Beyond the Bellman Recursion: A Pontryagin-Guided Framework for Non-Exponential Discounting

Anonymous Authors[1]

Abstract

Most value-based and actor–critic reinforcement learning methods rely on Bellman-style recursions, yet these recursions collapse under non-exponential discounting common in human preferences and survival processes. We show the breakdown is structural: exponential discounting sits at a fragile intersection of multiplicativity and time homogeneity, and violating either property breaks standard dynamic programming. To overcome this, we propose **Pontryagin-Guided Direct Policy Optimization (PG-DPO)**, a variational framework that abandons recursion and couples the Pontryagin Maximum Principle with Monte Carlo rollouts via an *Adjoint-MC projection* enforcing pointwise Hamiltonian maximization. Across multi-dimensional hyperbolic and survival-discount benchmarks, PG-DPO improves accuracy and stability where equation-driven solvers and critic-based baselines diverge.

1. Introduction

Reinforcement learning (RL) and continuous-time stochastic control hinge on a single principle: *Bellman recursion*. It encodes time consistency by requiring that every continuation problem preserves the same objective form. In continuous time, this aligns with *exponential discounting*, which underpins classical HJB equations and many modern deep solvers based on dynamic programming (Bellman, 1957; Sutton & Barto, 2018; Han et al., 2018). In contrast, non-exponential discounting is essential to capture present bias and preference reversals (Strotz, 1955; Laibson, 1997; Frederick et al., 2002). Indeed, empirical and theoretical work documents widespread departures from exponential discounting, including hyperbolic and survival-based patterns (Strotz, 1955; Laibson, 1997; Frederick et al., 2002; Schultheis et al., 2022).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

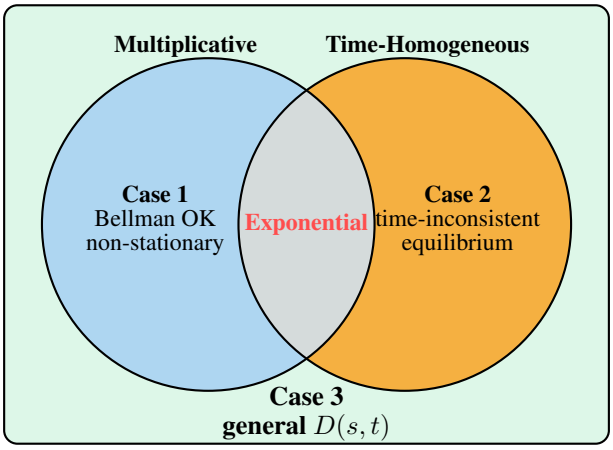


Figure 1. **Discount-kernel taxonomy.** Exponential discounting lies at the intersection of multiplicativity (1) and time homogeneity (2). Violating either property invalidates recursion-based methods.

To pinpoint the failure, let $D(s,t)$ denote the discount factor applied at evaluation time s to a payoff realized at time $t \geq s$. Bellman recursion corresponds to a *multiplicativity* of D, i.e.,

$$D(s,t) = D(s,u)D(u,t), \qquad \forall s \leq u \leq t. \tag{1}$$

Separately, *time homogeneity* assumes discounting determined only by the delay:

$$D(s,t) = \bar{D}(t-s). \tag{2}$$

Under mild regularity, (1) and (2) together imply the exponential form

$$D(s,t) = e^{-\delta(t-s)} \qquad \text{for some } \delta \geq 0. \tag{3}$$

Thus, "exponential" discounting represents a narrow intersection of these two independent properties, as visualized in Figure 1.

Departing from this exponential corner does not merely generalize the problem; it fundamentally invalidates the core assumptions of the standard pipeline. If D is multiplicative only (Case 1), while recursion technically survives, the failure of stationarity renders standard steady-state solvers obsolete, enforcing the computational burden of time-dependent

policies (Schultheis et al., 2022). Conversely, if D is time-homogeneous only (Case 2), the Bellman principle itself collapses, shattering the recursive structure required for dynamic programming. This breakdown forces a shift from optimality to time-consistent equilibrium, a problem typically patched via extended HJB systems (Ekeland & Lazrak, 2006a; Björk & Murgoci, 2014; Yong, 2012) or heuristic value pipelines like UGAE (Kwiatkowski et al., 2023), despite recent efforts to clarify well-posedness (Lei & Pun, 2023; Bayraktar et al., 2025). In Case 3, the pipeline suffers a complete structural failure where both recursion and stationarity are simultaneously lost.

Such failures are intrinsic to recursion-based methods. Existing remedies either try to reinstate recursion through state augmentation or relax optimality to time-consistent equilibrium notions (Ekeland & Lazrak, 2006a; Björk & Murgoci, 2014; Yong, 2012). However, both routes introduce additional time indices and/or nonlocal couplings, which quickly become brittle in high dimensions and undermine the scalability of global equation-driven solvers, including PDE-residual minimization methods such as PINNs (Raissi et al., 2019). Deep BSDE approaches face a closely related obstacle: they rely on global backward representations that are tailored to Bellman-consistent recursions and therefore mismatch non-multiplicative discounting (Han et al., 2018). Indeed, recent progress on deep solvers for genuinely nonlocal backward objects (e.g., BSVIEs) (Andersson et al., 2025) further underscores the need for recursion-free alternatives.

To address this, we adopt a variational approach based on adjoint sensitivity and Pontryagin's maximum principle (PMP) (Chen et al., 2018; Pontryagin et al., 1962), eschewing value-function decomposition entirely (Pontryagin et al., 1962; Yong & Zhou, 1999). While PMP has been revisited for continuous-time RL (Archibald et al., 2023; Eberhard et al., 2025), we propose a unified framework: *Pontryagin-Guided Direct Policy Optimization (PG-DPO)* (Huh et al., 2025).

Our contributions are: (i) a structural decomposition of recursion failures in non-exponential settings (Figure 1); (ii) the Bellman-free PG-DPO algorithm; and (iii) empirical gains over *(TD/Bellman-error) actor–critic* baselines and *global equation-driven (surrogate-fitting)* baselines across all non-exponential discounting cases (Cases 1–3).

2. Beyond Bellman Optimality: Pontryagin Optimality and PG-DPO

2.1. Pontryagin Maximum Principle (PMP)

Non-exponential discounting can invalidate Bellman recursion and thus obstruct a single Markovian value-function characterization. We therefore base our methodology on *Pontryagin optimality* (Pontryagin et al., 1962; Yong & Zhou, 1999). Our viewpoint is *decision-time anchoring*: at each decision time t_0, we treat the remaining-horizon problem on $[t_0, T]$ with the explicit kernel $D(t_0, \cdot)$ and enforce a pointwise Pontryagin condition. We record the PMP ingredients that we will enforce algorithmically in PG-DPO.

General discounted stochastic control. Fix a finite horizon $[t_0, T]$. Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[t_0,T]}, \mathbb{P})$ support a d-dimensional Brownian motion $W_t \in \mathbb{R}^d$. Consider the controlled diffusion

$$dX_t = b(t, X_t, u_t)\,dt + \sigma(t, X_t, u_t)\,dW_t, \quad X_{t_0} = x_0, \tag{4}$$

where $X_t \in \mathbb{R}^d$ and $u_t \in \mathcal{U}$ is an admissible control. Throughout, admissibility means (u_t) is progressively measurable and satisfies the usual integrability conditions that ensure (4) is well-posed.

Let $D : [t_0, T] \times [t_0, T] \to \mathbb{R}_+$ be a discount kernel with $D(s, s) = 1$. For a running reward ℓ and terminal reward g, define the anchored objective

$$J(t_0, x_0; u) := \mathbb{E}_{t_0,x_0}\Bigg[\int_{t_0}^{T} D(t_0, t)\,\ell(t, X_t, u_t)\,dt + D(t_0, T)\,g(x_T)\Bigg]. \tag{5}$$

where $\mathbb{E}_{t_0,x_0}[\cdot] := \mathbb{E}[\cdot \mid X_{t_0} = x_0]$.

Under standard smoothness and integrability assumptions, if $u^\star$ denotes the relevant time-consistent solution: when D is multiplicative, we use the term *optimal* $u^* \in \operatorname{argmax}_{u\in\mathcal{U}} J(t_0, x_0; u)$, otherwise *equilibrium*, characterized by $\liminf_{\epsilon\downarrow 0} \epsilon^{-1}(J(t, s; u^\star) - J(t, s; u_v^\epsilon)) \geq 0$ against any spike variation u_v^ϵ (Ekeland & Lazrak, 2006b; Björk et al., 2017; Yong, 2012).)

Anchored Hamiltonian and adjoint BSDE. For an anchor time t_0, define the (anchored) Hamiltonian

$$H(t_0, t, x, u, \lambda, Z) := D(t_0, t)\,\ell(t, x, u) + \langle \lambda, b(t, x, u)\rangle + \operatorname{Tr}\big(Z^\top \sigma(t, x, u)\big). \tag{6}$$

where $\lambda_t \in \mathbb{R}^d$ is the adjoint process and $Z_t \in \mathbb{R}^{d\times d}$ is the diffusion coefficient in the backward equation.

Let $X^\star$ be the induced state trajectory from $u^\star$. Then, there exist adapted processes $(\lambda^\star, Z^\star)$ satisfying the adjoint equation:

$$\begin{aligned} -d\lambda_t^\star &= \partial_x H\big(t_0, t, X_t^\star, u_t^\star, \lambda_t^\star, Z_t^\star\big)\,dt - Z_t^\star\,dW_t, \\ \lambda_T^\star &= D(t_0, T)\,\nabla g\big(X_T^\star\big). \end{aligned} \tag{7}$$

Maximum condition. PMP enforces a pointwise maximization of the Hamiltonian (Pontryagin et al., 1962; Yong & Zhou, 1999):

$$u_t^\star \in \arg\max_{u\in\mathcal{U}} H\big(t_0, t, X_t^\star, u, \lambda_t^\star, Z_t^\star\big). \tag{8}$$

This condition holds for a.e. $t \in [t_0, T]$, $\mathbb{P}$-a.s.

