Cevher. Finding mixed Nash equilibria of generative adversarial networks. In *International Conference on Machine Learning (ICML 2019)*, pp. 2810–2819, 2019.

Yanwei Jia and Xun Yu Zhou. Policy gradient and actor-critic learning in continuous time and space: Theory and algorithms. *arXiv preprint arXiv:2111.11232*, 2022.

Yanwei Jia and Xun Yu Zhou. Q-learning in continuous time, 2023.

Wanxin Jin, Zhaoran Wang, Zhuoran Yang, and Shaoshuai Mou. Pontryagin differentiable programming: An end-to-end learning and control framework. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7979–7992. Curran Associates, Inc., 2020.

Rudolf Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

Bekzhan Kerimkulov, James-Michael Leahy, David Siska, Lukasz Szpruch, and Yufei Zhang. A fisher-rao gradient flow for entropy-regularised markov decision processes in polish spaces. *arXiv preprint arXiv:2310.02951*, 2025.

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*, 22:1–34, 2021.

Toshinori Kitamura, Arnob Ghosh, Alex Ayoub, Thang D. Chu, and Csaba Szepesvári. Revisiting subgradient dominance in robust mdps: Counterexamples, hardness, and sufficient conditions. *arXiv preprint arXiv:2604.21177*, 2026.

Navdeep Kumar and Kfir Levy. Towards Faster Global Convergence of Robust Policy Gradient Methods. In *European Workshop on Reinforcement Learning (EWRL 2023)*, pp. 1–13, 2023.

Jongmin Lee and Ernest Ryu. Why Policy Gradient Algorithms Work for Undiscounted Total-Reward MDPs. *arXiv preprint arXiv:2510.18340*, 2026.

David Leeftink, Çağatay Yıldız, Steffen Ridderbusch, Max Hinne, and Marcel van Gerven. Optimal control of probabilistic dynamics models via mean hamiltonian minimization, 2025.

Mengmeng Li, Yifan Hu, Daniel Kuhn, and Yan Li. Robust markov decision processes on continuous state spaces. *arXiv preprint arXiv:2605.28706*, 2026a.

Mengmeng Li, Tobias Sutter, and Daniel Kuhn. Policy gradient algorithms for robust mdps with non-rectangular uncertainty sets. 36(1):120–151, 2026b. ISSN 1052-6234. doi: 10.1137/24M1631250.

Qianxiao Li, Long Chen, Cheng Tai, and E. Weinan. Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research*, 18:1–29, 2018.

Zhangyi Liu, Feng Liu, Rui Gao, and Shuang Li. Convergence of Mean-Field Langevin Stochastic Gradient Descent-Ascent for Distributional Minimax Optimization. (1), 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR 2019)*, 2019.

Yulong Lu. Two-scale gradient descent ascent dynamics finds mixed nash equilibria of continuous games: A mean-field perspective. In *International Conference on Machine Learning*, 2023.

Prashant Mehta and Sean Meyn. Q-learning and pontryagin's minimum principle. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 3598–3605, 2009. doi: 10.1109/CDC.2009.5399753.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvári, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning (ICML 2020)*, 2020.

Rémi Munos. Policy gradient in continuous time. *Journal of Machine Learning Research*, 7:771–791, 2006. ISSN 15337928.

Johannes Müller, Semih Çaycı, and Guido Montúfar. Fisher-rao gradient flows of linear programs and state-action natural policy gradients. *arXiv preprint arXiv:2403.19448*, 2025.

Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005. ISSN 0030364X. doi: 10.1287/opre.1050.0216.

Félix Otto and Cédric Villani. Generalization of an inequality by talagrand and links with the logarithmic sobolev inequality. *Journal of Functional Analysis*, 173(2):361–400, 2000. ISSN 0022-1236.

David Pfau, Ian Davies, Diana Borsa, Joao G. M. Araujo, Brendan Tracey, and Hado van Hasselt. Wasserstein policy optimization. *arXiv preprint arXiv:2505.00663*, 2025.

L.S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The mathematical theory of optimal processes.* Wiley, NY, 1962.

Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. DPPO : Diffusion Policy Policy Optimization. In *International Conference on Learning Representations (ICLR 2025)*, pp. 1–42, 2025.

Filippo Santambrogio. Euclidean , metric , and Wasserstein gradient flows : an overview. *Bulletin of Mathematical Sciences*, 7(1):87–154, 2017. ISSN 1664-3615. doi: 10.1007/s13373-017-0101-1.

David Silver, Guy Lever, Nicholas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic Policy Gradient Algorithms. In *International Conference on Machine Learning (ICML 2014)*, Bejing, China, 2014.

Justin Sirignano and Konstantinos Spiliopoulos. Stochastic gradient descent in continuous time. *SIAM Journal on Financial Mathematics*, 8(1):933–961, 2017. doi: 10.1137/17M1126825.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR 2021)*, 2021.

Xihong Su, Marek Petrik, Julien Grand-cl, H E C Paris, and Rue De Lib. Risk-averse Total-reward MDPs with ERM and EVaR. In *AAAI Conference on Artificial Intelligence (AAAI 25)*, pp. 20646–20654, 2025.

Zhongchang Sun. Policy Optimization for Robust Average Cost MDPs. In *Advances in Neural Information Processing (NeurIPS 2024)*, 2024.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning*, volume 3. MIT Press, 2012.

Taiji Suzuki, Denny Wu, and Atsushi Nitanda. Convergence of mean-field langevin dynamics: time-space discretization, stochastic gradient, and variance reduction. In *Advances in Neural Information Processing Systems (NeurIPS 2023)*, volume 36, pp. 15545–15577, 2023.

Bahar Taskesen, Dan A. Iancu, Çağıl Koçyiğit, and Daniel Kuhn. Distributionally robust linear quadratic control. In *Advances in Neural Information Processing Systems (NeurIPS 2023)*, 2023.

Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

E. Vinodh Kumar and Jovitha Jerome. Robust lqr controller design for stabilizing and trajectory tracking of inverted pendulum. *Procedia Engineering*, 64:169–178, 2013. ISSN 1877-7058. International Conference on Design and Manufacturing (IConDM2013).

Qiuhao Wang, Shaohang Xu, Chin Pang Ho, and Marek Petrik. Policy gradient for robust markov decision processes. *arXiv preprint arXiv:2410.22114*, 2024.

Qiuhao Wang, Yuqi Zha, Chin Pang, and Ho Marek. Provable Policy Gradient for Robust Average-Reward MDPs Beyond Rectangularity. In *International Conference on Machine Learning (ICML 2025)*, 2025.

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013. ISSN 0364765X. doi: 10.1287/moor.1120.0566.

Guojian Zhan, Yuxuan Jiang, Jingliang Duan, Shengbo Eben Li, Bo Cheng, and Keqiang Li. Continuous-time policy optimization. In *2023 American Control Conference (ACC)*, pp. 3382–3388, 2023. doi: 10.23919/ACC55779.2023.10156372.

Hanyang Zhao, Wenpin Tang, and David Yao. Policy optimization for continuous reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS 2023)*, 2023.

Mo Zhou and Jianfeng Lu. A policy gradient framework for stochastic optimal control problems with global convergence guarantee. *SIAM Journal on Control and Optimization*, 63(4):2605–2631, 2025. doi: 10.1137/23M1570739.

Xiangxin Zhou, Liang Wang, and Yichi Zhou. Stabilizing policy gradients for stochastic differential equations via consistency with perturbation process, 2024.

Denis Zorba, David Šiška, and Lukasz Szpruch. Convergence of actor-critic for entropy regularised mdps in general action spaces. *arXiv preprint arXiv:2510.14898*, 2025.

A Supporting lemmata

A.1 General lemmata

The discrete-time policy gradient can be derived for random ODEs according to the following lemma.

Lemma 8 (Discrete-time policy gradient) *Let h be the discretisation step size and $t_n := nh$. Recall that we work under a stochastic policy $u_n \sim \pi_\theta(\cdot|t_n, x_n)$ for the discrete cost $J_h(\theta, \xi) = \mathbf{E}\left[\sum_{n=1}^{N-1} r(t_n, x_n, u_n)h + R(x_N)\right]$. Let the trajectory be $\tau = (x_0, u_0, x_1, u_1, \ldots, x_N)$ obtained under a random ODE. Then the discrete-time policy gradient is given by*

$$\nabla_\theta J_h(\theta, \xi) = \mathbf{E}\left[\sum_{n=0}^{N-1} \nabla_\theta \log \pi_\theta(u_n|t_n, x_n) Q_h(t_n, x_n, u_n)\right], \tag{65}$$

where $Q_h(t_n, x_{t_n}, u_n) := \mathbf{E}[G(\tau)|t_n, x_n, u_n]$.

