Stochastic Differential Policy Optimization: A Rough Path Approach to Reinforcement Learning

Minh Nguyen MINHPNGUYEN@UTEXAS.EDU
University of Texas at Austin

Chandrajit Bajaj BAJAJ@CS.UTEXAS.EDU
University of Texas at Austin

Editors: Theory of AI for Scientific Computing Workshop

Abstract

We extend Differential Policy Optimization (DPO) to stochastic settings by deriving a discrete-time algorithm from the stochastic Pontryagin Maximum Principle using rough path theory. The framework preserves DPO's operator-based structure while incorporating stochasticity via Brownian and second-level rough path increments. We prove pointwise convergence, establish sample complexity bounds, and derive a regret bound of $O(K^{5/6})$. This provides a theoretically grounded approach to policy learning in continuous-time stochastic control settings.

Keywords: Stochastic Control, Reinforcement Learning, Rough Path Theory, Pontryagin Maximum Principle, Operator Learning.

1. Introduction

Reinforcement learning (RL) is a powerful method that achieves successes across domains such as robotics, biological sciences, and control systems [6, 9, 1]. However, sample complexity and the lack of physical bias prevent reinforcement learning from achieving good results in scientific computing. Model-based methods help mitigate this by improving sample efficiency and incorporating physical models, but they typically require access to analytic reward functions and their derivatives, or the ability to reset to intermediate timesteps [3, 5, 14], making them inapplicable to scientific settings. Differential Policy Optimization (DPO) [13] was recently proposed as a model-free framework that addresses these challenges by solving the differential dual of the continuous-time RL objective. DPO embeds physical structure through a symplectic operator derived from Pontryagin's Maximum Principle [7], and learns this operator directly from trajectory-level reward signals. This formulation avoids reliance on environment gradients or reset capabilities, while retaining the sample efficiency and inductive bias typically associated with model-based methods. In this work, we take a step further and extend DPO to the stochastic control settings. A naïve injection of stochasticity into the deterministic DPO breaks the theoretical connection to optimality conditions in stochastic control. To overcome this, we introduce a new framework that builds on rough path theory to construct differential operators that evolve along stochastic trajectories in a pathwise manner. This results in a stochastic extension of DPO that allows randomness and, at the same time, maintains the advantages of sample efficiency and physical bias for trajectory-level learning.

1.1. Related Works

Continuous-time RL. Given an Markov Decision Process (MDP) with state space $\mathcal{S}$ and action space $\mathcal{A}$, reinforcement learning (RL) aims to maximize the expected cumulative reward:

$$\mathcal{J} = \mathbb{E}_\pi \left[\sum_{k=0}^{H-1} r(s_k, a_k) \right], \quad s_{k+1} \sim \mathbb{P}(s_{k+1}|s_k, a_k), \quad a_k \sim \pi(a_k|s_k). \tag{1}$$

A continuous-time approximation of such formulation leads to the optimal control formulation:

$$\max_\pi \mathbb{E} \left[\int_0^T r(s_t, a_t) dt \right], \quad \text{subject to } \dot{s}_t = f(s_t, a_t). \tag{2}$$

Several works including DPO [13] operates on this continuous-time formulation or its variants.

Differential Policy Optimization (DPO). Pontryagin's Maximum Principle (PMP) [7], through the Hamiltonian function $HF(s, p, a)$, and its reduced version $hf(s, p) := HF(s, p, a^*(s, p))$, introduces the following differential dual system:

$$\begin{bmatrix} \dot{s} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} \nabla_p hf(s, p) \\ -\nabla_s hf(s, p) \end{bmatrix}. \tag{3}$$

By combining (s, p) into $x = (s, p)$, we obtain a dynamics operator:

$$x_{n+1} = G(x_n) := x_n + \Delta S \nabla hf(x_n), \tag{4}$$

where S is the canonical symplectic matrix. DPO reduces finding the optimal policy to approximating the operator G using the trajectory-level reward feedback g. In particular, DPO focuses on the abstract problem of finding/approximating $G : \Omega \to \Omega$:

$$x_0 = x \sim \rho_0, \quad x_1 = G(x_0) = G(x), \tag{5}$$

$$x_2 = G(x_1) = G^{(2)}(x), \cdots, x_{H-1} = G^{(H-1)}(x) \tag{6}$$

Here Ω is a compact domain in $\mathbb{R}^d$, H is the number of steps in an episode, and ρ_0 is the distribution of the starting point x_0. The framework learns a policy G_θ that approximates G by interacting with environment $\mathcal{B}$, which inputs a policy G_θ and outputs the trajectories $(G_\theta^{(k)}(x))_{k=0}^{H-1}$ together with their associated scores $(g(G_\theta^{(k)}(x)))_{k=0}^{H-1}$ for a sample x from distribution ρ_0. Due to Equation (4), we obtain the first-order relation: $G = \mathrm{Id} + \Delta S \nabla g$.

