A Differential and Pointwise Control Approach to Reinforcement Learning

Minh Nguyen Chandrajit Bajaj

University of Texas at Austin

November 6, 2025

Table of Content

Motivation

- **Reinforcement Learning (RL)** made strong progress in domains like robotics, biology, and language. But in **scientific applications with limited data**, RL struggles due to:
  1. low sample efficiency,
  2. weak alignment with physical laws, and
  3. limited theoretical guarantees.
- Even with reward shaping, **Model-free RL** struggles with sample efficiency and lacks built-in physics priors.
- **Model-based RL** can be more sample-efficient, but usually requires either:
  - access to exact reward functionals and/or their gradients, or
  - the ability to restart or modify trajectories mid-run.
- New approach: RL $\Rightarrow$ **continuous-time control formulation** $\Rightarrow$ Hamiltonian **differential dual** $\Rightarrow$ final algorithm to solve this dual.
  $\Rightarrow$ **physics priors**, sample-efficient policy updates, and **pointwise learning**.

Physics Intuition: From Newton to Hamiltonian Dual Control

- Newton's law $F = m\ddot{s}$ describes motion via *forces*.
- Lagrangian mechanics reformulates it through *energies* and the *principle of stationary action*:

$$\mathcal{S}[s] = \int_0^T \mathcal{L}(s, \dot{s}, t)\, dt, \quad \frac{\partial \mathcal{L}}{\partial s} = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{s}} \quad \text{(Euler-Lagrange)}. \tag{1}$$

  For $\mathcal{L} = \frac{1}{2}m\|\dot{s}\|^2 - \mathcal{V}(s)$, this reduces to Newton's second law $m\ddot{s} = -\nabla\mathcal{V}(s)$.
- Viewing velocity as control, $a = \dot{s}$, Lagrangian mechanics is a *continuous-time control problem*:

$$V(s, t) = \max_{a(\cdot)} \int_t^T -\mathcal{L}(w, a, u)\, du \quad \text{s.t. } \dot{w} = a, \ w(t) = s. \tag{2}$$

- Hamiltonian mechanics is the dual of Lagrangian mechanics, where value gradients act as momenta and dynamics unfold through symplectic flow.
- Through control theory, our differential-learning duality generalizes this physics correspondence and provides the bridge to continuous-time RL.

Revisit to TD error

For $s' = s + \Delta_t f(s,a)$, first-order expansion with $\Delta_t = 1$:

$$\underbrace{r(s,a) + V(s') - V(s)}_{\text{TD error}} \approx r(s,a) + f(s,a)\frac{\partial V}{\partial s}(s) = -\mathcal{H}\left(s, -\tfrac{\partial V}{\partial s}(s), a\right) \quad (3)$$

- The critic's local TD signal is (minus) the Hamiltonian at the value gradient.
- In the $\Delta_t \to 0$ limit: coincides with continuous-time q-function (Jia and Zhou 2023).
- Suggests *local*, physics-aligned learning targets.

Differential Control Formulation

From discrete reward to continuous control:

$$\max_{\pi} \mathbb{E}\left[\sum_{k=0}^{H-1} r(s_k, a_k)\right] \Rightarrow \max_{\pi} \mathbb{E}\left[\int_0^T r(s_t, a_t)\, dt\right] \quad \text{s.t. } \dot{s}_t = f(s_t, a_t; \epsilon) \tag{4}$$

Pontryagin dual system: Hamiltonian form

- Define Hamiltonian: $\mathcal{H}(s, p, a) := p^\top f(s, a) - r(s, a)$
- First-order optimality and reduced Hamiltonian: $a^*(s, p) : \frac{\partial \mathcal{H}}{\partial a} = 0$, and $h(s, p) = \mathcal{H}(s, p, a^*)$
- Dual dynamics: $\dot{s} = \frac{\partial h}{\partial p}, \qquad \dot{p} = -\frac{\partial h}{\partial s}$.
- In compact form with $x = (s, p)$ and symplectic $S = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$:

$$\dot{x} = S\,\nabla h(x) \quad \Rightarrow \quad x_{n+1} = x_n + \Delta_t S\,\nabla h(x_n) \tag{5}$$