Proof: Under the random ODE assumption of the state dynamics, the only θ-dependence in the trajectory distribution is though the policy, since the state transition is deterministic given u_n. Thus, we can decompose the trajectory density w.r.t. the action product measure as

$$p_\theta(\tau) = d_0(x_0)\Pi_{n=0}^{N-1}\pi_\theta(u_n, t_n, x_n),$$

where d_0 is the initial distribution. Let the total return be $G(\tau) = \sum_{n=0}^{N-1} r(t_n, x_n, u_n) + R(x_N)$. Then,

$$J_h(\theta, \xi) = \int G(\tau)p_\theta(\tau)\,\mathrm{d}\tau.$$

Differentiating w.r.t. θ,

$$\nabla_\theta J_h = \int G(\tau)\nabla_\theta p_\theta(\tau)\,\mathrm{d}\tau = \int G(\tau)p_\theta(\tau)\nabla_\theta \log p_\theta(\tau)\,\mathrm{d}\tau$$

which is simply $\mathbf{E}_\tau[G(\tau)\nabla_\theta \log p_\theta(\tau)]$. It follows that $\nabla_\theta \log p_\theta(\tau) = \sum_{n=0}^{N-1} \nabla_\theta \log \pi_\theta(u_n, t_n, x_n)$, implying

$$\nabla_\theta J_h(\theta, \xi) = \mathbf{E}\left[\sum_{n=0}^{N-1} \nabla_\theta \log \pi_\theta(u_n, t_n, x_n) G(\tau)\right].$$

Now for the policy gradient, we simply replace $G(\tau)$ for a cost-to-go function: $Q_h(t_n, x_{t_n}, u_n) := \mathbf{E}[G(\tau)|t_n, x_{t_n}, u_n]$. By using properties of the conditional expectation, we obtain

$$\mathbf{E}[\nabla_\theta \log \pi_\theta(u_n|t_n, x_n)G(\tau)|t_n, x_{t_n}, u_n] = \nabla_\theta \log \pi_\theta(u_n|t_n, x_{t_n})\mathbf{E}[G(\tau)|t_n, x_{t_n}, u_n],$$

which is equivalent to $\mathbf{E}[\nabla_\theta \log \pi_\theta(u_n|t_n, x_n)Q_h(t_n, x_n, u_n)]$. So the standard discrete-time identity is

$$\nabla_\theta J_h(\theta, \xi) = \mathbf{E}\left[\sum_{n=0}^{N-1} \nabla_\theta \log \pi_\theta(u_n|t_n, x_n) Q_h(t_n, x_n, u_n)\right].$$

□

Below is a standard descent lemma which is convenient in the regret analysis.

Lemma 9 (Standard descent lemma) *Let $J(\theta)$ be L-smooth. Then, for any $0 \leq \eta \leq 1/L$, we have*

$$J(\theta_{t+1}) \leq J(\theta_t) - \frac{\eta}{2}\left\|\nabla J(\theta)\right\|^2 .$$

Proof: This is a well-known result, and we include a brief sketch. Let $J(\theta)$ be L-smooth. Then, for any $\Delta\theta$,

$$J(\theta + \Delta\theta) \leq J(\theta) + \langle \nabla_\theta J(\theta), \Delta\theta \rangle + \frac{L}{2}\|\Delta\theta\|^2 ,$$

using a first-order Taylor expansion with a Lagrangian remainder bounded by $\frac{1}{2}\nabla^2 J(\theta_c)(\Delta\theta)^2 \leq \frac{L}{2}\|\Delta\theta\|^2$ where $\theta_c \in [\theta, \theta + \Delta\theta]$. The result follows from setting $\Delta\theta = -\eta\nabla J(\theta)$ and noting that $\eta \leq 1/L$ implies $1 - L\eta/2 \geq 1/2$. □

The lemma below, obtained from Theorem 4.2 of Wang et al. (2024), shows an upper bound can be established based on the iterates.

Lemma 10 (Upper bound on RMDP regret) *Let Φ be the objective in* (5) *and let $\theta^* \in \arg\min_{\theta\in\Theta} \Phi(\theta)$. Then for any macro-step $k \geq 1$,*

$$\Phi(\theta_k) - \Phi(\theta^*) \leq J(\theta_k, \xi_k) + \epsilon_k - J(\theta^*, \xi_k) , \tag{66}$$

where $\epsilon_k > 0$ is the error tolerance for the inner problem.

Performance difference lemmas are used to bound the value difference between two policies in discounted MDPs. Note that in discrete-time with $\Delta t \to 0$, our formulation is equivalent to the total cost MDP with states after the terminal state being absorbing states with cost 0. Hence we can use the so-called transient performance difference lemma for our formulation.

Lemma 11 (Transient performance difference lemma, Lee & Ryu (2026)) *For a total cost MDP $\mathcal{M} = (x_0, \mathbf{P}, r, \mathcal{X}, \mathcal{U})$, where x_0 is the starting state, $\mathbf{P}$ is the transition kernel, r is the cost function, and $\mathcal{X}$ and $\mathcal{U}$ are the state and action spaces. Let $\mathcal{M}$ have a finite value function in the sense of Assumption 2.1 in Lee & Ryu (2026). Then if $V(x|\pi') = 0$ for all recurrent states, the performance difference of two policies $\pi, \pi' \in \Pi$ is given by*

$$V(x_0|\pi) - V(x_0|\pi') = \sum_{x\in\mathcal{X}} \tilde{d}^{\pi}_{x_0,\mathbf{P}}(x) \sum_{u\in\mathcal{U}} (\pi(u|x) - \pi'(u|x)) Q(x, u|\pi') , \tag{67}$$

where $\tilde{d}^{\pi}_{x_0,p}$ is the transient visitation measure as defined in Definition 2.

Using the above transient visitation measure also leads to recovering results from discounted MDPs, in particular the performance difference lemmas for transition kernels (Wang et al., 2024; Li et al., 2026b).

Lemma 12 (Transient performance difference lemma for transition kernels) *Let $\theta \in \Theta$ and let $\mathbf{P}, \mathbf{P}' \in \mathcal{P}$. Under the conditions and definitions of Lemma 11, we have that*

$$J(\theta, \mathbf{P}) - J(\theta, \mathbf{P}') = \sum_{x\in\mathcal{X}} \tilde{d}^{\pi}_{d_0,\mathbf{P}'}(x) \sum_{(u,x')} \pi(u|x) V(x'|p) \left(\mathbf{P}(x'|x,u) - \mathbf{P}'(x'|x,u)\right) .$$

Proof: Let $\pi = \pi_\theta$. Then the value obtained under π and transition dynamics $p, p' \in \mathcal{P}$ satisfies

$$\begin{aligned}
&V(x|\mathbf{P}) - V(x|\mathbf{P}') \\
&= \sum_{u,x'} V(x'|\mathbf{P})\pi(u|x)\mathbf{P}(x'|x,u) - \sum_{(u,x')} V(x'|\mathbf{P}')\pi(u|x)\mathbf{P}'(x'|x,u) \\
&= \sum_{(u,x')} \pi(u|x)V(x'|\mathbf{P})\left(\mathbf{P}(x'|x,u) - \mathbf{P}'(x'|x,u)\right) + \sum_{(u,x')} \pi(u|x)\mathbf{P}'(x|x,u)(V(x'|\mathbf{P}) - V(x'|\mathbf{P}')) \\
&= \sum_{(u,x')} \pi(u|x)V(x'|\mathbf{P})\left(\mathbf{P}(x'|x,u) - \mathbf{P}'(x'|x,u)\right) + \sum_{(u,x')} \pi(u|x)\mathbf{P}'(x|x,u)(V(x'|\mathbf{P}) - V(x'|\mathbf{P}'))\,.
\end{aligned}$$

Let $u(x) = \sum_{u,x'} \pi(u|x)V(x'|\mathbf{P})\left(\mathbf{P}(x'|x,u) - \mathbf{P}'(x'|x,u)\right)$. The above implies

$$V(\cdot|\mathbf{P}) - V(\cdot|\mathbf{P}') = u + \mathcal{T}^{\pi,\mathbf{P}'}(V(\cdot|\mathbf{P}) - V(\cdot|\mathbf{P}'))$$

such that

$$V(\cdot|\mathbf{P}) - V(\cdot|\mathbf{P}') = (\mathbb{I} - \mathcal{T}^{\pi,\mathbf{P}})^{-1}u$$

and

$$\begin{aligned}
J(\theta,\mathbf{P}) - J(\theta,\mathbf{P}') &= d_0^\intercal(\mathbb{I} - \mathcal{T}^{\pi,\mathbf{P}'})^{-1}u \\
&= \sum_{x\in\mathcal{X}} d_0(x)\tilde{d}^{\pi}_{x,\mathbf{P}'}(x) \sum_{(u,x')} \pi(u|x)V(x'|\mathbf{P})\left(\mathbf{P}(x'|x,u) - \mathbf{P}'(x'|x,u)\right) \\
&= \sum_{x\in\mathcal{X}} \tilde{d}^{\pi}_{d_0,\mathbf{P}'}(x) \sum_{(u,x')} \pi(u|x)V(x'|\mathbf{P})\left(\mathbf{P}(x'|x,u) - \mathbf{P}'(x'|x,u)\right)\,.
\end{aligned}$$

□

The policy gradient for total cost MDPs is given as follows.