DPO's limitations and our contribution. DPO demonstrates competitive performance across diverse scientific computing tasks, including surface optimization, multiscale grid-based control, and molecular dynamics [13]. However, while we can augment the function G with stochastic terms to introduce exploration behavior, the resulting dynamics do not preserve the structural correspondence to the stochastic control formulation. Thus, in this work, we extend the DPO framework toward a more rigorous stochastic setting, with a proper differential dual for the stochastic control formulation. To achieve this, we investigate stochastic extensions of Pontryagin's Maximum Principle. A common method is to employ backward stochastic differential equations (BSDEs), which yield

powerful theoretical tools but are challenging to lift into an abstract operator framework like DPO. Moreover, the integration of BSDEs with deep learning architectures is computationally burdensome and algorithmically complex. To overcome these difficulties, we draw on recent advances in rough path theory, a more abstract but pathwise approach to stochastic calculus. Rough path theory allows for the construction of differential systems that evolve along irregular, stochastic trajectories in a fully pathwise manner. This aligns naturally with the original DPO perspective, enabling us to formulate a stochastic differential learning framework that is robust to noise and uncertainty.

Organization. The remainder of the paper is organized as follows. Section 2 reviews extensions of Pontryagin's Maximum Principle to the stochastic setting, focusing on formulations based on backward stochastic differential equations (BSDEs) and rough path theory. This discussion motivates and culminates in the update rule Equation (21). Section 3 presents the theoretical framework for our stochastic DPO algorithm based on this update rule. It includes convergence analysis, generalization bounds, and derives corresponding sample complexity and regret guarantees. We conclude in Section 4 with a brief summary.

2. Stochastic Pontryagin Maximum Principle

We now examine extensions from classical Pontryagin Maximum Principle (PMP) to stochastic systems, where the dynamics are governed by stochastic differential equation (SDE).

2.1. Backward Stochastic Differential Equation

One of the most prominent formulations of stochastic Pontryagin Maximum Principle (stochastic PMP) is based on the theory of backward stochastic differential equations (BSDEs), with an example of BSDE being shown below:

$$\begin{aligned} ds_t &= b(s_t, a_t)dt + \sigma(s_t, a_t)dW_t, \quad \text{(Forward)} \\ dp_t &= h(t, s_t, p_t, q_t)dt - q_t dW_t, \quad \text{(Backward)} \\ p_T &= g(s_T) \quad \text{(Terminal)}, \end{aligned} \tag{7}$$

where s_t is the state process, p_t the adjoint (costate) process, q_t arises from the martingale representation, and W_t is a standard Brownian motion.

In earlier versions of stochastic PMP, solving the dual problem involved two coupled FBSDE systems: one for (p_t, q_t) and another for correcting the diffusion through (P_t, Q_t). Under stronger assumptions, such as the concavity of the Hamiltonian, this can however be simplified. In particular, let's consider the general stochastic control problem of maximizing over policy $a = \{a_t\}_t$:

$$J(a) = \mathbb{E}\left[\int_0^T r(t, s_t, a_t)dt + g(s_T)\right], \tag{8}$$

subject to the controlled stochastic dynamics:

$$ds_t = b(s_t, a_t)dt + \sigma(s_t, a_t)dW_t, \quad s_0 = s. \tag{9}$$

Define the stochastic Hamiltonian:

$$H(t, s, a, p, q) = b(s, a)^\top p + \mathrm{tr}(\sigma(s, a)^\top q) + r(t, s, a). \tag{10}$$

Then the stochastic version of PMP [15] can be introduced through the adjoint BSDE system:

$$-dp_t = D_s H(t, s_t, a_t, p_t, q_t)^\top dt - q_t dW_t, \tag{11}$$

$$p_T = D_s g(s_T)^\top. \tag{12}$$

Here the optimality condition is simply $H(t, s_t, a_t, p_t, q_t) = \max_{a \in \mathcal{A}} H(t, s_t, a, p_t, q_t)$ almost surely for all $t \in [0, T]$. In practice, solving such BSDEs is challenging due to the hidden process q_t. Computing q_t often involves auxiliary PDEs or the more complex conditional probability. In the more straight-forward PDE approach [12], if $u(t, s)$ is a function that solves the corresponding PDE, then we can quantify the relation between processes p_t and q_t through the solution function u: $p_t = u(t, s_t)$ and $q_t = \sigma^\top(t, s_t) D_s u(t, s_t)$. However, parameterizing u in a deep learning context is nontrivial, and coupling it with FBSDEs further complicates implementation. These challenges motivate us to seek a more flexible, pathwise alternative.