Reinforcement learning and an abstract problem $\mathcal{D}$

Reframe RL as learning a dynamics operator $G : x \mapsto x + \Delta_t S \nabla g(x)$ such that $x, G(x), \ldots, G^{(H-1)}(x)$ trace an optimal trajectory. G can be called a policy

$$\boxed{G \;=\; \mathrm{Id} + \Delta_t\, S\, \nabla g} \tag{6}$$

- $g(x) \approx \hbar(x)$ is a learnable *score* over the extended phase space.
- **Query environment** $\mathcal{B}$
  - Given a policy (approximation) G_θ, and starting point/seed $x \sim \rho_0$, $\mathcal{B}$ returns rollout of $\left\{ G_\theta^{(k)}(x),\; g\big(G_\theta^{(k)}(x)\big) \right\}_{k=0}^{H-1}$.
  - Enables learning from trajectory segments + scores
- Physics prior via S while remaining model-free wrt reward gradients.

Solution to abstract problem $\mathcal{D}$

Algorithm dfPO (stage-wise, Dijkstra-like time expansion)

Initialize replay queue $\mathcal{M}$; random g_{θ_0}; set $G_{\theta_0} = \mathrm{Id} + \Delta_t S \nabla g_{\theta_0}$
for $k = 1$ to $H - 1$ **do**
    Query $\mathcal{B}$ with $G_{\theta_{k-1}}$ at N_k seeds $\{X^i\}$ to get trajectories and scores
    Add $(x, y) = \left(G_{\theta_{k-1}}^{(k-1)}(X^i),\ g(\cdot)\right)$ to $\mathcal{M}$
    Add stability samples $\left(G_{\theta_{k-1}}^{(j)}(X^i),\ g_{\theta_{k-1}}(\cdot)\right)$ for $j < k - 1$
    Fit g_{θ_k} on $\mathcal{M}$ with smooth L^1 loss; set $G_{\theta_k} = \mathrm{Id} + \Delta_t S \nabla g_{\theta_k}$
Output: $G_{\theta_{H-1}} = \mathrm{Id} + \Delta_t S \nabla g_{\theta_{H-1}}$

- Trust-region flavor via pointwise random samples
- Policy, dynamics and rewards related through gradient (automatic differentiation)

Experimental Setups

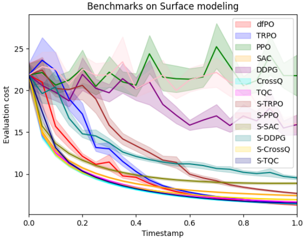
Representative tasks across scientific domains.

- **(1) Surface Modeling (single object-level)**: Control the geometry of an individual structure (e.g., airfoils, mechanical parts) via surface control points. Objectives include smoothness, curvature, and stress.
- **(2) Grid-Based Modeling (system-level)**: Macro-scale physical systems governed by PDEs. Coarse controls act on low-res grids; evaluation uses fine-grid PDE solvers.
- **(3) Molecular Dynamics (atomistic scale)**: Directly control atomic-scale systems governed by complex, nonlocal energy landscapes.

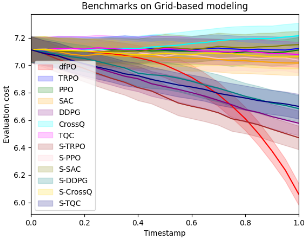
Setup.

- **Baselines (12 total):** TRPO, PPO, SAC, DDPG, CrossQ, TQC for both standard reward $r(s) = -\mathcal{F}(s)$ and reward reshaping $r(s, a) = \beta^{-t}(\frac{1}{2}\|a\|^2 - \mathcal{F}(s))$
- **Sample Budgets:** 100k steps for first 2 tasks and 5k steps for the last (due to high simulation cost)
- **Evaluation Metrics:** Terminal cost $\mathcal{F}(s)$ (lower is better).

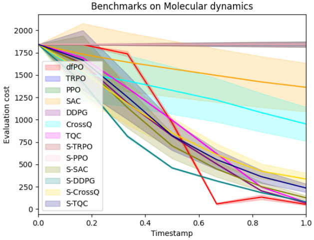
Results: visualization



(a) Surface modeling



(b) Grid-based modeling



(c) Molecular dynamics

Figure: Evaluation costs over episodes for 13 algorithms on 3 scientific computing tasks. dfPO (red curves) consistently achieves lower costs with more optimal and physically aligned trajectories.