Implications of PMP. The Maximum Condition (8) implies that the optimal/equilibrium control $u^\star$ is directly determined by the adjoint $\lambda^\star$, meaning that estimating $\lambda^\star$ is sufficient to synthesize $u^\star$.

Existing deep solvers typically do auxiliary function approximation to estimate $\lambda^\star$, based on Bellman recursion (HJB, BSDE). Even under time-inconsistency, this paradigm is rigidly maintained via Extended HJB systems or BSVIEs, despite the loss of standard recursion.

However, as is well known, global function approximation is often sensitive to heuristics and environmental. Furthermore, the inherent recursive structure suffers from error propagation, where initial approximation errors inevitably accumulate step-by-step.

We propose a fundamentally different algorithmic paradigm: we interpret Backpropagation Through Time (BPTT) as a *stochastic adjoint estimator*[1]. In our framework, (8) plays the role of a Bellman-free *time-consistent* condition and will be enforced by the Adjoint-MC projection step in Section 2.2.

2.2. Pontryagin-Guided Direct Policy Optimization

PG-DPO is a two-stage, Bellman-free procedure that couples Monte Carlo rollouts with a Pontryagin projection (Huh et al., 2025). Beyond prior PG-DPO tailored to time-consistent objectives, we enforce the decision-time–anchored (diagonal) Pontryagin condition to obtain a time-consistent equilibrium under non-exponential discounting. Stage 1 warm-starts a differentiable policy by direct rollout optimization; Stage 2 estimates costates via BPTT and performs action-space Hamiltonian maximization.

Policy class. We parameterize the control as a neural network $u_\theta(t, x)$ with explicit time input. This network does not serve as a final output of policy, but only intermediate proxy to guide state exploration.

Discretization and Monte Carlo objective. We discretize $[t_0, T]$ with $t_k = t_0 + k\Delta t$, $\Delta t = (T - t_0)/N$. Using Euler–Maruyama,

$$\begin{aligned} X_{k+1} = X_k &+ b\big(t_k, x_k, u_\theta(t_k, x_k)\big)\,\Delta t \\ &+ \sigma\big(t_k, x_k, u_\theta(t_k, X_k)\big)\,\Delta W_k, \end{aligned} \tag{9}$$

with $\Delta W_k \sim \mathcal{N}(0, \Delta t)$. For each simulated path, we approximate (5) by

$$\begin{aligned} \widehat{J}(t_0, x_0; \theta) := \sum_{k=0}^{N-1} D(t_0, t_k)\, \ell\big(t_k, X_k, u_\theta(t_k, X_k)\big)\,\Delta t \\ + D(t_0, T)\, g(X_N). \end{aligned} \tag{10}$$

[1]We assume access to a stochastic simulator (physics-based or statistically estimated), or a differentiable learned world model, that supports pathwise differentiation and thus enables BPTT-based adjoint estimation.

Algorithm 1 PG-DPO

Input: policy u_θ; anchor dist. ν; rollout steps N; batch M; iters K_0; stepsizes $\{\alpha_j\}$.
  projection params (M_{MC}, N'); query set $\mathcal{Q}$ (each (t, x)).
Stage 1 (rollout warm-start): initialize θ
for $j = 0, \dots, K_0 - 1$ **do**
  Sample $\{(t_0^{(i)}, x_0^{(i)})\}_{i=1}^M \sim \nu$ and simulate M rollouts via (9)
  $\widehat{g} \leftarrow \frac{1}{M} \sum_{i=1}^M \nabla_\theta \widehat{J}(t_0^{(i)}, x_0^{(i)}; \theta)$ (BPTT)
  $\theta \leftarrow \theta + \alpha_j\, \widehat{g}$
end for
$\theta^\star \leftarrow \theta$
Stage 2 (Adjoint-MC projection):
for each $(t, x) \in \mathcal{Q}$ **do**
  Simulate M_{MC} rollouts from (t, x) under $u_{\theta^\star}$ (horizon N'; anchor at t)
  $\widehat{\lambda}(t, x) \leftarrow \frac{1}{M_{\mathrm{MC}}} \sum_{m=1}^{M_{\mathrm{MC}}} \frac{\partial \widehat{J}^{(m)}(t, x; \theta^\star)}{\partial x}$ (BPTT)
  Compute $\hat{u}(t, x)$ via (12) (few Newton/barrier steps; warm-start at $u_{\theta^\star}(t, x)$)
end for
Return: $\theta^\star$ and $\hat{u}$ on $\mathcal{Q}$

Stage 1: rollout warm-start. We warm-start θ by maximizing $\widehat{J}(t_0, x_0; \theta)$ via BPTT on the rollout graph, optionally randomizing anchors $(t_0, x_0) \sim \nu$.

Stage 2: Adjoint-MC projection (control synthesis). Given a query (t, x) and the frozen warm-start policy $u_{\theta^\star}$, we discretize $[t, T]$ by $t_k^{(t)} = t + k\Delta t'$, $\Delta t' = (T - t)/N'$, simulate M_{MC} rollouts from (t, x), and evaluate an objective *anchored at* t:

$$\begin{aligned} \widehat{J}^{(j)}(t, x; \theta^\star) = \sum_{k=0}^{N'-1} D\big(t, t_k^{(t)}\big)\, \ell\big(t_k^{(t)}, X_k^{(j)}, u_{\theta^\star}(t_k^{(t)}, X_k^{(j)})\big)\Delta t' \\ + D(t, T)\, g\big(X_{N'}^{(j)}\big). \end{aligned}$$

For each rollout j, we compute a pathwise costate by BPTT, $\lambda^{(j)}(t, x) := \partial \widehat{J}^{(j)}(t, x; \theta^\star)/\partial x$, and average

$$\widehat{\lambda}(t, x) := \frac{1}{M_{\mathrm{MC}}} \sum_{j=1}^{M_{\mathrm{MC}}} \lambda^{(j)}(t, x). \tag{11}$$

Anchoring at t yields the diagonal Pontryagin condition (anchor = decision time) used for equilibrium control when multiplicativity fails; see Section 2.3.

We then finally compute the pointwise Hamiltonian maximizer (8)

$$\hat{u}(t,x) \in \arg\max_{u\in\mathcal{U}(x)} H\big(t,t,x,u,\widehat{\lambda}(t,x),\widehat{Z}(t,x)\big), \quad (12)$$

In many problems the maximizer depends only on $\widehat{\lambda}$ (so $\widehat{Z}$ is not needed); when Z is required, it can be estimated by a standard one-step L^2 projection/regression on ΔW.

Computing the projection (Newton / log-barrier). Equation (12) does not require a closed form in constraint problem: we solve the action-space maximization by a few warm-started Newton (or quasi-Newton) iterations. With inequality constraints $\mathcal{U}(x) = \{u : g_i(u,x) \le 0\}$, we use the interior-point objective

$$\max_u \; H\big(t,t,x,u,\widehat{\lambda}(t,x),\widehat{Z}(t,x)\big) + \mu \sum_i \log\big(-g_i(u,x)\big), \quad (13)$$

together with backtracking line search to maintain feasibility. Optionally, this projection step can be amortized via offline distillation of $\hat{u}(t,x)$ into a student policy $\pi_\phi(t,x)$ for faster deployment.

2.3. BPTT as a stochastic adjoint estimator

A key reason PG-DPO remains effective under non-exponential discounting is that it does *not* learn a value function (critic) or rely on Bellman recursion. Instead, it computes *value-gradient information on-the-fly* from differentiable Monte Carlo rollouts, and then turns this gradient into an action by enforcing a Pontryagin (Hamiltonian) condition.

(1) BPTT gives *marginal value* with respect to the state. Fix an anchor time t_0 and consider the anchored rollout return $\widehat{J}(t_0,s_0;\theta)$ in (10). Reverse-mode differentiation through the rollout graph produces

$$\lambda_k^{\mathrm{pw}} := \frac{\partial \widehat{J}(t_0,x_0;\theta)}{\partial X_k},$$

which measures how much the anchored objective would change under an infinitesimal perturbation of the state at time t_k. In continuous-time PMP language, this is exactly the *costate* interpretation: λ is the marginal value of the state (Pontryagin et al., 1962; Yong & Zhou, 1999).

(2) Why averaging is the point (variance reduction & adaptedization). A single pathwise gradient λ^{pw} is noisy because it depends on one Brownian realization. Stage 2 therefore computes *stabilized* costate estimates by Monte Carlo averaging of BPTT state-gradients,

$$\widehat{\lambda}(t,x) := \frac{1}{M_{\mathrm{MC}}} \sum_{j=1}^{M_{\mathrm{MC}}} \lambda^{(j)}(t,x), \quad (14)$$

as already defined in (11). Intuitively, $\widehat{\lambda}(t,x)$ plays the role of a *value-gradient critic* evaluated at (t,x), but obtained by autodiff through the simulator rather than by training a separate network.

(3) How Averaged BPTT sensitivities ($\widehat{\lambda}$) become Pontryagin-style costates ($\lambda^\star$) Because BPTT differentiates through the feedback dependence $u_k = u_\theta(t_k, X_k)$, the exact state-sensitivity recursion contains an extra closed-loop term:

$$\lambda_k^{\mathrm{pw}} = \partial_{X_k} r_k + (\partial_{X_k} F_k)^\top \lambda_{k+1}^{\mathrm{pw}} + (\partial_{X_k} u_k)^\top G_k,$$

where G_k is a discrete stationarity residual (a discrete analogue of $\partial_u H$; Appendix A). As the Hamiltonian residual is becoming small by Stage 1, the closed-loop term vanishes (envelope cancellation), and the recursion reduces to the standard adjoint form. Consequently, the stabilized estimate $\widehat{\lambda}$ obtained by MC-averaging of λ^{pw} aligns with the Pontryagin-style costate $\lambda^\star$ near the projection step; see Appendix A for the formal statement.