2.2. Rough Path Theory

Rough path theory, pioneered by Lyons [10, 11], provides a pathwise framework for integrating and solving differential equations driven by irregular signals, such as Brownian motion. It enhances a trajectory $x : [0, T] \to \mathbb{R}^d$ by embedding it in a higher-order structure. Intuitively, a rough path $\mathbf{X}$ is defined by N structures: $(X^1_{s,t}, X^2_{s,t}, \ldots, X^N_{s,t}) \in G$ for a suitable algebraic space G, where each component $X^n_{s,t}$ is heuristically similar to n-th order iterated integral $\int_{s<t_1<\cdots<t_n<t} dx_{t_1} \cdots dx_{t_n}$. More rigorously, an α-Hölder rough path over a Banach space V is a multiplicative functional:

$$\mathbf{X} : \{(s, t) \mid 0 \le s \le t \le T\} \to T^N(V), \tag{13}$$

where $\mathbf{X}_{s,t} = (X^0_{s,t}, X^1_{s,t}, \ldots, X^N_{s,t})$ lies in the truncated tensor algebra $T^N(V)$ with $X^0_{s,t} = 1$. It satisfies the Chen (multiplicativity) identity:

$$\mathbf{X}_{s,u} = \mathbf{X}_{s,t} \otimes \mathbf{X}_{t,u} \quad \text{for all } s \le t \le u, \tag{14}$$

and the α-Hölder continuity condition:

$$\|X^n\|_{n\alpha} := \sup_{0 \le s < t \le T} \frac{|X^n_{s,t}|}{|t - s|^{n\alpha}} < \infty. \tag{15}$$

This formal structure allows one to define integration and differential equations for paths that are too irregular for classical calculus, such as sample paths of Brownian motion. The rough integral against such a path is defined via enhanced integrands $\mathcal{Y}_t$, which is also a multi-level object $(Y^0_t, Y^1_t, \ldots, Y^{N-1}_t)$ with controlled structure. The integral is then approximated by generalized Riemann sums:

$$\int_0^t \mathcal{Y}_s d\mathbf{X}_s := \lim_{|\mathcal{P}| \to 0} \sum_{t_i \in \mathcal{P}} \sum_{k=1}^{N} Y^{k-1}_{t_{i-1}} X^k_{t_{i-1}, t_i}. \tag{16}$$

This allows us to define the rough differential equation (RDE):

$$d\mathcal{Y}_t = F(\mathcal{Y}_t) d\mathbf{X}_t. \tag{17}$$

Lyons' universal limit theorem [10, 11] guarantees continuity of the solution map and provides a stable framework under pathwise irregularity. Notably, for Brownian motion W_t, its rough path $\mathbf{B}_t$

lift includes both the original Brownian motion and a second-level iterated integral [2]. When $i \neq j$, the component $\mathbf{B}_{s,t}^{2;i,j}$ of this lift corresponds to the Stratonovich integral $\int_s^t W_{s,v}^i \circ dW_v^j$, which coincides with the Itô integral due to independency. Here W^i is the i^{th} component of W_t, and $W_{s,v}^i$ is simply the increment $W_v^i - W_s^i$. Additionally, the diagonal term $\mathbf{B}_{s,t}^{2;i,i}$ is the quadratic increment: $\frac{1}{2}(W_{s,t}^i)^2$.

Building on this rough path theory, Lew [8] introduces a maximum principle version for the (rough) stochastic control problem:

$$\max_{a=(a_t)_{t\geq 0}} \mathbb{E}\left[\int_0^T r(t, s_t, a_t)dt\right] \tag{18}$$

$$\text{subject to } s_t = s_0 + \int_0^t b(v, s_v, a_v)d_v + \int_0^t \sigma(v, s_v)d\mathbf{B}_v, \quad t \in [0, T] \tag{19}$$

Here $\mathbf{B}_t$ is the rough path for Brownian motion W_t. More specifically, under a smooth assumption on σ, with the following Hamiltonian $H : [0, T] \times \mathcal{S} \times \mathcal{A} \times \mathbb{R}^{d_\mathcal{A}} \to \mathbb{R}$: $H(t, s, a, p) = p^T b(t, s, a) + r(t, s, a)$, the maximum principle $a_t = \arg\max_{a\in\mathcal{A}} \mathbb{E}[H(t, s_t, a, p_t)]$ holds with the following RDEs for the adjoint process:

$$p_t = p_0 - \int_0^t \frac{\partial H}{\partial s}(v, s_v, a_v, p_v)dv - \int_0^t \left(\frac{\partial \sigma}{\partial s}(s_v)\right)^\top p_v d\mathbf{B}_v, \tag{20}$$