Lemma 13 *For a total cost MDP $\mathcal{M} = (d_0, \mathbf{P}, r, \mathcal{X}, \mathcal{U})$ the policy gradient for a directly parametrised $\pi = \theta$ is given by*

$$\partial_{x',u'} J(\theta,\xi) = \tilde{d}^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x')Q(x',u'|\pi_{\theta'})\,. \tag{68}$$

and for a general parametrisation by

$$\nabla_\theta J(\theta,\xi) = \sum_{x\in\mathcal{X}} \tilde{d}^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x) \sum_{u\in\mathcal{U}} \pi_\theta(u|x)Q(x,u|\pi_{\theta'})\frac{\nabla_\theta \pi_\theta(u|x)}{\pi_\theta(u|x)})\,. \tag{69}$$

Proof: Applying the policy gradient definition (e.g. Theorem 4.7 in Lee & Ryu (2026)) to the general parametrisation gives the result in Equation (69). Under the direct parametrisation, we can derive that

$$\begin{aligned}
\partial_{x',u'} J(\theta,\xi) &= \sum_{x\in\mathcal{X}} \tilde{d}^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x) \sum_{u\in\mathcal{U}} \pi_\theta(u|x)Q(x,u|\pi_{\theta'})\frac{\nabla_\theta \pi_\theta(u|x)}{\pi_\theta(u|x)})\mathbb{I}((x,u) = (x',u')) \\
&= \tilde{d}^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x')Q(x',u'|\pi_{\theta'})\,.
\end{aligned}$$

□

The adversary gradient for a direct parametrisation is given by the below lemma.

Lemma 14 (Adversary gradient for total cost MDP) *For a total cost MDP $\mathcal{M} = (d_0, p, r, \mathcal{X}, \mathcal{U})$ with directly parametrised p, we have*

$$\partial_{(x',u',x'')} J(\theta,\mathbf{P}) = \tilde{d}^{\pi_\theta}_{d_0,p}(x')\pi(u'|x')V(x'|p)\,. \tag{70}$$

Proof: Using comparable calculations as in Lemma 12,

$$J(\theta, \mathbf{P}) = \sum_{x \in \mathcal{X}} \tilde{d}^{\pi}_{d_0,p}(x) \sum_{(u,x')} \pi(u|x) V(x'|p) \mathbf{P}(x'|x,u)$$

such that

$$\partial_{(x,u,x')} J(\theta, \mathbf{P}) = \tilde{d}^{\pi}_{d_0,p}(x) \pi(u|x) V(x'|p)\,.$$

□

A.2 Smoothness derivation

Below is a proof of the validity of the L-smoothness assumption under our general assumptions.

Lemma 15 (L-smoothness of the deterministic policy objective) *Let $J : \Theta \to \mathbb{R}^+$ be the expected total cost as in (2) for fixed $\xi \in \Xi$, under the ODE dynamics*

$$\dot{x}_t = f_\xi(t, x_t, u_t), \qquad u_t = \mu_\theta(t, x_t).$$

Assume the following:

1. *f_ξ, r, and R are C^2 in their arguments on a compact invariant set, and their first derivatives are bounded and Lipschitz in x and in u;*
2. *$\mu_\theta(t, x)$ is C^2 in (θ, x);*
3. *μ_θ, $\nabla_\theta \mu_\theta$, and $\nabla_x \mu_\theta$ are uniformly bounded;*
4. *$\nabla_\theta \mu_\theta$ and $\nabla_x \mu_\theta$ are Lipschitz in θ and in x;*
5. *the state equation, the sensitivity equation, and the adjoint equation admit unique solutions on $[0, T]$.*

Then there exists a constant $L > 0$ such that for all $\theta, \theta' \in \Theta$,

$$\|\nabla_\theta J(\theta', \xi) - \nabla_\theta J(\theta, \xi)\| \le L \|\theta' - \theta\|.$$

Hence, $J(\cdot, \xi)$ is L-smooth.

Proof: For fixed ξ, recall that the deterministic adjoint-based policy gradient is

$$\nabla_\theta J(\theta, \xi) = \mathbb{E}\left[\int_0^T \left(\nabla_\theta \mu_\theta(t, x_t) + \nabla_x \mu_\theta(t, x_t)\, z_t\right)^\top \nabla_u H(t, x_t, u_t, p_t)\, dt\right],$$

where

$$H(t, x_t, u_t, p_t) = r(t, x_t, u_t) + p_t^\top f_\xi(t, x_t, u_t),$$

the state sensitivity process $z_t := \nabla_\theta x_t$ satisfies

$$\dot{z}_t = \nabla_x f_\xi(t, x_t, u_t)\, z_t + \nabla_u f_\xi(t, x_t, u_t)\big(\nabla_x \mu_\theta(t, x_t)\, z_t + \nabla_\theta \mu_\theta(t, x_t)\big), \qquad z_0 = 0,$$

and the costate p_t satisfies

$$-\dot{p}_t = \nabla_x r(t, x_t, u_t) + \nabla_x f_\xi(t, x_t, u_t)^\top p_t, \qquad p_T = \nabla_x R(x_T).$$

For convenience, define

$$a_\theta(t) := \nabla_\theta \mu_\theta(t, x_t) + \nabla_x \mu_\theta(t, x_t)\, z_t,$$

and

$$h_\theta(t) := \nabla_u H(t, x_t, u_t, p_t) = \nabla_u r(t, x_t, u_t) + \nabla_u f_\xi(t, x_t, u_t)^\top p_t.$$

Then

$$\nabla_\theta J(\theta, \xi) = \mathbb{E}\left[\int_0^T a_\theta(t)^\top h_\theta(t)\, dt\right].$$

We will show that both $a_\theta(t)$ and $h_\theta(t)$ are Lipschitz in θ, uniformly in t.

Uniform bounds on z_t and p_t. Since the dynamics evolve on a compact invariant set and all derivatives are bounded, there exist constants

$$C_{fx}, \quad C_{fu}, \quad C_{\mu\theta}, \quad C_{\mu x}$$

such that

$$\|\nabla_x f_\xi(t, x_t, u_t)\| \le C_{fx}, \qquad \|\nabla_u f_\xi(t, x_t, u_t)\| \le C_{fu},$$
$$\|\nabla_\theta \mu_\theta(t, x_t)\| \le C_{\mu\theta}, \qquad \|\nabla_x \mu_\theta(t, x_t)\| \le C_{\mu x}.$$

Hence

$$\|\dot{z}_t\| \le \big(C_{fx} + C_{fu}C_{\mu x}\big)\,\|z_t\| + C_{fu}C_{\mu\theta}.$$

By Grönwall's inequality, there exists a constant $C_z > 0$ such that

$$\sup_{t\in[0,T]} \|z_t\| \le C_z.$$

Next, write the costate equation as

$$\dot{p}_t = -A_t p_t - b_t, \qquad p_T = \nabla_x R(x_T),$$

where

$$A_t := \nabla_x f_\xi(t, x_t, u_t)^\top, \qquad b_t := \nabla_x r(t, x_t, u_t).$$