(4) Stage 2: action synthesis by (approximately) killing the Hamiltonian residual. Stage 2 is an action-synthesis step: it constructs a *new* control $u^{\mathrm{proj}}(t,x)$ by maximizing the Hamiltonian anchored at the decision time t,

$$u^{\mathrm{proj}}(t,x) \in \arg\max_{u\in\mathcal{U}(x)} H\big(t,t,x,u,\widehat{\lambda}(t,x),\widehat{Z}(t,x)\big).$$

In this sense, Stage 2 enforces a near-zero Hamiltonian stationarity residual *with respect to the plug-in costate estimate* at the query point,

$$\partial_u H\big(t,t,x,u^{\mathrm{proj}}(t,x),\widehat{\lambda}(t,x),\widehat{Z}(t,x)\big) \approx 0.$$

This can be viewed as a Pontryagin-style *policy-improvement* step: given a costate estimate, we synthesize an action that is stationary for the corresponding anchored Hamiltonian.

2.4. Technical Advantage

Bypassing the Global Approximation Bottleneck.

Existing deep RL/solvers generally struggle with a trade-off between computational cost and control precision. Purely end-to-end actor–critic training is often brittle because it uses a global averaged objective to enforce a *pointwise* stationarity condition ($\partial_u H \approx 0$), so satisfying this local condition uniformly over the state space can require excessive iterations. Recent *model-based* trends partially mitigate sample inefficiency by learning a stochastic "world model" and planning through it; however, the difficulty of turning global training signals into uniformly accurate pointwise optimality conditions remains.

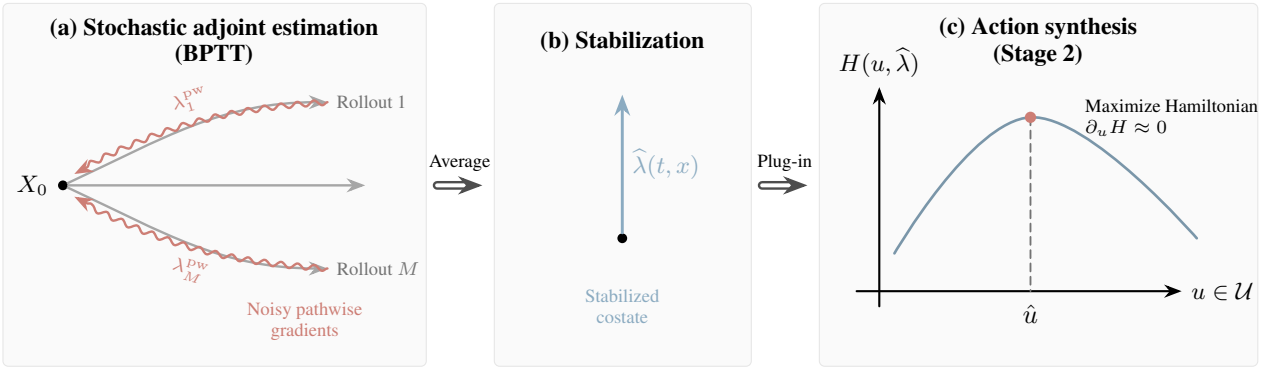


Figure 2. **Mechanism of Adjoint-MC Projection.** (a) BPTT computes noisy pathwise state-gradients (λ^{pw}) from anchored rollouts. (b) Monte Carlo averaging stabilizes these gradients into a robust costate estimate $\widehat{\lambda}(t,x)$. (c) This estimate defines the local Hamiltonian $H(\cdot,\widehat{\lambda})$, which is maximized in action space to synthesize u^{proj}, enforcing the Pontryagin condition directly.

Global equation-driven approaches (e.g., PINNs, Deep BSDEs) face a different bottleneck: they rely on global function approximation. A small error in the surrogate does not guarantee accurate recovery of its gradients, so expensive training can still yield unreliable controls.

PG-DPO fundamentally breaks this dependency. Even if the Stage 1 policy provides only a coarse approximation (weak optimality), the Stage 2 projection effectively minimizes the Hamiltonian residual via direct optimization (Appendix B). Since a sufficiently small Hamiltonian residual theoretically guarantees proximity to the optimal/equilibrium control (Appendix A), PG-DPO achieves high-confidence control synthesis with low training cost, bypassing the need for perfect global approximation.

Computational Efficiency. For a *single* query (t,x), the dominant cost of Stage 2 is linear in the total number of simulated steps,

$$T(t,x) = \mathcal{O}\Big(M_{\mathrm{MC}}\, N'\, C_{\mathrm{step}} \;+\; I_{\mathrm{proj}}\, C_{\mathrm{proj}}\Big), \qquad (15)$$

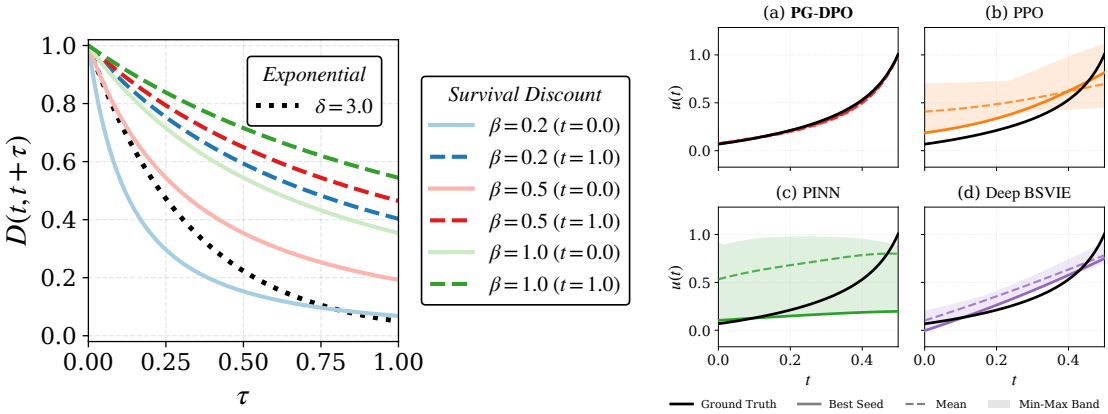
where C_{step} is the per-step cost of evaluating the frozen warm-start policy $u_{\theta^\star}$ and propagating the Euler–Maruyama dynamics. C_{proj} is the cost per optimization iteration (e.g., evaluating the Hamiltonian gradient and Hessian), and I_{proj} is the number of Newton/quasi-Newton (or barrier) iterations used to solve (12). In typical applications I_{proj} is a small constant (e.g., a handful of warm-started steps), so the overall runtime is dominated by costate estimation (BPTT). Our empirical experiments in Appendix C validate this linear scaling, demonstrating that our method achieves sub-second inference latency (< 0.02s) for our base configuration even on commodity CPUs. This efficiency implies that PG-DPO is not merely an offline solver but is computationally viable for online real-time control synthesis, where rapid re-planning is critical.

If even lower latency is desired, the per-query Stage 2 projection can be amortized offline by distilling the synthesized actions $\hat{u}(t,x)$ over a representative query set into a lightweight student policy, reducing deployment to a single forward pass.

3. Numerical Results

Our experiments cover the entire spectrum of discount-kernel regimes depicted in Figure 1. This confirms that **PG-DPO** generalizes effectively across diverse non-exponential discounting landscapes, verifying its comprehensive applicability. Throughout, we use $d = 5$ and visualize only the first control coordinate (u_1 or π_1) for clarity.

Baselines include PINN (Raissi et al., 2019) and Deep BSDE/BSVIE (Han et al., 2018), *global equation-driven* solvers that first fit a surrogate object (e.g., a value function or a PDE/BSDE/BSVIE representation) over the state space and then recover the control via first-order optimality conditions. PG-DPO is likewise equation-driven, but bypasses global surrogate fitting by using stochastic-adjoint information and enforcing pointwise Hamiltonian maximization, enabling a like-for-like comparison among equation-driven baselines. For completeness, we additionally report PPO (Schulman et al., 2017) as a representative *critic-based (TD/Bellman-error) actor–critic* baseline, to assess whether the theoretically motivated projection step yields a practical performance advantage. In Cases 2–3, PPO is trained with explicit time input and the same decision-time anchored Monte Carlo objective used for evaluation; it remains inferior because it does not enforce the diagonal (time-consistent) stationarity condition. Grid-based DP is omitted due to the curse of dimensionality inherent in our multidimensional settings. *Ground Truth* denotes reference policies from the standard HJB in the multiplicative regime (Case 1) or from the extended/equilibrium HJB in



(a) Survival discount $D(t, t+\tau)$ for varying β_0.

(b) Policy $u_1(t)$ for $\beta_0 = 1.0$.

Figure 3. **Case 1 (survival discounting).** (a) The survival-based kernel is multiplicative but time-inhomogeneous. (b) Learned controls compared to the analytic policy along a representative trajectory.

Table 1. **Quantitative comparison of control policy errors.** Values are reported as Mean ± Std over 10 seeds.

	Global L_1 Error ($\iint \lvert u^\star - \hat{u} \rvert dxdt$)		
Method	$\beta_0 = 0.2$	$\beta_0 = 0.5$	$\beta_0 = 1.0$
PPO	$2.51\text{e}{-1} \pm 4.48\text{e}{-2}$	$2.18\text{e}{-1} \pm 5.06\text{e}{-2}$	$2.23\text{e}{-1} \pm 4.90\text{e}{-2}$
PINN	$1.11\text{e}{0} \pm 2.07\text{e}{-1}$	$1.16\text{e}{0} \pm 1.20\text{e}{-1}$	$1.65\text{e}{0} \pm 2.53\text{e}{-1}$
Deep BSDE	$1.95\text{e}{-1} \pm 4.59\text{e}{-2}$	$3.71\text{e}{-1} \pm 2.90\text{e}{-2}$	$4.17\text{e}{-1} \pm 1.52\text{e}{-1}$
PGDPO (Ours)	$\mathbf{1.45\text{e}{-2} \pm 5.14\text{e}{-3}}$	$\mathbf{2.97\text{e}{-2} \pm 1.29\text{e}{-2}}$	$\mathbf{1.80\text{e}{-2} \pm 8.10\text{e}{-3}}$

the time-inconsistent regimes (Cases 2–3).