By discretizing and coupling the above RDE system, we obtain the following update rules:

$$\begin{bmatrix} s_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} s_k \\ p_k \end{bmatrix} + \Delta\, \bar{b}\left(\begin{bmatrix} s_k \\ p_k \end{bmatrix}, a_k\right) + \bar{\sigma}\left(\begin{bmatrix} s_k \\ p_k \end{bmatrix}\right) W_{t_k,t_{k+1}} + (\nabla\bar{\sigma}\bar{\sigma})\left(\begin{bmatrix} s_k \\ p_k \end{bmatrix}\right) \mathbb{B}_{t_k,t_{k+1}} \tag{21}$$

where the augmented drift term is $\bar{b}(x, a) = (b(s, a), -\frac{\partial H}{\partial s}(s, a, p))$ with $x = (s, p)$, and the diffusion term is $\bar{\sigma}(s, p) = (\sigma(s), -(\partial_s\sigma(s))^\top p)$. Also, Δ is the discretization time step, and $W_{v,t} = W_t - W_v$ is simply Brownian motion increment. Additionally, $\mathbb{B}$ corresponds to the second-level of the Brownian motion's rough path $\mathbf{B}$, with detailed formula given above.

3. Theoretical framework

Framework and algorithm. Our stochastic algorithm still follows the main flow of DPO, but with significant modification to the dynamic operator, i.e. definition of G and its updates, since the original relation: $G = \mathrm{Id} + \Delta S\nabla g$ is no longer valid (see Algorithm 1). The new "stochastic" update rule Equation (21) allows us to modify the differential RL dynamics to align with the rigorous stochastic differential dual. In particular, we now replace the dynamic operator $x_{k+1} = G(x_k)$ by $x_{k+1} = G(x_k, \xi, \chi)$, where ξ corresponds to the random noise of the (first-level) standard Brownian motion, and χ corresponds to the noise of the second-level term of the rough path $\mathbf{B}_t$. Here x_k is still the composite vector of the state s_k and the adjoint p_k. This leads to the following update equation for the optimal dynamic operator G and its neural network approximator G_{θ_k} at stage k, both from step $n-1$ to step n:

$$G^{(n)}(X) = \mathcal{F}(G^{(n-1)}(X), \xi^n, \chi^n) \tag{22}$$
$$:= G^{(n-1)}(X) + \Delta S \nabla g(G^{(n-1)}(X)) + h_1(G^{(n-1)}(X))\xi^n + h_2(G^{(n-1)}(X))\chi^n$$

And

$$G^{(n)}_{\theta_k}(X) = \mathcal{F}(G^{(n-1)}_{\theta_k}(X), \xi^n, \chi^n) \tag{23}$$
$$:= G^{(n-1)}_{\theta_k}(X) + \Delta S \nabla g_{\theta_k}(G^{(n-1)}_{\theta_k}(X)) + h_1(G^{(n-1)}_{\theta_k}(X))\xi^n + h_2(G^{(n-1)}_{\theta_k}(X))\chi^n$$

Here, functions h_1 and h_2 depends on σ according to Equation (21). From now on, for notational convenience, we use $\xi := \xi^n$ and $\chi := \chi^n$, where step n can be inferred from the context. In our setting, ξ is the discretization of Brownian motion increment and is equal to $W_{t_n} - W_{t_{n-1}}$, while χ is the discretization of the second-level rough path with the definition:

$$\chi_{i,j} := \int_{t_{n-1}}^{t_n} W^i_{t_{n-1},u} dW^j_u \quad \text{for } i \neq j, \qquad \chi_{i,i} = \frac{1}{2}(W^i_{t_{n-1},t_n})^2 \tag{24}$$

Again, W^i is the i^{th} component of multidimensional Brownian motion W.

Algorithm 1 (**Main algorithm**) Stochastic DPO for a generic environment $\mathcal{B}$

Input: a generic environment $\mathcal{B}$, the number of steps per episode H, time step Δ, and the number of samples N_k at stage k with $k \in \overline{1, H-1}$. Here N_k can be chosen based on Theorem 2. We also assume that the hypothesis space for the policy approximator G_{θ_k} in stage k is $\mathcal{H}_k$ for $k \in \overline{1, H}$.
Output: a neural network approximation G_θ that approximates the optimal policy G