By boundedness of $\nabla_x f_\xi$, $\nabla_x r$, and $\nabla_x R$, there exist constants C_A, C_b, C_R such that

$$\|A_t\| \le C_A, \qquad \|b_t\| \le C_b, \qquad \|p_T\| \le C_R.$$

By variation of constants,

$$p_t = \Phi(t, T)p_T - \int_t^T \Phi(t, s)b_s\, ds,$$

where $\Phi(t, s)$ is the state transition matrix for $\dot{y} = -A_t y$. Using the standard bound

$$\|\Phi(t, s)\| \le e^{C_A(s-t)},$$

we get

$$\|p_t\| \le C_R e^{C_A(T-t)} + \int_t^T e^{C_A(s-t)} C_b\, ds.$$

Thus, there exists a constant $C_p > 0$ such that

$$\sup_{t\in[0,T]} \|p_t\| \le C_p.$$

Lipschitz continuity of the state trajectory in θ. Let x_t and x'_t denote the trajectories induced by θ and θ', respectively, and define

$$\delta x_t := x'_t - x_t, \qquad \delta u_t := u'_t - u_t.$$

Then

$$\dot{\delta x}_t = f_\xi(t, x'_t, u'_t) - f_\xi(t, x_t, u_t).$$

By Lipschitz continuity of f_ξ in (x, u),

$$\|\dot{\delta x}_t\| \le L_{fx}\|\delta x_t\| + L_{fu}\|\delta u_t\|.$$

Since

$$u_t = \mu_\theta(t, x_t), \qquad u'_t = \mu_{\theta'}(t, x'_t),$$

we have

$$\|\delta u_t\| \le \|\mu_{\theta'}(t, x'_t) - \mu_{\theta'}(t, x_t)\| + \|\mu_{\theta'}(t, x_t) - \mu_\theta(t, x_t)\| \le L_{\mu x}\|\delta x_t\| + L_{\mu\theta}\|\theta' - \theta\|.$$

Therefore

$$\|\dot{\delta x_t}\| \leq \big(L_{fx} + L_{fu}L_{\mu x}\big)\,\|\delta x_t\| + L_{fu}L_{\mu\theta}\|\theta' - \theta\|.$$

Since $\delta x_0 = 0$, Grönwall's inequality yields

$$\sup_{t\in[0,T]} \|x'_t - x_t\| \leq C_x\|\theta' - \theta\|$$

for some constant $C_x > 0$.

Lipschitz continuity of the sensitivity process in θ. Let z_t and z'_t denote the sensitivity processes induced by θ and θ', and define

$$\delta z_t := z'_t - z_t.$$

Using the sensitivity equation for z_t and subtracting the two equations, we obtain

$$\begin{aligned}\dot{\delta z_t} = &\Big(\nabla_x f_\xi(t, x'_t, u'_t) - \nabla_x f_\xi(t, x_t, u_t)\Big) z'_t + \nabla_x f_\xi(t, x_t, u_t)\,\delta z_t \\ &+\Big(\nabla_u f_\xi(t, x'_t, u'_t) - \nabla_u f_\xi(t, x_t, u_t)\Big)\Big(\nabla_x \mu_{\theta'}(t, x'_t)\, z'_t + \nabla_\theta \mu_{\theta'}(t, x'_t)\Big) \\ &+\nabla_u f_\xi(t, x_t, u_t)\Big(\nabla_x \mu_{\theta'}(t, x'_t)\, z'_t - \nabla_x \mu_\theta(t, x_t)\, z_t + \nabla_\theta \mu_{\theta'}(t, x'_t) - \nabla_\theta \mu_\theta(t, x_t)\Big).\end{aligned}$$

Using boundedness of z_t, z'_t, boundedness of first derivatives, and Lipschitz continuity of $\nabla_x f_\xi$, $\nabla_u f_\xi$, $\nabla_x \mu_\theta$, and $\nabla_\theta \mu_\theta$, it follows that

$$\|\dot{\delta z_t}\| \leq L_z\|\delta z_t\| + K_z\big(\|x'_t - x_t\| + \|\theta' - \theta\|\big)$$

for some constants $L_z, K_z > 0$. Using Lipschtz continuity of the state and $\delta z_0 = 0$, Grönwall's inequality yields

$$\sup_{t\in[0,T]} \|z'_t - z_t\| \leq C'_z\|\theta' - \theta\|$$

for some constant $C'_z > 0$.

Lipschitz continuity of the costate in θ. Let

$$\delta p_t := p'_t - p_t.$$

Subtracting the two costate equations gives

$$-\dot{\delta p_t} = \Big(\nabla_x r(t, x'_t, u'_t) - \nabla_x r(t, x_t, u_t)\Big) + \Big(\nabla_x f_\xi(t, x'_t, u'_t)^\top p'_t - \nabla_x f_\xi(t, x_t, u_t)^\top p_t\Big).$$

Rearranging,

$$\dot{\delta p_t} = -A_t \delta p_t - c_t,$$

where $A_t = \nabla_x f_\xi(t, x_t, u_t)^\top$ and

$$c_t = \Big(\nabla_x r(t, x'_t, u'_t) - \nabla_x r(t, x_t, u_t)\Big) + \Big(\nabla_x f_\xi(t, x'_t, u'_t) - \nabla_x f_\xi(t, x_t, u_t)\Big)^\top p'_t.$$

By Lipschitz continuity of $\nabla_x r$ and $\nabla_x f_\xi$, and boundedness of p'_t, there exists $K_p > 0$ such that

$$\|c_t\| \leq K_p\big(\|x'_t - x_t\| + \|u'_t - u_t\|\big).$$

Using Lipschitz continuity of the state and the corresponding bound on δu_t,

$$\|c_t\| \leq K'_p\|\theta' - \theta\|$$

for some $K'_p > 0$. Also,

$$\|\delta p_T\| = \|\nabla_x R(x'_T) - \nabla_x R(x_T)\| \leq L_{R,x}\|x'_T - x_T\| \leq L_{R,x}C_x\|\theta' - \theta\|.$$

Applying variation of constants and Grönwall's inequality, we conclude that there exists $C'_p > 0$ such that

$$\sup_{t\in[0,T]} \|p'_t - p_t\| \leq C'_p\|\theta' - \theta\|.$$

Lipschitz continuity of $a_\theta(t)$. Recall

$$a_\theta(t) = \nabla_\theta \mu_\theta(t, x_t) + \nabla_x \mu_\theta(t, x_t)\, z_t.$$

Hence

$$a_{\theta'}(t) - a_\theta(t) = \Big(\nabla_\theta \mu_{\theta'}(t, x'_t) - \nabla_\theta \mu_\theta(t, x_t)\Big) + \Big(\nabla_x \mu_{\theta'}(t, x'_t)\, z'_t - \nabla_x \mu_\theta(t, x_t)\, z_t\Big).$$

Therefore,

$$\|a_{\theta'}(t) - a_\theta(t)\| \le \|\nabla_\theta \mu_{\theta'}(t, x'_t) - \nabla_\theta \mu_\theta(t, x_t)\| + \|\nabla_x \mu_{\theta'}(t, x'_t)\, z'_t - \nabla_x \mu_\theta(t, x_t)\, z_t\|.$$

Split the second term:

$$\|\nabla_x \mu_{\theta'}(t, x'_t)\, z'_t - \nabla_x \mu_\theta(t, x_t)\, z_t\| \le \|\nabla_x \mu_{\theta'}(t, x'_t)\|\, \|z'_t - z_t\| + \|\nabla_x \mu_{\theta'}(t, x'_t) - \nabla_x \mu_\theta(t, x_t)\|\, \|z_t\|.$$

Using boundedness of $\nabla_x \mu_\theta$, Lipschitz continuity of $\nabla_x \mu_\theta$, $\nabla_\theta \mu_\theta$, the state and sensitivity process, there exists $C_a > 0$ such that

$$\sup_{t\in[0,T]} \|a_{\theta'}(t) - a_\theta(t)\| \le C_a \|\theta' - \theta\|.$$

Also, by boundedness,

$$\sup_{t\in[0,T]} \|a_\theta(t)\| \le C_{\mu\theta} + C_{\mu x} C_z =: \bar{C}_a.$$

Lipschitz continuity of $h_\theta(t) = \nabla_u H(t, x_t, u_t, p_t)$. Recall

$$h_\theta(t) = \nabla_u r(t, x_t, u_t) + \nabla_u f_\xi(t, x_t, u_t)^\top p_t.$$

Thus

$$h_{\theta'}(t) - h_\theta(t) = \Big(\nabla_u r(t, x'_t, u'_t) - \nabla_u r(t, x_t, u_t)\Big) + \Big(\nabla_u f_\xi(t, x'_t, u'_t)^\top p'_t - \nabla_u f_\xi(t, x_t, u_t)^\top p_t\Big).$$