We demonstrate the high accuracy and robustness of **PG-DPO** by reporting the L_1 error and standard deviation across multiple random seeds. To ensure that the observed performance gap is purely algorithmic, we allocated a larger computational budget and model complexity exclusively to the approximation step of the baselines (Semi-details: Appendix D). [2]

3.1. Case 1 Results: Survival-Discounted Target Control

Discounting and task. Following Schultheis et al. (2022) (see also Aalen et al., 2008), we model discounting via survival under a Gamma prior on the hazard rate:

$$S(t) = \left(\frac{\beta_0}{\beta_0 + t}\right)^{\alpha_0}, \quad D(s,t) = \frac{S(t)}{S(s)} = \left(\frac{\beta_0 + s}{\beta_0 + t}\right)^{\alpha_0},$$

for $0 < s < t$. This preserves multiplicativity but destroys stationarity. In Case 1, the agent drives a controlled diffusion toward a target with a quadratic control-energy penalty. The survival kernel admits a natural *random termination* interpretation: when survival drops rapidly early on (Figure 3(a), small β_0), the objective becomes urgent and explicitly time-inhomogeneous. Overall, Case 1 should be viewed as an abstraction aligned with standard continuous-control objectives (stabilization/goal reaching/tracking under termination), while isolating the effect of non-stationary discounting.

Why PG-DPO succeeds. Figure 3(b) confirms the superiority of our approach. Panel (a) shows that PG-DPO tracks the ground-truth policy almost perfectly, and the very narrow Min–Max band indicates strong robustness across random seeds. In contrast, competing solvers (b–d) exhibit either large variance (PPO, PINN) or systematic bias (Deep BSDE), failing to match the correct curvature. Table 1 reports the quantitative comparison along β_0. PG-DPO attains the lowest L_1 error with the smallest standard deviation. This gap arises because Stage 2 enforces the Pontryagin condition locally in time: decision-time–anchored rollouts yield a costate that captures steep early discounting, and Hamiltonian maximization maps this local marginal value to an action without requiring a globally consistent value-function fit.

[2] For full details and reproducibility: https://github.com/Non-exponential/Beyond-the-Bellman-Recursion

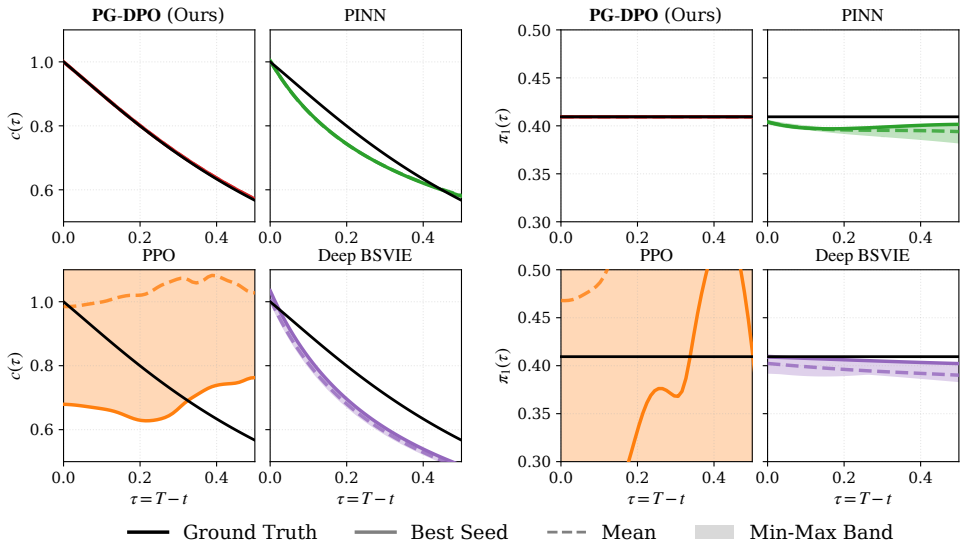


Figure 4. **Case 2 equilibrium policies.** Semi-analytic (extended-HJB) equilibrium vs. learned controls. (Left 2x2: Consumption Policy / Right 2x2: Investment Policy)

Table 2. **Quantitative comparison of errors**. The table reports Global L_1, L_∞ error as Mean ± Std over 10 seeds.

	Consumption (c)		**Investment** (π)	
Method	**MAE** (L_1)	**Max** (L_∞)	**MAE** (L_1)	**Max** (L_∞)
PPO	4.66e-01 ± 1.35e-01	1.03e+00 ± 2.79e-01	5.39e-01 ± 6.34e-02	2.22e+00 ± 4.03e-01
PINN	6.67e-02 ± 3.01e-03	2.17e-01 ± 1.83e-02	4.73e-02 ± 5.95e-03	3.05e-01 ± 2.57e-02
BSVIE	6.41e-02 ± 4.86e-03	1.46e-01 ± 1.13e-02	4.08e-02 ± 5.01e-03	2.44e-01 ± 1.25e-02
PG-DPO	**3.47e-03 ± 2.41e-10**	**6.11e-03 ± 1.19e-08**	**6.36e-08 ± 3.37e-10**	**1.92e-06 ± 1.46e-07**

3.2. Case 2 Results: Merton Problem with Hyperbolic Discounting

Discounting and task. We use the classical hyperbolic kernel (Strotz, 1955; Laibson, 1997; Frederick et al., 2002)

$$D(s,t) = \frac{1}{1+\kappa(t-s)} \qquad (\tau := t-s).$$

This kernel is time-homogeneous but non-multiplicative, hence time-inconsistent. Accordingly, performance is evaluated against the time-consistent (equilibrium) solution of the extended HJB (Ekeland & Lazrak, 2006a; Björk & Murgoci, 2014; Yong, 2012).

Why PG-DPO succeeds. In this case, restoring timehomogeneity simplifies the control problem to a stationary regime, generally reducing variance across all baselines compared to the non-stationary setting (Case 1). Despite this shared benefit, a clear performance hierarchy remains evident. As shown in Figure 4, *global equation-driven* baselines (PINN, Deep BSVIE) outperform the *critic-based (TD/Bellman-error) actor–critic* baseline (PPO), while PG-DPO achieves near-perfect alignment with the ground truth for both consumption and investment policies, exhibiting overwhelming accuracy and significantly tightest confidence bands. This superiority is quantitatively confirmed in Table 2, where PG-DPO reduces L_1 and L_∞ errors by several orders of magnitude. This precise alignment stems from Stage 2, which leverages the stationary structure to stabilize local costates via anchored rollouts, allowing Hamiltonian maximization to synthesize the exact equilibrium action.

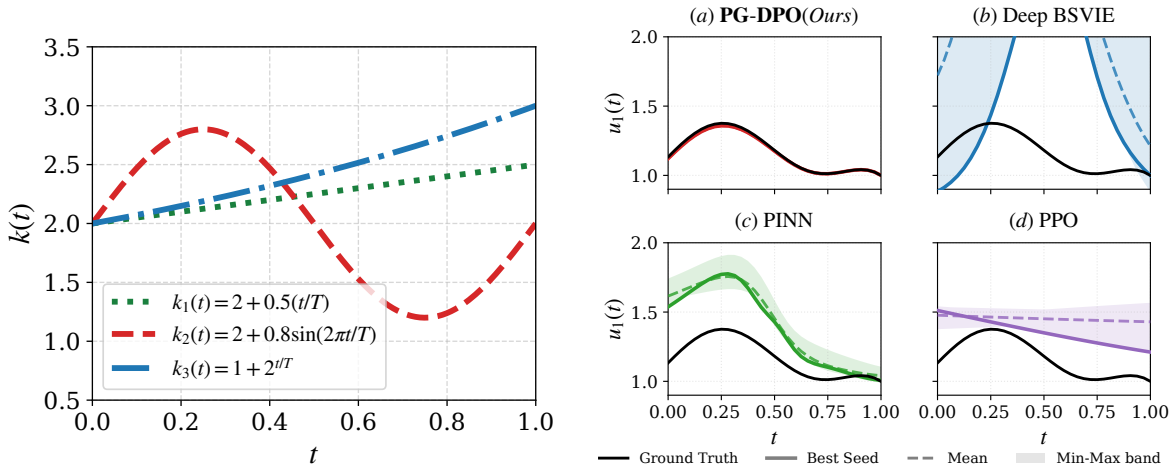
3.3. Case 3 Results: Stochastic Resource with Time-Varying Impatience

Discounting and task. We generalize Case 2 by allowing the hyperbolic parameter to vary with the decision time (Björk & Murgoci, 2014; Yong, 2012):

$$D(s,t) = \frac{1}{1+k(s)(t-s)}.$$

This kernel is non-multiplicative and explicitly non-stationary, inducing time inconsistency. Case 3 considers a stochastic resource/consumption problem where impatience fluctuates over time.

Why PG-DPO succeeds. Figure 5 shows that equilibrium policy becomes *non-monotone* and co-moves with $k(t)$.



(a) Impatience profiles $k(t)$.

(b) Equilibrium policy (analytic vs. learned).

Figure 5. **Case 3 (time-varying hyperbolic discounting).** (a) Time-varying impatience profiles $k(t)$ we used. (b) Equilibrium consumption under non-stationary discounting in case of $k_2(t)$.

Table 3. **Quantitative comparison of errors (Case 3).** The table reports Global L_1 as Mean ±Std under three different time-varying impatience profiles $k(t)$.

Method	Linear	Sinusoidal	Exponential
Deep BSVIE	1.29e0 ± 3.85e-01	1.02e0 ± 2.30e-01	1.36e0 ± 4.11e-01
PINN	2.64e-01 ± 3.60e-02	2.67e-01 ± 3.66e-02	2.93e-01 ± 5.97e-02
PPO	2.68e-01 ± 5.84e-02	2.89e-01 ± 5.14e-02	2.42e-01 ± 5.62e-02
PG-DPO	**6.35e-03 ± 2.51e-05**	**7.39e-03 ± 3.88e-05**	**6.70e-03 ± 3.19e-05**

PG-DPO tracks these regime changes because Stage 2 is decision-time anchored: when $k(t)$ changes, the anchored kernel $D(t,\cdot)$ and the corresponding Hamiltonian change immediately, and the action is synthesized by local Hamiltonian maximization rather than by relying on a globally trained value function. Across the whole $k(t)$, Table 3 shows uniformly smallest errors, supporting stable recovery of the time-consistent equilibrium mechanism under explicit non-stationarity.