1: Initialize an empty replay memory queue $\mathcal{M}$.
2: Initialize $k = 1$ as the current stage and a random scoring function g_{θ_0}. Set the initial policy through Equation (23) via automatic differentiation.
3: **repeat**
4: Use N_k starting points $\{X^i\}_{i=1}^{N_k}$ and previous policy $G_{\theta_{k-1}}$ to query $\mathcal{B}$ and get N_k sample trajectories $\{G^{(n)}_{\theta_{k-1}}(X^i)\}_{n=0}^{H-1}$ together with their scores $\{g(G^{(n)}_{\theta_{k-1}}(X^i))\}_{n=0}^{H-1}$ for $i \in \overline{1, N_k}$
5: Add the labeled samples of the form $(x, y) = (G^{(n)}_{\theta_{k-1}}(X^i), g(G^{(n)}_{\theta_{k-1}}(X^i)))$ to $\mathcal{M}$. Also add labeled samples $(x, y) = (G^{(n)}_{\theta_{k-1}}(X^i), g_{\theta_{k-1}}(G^{(n)}_{\theta_{k-1}}(X^i)))$ for $n \in \overline{1, k-2}$ and $i \in \overline{1, N_k}$ to $\mathcal{M}$. The latter addition step is to ensure that the new policy doesn't deviate from the previous policy on samples on which the previous policy already performs well.
6: Train the neural network $g_{\theta_k} \in \mathcal{H}_k$ at stage k using labeled sample from $\mathcal{M}$ with smooth L^1 loss function [4].
7: Set G_{θ_k} based on Equation (23) via automatic differentiation. Update $k \to k + 1$.
8: **until** $k \geq H$
9: Output $G_{\theta_{H-1}}$ via automatic differentiation baased on Equation (23).

Theoretical analysis. Similar to DPO [13], for this stochastic extension, we will prove a pointwise estimate that enables us to prove the algorithm's convergence, its sample complexity and derive a

regret bound. First of all, we recall from DPO the definition that defines the number of training samples needed to allow derivative approximation transfer:

Definition 1 *[13] For a function $g : \Omega \to \mathbb{R}$, a hypothesis space $\mathcal{H}$ consists of the function $h \in \mathcal{H}$ that approximates g, two positive constants ϵ and δ, we define the function $N(g, \mathcal{H}, \epsilon, \delta)$ to be the number of samples needed such that if we approximate g by $h \in \mathcal{H}$ via $N(g, \mathcal{H}, \epsilon, \delta)$ training samples, then with probability of at least $1 - \delta$, we also have the following estimate on two function gradients:*

$$\|\nabla g(X) - \nabla h(X)\| < \epsilon \tag{25}$$

In other words, we want $N(g, \mathcal{H}, \epsilon, \delta)$ to be large enough so that the original approximation can transfer to the derivative approximation above. If no such $N(g, \mathcal{H}, \epsilon, \delta)$ exists, let $N(g, \mathcal{H}, \epsilon, \delta) = \infty$.

This definition will be used to derive the number of samples needed for Algorithm 1. Below is the pointwise convergence theorem for Algorithm 1 and its proof.

Theorem 2 *Suppose that we are given a threshold error ϵ, a probability threshold δ, and a number of steps per episode H. Assume that $\{N_k\}_{k=1}^{H-1}$ is the sequence of numbers of samples used at each stage in Algorithm 1 (Stochastic DPO) so that:*

$$N_1 = N(g, \mathcal{H}_1, \epsilon, \delta), \tag{26}$$

$$N_k = \max\{N(g_{\theta_{k-1}}, \mathcal{H}_k, \epsilon, \delta/(2(k-1))), N(g, \mathcal{H}_k, \epsilon, \delta/(2(k-1)))\} \text{ for } k \in \overline{2, H-1} \tag{27}$$

We further assume that there exists a constant $L > 0$ such that relevant functions such as $h_1, h_2, \nabla g$, and the policy neural network approximator ∇g_{θ_k} at step k with regularized parameters have their Lipschitz constants at most L for each $k \in \overline{1, H}$. Then, for a general starting point X, with probability at least $1 - \delta$, the following generalization bound for the trained policy G_{θ_k} holds for all $k \in \overline{1, H-1}$:

$$\mathbb{E}_{X,\xi,\chi}\|G_{\theta_k}^{(n)}(X) - G^{(n)}(X)\| < \frac{C^n}{C-1}(C\Delta + n\Delta^2 L)\epsilon \text{ for all } 1 \le n \le k \tag{28}$$

Here $C = 1 + L\Delta + L\sqrt{\Delta}$. Note that when $N_k \to \infty$, the errors approach 0 uniformly for all n given a finite terminal time T.