Results: numerical (lower is better)

Numerical results:

		Standard Algorithms						Reward-Shaping Variants					
Task	dfPO	S-TRPO	S-PPO	S-SAC	S-DDPG	S-CrossQ	S-TQC	TRPO	PPO	SAC	DDPG	CrossQ	TQC
Surface	**6.32**	7.74	19.17	8.89	9.54	6.93	6.51	**6.48**	20.61	7.41	15.92	**6.42**	6.67
Grid	**6.06**	**6.48**	7.05	7.17	6.68	7.07	6.71	7.10	7.11	7.00	**6.58**	7.23	7.12
Mol.	**53.34**	1842.30	1842.30	126.73	82.95	338.07	231.98	1842.28	1842.31	1361.31	**68.20**	923.90	**76.87**

Highlights:

- dfPO attains the best (lowest) terminal costs on all 3 tasks; next-best varies (CrossQ/TQC/DDPG/TRPO).
- Reward-shaping generally helps baselines but remains below dfPO.
- Visualization: dfPO explores to lower costs with moderate variance with pattern similar to TRPO; SAC smooth but biased; PPO underperforms.

Pointwise Convergence

Derivative-transfer objective: Learn g_{θ_k} such that $G_{\theta_k} = \mathrm{Id} + \Delta_t S \nabla g_{\theta_k}$ approximates $G = \mathrm{Id} + \Delta_t S \nabla g$ pointwise.

Theorem (Pointwise convergence). Assume G, G_{θ_k} are L-Lipschitz and budgets N_k satisfy stagewise transfer criteria. Then with probability $\geq 1 - \delta$:

$$\mathbb{E}_X \| G_{\theta_k}^{(j)}(X) - G^{(j)}(X) \| < \frac{jL^j \epsilon}{L-1} \quad (1 \leq j \leq k) \tag{7}$$

Key idea: 3-term decomposition (at stage $k+1$):

$$\begin{aligned}\mathbb{E}\|G_{\theta_{k+1}}^{(k+1)}(X) - G^{(k+1)}(X)\| &\leq \|G_{\theta_{k+1}}(G_{\theta_{k+1}}^{(k)}(X)) - G_{\theta_{k+1}}(G_{\theta_k}^{(k)}(X))\| + \|G_{\theta_{k+1}}(G_{\theta_k}^{(k)}(X)) - G(G_{\theta_k}^{(k)}(X))\| \\ &\quad + \|G(G_{\theta_k}^{(k)}(X)) - G(G^{(k)}(X))\| \\ &\leq L \underbrace{\|G_{\theta_{k+1}}^{(k)}(X) - G_{\theta_k}^{(k)}(X)\|}_{\text{replay drift}} + \underbrace{\|G_{\theta_{k+1}}(G_{\theta_k}^{(k)}(X)) - G(G_{\theta_k}^{(k)}(X))\|}_{\text{supervised error}} + L \underbrace{\|G_{\theta_k}^{(k)}(X) - G^{(k)}(X)\|}_{\text{inductive propagation}}\end{aligned}$$

Sample Complexity and Regret Bounds

Goal. Bound cumulative regret: $\mathrm{Regret}(K) = \sum_{k=1}^{K} \left(V(s^k) - V_{\pi^k}(s^k)\right)$ using stagewise sample budgets $N_k = \mathcal{O}(\epsilon^{-\mu})$ with fixed rollout horizon H.

Stagewise Sample Budget. Let $N(g, \mathcal{H}, \epsilon, \delta)$ be the minimal number of samples required to train $h \in \mathcal{H}$ on $X \sim \rho_0$ such that $\mathbb{P}\left(\|\nabla g(X) - \nabla h(X)\| < \epsilon\right) \geq 1 - \delta$. Then define:

$$N_1 = N(g, \mathcal{H}_1, \epsilon, \delta), \quad N_k = \max\left\{N(g_{\theta_{k-1}}, \mathcal{H}_k, \epsilon, \delta_{k-1}/(k-1)), N(g, \mathcal{H}_k, \epsilon, \delta_{k-1}/(k-1))\right\} \tag{8}$$

Two Hypothesis Settings:

- **General $\mathcal{H}_k$:** usual neural network class, $g, h \in C^2$ with bounded weights

$$N_k = \mathcal{O}(\epsilon^{-(2d+4)}) \Rightarrow \mathrm{Regret}(K) = \mathcal{O}(K^{(2d+3)/(2d+4)}) \tag{9}$$

- **Restricted $\mathcal{H}_k$:** $h - g_k$ is linearly bounded and p-weakly convex, $p \geq 2d$

$$N_k = \mathcal{O}(\epsilon^{-6}) \Rightarrow \mathrm{Regret}(K) = \mathcal{O}(K^{5/6}) \tag{10}$$

Sketch. Sample complexity $\Rightarrow$ pointwise generalization bound $\Rightarrow$ $\mathcal{O}(\epsilon)$ per-step loss $\Rightarrow$ regret via H-step rollout across K episodes. *Next: we show how these N_k bounds arise under the two settings (next 2 slides).*

Setting 1: General $\mathcal{H}_k$, $N_k = \mathcal{O}(\epsilon^{-(2d+4)})$

Goal: Control $\|\nabla h - \nabla g\|$ via pointwise bounds on $|h - g|$.
Flow: Local bounds on $|h - g|$ at random samples $X_1, \cdots, X_m$ and $Y \Rightarrow$ gradient control at anchor $Y \Rightarrow$ global expectation via Rademacher complexity $\Rightarrow$ Yields dimension-dependent rate: $N_k = \mathcal{O}(\epsilon^{-(2d+4)})$.

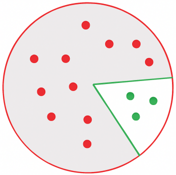
- **Second-order Taylor expansion:** around random anchor Y and nearby X_k for one of $k \in \overline{1, m}$:

$$h(X_k) - g(X_k) \approx h(Y) - g(Y) + \langle \nabla h(Y) - \nabla g(Y), X_k - Y\rangle + \mathcal{O}(\|X_k - Y\|^2) \quad (11)$$

- **Bound gradient gap:** Since $|h(X_k) - g(X_k)|$ and $|h(Y) - g(Y)|$ is small, if the gradient gap $\|\nabla h(Y) - \nabla g(Y)\|$ can be upper-bounded by $\langle \nabla h(Y) - \nabla g(Y), X_k - Y\rangle$, then the gap is small.
- **Conic technique:** probabilistically provide such an upper-bound inequality

$$\langle \nabla h(Y) - \nabla g(Y), X_k - Y\rangle \geq (1 - \epsilon_2)\|\nabla h(Y) - \nabla g(Y)\|\|X_k - Y\|$$

  This holds when $X_k - Y$ is well-aligned with $\nabla h(Y) - \nabla g(Y)$, i.e., falls inside a narrow favorable green cone and avoids the wide gray-red region (see figure).



- **Cone hit probability:** For $\|X_k - Y\| \in [\epsilon_1/2, \epsilon_1]$, the chance of hitting the favorable cone is at least $c_1\epsilon_1^d\epsilon_2$, so $m = \mathcal{O}(\epsilon_1^{-d})$ suffices.

Setting 2: Restricted $\mathcal{H}_k$, $N_k = \mathcal{O}(\epsilon^{-6})$

Assumptions. $h - g$ is both **linearly bounded** and p-**weakly convex** with $p \geq 2d$:

$$|u(y) - u(x)| \leq C \,\|\nabla u(x)\|\, \|y - x\| \qquad \text{(linearly bounded)}$$

$$u(y) \geq u(x) + \nabla u(x)^\top (y - x) - C\|y - x\|^p \qquad \text{(weakly convex)}$$

Inductive Scheme. At each step k, aim to **incrementally** bound $\|\nabla h - \nabla g\| \lesssim \epsilon^{\beta_k}$ at anchor points using losses:

$$\phi_k(y, \hat{y}) = \operatorname{clip}\big(|y - \hat{y}|,\, 0,\, C\epsilon^{\alpha+\beta_k}\big)^d, \qquad \alpha = 1/d \tag{12}$$

Main Steps.