Takeaway. Taken together with Cases 2–3, the results support the central message: when multiplicativity fails, PG-DPO benefits from Stage 2 by enforcing the decision-time Pontryagin condition through local action synthesis.

4. Conclusion

We showed that the ubiquity of exponential discounting in RL is structural: it arises from the intersection of multiplicativity and time homogeneity (Figure 1; Equations (1) to (3)). Violating either property breaks the recursive machinery behind dynamic programming and single-time BSDE formulations, helping explain the empirical instability of standard methods in non-exponential settings.

To bridge this gap, we introduced Pontryagin-Guided Direct Policy Optimization (PG-DPO; Section 2.2). PG-DPO is a model-based, simulator-access method tailored to settings with a stochastic simulator (physics-based or statistically estimated). By interpreting BPTT as a stochastic adjoint estimator, PG-DPO recovers costate information without a Markovian value function and replaces broken Bellman recursion with a variational principle.

Our approach advances the optimization landscape in two views. First, it achieves structural flexibility by liberating the optimization process from the Bellman recursion with mathematical consistency. Second, it offers technical robustness by moving beyond the standard frame of function approximation. This shift significantly reduces the reliance on heuristic tuning and stochastic environment.

Looking forward, promising directions include relaxing regularity to accommodate data-driven or non-smooth discount kernels and extending the projection machinery to richer constraints and frictions.

Impact Statement

This paper introduces PG-DPO, a method for optimizing policies under non-exponential discounting. Our work is primarily methodological, aiming to bridge the gap between theoretical control and practical applications. While effective control algorithms can have broad societal impacts, we believe this specific work does not introduce new ethical concerns beyond those already present in the fields of reinforcement learning and optimal control.

References

Aalen, O., Borgan, O., and Gjessing, H. *Survival and Event History Analysis: A Process Point of View*. Springer, 2008. doi: 10.1007/978-0-387-68560-1.

Andersson, A., Gnoatto, A., and García Trillos, N. Deep learning for backward stochastic volterra integral equations. *arXiv preprint arXiv:2505.18297*, 2025.

Archibald, R., Bao, F., and Yong, J. A stochastic maximum principle approach for reinforcement learning with parameterized environment. *Journal of Computational Physics*, 488:112238, 2023. doi: 10.1016/j.jcp.2023.112238.

Bayraktar, E., Huang, Y.-J., Wang, Z., and Zhou, Z. Relaxed equilibria for time-inconsistent markov decision processes. *Mathematics of Operations Research*, 50(4): 2666–2687, 2025. doi: 10.1287/moor.2023.0209.

Bellman, R. *Dynamic Programming*. Princeton University Press, 1957.

Björk, T. and Murgoci, A. A theory of markovian time-inconsistent stochastic control in discrete time. *Finance and Stochastics*, 18(3):545–592, 2014.

Björk, T., Khapko, M., and Murgoci, A. On time-inconsistent stochastic control in continuous time. *Finance and Stochastics*, 21(2):331–360, April 2017. doi: 10.1007/s00780-017-0327-5.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Eberhard, O., Vernade, C., and Muehlebach, M. A pontryagin perspective on reinforcement learning. In *Proceedings of the Seventh Annual Learning for Dynamics & Control Conference*, volume 283 of *Proceedings of Machine Learning Research*, pp. 233–244, 2025.

Ekeland, I. and Lazrak, A. Being serious about non-commitment: subgame perfect equilibrium in continuous time. *arXiv preprint math/0604264*, 2006a.

Ekeland, I. and Lazrak, A. Being serious about non-commitment: subgame perfect equilibrium in continuous time, April 2006b. arXiv:math/0604264.

Frederick, S., Loewenstein, G., and O'Donoghue, T. Time discounting and time preference: A critical review. *Journal of Economic Literature*, 40(2):351–401, 2002.

Han, J., Jentzen, A., and E, W. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34): 8505–8510, 2018.

Huh, J., Jeon, J., Koo, H. K., and Lim, B. H. Breaking the dimensional barrier: A pontryagin-guided direct policy optimization for continuous-time multi-asset portfolio choice. arXiv preprint arXiv:2504.11116, 2025.

Kwiatkowski, A., Kalogeiton, V., Pettré, J., and Cani, M.-P. Ugae: A novel approach to non-exponential discounting. *arXiv preprint arXiv:2302.05740*, 2023.

Laibson, D. Golden eggs and hyperbolic discounting. *The Quarterly Journal of Economics*, 112(2):443–478, 1997.

Lei, K. L. and Pun, C. S. On the well-posedness of hamilton-jacobi-bellman equations of the equilibrium type. *arXiv preprint arXiv:2307.01986*, 2023.

Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F. *The Mathematical Theory of Optimal Processes*. Interscience, 1962.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. doi: 10.48550/arXiv.1707.06347.

Schultheis, M., Rothkopf, C. A., and Koeppl, H. Reinforcement learning with non-exponential discounting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Strotz, R. H. Myopia and inconsistency in dynamic utility maximization. *The Review of Economic Studies*, 23(3): 165–180, 1955.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

Yong, J. Time-inconsistent optimal control problems and the equilibrium HJB equation. *Mathematical Control and Related Fields*, 2(3):271–329, 2012.

Yong, J. and Zhou, X. Y. *Stochastic Controls: Hamiltonian Systems and HJB Equations*. Springer, 1999.

A. PG-DPO, BPTT costates, and diagonal Pontryagin projection

This appendix clarifies (i) what backpropagation through time (BPTT) computes when differentiating anchored Monte Carlo rollouts, and (ii) which optimality/equilibrium condition is *explicitly enforced* by Stage 2 in PG-DPO. A key point is that, under non-multiplicative discounting, Stage 1 optimizes a *cooperative surrogate* (random-anchor objective) and is *not* claimed to yield a time-consistent equilibrium by itself. Time consistency is restored by Stage 2, which enforces a *diagonal* Pontryagin condition (anchor = decision time); see Ekeland & Lazrak (2006a); Björk & Murgoci (2014); Yong (2012) for equilibrium notions in time-inconsistent control and Pontryagin et al. (1962); Yong & Zhou (1999) for the Pontryagin maximum principle.

All statements in this appendix are formulated for a generic bounded continuous kernel $D(t_0, t)$ (two-time discount), covering: (i) multiplicative but time-inhomogeneous survival discounting (Case 1), (ii) time-homogeneous but non-multiplicative hyperbolic discounting (Case 2), and (iii) time-varying hyperbolic discounting (Case 3). For each fixed anchor t_0, BPTT yields an anchored closed-loop sensitivity recursion. In the multiplicative regime (Case 1), this anchored recursion is consistent with the classical anchored PMP adjoint (up to the usual envelope conditions). When multiplicativity fails (Cases 2–3), time-consistent equilibrium (in the spike-variation sense) is characterized by a diagonal (anchor = decision time) Pontryagin condition, which is explicitly enforced by Stage 2.

A.1. Setup: anchored objective, Hamiltonian, and assumptions

We consider the controlled diffusion on $[0, T]$:

$$dS_t = b(t, S_t, u_t)\, dt + \sigma(t, S_t, u_t)\, dW_t, \qquad S_{t_0} = s_0,$$

with an admissible action set $\mathcal{U}(s)$ and a (possibly non-multiplicative) discount kernel $D(t_0, t)$. For a fixed anchor (t_0, s_0), the anchored objective is

$$J(t_0, s_0; u) := \mathbb{E}\left[\int_{t_0}^{T} D(t_0, t)\, \ell(t, S_t, u_t)\, dt + D(t_0, T)\, g(S_T)\right].$$

Define the anchored Hamiltonian

$$H(t_0, t, s, u, y, z) := D(t_0, t)\, \ell(t, s, u) + \langle y, b(t, s, u)\rangle + \mathrm{Tr}\big(z^\top \sigma(t, s, u)\big). \tag{16}$$

Standing assumptions (for discrete-to-continuous statements). We assume:

1. (**Regularity of b, σ.**) b, σ are globally Lipschitz in s with linear growth, and $\sigma\sigma^\top$ is uniformly non-degenerate. Moreover, b, σ are C^1 in (s, u) with derivatives $\partial_s b, \partial_u b, \partial_s \sigma, \partial_u \sigma$ that are globally Lipschitz (or at least of polynomial growth ensuring the L^2 expansions below).
2. (**Regularity of costs.**) $\ell(\cdot, \cdot, \cdot)$ and $g(\cdot)$ are C^1 in s (and differentiable in u when needed) with polynomial growth.
3. (**Discount kernel.**) $D(t_0, t)$ is bounded and continuous on $\{(t_0, t) : 0 \le t_0 \le t \le T\}$ with $D(t, t) = 1$.
4. (**Differentiable policy class.**) The policy class is differentiable in state: $u_\theta(t, s)$ is C^1 in s with square-integrable Jacobians, and $u_\theta(t, s) \in \mathcal{U}(s)$.

A.2. Discrete rollouts and the exact closed-loop BPTT recursion

Fix an anchor (t_0, s_0) and discretize $[t_0, T]$ by $t_k = t_0 + k\Delta t$, $k = 0, \dots, N$. Consider the Euler–Maruyama rollout under a differentiable feedback policy $u_\theta(t, s)$:

$$S_{k+1} = S_k + b(t_k, S_k, u_k)\, \Delta t + \sigma(t_k, S_k, u_k)\, \Delta W_k, \qquad u_k := u_\theta(t_k, S_k). \tag{17}$$

Define the discrete anchored return

$$\widehat{J}(t_0, s_0; \theta) := \sum_{k=0}^{N-1} D(t_0, t_k)\, \ell(t_k, S_k, u_k)\, \Delta t + D(t_0, T)\, g(S_N). \tag{18}$$

Let the *pathwise BPTT costates* be the state sensitivities

$$\lambda_k^{\mathrm{pw}} := \frac{\partial \widehat{J}(t_0, s_0; \theta)}{\partial S_k}, \qquad k = 0, \dots, N.$$

One-step notation. Define the one-step reward and transition map by

$$r_k := D(t_0, t_k)\, \ell(t_k, S_k, u_k)\, \Delta t, \qquad u_k := u_\theta(t_k, S_k),$$

$$S_{k+1} := F_k(S_k, u_k, \Delta W_k),$$

where F_k is the Euler update in (17).