Split the second term:

$$\|\nabla_u f_\xi(t, x'_t, u'_t)^\top p'_t - \nabla_u f_\xi(t, x_t, u_t)^\top p_t\| \le \|\nabla_u f_\xi(t, x'_t, u'_t)\|\, \|p'_t - p_t\|$$
$$+ \|\nabla_u f_\xi(t, x'_t, u'_t) - \nabla_u f_\xi(t, x_t, u_t)\|\, \|p_t\|.$$

Using boundedness of $\nabla_u f_\xi$, Lipschitz continuity of $\nabla_u r$ and $\nabla_u f_\xi$, boundedness of p_t, and Lipschitz continuity of the state and costate, there exists $C_h > 0$ such that

$$\sup_{t\in[0,T]} \|h_{\theta'}(t) - h_\theta(t)\| \le C_h \|\theta' - \theta\|.$$

Also,

$$\sup_{t\in[0,T]} \|h_\theta(t)\| \le C_{\nabla_u r} + C_{fu} C_p =: \bar{C}_h.$$

Conclusion. We now estimate the difference of the gradients:

$$\nabla_\theta J(\theta', \xi) - \nabla_\theta J(\theta, \xi) = \mathbb{E}\left[\int_0^T \Big(a_{\theta'}(t)^\top h_{\theta'}(t) - a_\theta(t)^\top h_\theta(t)\Big)\, dt\right].$$

Insert and subtract $a_{\theta'}(t)^\top h_\theta(t)$:

$$a_{\theta'}^\top h_{\theta'} - a_\theta^\top h_\theta = a_{\theta'}^\top (h_{\theta'} - h_\theta) + (a_{\theta'} - a_\theta)^\top h_\theta.$$

Therefore,

$$\|\nabla_\theta J(\theta', \xi) - \nabla_\theta J(\theta, \xi)\| \le \mathbb{E}\left[\int_0^T \Big(\|a_{\theta'}(t)\|\, \|h_{\theta'}(t) - h_\theta(t)\| + \|a_{\theta'}(t) - a_\theta(t)\|\, \|h_\theta(t)\|\Big)\, dt\right].$$

Using the uniform bounds established above,

$$\|\nabla_\theta J(\theta', \xi) - \nabla_\theta J(\theta, \xi)\| \leq T\left(\bar{C}_a C_h + C_a \bar{C}_h\right)\|\theta' - \theta\|.$$

Hence $J(\cdot, \xi)$ is L-smooth with

$$L := T\left(\bar{C}_a C_h + C_a \bar{C}_h\right).$$

□

A.3 Proof of Lemma 1

Recall that the exact trajectory be denoted by $\bar{x}(t) = f_\xi(t, \bar{x}(t), \bar{u}_n)$ for $t \in [t_n, t_{n+1})$, $\bar{x}(t_0) = x_0$ where $\bar{u}_n \sim \pi_\theta(\cdot|t_n, \bar{x}(t_n))$. That is, the exact step is given by

$$\bar{x}(t_{n+1}) = \bar{x}(t_n) + \int_{t_n}^{t_{n+1}} f_\xi(s, \bar{x}(s), \bar{u}_n)\,\mathrm{d}s \tag{71}$$

The above corresponds to a sample-and-hold continuous-time model (i.e. we sample an action and hold it constant on the time interval $[t_n, t_{n+1})$. Essentially, this is a random ODE with piecewise-constant control. The Euler scheme is denoted by

$$x_{n+1} = x_n + h f_\xi(t_n, x_n, u_n), \quad u_n \sim \pi_\theta(\cdot|t_n, x_n). \tag{72}$$

We will bound the grid error $e_n := \bar{x}(t_n) - x(t_n)$ in expectation.

One-step error The error is given by accumulating the difference between Equation (71) and Equation (72) according to

$$e_{n+1} = e_n + \int_{t_n}^{t_{n+1}} f_\xi(s, \bar{x}(s), \bar{u}_n) - f_\xi(t_n, x_n, u_n)\,\mathrm{d}s\,.$$

Splitting the integrand gives the expression

$$\begin{aligned} f_\xi(s, \bar{x}(s), \bar{u}_n) - f_\xi(t_n, x_n, u_n) = &\underbrace{f_\xi(s, \bar{x}(s), \bar{u}_n) - f_\xi(t_n, \bar{x}(t_n), \bar{u}_n)}_{\text{(A) local truncation}} \\ &+ \underbrace{f_\xi(t_n, \bar{x}(t_n), \bar{u}_n) - f_\xi(t_n, x_n, \bar{u}_n)}_{\text{(B) state error}} \\ &+ \underbrace{f_\xi(t_n, x_n, \bar{u}_n) - f_\xi(t_n, x_n, u_n)}_{\text{(C) action mismatch}}\,. \end{aligned}$$

(B) State error By the stated assumptions, $\|(\mathrm{B})\| \leq L_x \|e_n\|$.

(C) Action mismatch Likewise, $\|(\mathrm{C})\| \leq L_u \|\bar{u}_n - u_n\|$. By a coupling argument (using an optimal W_1 coupling), we can choose a coupling and obtain

$$\mathbf{E}\|\bar{u}_n - u_n\| \leq W_1(\pi_\theta(\cdot|t_n, \bar{x}(t_n)), \pi_\theta(\cdot|t_n, x_n)) \leq L_\pi \mathbf{E}\|e_n\|\,. \tag{73}$$

by the aforementioned assumption of the policy being Lipschitz in the state.

(A) Local truncation We have $\|f_\xi(s, \bar{x}(s), \bar{u}_n) - f_\xi(t_n, \bar{x}(t_n), \bar{u}_n)\| \leq L_x \|\bar{x}(s) - \bar{x}(t_n)\| + L_t(s - t_n)$. By the boundedness of f_ξ, we also have $\|\bar{x}(s) - \bar{x}(t_n)\| \leq M(s - t_n)$, implying

$$\|f_\xi(s, \bar{x}(s), \bar{u}_n) - f_\xi(t_n, \bar{x}(t_n), \bar{u}_n)\| \leq (L_x M + L_t)(s - t_n)$$

Integrating $s \in [t_n, t_{n+1})$ yields

$$\int_{t_n}^{t_{n+1}} (\mathrm{A})\, \mathrm{d}s \leq \frac{1}{2}(L_x M + L_t)h^2 \,.$$

We now perform recursion for the error e_n and obtain an explicit global bound. From the three bounds above,

$$\|e_{n+1}\| \leq \|e_n\| + \int_{t_n}^{t_{n+1}} L_x\|e_n\| + L_u\|\bar{u}_n - u_n\| \,\mathrm{d}s + \frac{1}{2}(L_x M + L_t)h^2 \tag{74}$$

$$\leq (1 + L_x h)\|e_n\| + hL_u\|\bar{u}_n - u_n\| + \frac{1}{2}(L_x M + L_t)h^2 \tag{75}$$

Taking the expectation, and bounding the action mismatch as in part (C),

$$\mathbf{E}\|e_{n+1}\| \leq (1 + h\underbrace{(L_x + L_u L_\pi)}_{L_{\mathrm{cl}}})\mathbf{E}\|e_n\| + \frac{1}{2}(L_x M + L_t)h^2 \,. \tag{76}$$

With $e_0 = 0$, by unrolling the linear recursion with a geometric-series sum,

$$\mathbf{E}\|e_n\| \leq \frac{1}{2}(L_x M + L_t)h^2 \sum_{k=0}^{n-1}(1 - L_{\mathrm{cl}}h)^k = \frac{(1 + L_{\mathrm{cl}})^n - 1}{2L_{\mathrm{cl}}}(L_x M + L_t)h.$$

At the horizon $T = t_0 + Nh$ (where $t_0 = 0$ w.l.o.g.),

$$\max_{0 \leq n \leq N} \mathbf{E}\big\|\bar{x}(t_n) - x_n\big\| \leq \underbrace{\frac{e^{L_{\mathrm{cl}}(T - t_0)} - 1}{2L_{\mathrm{cl}}}(L_x M + L_t)}_{C_x(T)}\, h = \mathcal{O}(h) \tag{77}$$

This is the first-order global one-step Euler discretisation error under a general stochastic policy.