Proof . We use the notation H and H_k for ∇g and ∇g_{θ_k} respectively. First, g is approximated by $g_{\theta_{k+1}} \in \mathcal{H}_{k+1}$ on sample points $\{G_{\theta_k}^{(n)}(X^i)\}_{i=1}^{N_{k+1}}$ with $n \in \overline{1, k-1}$. Definition of $N_{k+1} \ge N(g, \mathcal{H}_{k+1}, \epsilon, \delta_k/(2k))$ allows derivative approximation transfer so that for a general starting point X, with probability of at least $1 - \delta_k/(2k)$, the following estimate holds:

$$\mathbb{E}_X\|H_{k+1}(G_{\theta_k}^{(n)}(X)) - H(G_{\theta_k}^{(n)}(X))\| < \epsilon \tag{29}$$

Second, $g_{\theta_{k+1}}$ is trained to approximate g_{θ_k} to ensure that the updated policy doesn't deviate too much from current policy. For $n \in \overline{1, k-1}$, $g_{\theta_{k+1}} \in \mathcal{H}_{k+1}$ approximates g_{θ_k} on N_{k+1} samples of the form $G_{\theta_k}^{(n)}(X^i)$ for $i \in \{1, \cdots, N_{k+1}\}$. Since $N_{k+1} \ge N(g_{\theta_k}, \mathcal{H}_{k+1}, \epsilon, \delta_k/(2k))$ allows derivative approximation transfer, for probability of at least $1 - \delta_k/(2k)$:

$$\mathbb{E}_X\|H_{k+1}(G_{\theta_k}^{(n)}(X)) - H_k(G_{\theta_k}^{(n)}(X))\| < \epsilon. \tag{30}$$

Next, for notational convenience, define the following errors:

$$e_k^n = \mathbb{E}_{X,\xi,\chi}\|G_{\theta_k}^{(n)}(X) - G^{(n)}(X)\| \tag{31}$$

$$e_{k,k-1}^n = \mathbb{E}_{X,\xi,\chi}\|G_{\theta_k}^{(n)}(X) - G_{\theta_{k-1}}^{(n)}(X)\| \tag{32}$$

We now bound $\mathbb{E}_{X,\xi,\chi}\|G_{\theta_{k+1}}^{(n)}(X) - G^{(n)}(X)\|$ by taking the difference between Equation (22) and Equation (23):

$$\begin{aligned}\|G_{\theta_{k+1}}^{(n)}(X) - G^{(n)}(X)\| &\leq \|G_{\theta_{k+1}}^{(n-1)}(X) - G^{(n-1)}(X)\| \\ &+ \Delta\|H_{k+1}(G_{\theta_{k+1}}^{(n-1)}(X)) - H(G^{(n-1)}(X))\| \\ &+ \|h_1(G_{\theta_{k+1}}^{(n-1)}(X)) - h_1(G^{(n-1)}(X))\|\|\xi\| \\ &+ \|h_2(G_{\theta_{k+1}}^{(n-1)}(X)) - h_2(G^{(n-1)}(X))\|\|\chi\| \\ &\leq \|G_{\theta_{k+1}}^{(n-1)}(X) - G^{(n-1)}(X)\| \\ &\quad + \Delta\|H_{k+1}(G_{\theta_{k+1}}^{(n-1)}(X)) - H(G^{(n-1)}(X))\| \\ &\quad + (L_{h_1}\|\xi\| + L_{h_2}\|\chi\|)\|G_{\theta_{k+1}}^{(n-1)}(X) - G^{(n-1)}(X)\| \\ &= T_1 + T_2 + T_3\end{aligned} \tag{33}$$

Here we denote L_{h_1} and L_{h_2} as Lipschitz constant of h_1 and h_2 and by theorem assumption they are also at most L. For the second term T_2, we estimate through the following 3-term decomposition:

$$\begin{aligned}&\|H_{k+1}(G_{\theta_{k+1}}^{(n-1)}(X)) - H(G^{(n-1)}(X))\| \leq \|H_{k+1}(G_{\theta_{k+1}}^{(n-1)}(X)) - H_{k+1}(G_{\theta_k}^{(n-1)}(X))\| \\ &+ \|H_{k+1}(G_{\theta_k}^{(n-1)}(X)) - H(G_{\theta_k}^{(n-1)}(X))\| + \|H(G_{\theta_k}^{(n-1)}(X)) - H(G^{(n-1)}(X))\| \\ &\leq L\|G_{\theta_{k+1}}^{(n-1)}(X) - G_{\theta_k}^{(n-1)}(X)\| + \epsilon + L\|G_{\theta_k}^{(n-1)}(X) - G^{(n-1)}(X)\|\end{aligned} \tag{34}$$