Proposition A.1 (Exact BPTT recursion (closed-loop adjoint with a policy-Jacobian term))**.** *The pathwise BPTT costates satisfy*

$$\begin{aligned} \lambda_N^{\mathrm{pw}} &= D(t_0, T)\, \nabla g(S_N), \\ \lambda_k^{\mathrm{pw}} &= \partial_{S_k} r_k + (\partial_{S_k} F_k)^\top \lambda_{k+1}^{\mathrm{pw}} + (\partial_{S_k} u_k)^\top G_k, \end{aligned} \tag{19}$$

for $k = 0, \dots, N-1$*, where*

$$G_k := \partial_{u_k} r_k + (\partial_{u_k} F_k)^\top \lambda_{k+1}^{\mathrm{pw}}. \tag{20}$$

Proof. Fix a single Monte Carlo rollout, i.e., condition on a realization of the noise increments $\{\Delta W_k\}_{k=0}^{N-1}$. Then the rollout induces a deterministic computation graph with the closed-loop dependence $u_k = u_\theta(t_k, S_k)$ and the one-step transition $S_{k+1} = F_k(S_k, u_k, \Delta W_k)$. For $k = 0, \dots, N$, define the *tail return* from time index k:

$$\widehat{J}_k := \sum_{i=k}^{N-1} r_i + D(t_0, T)\, g(S_N), \qquad \text{so that} \qquad \widehat{J}_0 = \widehat{J}(t_0, s_0; \theta).$$

By construction, $\widehat{J}_k$ depends on S_k only through the subgraph $S_k \to (u_k, r_k, S_{k+1}) \to \cdots \to S_N$, hence

$$\lambda_k^{\mathrm{pw}} = \frac{\partial \widehat{J}(t_0, s_0; \theta)}{\partial S_k} = \frac{\partial \widehat{J}_k}{\partial S_k}.$$

Terminal condition. At $k = N$, $\widehat{J}_N = D(t_0, T)\, g(S_N)$, hence

$$\lambda_N^{\mathrm{pw}} = \frac{\partial \widehat{J}_N}{\partial S_N} = D(t_0, T)\, \nabla g(S_N),$$

which proves the terminal condition in (19).

Backward recursion. For $k = 0, \dots, N-1$, the tail return satisfies

$$\widehat{J}_k = r_k + \widehat{J}_{k+1}, \qquad \text{with} \qquad S_{k+1} = F_k(S_k, u_k, \Delta W_k), \;\; u_k = u_\theta(t_k, S_k).$$

Differentiate both sides with respect to S_k and use the chain rule:

$$\begin{aligned} \lambda_k^{\mathrm{pw}} = \frac{\partial \widehat{J}_k}{\partial S_k} &= \frac{\partial r_k}{\partial S_k} + \left(\frac{\partial S_{k+1}}{\partial S_k}\right)^\top \frac{\partial \widehat{J}_{k+1}}{\partial S_{k+1}} \\ &= \frac{\partial r_k}{\partial S_k} + \left(\frac{\partial S_{k+1}}{\partial S_k}\right)^\top \lambda_{k+1}^{\mathrm{pw}}. \end{aligned}$$

Expand the two Jacobians under the closed-loop dependence $u_k = u_\theta(t_k, S_k)$.

(i) Reward term. Viewing $r_k = r_k(S_k, u_k)$,

$$\frac{\partial r_k}{\partial S_k} = \partial_{S_k} r_k + (\partial_{S_k} u_k)^\top \partial_{u_k} r_k.$$

(ii) Transition term. Viewing $S_{k+1} = F_k(S_k, u_k, \Delta W_k)$ (with ΔW_k fixed),

$$\frac{\partial S_{k+1}}{\partial S_k} = \partial_{S_k} F_k + \partial_{u_k} F_k \; \partial_{S_k} u_k.$$

Substituting (i) and (ii) gives

$$\begin{aligned}\lambda_k^{\mathrm{pw}} &= \partial_{S_k} r_k + (\partial_{S_k} u_k)^\top \partial_{u_k} r_k + \Big(\partial_{S_k} F_k + \partial_{u_k} F_k\, \partial_{S_k} u_k\Big)^\top \lambda_{k+1}^{\mathrm{pw}} \\ &= \partial_{S_k} r_k + (\partial_{S_k} F_k)^\top \lambda_{k+1}^{\mathrm{pw}} + (\partial_{S_k} u_k)^\top \Big(\partial_{u_k} r_k + (\partial_{u_k} F_k)^\top \lambda_{k+1}^{\mathrm{pw}}\Big),\end{aligned}$$

and defining G_k as in (20) completes the proof. □

Interpretation. The additional term $(\partial_{S_k} u_k)^\top G_k$ is the standard policy-Jacobian contribution in closed-loop differentiation. The quantity G_k is the discrete analogue of a Hamiltonian stationarity residual: it vanishes at an interior Pontryagin-stationary point (discrete analogue of $\partial_u H = 0$).

Corollary A.2 (Envelope cancellation at Pontryagin-stationary points)**.** *If $G_k = 0$ along the rollout (e.g., at an interior maximizer in action space), then the BPTT recursion reduces to the standard discrete adjoint recursion without the policy-Jacobian term:*

$$\lambda_k^{\mathrm{pw}} = \partial_{S_k} r_k + (\partial_{S_k} F_k)^\top \lambda_{k+1}^{\mathrm{pw}}.$$

A.3. Adapted projections and a BSDE limit (diagonal/anchored adjoint)

BPTT produces *pathwise* costates λ_k^{pw}. For Pontryagin-type synthesis we use their predictable (adapted) projection and the associated martingale coefficient. Define the one-step predictable projections

$$\lambda_k := \mathbb{E}[\lambda_k^{\mathrm{pw}} \mid \mathcal{F}_{t_k}], \qquad Z_k := \frac{1}{\Delta t}\,\mathbb{E}\big[\lambda_{k+1}^{\mathrm{pw}} (\Delta W_k)^\top \mid \mathcal{F}_{t_k}\big], \tag{21}$$

where $\Delta W_k := W_{t_{k+1}} - W_{t_k}$ and $Z_k \in \mathbb{R}^{d\times q}$.

Proposition A.3 (Continuous-time limit: closed-loop adjoint BSDE (anchored at t_0))**.** *Under the standing assumptions, as $\Delta t \to 0$ the piecewise-constant interpolations of (λ_k, Z_k) converge in L^2 (along a refining sequence of grids) to an adapted pair (λ_t, Z_t) that satisfies a closed-loop adjoint BSDE for the anchored objective $J(t_0, s_0; u_\theta)$:*

$$\begin{aligned} d\lambda_t &= -\Big(\partial_s H(t_0, t, S_t, u_\theta(t,S_t), \lambda_t, Z_t) + (\partial_s u_\theta(t,S_t))^\top\, \partial_u H(t_0, t, S_t, u_\theta(t,S_t), \lambda_t, Z_t)\Big)\, dt \\ &\quad + Z_t\, dW_t, \\ \lambda_T &= D(t_0, T)\, \nabla g(S_T). \end{aligned} \tag{22}$$

Moreover, if the diagonal Pontryagin stationarity holds (interior case) so that $\partial_u H(t_0, t, S_t, u_\theta(t,S_t), \lambda_t, Z_t) = 0$ a.s. for a.e. t, then (22) reduces to the standard anchored adjoint BSDE (without the policy-Jacobian term), consistent with the classical PMP (Pontryagin et al., 1962; Yong & Zhou, 1999).

Proof. We work on a fixed anchor (t_0, s_0) and a refining sequence of uniform grids $t_k = t_0 + k\Delta t$, $\Delta t = (T - t_0)/N$. For clarity write $u_k := u_\theta(t_k, S_k)$, $b_k := b(t_k, S_k, u_k)$ and $\sigma_k := \sigma(t_k, S_k, u_k)$.

Step 1: recall the closed-loop BPTT recursion. From Proposition A.1,

$$\lambda_k^{\mathrm{pw}} = \partial_{S_k} r_k + (\partial_{S_k} F_k)^\top \lambda_{k+1}^{\mathrm{pw}} + (\partial_{S_k} u_k)^\top G_k, \qquad k = 0, \dots, N-1, \tag{23}$$

with terminal condition $\lambda_N^{\mathrm{pw}} = D(t_0, T)\nabla g(S_N)$. Here $r_k = D(t_0, t_k)\ell(t_k, S_k, u_k)\Delta t$ and the Euler step is

$$F_k(s, u, \Delta W) = s + b(t_k, s, u)\Delta t + \sigma(t_k, s, u)\Delta W.$$

By construction, $\partial_{S_k} F_k$ and $\partial_{u_k} F_k$ denote *partial* derivatives of F_k with respect to its first and second arguments.