A.4 Proof of Lemma 2

Define $D(x, u, \xi) := \int |\mathbf{P}_\xi(x'|x, u) - \mathbf{P_0}(x'|x, u)| \,\mathrm{d}x'$. Denoting $\mathcal{N}(x|y)$ as the density of the normal distribution with mean y and standard deviation $\Delta t\sigma(x, u)$, note that

$$\begin{aligned} D(x, u, \xi) \leq \max_{\xi \in \Xi} D(x, u, \xi) &= \max_{\xi \in \Xi} \int_{\mathcal{X}} |\mathbf{P}_\xi(x'|x, u) - \mathbf{P_0}(x'|x, u)| \,\mathrm{d}x' \\ &= \max_{\xi \in \Xi} \int_{\mathcal{X}} |\mathcal{N}(x'|\Delta t b_\xi(x, u)) - \mathcal{N}(x'|\mathbf{0})| \,\mathrm{d}x' \,. \end{aligned}$$

The two distributions' densities will intersect at at $M(x, u; \xi) = \frac{1}{2}\Delta t b_\xi(x, u)$. Denoting $\Phi(x|y)$ for the cumulative density given mean y, again with standard deviation $\Delta t\sigma(x, u)$, the above leads to an upper bound on the L_1 budget for (x, u):

$$\begin{aligned} \max_{\xi \in \Xi} D(x, u, \xi) &= \max_{\xi \in \Xi} \int_{-\infty}^{M(x,u;\xi)} \big(\mathcal{N}(x'|\mathbf{0}) - \mathcal{N}(x'|\Delta t b_\xi(x, u))\big) \,\mathrm{d}x' \\ &\quad + \int_{M(x,u;\xi)}^{\infty} \big(\mathcal{N}(x'|\Delta t b_\xi(x, u)) - \mathcal{N}(x'|\mathbf{0})\big) \,\mathrm{d}x' \\ &= \max_{\xi \in \Xi} \Phi(M(x, u; \xi)|\mathbf{0}) - \Phi(M(x, u)|\Delta t b_\xi(x, u)) \\ &\quad + \big((1 - \Phi(M(x, u; \xi)|\Delta t b_\xi(x, u))) - (1 - \Phi(M(x, u; \xi)|\mathbf{0}))\big) \\ &= \max_{\xi \in \Xi} 2\left(\Phi(M(x, u; \xi)|\mathbf{0}) - \Phi(M(x, u; \xi)|\Delta t b_\xi(x, u))\right) =: \psi(x, u) \,. \end{aligned}$$

Note then that $\mathcal{P}_\Xi = \bigotimes_{(x,u) \in \mathcal{Z}} \{\mathbf{P}_\xi \in \mathbb{P}[\mathcal{X}]^{\mathcal{X} \times \mathcal{U}} : \xi \in \Xi\}$. Since for any $(x, u) \in \mathcal{Z}$, $\phi(x, u)$ is linearly independent from all other $v \in \mathcal{B}$, a single parameter $\xi \in \Xi$ can be chosen to define the drifts, and thereby

the ℓ_1-maximizing probability distributions, independently for all $z \in \mathcal{Z}$. Consequently, $\mathcal{P}_\Xi$ is a well-defined rectangular uncertainty set with respect to Z.

Now further let the drifts be contained in $B_{x,u}$ for all $(x,u) \in \mathcal{Z}$, and let $(x,u) \in \mathcal{Z}$. Note that the Gaussian likelihood of the $\mathbf{0}$ model will decrease monotonically (exponentially in particular) with increasing ℓ_2 distance. Thus $D(x,u,\xi)$ will increase monotonically with $\left\|\Delta t b_\xi(x,u)\right\|$. Due to monotonicity, we can define $\mathcal{P}_\Xi(x,u)$ with a particular budget for the drift norm, $\rho(x,u)$, such that $\int_{\mathcal{X}} |\mathcal{N}(x'|\Delta t b_\xi(x,u)) - \mathcal{N}(x'|\mathbf{0})| \, \mathrm{d}x' \leq \psi(x,u)$. Since $\mathcal{P}_\Xi(x,u)$ densely covers the L_1 ball, it follows that $\mathcal{P}_\Xi(x,u) = \mathcal{P} = \bigotimes_{(x,u)\in\mathcal{Z}} \{\mathbf{P} \in \mathbb{P}[\mathcal{X}]^{\mathcal{X}\times\mathcal{U}} : D(\mathbf{P}(\cdot|x,u), \mathbf{P_0}(\cdot|x,u)) \leq \psi(x,u)\}$.

A.5 Proof of Lemma 3

The proof follows formalisms within the total MDP framework as in Lemma 11, considering the MDP obtained by the discretised ODE or SDE dynamics parametrised by ξ. Using the notation $Q_h(x,u|\pi) = Q(x,u|\pi)\Delta_t$ and $d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}$ to denote the transient visitation measure from starting distribution d_0 and the transition kernel $\mathbf{P}_\xi$ induced by the discretised dynamics. It is assumed no projection is needed as the policy parameters remain within the interior, either due to regularisation, small learning rates, or policy class definition (e.g. softmax).

a) direct parametrisation. Making use of the transient performance difference lemma (Lemma 11) on the restricted transient visitation measure, the setting $M = \max_{\pi,\pi^*\in\Pi} \left\| \frac{d^{\pi}_{d_0,\mathbf{P}_\xi}}{d^{\pi'}_{d_0,\mathbf{P}_\xi}} \right\|_\infty$, and Equation (68), we have that

$$
\begin{aligned}
[J(\theta,\xi) - J(\theta^*,\xi)]^{1/2} &= \left[\sum_{x\in\mathcal{X}} d^{\pi_{\theta^*}}_{d_0,\mathbf{P}_\xi}(x) \langle Q_h(x,\cdot|\pi_\theta), \pi_\theta(\cdot|x) - \pi_{\theta^*}(\cdot|x) \rangle \right]^{1/2} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(performance difference lemma)} \\
&= \left[\sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x) \frac{d^{\pi_{\theta^*}}_{d_0,\mathbf{P}_\xi}(x)}{d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x)} \langle Q_h(x,\cdot|\pi_\theta), \pi_\theta(\cdot|x) - \pi_{\theta^*}(\cdot|x) \rangle \right]^{1/2} \\
&\leq \left[M \sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x) \langle Q_h(x,\cdot|\pi_\theta), \pi_\theta(\cdot|x) - \pi_{\theta^*}(\cdot|x) \rangle \right]^{1/2}
\end{aligned}
$$

$$
\begin{aligned}
&\leq \left[M \sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x) \sum_{u\in\mathcal{U}} Q_h(x,u|\pi_\theta)\pi_\theta(u|x) \right]^{1/2} && \text{(positive costs)} \\
&\leq \sqrt{M} \max_{x'\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x')^{-1/2} \max_{x\in\mathcal{X}} V_h(x|\pi_\theta)^{-1/2} \sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x) \sum_{u\in\mathcal{U}} \pi_\theta(u|x) Q_h(x,u|\pi_{\theta'}) \\
&= \sqrt{M} \max_{x'\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x')^{-1/2} \max_{x\in\mathcal{X}} V_h(x|\pi_\theta)^{-1/2} \left\|\nabla_\theta J(\theta,\xi)\right\|_1 && \text{(Equation (68))} \\
&\leq \sqrt{M d_\theta} \max_{x'\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x')^{-1/2} \max_{x\in\mathcal{X}} V_h(x|\pi_\theta)^{-1/2} \left\|\nabla_\theta J(\theta,\xi)\right\| \\
&\leq \frac{1}{c} \left\|\nabla_\theta J(\theta,\xi)\right\| \, .
\end{aligned}
$$

It follows that $\frac{\left\|\nabla_\theta J(\theta,\xi)\right\|^2}{2} \geq \frac{c^2}{2} \left(J(\theta,\xi) - J(\theta^*,\xi)\right) = \mu \left(J(\theta,\xi) - J(\theta^*,\xi)\right)$.