1. Use weak convexity + linear bound + conic argument from Setting 1 to convert $|h - g|$ at nearby points into first-order control at Y.
2. Apply Rademacher complexity bounds to the previous inequalities via Lipschitz coefficients of ϕ_k.
3. From here, inductively shows: $\beta_{k+1} = \beta_k + 1/d \Rightarrow$ finer bound: $\|\nabla h(Y) - \nabla g(Y)\| \leq C\epsilon^{\alpha+\beta_{k+1}}$.
4. After d steps, $\beta_d = 1 \Rightarrow N = \mathcal{O}(\epsilon^{-6})$, independent of d.

Two Settings: Rates and Assumptions

- In continuous domains, under mild Lipschitz–MDP conditions, the minimax lower bound is $\Omega(K^{\frac{d+1}{d+2}})$ (Slivkins 2024). Our regret $\mathcal{O}(K^{\frac{2d+3}{2d+4}})$ in general case (Setting 1) is reasonable and surprisingly close, despite using a very different approach.
- Faster, dimension-free rates require stronger smoothness assumptions (e.g., Maran et al. 2024; Vakili and Olkhovskaya 2023).
- Our **linearly bounded** assumption is mild: it holds for any Lipschitz h, g outside the zero-gradient region, which has measure zero if $h - g$ is C^∞ (covering argument $\Rightarrow$ satisfy).
- The **weak convexity** condition holds for convex functions and neural networks with convex activations (e.g., ReLU). The Hamiltonian structure can further be used to establish this condition, making C^∞ smoothness a candidate under our setting. Tighter rates and less restricted assumptions can be achieved with further refinement of our proof.

Conclusion

1. **Differential Reinforcement Learning (Differential RL)** reinterprets RL through the lens of continuous-time control theory with Hamiltonian dual formulation, offering:
   - **Physics priors**
   - **Sample efficiency**
   - **Pointwise updates**
2. We instantiate this framework via **Differential Policy Optimization (dfPO)**:
   - $\mathcal{O}(K^{5/6})$ regret bound with pointwise convergence guarantees.
   - Strong empirical gains over RL baselines on scientific domains across physical scales: object-level, macroscopic system-level, and atomistic-level control tasks.
3. **Future directions**: Adaptive discretization, broader applicable domains, more unified framework bridging RL and control theory.

Acknowledgments: Supported by the Peter O'Donnell Foundation, Michael J. Fox Foundation, Jim Holland–Backcountry Foundation (AI in Parkinson's), and Army Research Office (W911NF-19-2-0333).

References (Part I)

- Jia, Yanwei and Xun Yu Zhou (2023). "q-Learning in Continuous Time". In: *Journal of Machine Learning Research* 24.161, pp. 1–61.
- Fleming, Wendell H and H Mete Soner (2006). *Controlled Markov processes and viscosity solutions*. en. 2nd. New York, NY: Springer.
- Wang, Haoran, Thaleia Zariphopoulou, and Xun Yu Zhou (2020). "Reinforcement Learning in Continuous Time and Space: A Stochastic Control Approach". In: *Journal of Machine Learning Research* 21.198, pp. 1–34.
- Schulman, John et al. (July 2015). "Trust Region Policy Optimization". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1889–1897.
- Maran, Davide et al. (2024). "Local Linearity: the Key for No-regret Reinforcement Learning in Continuous MDPs". In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., pp. 75986–76029.
- Slivkins, Aleksandrs (2024). "Introduction to Multi-Armed Bandits". In: *arXiv preprint arXiv:1904.07272*. arXiv: `1904.07272 [cs.LG]`.

References (Part II)

- Vakili, Sattar and Julia Olkhovskaya (2023). "Kernelized Reinforcement Learning with Order Optimal Regret Bounds". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., pp. 4225–4247.
- Zhao, Hanyang, Wenpin Tang, and David Yao (2023). "Policy Optimization for Continuous Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., pp. 13637–13663.
- Kirk, Donald E (1971). *Optimal Control Theory: An Introduction*. en. London, England: Prentice-Hall.
- Bajaj, Chandrajit, Minh Nguyen, and Conrad Li (2025). "Reinforcement Learning for Molecular Dynamics Optimization: A Stochastic Pontryagin Maximum Principle Approach". In: *Neural Information Processing*. Singapore: Springer Nature Singapore, pp. 310–323.
- Bajaj, Chandrajit and Minh Nguyen (2024). "Physics-Informed Neural Networks via Stochastic Hamiltonian Dynamics Learning". In: *Intelligent Systems and Applications*. Springer Nature Switzerland, pp. 182–197.