Step 2: define the predictable projections and the conditional L^2 decomposition. Let $\mathcal{F}_{t_k}$ be the sigma-field generated by the Brownian path up to t_k. Define

$$\lambda_k := \mathbb{E}[\lambda_k^{\mathrm{pw}} \mid \mathcal{F}_{t_k}], \qquad \tilde{\lambda}_{k+1} := \mathbb{E}[\lambda_{k+1}^{\mathrm{pw}} \mid \mathcal{F}_{t_k}] = \mathbb{E}[\lambda_{k+1} \mid \mathcal{F}_{t_k}],$$

and define Z_k as in (21). Then the conditional L^2 projection yields

$$\lambda_{k+1}^{\mathrm{pw}} = \tilde{\lambda}_{k+1} + Z_k\, \Delta W_k + R_k, \qquad \mathbb{E}[R_k \mid \mathcal{F}_{t_k}] = 0, \quad \mathbb{E}[R_k (\Delta W_k)^\top \mid \mathcal{F}_{t_k}] = 0. \tag{24}$$

Step 3: take conditional expectation of (23). Taking $\mathbb{E}[\cdot \mid \mathcal{F}_{t_k}]$ in (23) gives

$$\lambda_k = \mathbb{E}[\partial_{S_k} r_k \mid \mathcal{F}_{t_k}] + \mathbb{E}[(\partial_{S_k} F_k)^\top \lambda_{k+1}^{\mathrm{pw}} \mid \mathcal{F}_{t_k}] + \mathbb{E}[(\partial_{S_k} u_k)^\top G_k \mid \mathcal{F}_{t_k}]. \tag{25}$$

Since $\partial_{S_k} u_k = \partial_s u_\theta(t_k, S_k)$ is $\mathcal{F}_{t_k}$-measurable, we will use

$$\mathbb{E}[(\partial_{S_k} u_k)^\top G_k \mid \mathcal{F}_{t_k}] = (\partial_s u_\theta(t_k, S_k))^\top \mathbb{E}[G_k \mid \mathcal{F}_{t_k}].$$

Step 4: expand $\mathbb{E}[(\partial_{S_k} F_k)^\top \lambda_{k+1}^{\mathrm{pw}} \mid \mathcal{F}_{t_k}]$. From the Euler map,

$$\partial_{S_k} F_k = I_d + \partial_s b(t_k, S_k, u_k)\,\Delta t + \sum_{\ell=1}^{q} \partial_s \sigma^{(\ell)}(t_k, S_k, u_k)\,\Delta W_k^{(\ell)},$$

where $\sigma^{(\ell)}$ is the ℓ-th column of σ. Insert (24) and use conditional moments $\mathbb{E}[\Delta W_k \mid \mathcal{F}_{t_k}] = 0$ and $\mathbb{E}[\Delta W_k (\Delta W_k)^\top \mid \mathcal{F}_{t_k}] = \Delta t\, I_q$. A direct computation yields

$$\begin{aligned}\mathbb{E}[(\partial_{S_k} F_k)^\top \lambda_{k+1}^{\mathrm{pw}} \mid \mathcal{F}_{t_k}] = \tilde{\lambda}_{k+1} + \big(\partial_s b(t_k, S_k, u_k)\big)^\top \tilde{\lambda}_{k+1}\,\Delta t \\ + \sum_{\ell=1}^{q} \big(\partial_s \sigma^{(\ell)}(t_k, S_k, u_k)\big)^\top Z_k^{(\ell)}\,\Delta t + o_{L^2}(\Delta t),\end{aligned} \tag{26}$$

where $Z_k^{(\ell)}$ denotes the ℓ-th column of Z_k, and $o_{L^2}(\Delta t)$ collects higher-order Euler remainders and terms controlled under the standing assumptions.

Step 5: expand $\mathbb{E}[G_k \mid \mathcal{F}_{t_k}]$. Recall

$$G_k = \partial_{u_k} r_k + (\partial_{u_k} F_k)^\top \lambda_{k+1}^{\mathrm{pw}}, \qquad \partial_{u_k} F_k = \partial_u b(t_k, S_k, u_k)\,\Delta t + \sum_{\ell=1}^{q} \partial_u \sigma^{(\ell)}(t_k, S_k, u_k)\,\Delta W_k^{(\ell)}.$$

Using (24) and conditional moments of ΔW_k,

$$\begin{aligned}\mathbb{E}[G_k \mid \mathcal{F}_{t_k}] = \partial_{u_k} r_k + \big(\partial_u b(t_k, S_k, u_k)\big)^\top \tilde{\lambda}_{k+1}\,\Delta t \\ + \sum_{\ell=1}^{q} \big(\partial_u \sigma^{(\ell)}(t_k, S_k, u_k)\big)^\top Z_k^{(\ell)}\,\Delta t + o_{L^2}(\Delta t).\end{aligned} \tag{27}$$

Step 6: collect terms and match the Hamiltonian derivatives. First,

$$\partial_{S_k} r_k = D(t_0, t_k)\,\partial_s \ell(t_k, S_k, u_k)\,\Delta t, \qquad \partial_{u_k} r_k = D(t_0, t_k)\,\partial_u \ell(t_k, S_k, u_k)\,\Delta t.$$

Plug (26) and (27) into (25) to obtain

$$\lambda_k = \tilde{\lambda}_{k+1} + \Big(\partial_s H(t_0, t_k, S_k, u_k, \tilde{\lambda}_{k+1}, Z_k) + (\partial_s u_\theta(t_k, S_k))^\top\, \partial_u H(t_0, t_k, S_k, u_k, \tilde{\lambda}_{k+1}, Z_k)\Big)\Delta t + o_{L^2}(\Delta t), \tag{28}$$

where H is the anchored Hamiltonian (16) and the identities follow by direct differentiation of H with respect to s and u.

Step 7: continuous-time limit. Let $\bar{\lambda}^{\Delta t}$ and $\bar{Z}^{\Delta t}$ be the piecewise-constant interpolations on $[t_0, T]$ defined by $\bar{\lambda}_t^{\Delta t} := \lambda_k$ and $\bar{Z}_t^{\Delta t} := Z_k$ for $t \in [t_k, t_{k+1})$. Under the standing assumptions, Euler–Maruyama satisfies strong L^2 convergence, and the backward scheme (28) is stable in L^2. Therefore, along a refining sequence $\Delta t \downarrow 0$, the interpolations converge in L^2 to an adapted pair (λ_t, Z_t) satisfying

$$d\lambda_t = -\Big(\partial_s H(t_0, t, S_t, u_\theta(t, S_t), \lambda_t, Z_t) + (\partial_s u_\theta(t, S_t))^\top\, \partial_u H(t_0, t, S_t, u_\theta(t, S_t), \lambda_t, Z_t)\Big)\,dt + Z_t\,dW_t,$$

with terminal condition $\lambda_T = D(t_0, T)\nabla g(S_T)$, i.e., (22). (See, e.g., BSDE stability/discretization results in Yong & Zhou (1999) and references therein.)

Step 8: envelope cancellation under Pontryagin stationarity. If the (interior) Pontryagin stationarity condition holds so that $\partial_u H(t_0, t, S_t, u_\theta(t, S_t), \lambda_t, Z_t) = 0$ a.s. for a.e. t, then the policy-Jacobian drift term vanishes and (22) reduces to the standard anchored adjoint BSDE of the Pontryagin maximum principle (Pontryagin et al., 1962; Yong & Zhou, 1999). □

A.4. What Stage 1 optimizes: a random-anchor cooperative surrogate

Stage 1 in PG-DPO maximizes the random-anchor surrogate

$$J^{\mathrm{sur}}(\theta) := \mathbb{E}_{(t_0,s_0)\sim\nu}\big[\widehat{J}(t_0, s_0; \theta)\big], \tag{29}$$

via BPTT through rollouts.

Proposition A.4 (Unbiased random-anchor gradient estimator)**.** *Sampling $(t_0, s_0) \sim \nu$ and backpropagating $\widehat{J}(t_0, s_0; \theta)$ yields an unbiased estimator of $\nabla_\theta J^{\mathrm{sur}}(\theta)$.*

Proof. Immediate from linearity of expectation and i.i.d. sampling of anchors. □

Remark A.5 (Why Stage 1 alone does not imply time-consistent equilibrium (non-multiplicative D)). When discounting is non-multiplicative, time-consistent equilibrium conditions are *diagonal* conditions with anchor = decision time (extended HJB / equilibrium control; see Ekeland & Lazrak, 2006a; Björk & Murgoci, 2014; Yong, 2012). Stationarity of the *averaged* surrogate (29) generally enforces a weighted mixture of anchor-dependent stationarity residuals and does not, in general, imply the diagonal equilibrium condition. Accordingly, we use Stage 1 as a robust warm-start that produces a stable differentiable rollout policy, while equilibrium/optimality is enforced by Stage 2.

A.5. What Stage 2 enforces: diagonal Pontryagin projection (time-consistency restoration)

Stage 2 is designed to enforce a *diagonal* Pontryagin condition (anchor = decision time). At a query point (t, s), we construct Monte Carlo rollouts anchored at time t and define anchored returns $\widehat{J}^{(j)}(t, s; \theta^\star)$. BPTT state-gradients yield pathwise diagonal costate estimates

$$\lambda^{(j)}(t, s) := \frac{\partial \widehat{J}^{(j)}(t, s; \theta^\star)}{\partial s}, \qquad \widehat{\lambda}(t, s) := \frac{1}{M_{\mathrm{MC}}} \sum_{j=1}^{M_{\mathrm{MC}}} \lambda^{(j)}(t, s).$$

Optionally, when σ depends on u, we also estimate the diagonal martingale coefficient $\widehat{Z}(t, s)$ by one-step L^2 regression (cf. (21)). We then synthesize the action by diagonal Hamiltonian maximization (or a constrained Newton/log-barrier solve):

$$u^{\mathrm{proj}}(t, s) \in \arg\max_{u\in\mathcal{U}(s)} H\big(t, t, s, u, \widehat{\lambda}(t, s), \widehat{Z}(t, s)\big), \tag{30}$$

where H is the anchored Hamiltonian (16).

Proposition A.6 (Diagonal Pontryagin enforcement by Stage 2)**.** *Assume (i) the inner action-space maximization (30) is solved exactly (or to a prescribed tolerance) and (ii) the diagonal estimators $\widehat{\lambda}(t, s)$ (and $\widehat{Z}(t, s)$ when needed) are accurate for the anchored-at-t objective. Then $u^{\mathrm{proj}}(t, s)$ satisfies the diagonal Pontryagin* equilibrium *condition at (t, s) up to solver/statistical tolerance. When D is multiplicative, this diagonal condition reduces to the classical Pontryagin optimality condition. Under standard concavity assumptions, this diagonal condition is the appropriate first-order characterization of time-consistent equilibrium for non-multiplicative discounting (Ekeland & Lazrak, 2006a; Björk & Murgoci, 2014; Yong, 2012).*

Proof. Fix a query point (t, s).