b) general parametrisation. Define $c = \min_{x\in\mathcal{X}} \sqrt{d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x)} \min_{x'\in\mathcal{X},a'\in\mathcal{U}} \pi_\theta(u'|x') \min_{x'\in\mathcal{X},u'\in\mathcal{U}} \left\|\nabla_\theta \log(\pi_\theta(u|x))\right\|_\infty$ and set $\mu = c^2/2$. Then

$$
\begin{aligned}
\left[J(\theta,\xi) - J(\theta^*,\xi)\right]^{1/2} &= \left[\sum_{x\in\mathcal{X}} d^{\pi_{\theta^*}}_{d_0,\mathbf{P}_\xi}(x)\langle Q_h(x,\cdot|\pi_\theta), \pi_\theta(\cdot|x) - \pi_{\theta^*}(\cdot|x)\rangle\right]^{1/2} \\
&\qquad \text{(performance difference lemma)} \\
&= \left[\sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x)\frac{d^{\pi_{\theta^*}}_{d_0,\mathbf{P}_\xi}(x)}{d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x)}\langle Q_h(x,\cdot|\pi_\theta), \pi_\theta(\cdot|x) - \pi_{\theta^*}(\cdot|x)\rangle\right]^{1/2} \\
&\leq \left[M\sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x)\langle Q_h(x,\cdot|\pi_\theta), \pi_\theta(\cdot|x) - \pi_{\theta^*}(\cdot|x)\rangle\right]^{1/2} \\
&\leq \left[M\sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x)\sum_{u\in\mathcal{U}} \pi_\theta(\cdot|x) Q_h(x,u|\pi_{\theta'})\right]^{1/2} \qquad \leq M^{1/2}\max_{x'\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x')^{-1} \\
&\qquad \text{(rearranging and positive costs)} \\
&\leq M^{1/2}\max_{x'\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x')^{-1/2}\max_{x\in\mathcal{X}} V_h(x|\pi_\theta)^{-1/2}\mathbf{E}[Q_h(x,u)] \\
&\leq M^{1/2}\max_{x'\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x')^{-1/2}\max_{x\in\mathcal{X}} V_h(x|\pi_\theta)^{-1/2}\left\|\mathbf{E}[Q_h(x,u)\nabla_\theta\log(\pi(a|s))]\right\|_\infty \\
&\qquad \left(\left\|\mathbf{E}[Q_h(x,u)\nabla_\theta\log(\pi(a|s))]\right\|_\infty \geq \mathbf{E}[Q_h(x,u)]\right) \\
&\leq M^{1/2}\max_{x'\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x')^{-1/2}\max_{x\in\mathcal{X}} V_h(x|\pi_\theta)^{-1/2}\left\|\nabla_\theta J(\theta,\xi)\right\|_\infty \qquad \text{(Equation (69))} \\
&\leq M^{1/2}\max_{x'\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x')^{-1/2}\max_{x\in\mathcal{X}} V_h(x|\pi_\theta)^{-1/2}\left\|\nabla_\theta J(\theta,\xi)\right\| \\
&\leq \frac{1}{c}\left\|\nabla_\theta J(\theta,\xi)\right\| \,.
\end{aligned}
$$

It follows that $\frac{\left\|\nabla_\theta J(\theta,\xi)\right\|^2}{2} \geq \frac{c^2}{2}\left[J(\theta,\xi) - J(\theta^*,\xi)\right] = \mu\left[J(\theta,\xi) - J(\theta^*,\xi)\right]$.

A.6 Proof of Lemma 6

Making use of the transient performance difference lemma for transition kernels (Lemma 12) on the restricted transient visitation measure, the setting $M = \max_{\mathbf{P},\mathbf{P}'\in\mathcal{P}} \left\| \frac{d^{\pi}_{d_0,\mathbf{P}}}{d^{\pi}_{d_0,\mathbf{P}'}} \right\|_\infty$, and Lemma 14, we have that

$$
\begin{aligned}
[J(\theta,\mathbf{P}^*) - J(\theta,\mathbf{P})]^{1/2} &= \left[\sum_{x\in\mathcal{X}} d^{\pi}_{d_0,\mathbf{P}^*}(x) \sum_{(u,x')} \pi(u|x) V_h(x'|\mathbf{P}) \left(\mathbf{P}^*(x'|x,u) - \mathbf{P}(x'|x,u) \right) \right]^{1/2} \\
&\qquad\qquad \text{(performance difference lemma)} \\
&= \left[\sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}}(x) \frac{d^{\pi_\theta}_{d_0,\mathbf{P}^*}(x)}{d^{\pi_\theta}_{d_0,\mathbf{P}}(x)} \sum_{u\in\mathcal{U}} \pi(u|x) \langle V_h(x\cdot|p), \mathbf{P}^*(\cdot|x,u) - \mathbf{P}(\cdot|x,u) \rangle \right]^{1/2} \\
&\le \left[M \sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}}(x) \sum_{u\in\mathcal{U}} \pi_\theta(u|x) \sum_{x'\in\mathcal{X}} \mathbf{P}(x'|x,u) V_h(x'|p) \right]^{1/2} \qquad \text{(positive costs)} \\
&\le M^{1/2} \max_{x''} d^{\pi_\theta}_{d_0,\mathbf{P}}(x'')^{-1/2} \max_{x'} V_h(x'|\mathbf{P})^{-1/2} \left(\sum_{x\in\mathcal{X}} d^{\pi_\theta}_{d_0,\mathbf{P}_\xi}(x) \sum_{u\in\mathcal{U}} \pi(u|x) \sum_{x'\in\mathcal{X}} \mathbf{P}(x'|x,u) V_h(x'|\mathbf{P}) \right) \\
&\le M^{1/2} \max_{x''} d^{\pi_\theta}_{d_0,\mathbf{P}}(x'')^{-1/2} \max_{x'} V_h(x'|\mathbf{P})^{-1/2} \left\| \nabla_{\mathbf{P}} J(\theta,\mathbf{P}) \right\|_1 \qquad \text{(Lemma 14)} \\
&\le \sqrt{M d_\xi} \max_{x''} d^{\pi_\theta}_{d_0,\mathbf{P}}(x'')^{-1/2} \max_{x'} V_h(x'|\mathbf{P})^{-1/2} \left\| \nabla_{\mathbf{P}} J(\theta,\mathbf{P}) \right\| \\
&\le \sqrt{M d_\xi} \max_{x''} d^{\pi_\theta}_{d_0,\mathbf{P}}(x'')^{-1/2} \max_{x'} V_h(x'|\mathbf{P})^{-1/2} \beta^{-1} \left\| h_\eta(\mathbf{P}) \right\| \\
&\le \frac{1}{c} \left\| h_\eta(\mathbf{P}) \right\| .
\end{aligned}
$$

Setting $\mu = c^2/2$ and noting that $\xi = \mathbf{P}$ it follows that $\frac{\|\nabla_\theta J(\theta,\xi)\|^2}{2} \ge \frac{c^2}{2}\left(J(\theta,\xi) - J(\theta^*,\xi)\right) = \mu\left(J(\theta,\xi) - J(\theta^*,\xi)\right)$.

A.7 Proof of Lemma 7

Define the step $\Delta\xi = \eta_2(k,t)\nabla_\xi J(\theta_k, \xi^{t-1}_{k-1})$. Then applying Lemma 9 to $-J$ as a cost function with traditional gradient descent, we have that

$$
\begin{aligned}
-J\left(\theta_k, \xi^{t-1}_{k-1} + \Delta\xi\right) &\le -J(\theta_k, \xi^{t-1}_{k-1}) - \langle \nabla_\theta J(\theta_k, \xi^{t-1}_{k-1}), \Delta\xi \rangle + \frac{L}{2}\|\Delta\xi\|^2 \\
&\le -J(\theta_k, \xi^{t-1}_{k-1}) - \eta_2(k-1,t) - 1(1 - \eta_2(k-1,t-1)\frac{L}{2}) \left\| \nabla_\xi J(\theta_k, \xi^{t-1}_{k-1}) \right\|^2 \\
&\le -J(\theta_k, \xi^{t-1}_{k-1}) - \frac{1}{2}\eta_2(k-1,t-1) \left\| \nabla_\xi J(\theta_k, \xi^{t-1}_{k-1}) \right\|^2 \\
&\le -J(\theta_k, \xi^{t-1}_{k-1}) - \eta_2(k,t-1)\mu(\theta_{k-1}) \left(J(\theta_k, \xi^*_{k-1} - J(\theta_k, \xi^{t-1}_{k-1}) \right) ,
\end{aligned}
$$

where the penultimate step follows from the PL condition (Assumption 1). Adding the maximal cost on both sides and switching the iterates, we obtain

$$
J(\theta_k, \xi^*_{k-1}) - J(\theta_k, \xi^t_{k-1}) \le (1 - \eta_2(k-1,t-1)\mu_k)(J(\theta_k, \xi^*_{k-1}) - J(\theta_k, \xi^{t-1}_{k-1})) .
$$

Now consider the case with projected gradient descent. Define the projected gradient mapping

$$
h(k,t) = \frac{1}{\eta_2(k-1,t-1)} \left(\mathrm{proj}_\Xi \left(\xi^{t-1}_{k-1} + \eta_2(k-1,t-1)\nabla_\xi J(\theta_k, \xi^{t-1}_{k-1}) \right) - \xi^{t-1}_{k-1} \right) .
$$

Based on our Modified PL requirement,

$$\frac{1}{2}\left\|h(k,t)\right\|^2 \geq \mu(\theta_k)\left(J(\theta_k,\xi_{k-1}^*) - J(\theta_k,\xi_{k-1}^{t-1})\right) .$$

Following analogous computations to the above-mentioned gradient ascent derivations, but now replacing $\Delta\xi = \eta_2(k-1,t-1)h(k,t)$, we obtain the desired result.