The middle inequality holds with probability of at least $1 - \delta/(2k)$ thanks to the first paragraph. Now combining with two (easier terms) T_1 and T_3, and then taking expectation over X, ξ and χ. Since $\mathbb{E}\|\xi\|$ and $\mathbb{E}\|\chi\|$ correspond to Gaussian random variables with standard deviations that scale with $\sqrt{\Delta}$ and Δ respectively, for $n \in \overline{1,k}$, with probability of at least $1 - \delta/(2k)$, we have the estimate:

$$e_{k+1}^n \leq e_{k+1}^{n-1} + \Delta(Le_{k+1,k}^{n-1} + \epsilon + Le_k^{n-1}) + L\sqrt{\Delta}e_{k+1}^{n-1} \tag{35}$$

By a similar estimate but with a two-term decomposition for the middle inequality, with probability of at least $1 - \delta/(2k)$, for $n \in \overline{1,k}$, we obtain estimate for $\mathbb{E}_{X,\xi,\chi}\|G_{\theta_{k+1}}^{(n)} - G_{\theta_k}^{(n)}\|$:

$$e_{k+1,k}^n \leq e_{k+1,k}^{n-1} + \Delta(Le_{k+1,k}^{n-1} + \epsilon) + L\sqrt{\Delta}e_{k+1,k}^{n-1} \tag{36}$$

Hence, with probability of at least $1 - \delta$, Equation (35) and Equation (36) holds for all $n \in \overline{1,k}$. By induction, we have $e_{k+1,k}^n \leq \dfrac{\Delta\epsilon(C^n - 1)}{C - 1}$. Now we also prove by induction on n that for any $n \in \overline{1,k}$, $e_k^n \leq u_n$, where u_n is the sequence that satisfy $u_n = (\alpha + \beta)u_{n-1} + \gamma_n = Cu_{n-1} + \gamma_n$,

where $\alpha = (1 + L\sqrt{\Delta}), \beta = L\Delta$, and $\gamma_n = \Delta L \dfrac{\Delta\epsilon(C^n - 1)}{C - 1} + \Delta\epsilon$. Indeed, for the induction step, Equation (35) together with the previous bound on two-indices term $e^n_{k+1,k}$ yields:

$$e^n_{k+1} \leq \alpha e^{n-1}_{k+1} + \beta e^{n-1}_k + \Delta L e^{n-1}_{k+1,k} + \Delta\epsilon \leq \alpha u_{n-1} + \beta u_{n-1} + \gamma_n \tag{37}$$

Now, by simple induction, $u_n = \sum_{i=0}^n C^i \gamma_{n-i}$, and hence u_n is upper-bounded by:

$$\begin{aligned} u_n &= \Delta\epsilon \left(\sum_{i=0}^n C^i \right) + \Delta^2 L\epsilon \left(\sum_{i=1}^n C^i \frac{C^{n-i} - 1}{C - 1} \right) \\ &\leq \Delta\epsilon \frac{C^{n+1}}{C - 1} + \Delta^2 L\epsilon \sum_{i=1}^n C^i \frac{C^{n-i}}{C - 1} = \frac{C^n}{C - 1}(C\Delta + n\Delta^2 L)\epsilon \end{aligned} \tag{38}$$

Therefore, we obtain Equation (28) as needed. ■

Sample complexity. Similar to DPO [13], the pointwise estimates for the stochastic DPO version in Theorem 2 allow us to explicitly state the number of training episodes required for two scenarios considered in this work: one works with general neural network approximators and the other with more restricted (weakly convex and linearly bounded) functions. Detailed definitions of weakly convex and linearly bounded are given in [13]. Furthermore, DPO [13] shows that $N(g, \mathcal{H}, \epsilon, \delta) = O(\epsilon^{-(2d+4)})$ for general case, and $O(\epsilon^{-6})$ for restricted case. Hence, the following corollaries regarding sample complexity for our stochastic extension also holds through similar proofs:

Corollary 3 *In Algorithm 1 (Stochastic DPO), suppose we are given fixed step size and fixed number of steps per episode H. Further assume that for all $k \in \overline{1, H-1}$, $\mathcal{H}_k$ is the same everywhere and is the hypothesis space $\mathcal{H}$ consisting of neural network approximators with bounded weights and biases. Then with the sequence of numbers of training episodes $N_k = O(\epsilon^{-(2d+4)})$, the pointwise estimates Equation (28) hold.*

Corollary 4 *If in Algorithm 1 (Stochastic DPO), $\mathcal{H}_k$ is the **special** hypothesis subspace consisting of $h \in \mathcal{H}_k$ so that $h - g$ and $h - g_{\theta_{k-1}}$ are both p-weakly convex and linearly bounded instead. Then with $N_k = O(\epsilon^{-6})$, we obtain Equation (28).*