A.8 The number of trajectories

Since the expectation of the policy gradient is unbiased, with a sufficient number of samples it is possible to closely approximate the true gradient, as shown in the lemma below. The proof relies on ℓ_2 norm bounds on the gradients, which is a reasonable assumption based on our general assumption. For instance, using notations from the proof of Lemma 15, the deterministic policy gradient norm is bounded by

$$\left\|\nabla_\theta J(\theta)\right\| \leq \int_0^T \|\partial_\theta a_\theta\|\left\|H(t)\right\| \mathrm{d}t = T\bar{C}_a(C_{ru} + C_{fu}C_p) =: B_\theta \tag{78}$$

Lemma 16 (Number of trajectories) *Let $\varepsilon > 0$ and let $\delta_1, \delta_2 \in (0,1)$ represent the failure probability. Then with probability $1-\delta_1$, the number of trajectories to obtain an ε-precise policy gradient estimate is bounded by $n = \tilde{\mathcal{O}}(\frac{B_\theta}{\varepsilon^2})$, where $B_\theta \geq \left\|\nabla_\theta J(\theta,\xi)\right\|$, and with probability $1-\delta_2$, the number of trajectories to obtain an ε-precise projected adversary gradient estimate is bounded by $n = \tilde{\mathcal{O}}(\frac{B_\xi}{\varepsilon^2})$, where $B_\xi \geq \left\|\nabla_\xi J(\theta,\xi)\right\|$.*

Proof: Sample n independent trajectories and compute the policy/adversary gradient expressions as $X_1,\ldots,X_n$. The arithmetic mean $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$ then represents the estimate of the policy/adversary gradient. We will evaluate the four mentioned estimates, with failure probabilities δ_i, $i = 1,\ldots,4$, and then

The policy gradient. Note that for any $(\theta,\xi) \in \Theta \times \Xi$,

$$\left\|\nabla_\theta J(\theta,\xi)\right\|_\infty \leq \left\|\nabla_\theta J(\theta,\xi)\right\| \leq B_\theta .$$

Via Hoeffding's inequality with X in $[-B_\theta, B_\theta]$, it follows for any dimension $j \in 1,\ldots,d$ that

$$\mathbb{P}(|\bar{X}^j - \mathbf{E}[X^j]| \geq \varepsilon) \leq 2\exp\left(-\frac{2n\varepsilon^2}{2B_\theta}\right) .$$

Applying union bound, it follows that the failure probability is bounded by

$$\sum_{j=1}^{d_\theta} \mathbb{P}(|\bar{X}^j - \mathbf{E}[X^j]| \geq \varepsilon) \leq 2d_\theta \exp\left(-\frac{n\varepsilon^2}{B_\theta}\right) =: \delta_1 ,$$

such that $n = \frac{B_\theta}{\varepsilon^2}\log\left(\frac{2d_\theta}{\delta_1}\right)$.

The adversary. Define the projected gradient mappings

$$\hat{h}_\eta(\xi) = \frac{1}{\eta}\left(\mathrm{proj}_\Xi\left(\xi + \eta\hat{X}\right) - \xi\right)$$

and

$$h_\eta(\xi) = \frac{1}{\eta}\left(\mathrm{proj}_\Xi\left(\xi + \eta\mathbf{E}[X]\right) - \xi\right) .$$

Their distance is upper bounded by

$$
\begin{aligned}
&\left\|\hat{h}_\eta(\xi) - h_\eta(\xi)\right\|_\infty \\
&= \left\|\frac{1}{\eta}\left(\mathrm{proj}_\Xi\left(\xi + \eta\hat{X}\right) - \xi\right) - \frac{1}{\eta}\left(\mathrm{proj}_\Xi\left(\xi + \eta\mathbf{E}[X]\right) - \xi\right)\right\|_\infty \\
&= \frac{1}{\eta}\left\|\mathrm{proj}_\Xi\left(\xi + \eta\hat{X}\right) - \mathrm{proj}_\Xi\left(\xi + \eta\mathbf{E}[X]\right)\right\|_\infty \\
&\leq \left\|\hat{X} - \mathbf{E}[X]\right\|_\infty && \left(\left\|\mathrm{proj}_\Xi(v) - \mathrm{proj}_\Xi(w)\right\| \leq \|v - w\|\right) \\
&\leq \varepsilon\,.
\end{aligned}
$$

Applying Hoeffing's inequality analogous to the derivations for the policy gradient, it follows that the failure probability is bounded by

$$
\sum_{j=1}^{d} \mathbb{P}(|\hat{h}_\eta(\xi)^j - h_\eta(\xi)^j| \geq \varepsilon) \leq \sum_{j=1}^{d} \mathbb{P}(|\hat{X}^j - \mathbf{E}[X^j]| \geq \varepsilon) \leq 2d_\xi \exp\left(-\frac{n\varepsilon^2}{B_\xi}\right) =: \delta_2\,,
$$

such that $n = \frac{B_\xi}{\varepsilon^2}\log\left(\frac{2d_\xi}{\delta_2}\right)$. □

B Further experiment details

B.1 Double-loop algorithm experiments

The experiment hyperparameters with the best performance for the different learners is shown in Table 4; these settings form the basis of Table 2. The AdamW optimizer (Loshchilov & Hutter, 2019) is used for the stochastic policy gradient based optimizers (the discrete-time baseline and the stochastic policy gradient algorithm) as this improved the performance compared to SGD. For deterministic policy gradient based algorithms, both optimizers worked well and so the SGD was chosen. The policy learning rate was tuned separately for the different optimizers, resulting in settings of $1\mathrm{e}^{-3}$ and $3\mathrm{e}^{-4}$ for SGD and AdamW, respectively. For the adversary, an aggressive step size was found to work best, setting it to 0.5 for double-loop algorithms (which take 20 steps per macro-iteration) and to 0.1 for the adversary-only experiments (which take 100 steps). A small amount of noise was added to the inner parameters after each update to prevent getting stuck at the boundary of the uncertainty set. In preliminary experiments, adversaries converged quickly to a particular environment such that the policy only experiences a single environment. To mitigate this issue, we run the experiments with the nominal flag, so that the adversaries restart from the nominal at each macro-iteration, a technique previously used in Wang et al. (2024). The stochastic policies do not seem to be hampered with this issue. For the Stochastic Hamiltonian algorithm, the from nominal flag does not improve performance, so the adversary is continued from the previous iteration. For the discrete-time algorithm, both flags perform poorly, and so we report the result based on the from nominal flag, which has the best robust optimization score.

B.2 MFL-SDA experiments

The hyperparameters for the mean-field optimization experiments are given in Table 5. Many settings were taken from the double-loop experiments. Other settings were modified to account for the changed algorithm and increased computational expense. As no tuning was done, all algorithms are evaluated on SGD with the same $\eta_1 = 1\mathrm{e}^{-3}$ and $\eta_2 = 1\mathrm{e}^{-1}$ learning rates, where the adversary learning rate is set to less than that in double-loop experiments since no near-optimality is required at each iteration. The regularisation constant was set to $\tau = 1$, which leads to $\sqrt{2\eta_1}$ and $\sqrt{2\eta_2}$ noise factors. The number of iterations was reduced to 50.

Table 4: Hyperparameter settings for double-loop algorithm experiments.

Hyperparameter	**Setting**
Policy learning rate (η_1)	$1e^{-3}$ for deterministic and $3e^{-4}$ for stochastic
Policy optimizer	SGD for deterministic and AdamW for stochastic
Policy updates per macro-iteration	4
Macro-iterations (K)	100
Policy gradient norm clipping	10.0
Number of trajectories (n)	1 for deterministic and 10 for stochastic
Inner learning rate (η_2)	$5e^{-1}$ for double-loop training, e^{-1} for adversary-only (test and training)
From nominal	False for Stochastic Hamiltonian, True otherwise
Transition kernel updates per macro-iteration	20
Inner parameter noise	0.001
Inner gradient norm clipping	1.0

Table 5: Hyperparameter settings for MFL-SDA experiments.

Hyperparameter	**Setting**
Policy learning rate (η_1)	$1e^{-3}$
Policy optimizer	SGD
Iterations (K)	50
Policy gradient norm clipping	10.0
Number of trajectories (n)	1 for deterministic and 10 for stochastic
Inner learning rate (η_2)	e^{-1}
Inner gradient norm clipping	1.0
Regularisation constant (τ)	1.0
Number of particles	25