Regret bound. For a given policy π, define $V_\pi(s) := \mathbb{E}_{a,s_1,\cdots}\left[\sum_{k=0}^{H-1} r(s_k, a_k) | s_0 = s\right]$ and optimal value function $V(s) := \arg\max_\pi V_\pi(s)$. Suppose K episodes are used during the training process and suppose a policy π^k is applied at the beginning of the k-th episode with the starting state s^k for $k \in \overline{1, K}$. Then **Regret** is defined as the following function of the number of episodes K:

$$\textbf{Regret}(K) = \sum_{k=1}^K (V(s^k) - V_{\pi^k}(s^k)) \tag{39}$$

The above corollaries (3 and 4) result in the regret bound estimate below directly:

Corollary 5 *Suppose that number of steps per episode H is fixed and relatively small. If in Algorithm 1 (Stochastic DPO), the number of training samples N_k has the scale of $O(\epsilon^{-\mu})$, the **Regret** for stochastic DPO is upper-bounded by $O(K^{(\mu-1)/\mu})$. In other words, we obtain a regret bound of $O(K^{(2d+3)/(2d+4)})$ for regular and $O(K^{5/6})$ for restricted hypothesis spaces respectively.*

4. Conclusion

We present a stochastic extension of DPO by embedding rough path formulations of the stochastic PMP into an operator-based RL framework. Theoretical contributions include pointwise convergence, sample complexity estimates, and a regret bound under stochastic dynamics. This work bridges trajectory-level reinforcement learning and stochastic control theory with rigorous guarantees.

Acknowledgments

This research was supported in part by a grant from the Peter O'Donnell Foundation, the Michael J. Fox Foundation, Jim Holland-Backcountry Foundation to support AI in Parkinson, and in part from a grant from the Army Research Office accomplished under Cooperative Agreement Number W911NF-19-2-0333.

References

[1] David Biagioni, Xiangyu Zhang, Dylan Wald, Deepthi Vaidhynathan, Rohit Chintala, Jennifer King, and Ahmed S. Zamzam. Powergridworld: a framework for multi-agent reinforcement learning in power systems. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, e-Energy '22, page 565–570, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393973.

[2] Laure Coutin and Zhongmin Qian. Stochastic analysis, rough path analysis and fractional brownian motions. *Probab. Theory Relat. Fields*, 122(1):108–140, 2002.

[3] Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472. Citeseer, 2011.

[4] Ross Girshick. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*, 2015.

[5] Nicolas Heess, Greg Wayne, David Silver, Timothy Lillicrap, Yuval Tassa, and Tom Erez. Learning continuous control policies by stochastic value gradients. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2944–2952, Cambridge, MA, USA, 2015. MIT Press.

[6] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Scalable deep reinforcement learning for vision-based robotic manipulation. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 651–673, Zurich, Switzerland, 29–31 Oct 2018. PMLR.

[7] Donald E Kirk. *Optimal Control Theory: An Introduction*. Prentice-Hall, London, England, 1971.

[8] Thomas Lew. Rough stochastic pontryagin maximum principle and an indirect shooting method. *arXiv preprint arXiv:2502.06726*, 2025.

[9] Isaac D. Lutz, Shunzhi Wang, Christoffer Norn, Alexis Courbet, Andrew J. Borst, Yan Ting Zhao, Annie Dosey, Longxing Cao, Jinwei Xu, Elizabeth M. Leaf, Catherine Treichel, Patrisia Litvicov, Zhe Li, Alexander D. Goodson, Paula Rivera-Sánchez, Ana-Maria Bratovianu, Minkyung Baek, Neil P. King, Hannele Ruohola-Baker, and David Baker. Top-down design of protein architectures with reinforcement learning. *Science*, 380(6642):266–273, 2023.

[10] T J Lyons. Differential equations driven by rough signals. i. an extension of an inequality of L. *C. Young. Math. Res. Lett*, 1(4):451–464, 1994.

[11] Terry J Lyons. Differential equations driven by rough signals. *Bibl. Rev. Mat. Iberoamericana*, 14(2):215–310, 1998.

[12] G N Milstein and M V Tretyakov. Numerical algorithms for forward-backward stochastic differential equations. *SIAM J. Sci. Comput.*, 28(2):561–582, 2006.

[13] Minh Nguyen and Chandrajit Bajaj. Dpo: A differential and pointwise control approach to reinforcement learning. *arXiv preprint arXiv:2404.15617*, 2025.

[14] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913, 2012.

[15] Jiongmin Yong and Xun Yu Zhou. *Stochastic controls: Hamiltonian systems and HJB equations*. Springer, New York, NY, 1999 edition, 2012.