Step 1: Stage 2 action-space problem. By definition of Stage 2, we form Monte Carlo rollouts starting from (t, s) under the frozen warm-start policy $u_{\theta^\star}$ and evaluate an objective *anchored at the decision time* t. From these rollouts we construct the diagonal costate estimator $\widehat{\lambda}(t, s)$ (and $\widehat{Z}(t, s)$ when needed), and then compute

$$u^{\mathrm{proj}}(t, s) \in \arg\max_{u\in\mathcal{U}(s)} H\big(t, t, s, u, \widehat{\lambda}(t, s), \widehat{Z}(t, s)\big). \tag{31}$$

Assumption (i) states that the numerical solver returns an $\varepsilon_{\mathrm{opt}}$-accurate maximizer in the sense that

$$H\big(t,t,s,u^{\mathrm{proj}}(t,s),\widehat{\lambda}(t,s),\widehat{Z}(t,s)\big) \geq \sup_{u\in\mathcal{U}(s)} H\big(t,t,s,u,\widehat{\lambda}(t,s),\widehat{Z}(t,s)\big) - \varepsilon_{\mathrm{opt}}. \tag{32}$$

Assumption (ii) states that $(\widehat{\lambda},\widehat{Z})$ are accurate for the anchored-at-t continuation problem; for instance,

$$\|\widehat{\lambda}(t,s)-\lambda(t,s)\| \leq \varepsilon_\lambda, \qquad \|\widehat{Z}(t,s)-Z(t,s)\| \leq \varepsilon_Z,$$

where (λ, Z) denote the true (diagonal) adjoint variables for the anchored objective at t.

Step 2: diagonal maximization implies (approximate) diagonal stationarity. Consider first the unconstrained or interior case. Suppose $u^{\mathrm{proj}}(t,s)$ lies in the interior of $\mathcal{U}(s)$. If the map $u \mapsto H(t,t,s,u,\widehat{\lambda}(t,s),\widehat{Z}(t,s))$ is concave, differentiable, and has L-Lipschitz gradient in u (local L-smoothness), then an exact maximizer satisfies

$$\partial_u H\big(t,t,s,u^{\mathrm{proj}}(t,s),\widehat{\lambda}(t,s),\widehat{Z}(t,s)\big) = 0. \tag{33}$$

If we solve only to tolerance (32), standard smooth concave maximization arguments yield an *approximate* stationarity statement of the form

$$\big\|\partial_u H\big(t,t,s,u^{\mathrm{proj}}(t,s),\widehat{\lambda}(t,s),\widehat{Z}(t,s)\big)\big\| \;\leq\; \sqrt{2L\,\varepsilon_{\mathrm{opt}}}. \tag{34}$$

In the constrained case $\mathcal{U}(s)=\{u:\, g_i(u,s)\leq 0\}$, Stage 2 is implemented by an interior-point / log-barrier or projected Newton solve, so the appropriate first-order condition is the KKT condition. Under standard constraint qualification and concavity in u, the maximizer satisfies

$$\partial_u H(\cdots) + \sum_i \eta_i\, \partial_u g_i(u^{\mathrm{proj}}(t,s),s) = 0, \qquad \eta_i \geq 0, \quad \eta_i g_i(u^{\mathrm{proj}}(t,s),s) = 0,$$

up to solver tolerance (and, if a log-barrier with parameter $\mu>0$ is used, up to the usual barrier bias that vanishes as $\mu \downarrow 0$).

Step 3: transfer from estimated to true diagonal adjoints. Let $(\lambda(t,s), Z(t,s))$ denote the true diagonal adjoint variables for the anchored-at-t continuation problem. By smoothness of H in (λ, Z), we have a stability bound

$$\begin{aligned}\Big\|\partial_u H\big(t,t,s,u^{\mathrm{proj}},\lambda(t,s),Z(t,s)\big)\Big\| \leq{}& \Big\|\partial_u H\big(t,t,s,u^{\mathrm{proj}},\widehat{\lambda}(t,s),\widehat{Z}(t,s)\big)\Big\| \\ &+ L_\lambda\,\|\widehat{\lambda}(t,s)-\lambda(t,s)\| + L_Z\,\|\widehat{Z}(t,s)-Z(t,s)\|,\end{aligned} \tag{35}$$

for appropriate local Lipschitz constants L_λ, L_Z. Combining (34) with (35) yields the claimed diagonal Pontryagin optimality/equilibrium condition up to *solver* and *statistical* tolerances.

Step 4: relation to time-consistent equilibrium under non-multiplicative discounting. For non-multiplicative kernels, the continuation problem depends on the anchor time, and equilibrium notions in time-inconsistent control are characterized by *diagonal* (anchor = decision time) first-order conditions (extended/equilibrium HJB / extended PMP; see Ekeland & Lazrak, 2006a; Björk & Murgoci, 2014; Yong, 2012). Stage 2 enforces exactly this diagonal condition by anchoring rollouts at t and maximizing the corresponding Hamiltonian. This completes the proof. □

B. Stage 2 kills Hamiltonian Residual despite of the weak optimality of Stage 1

We report two curves in Figure 6: (i) *Stage 1 (warm-up)* Hamiltonian residual under the parametric policy u_θ; and (ii) *Stage 2 (Adjoint-MC projection)* Hamiltonian residual under the projected control $\hat{u}$ obtained from the BPTT-estimated adjoint. A key observation is that even when Stage 1 is trained only coarsely (i.e., yields a weakly optimal and not yet PMP-consistent policy), the Stage 2 projection step still produces a pronounced reduction in the Hamiltonian residual. This indicates that the projection acts as a structure-enforcing correction: it can substantially tighten the PMP stationarity condition beyond what is achieved by additional warm-up alone, thereby empirically supporting that BPTT-based adjoint estimation combined with projection enforces the Bellman-free time-consistent optimality condition in practice.

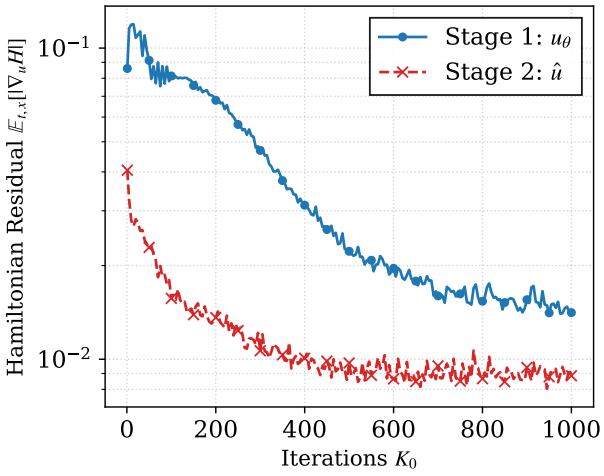


Figure 6. **Hamiltonian stationarity residual across iterations.** We plot the expected Hamiltonian residual $\mathcal{R} = \mathbb{E}[\|\nabla_u H\|_1]$ (log scale) during Stage 1 warm-up and after Stage 2 Adjoint-MC projection, while targeting Case 1 task 3.1.

Table 4. Computational runtime analysis per query state (t, s). We report the total wall-clock time for Stage 2 inference (including BPTT and action synthesis) across different hardware configurations, demonstrating sub-second latency even on CPUs.

Hardware	**Configuration**	**Horizon** (N')	**Batch** (M_{MC})	**Time / Query (s)**
CPU (Intel Xeon)	Base	16	256	0.018
	Scalability Test A	50	1024	0.25
	Scalability Test B	100	4096	1.80
GPU (NVIDIA A100)	Base	16	256	0.03
	Scalability Test A	50	1024	0.04
	Scalability Test B	100	4096	0.17

C. Computational Efficiency.

Table 4 summarizes the wall-clock runtime of the proposed PG-DPO inference in Stage 2. Our Method is computationally extremely efficient, requiring only **0.018 seconds** on a standard CPU and **0.003 seconds** on a GPU per query. This implies that our framework can support high-frequency decision-making (e.g., > 50Hz) on commodity hardware without requiring specialized accelerators.

As shown in Table 4, leveraging the massive parallelism of modern GPUs allows PG-DPO to maintain a **sub-second latency (0.17s)** even in these intensive settings. This confirms that the inference cost scales linearly with problem complexity ($\mathcal{O}(M \cdot N)$), making the method feasible for both lightweight deployment on CPUs and high-precision tasks on GPUs.

D. Additional Experimental Details and Baseline Fairness (Case 1)

D.1. Compute budget and hyperparameters (Case 1)

Table 5 reports the exact training budgets and hyperparameters used in Case 1. These values are fixed *a priori* and applied consistently across random seeds.

Network architectures. We list the exact architectures used in Case 1 to prevent capacity mismatch claims:

- PG-DPO policy network: MLP with input dimension 2 and hidden width 128 for 2 layers (tanh activations), output dimension m.
- PPO policy: Gaussian MLP with shared trunk (2→164→164→164), mean head $\mu(s) \in \mathbb{R}^m$ and global log-std parameter $\log \sigma \in \mathbb{R}^m$.

Table 5. **Case 1 budgets and hyperparameters.** We report (i) training update counts, (ii) batch sizes, (iii) rollout lengths, and (iv) major optimizer settings.

Method	Budget / hyperparameters
PG-DPO (Stage 1 warm-up)	Adam steps: 500; batch trajectories: 256 per step; rollout steps $M = 64$; learning rate 10^{-3}; grad clip 1.0. Variance reduction: antithetic ($\pm$) and Richardson extrapolation enabled; control variate enabled with EMA coefficient 0.98 and coefficient clipping 10.0.
PG-DPO (Stage 2 projection / costate)	For each time grid row ($N_T = 64$): costate repeats 256 with sub-batch 256. Each repeat uses antithetic pairing ($\pm$) when estimating pathwise sensitivities.
PPO	Iterations: 1000; parallel environments 256; rollout steps $M = 64$ (total env-steps: $1000 \times 256 \times 64 = 16.384$M). Update epochs: 5; minibatch size: 2048; learning rate $3 \cdot 10^{-4}$; GAE $\lambda = 0.95$; clipping $\epsilon = 0.2$; value coefficient 0.5; entropy coefficient 0.0.
PINN (KAN)	Adam steps 5000 with batch 4096; L-BFGS steps 1000 (strong Wolfe line search); learning rate $5 \cdot 10^{-4}$; terminal penalty weight 20; causal weighting parameter 100.
Deep BSDE	Adam steps 3000; batch 256; time steps 64; learning rate 10^{-4}.

- PPO value network: MLP 2→164→164→1 (tanh).
- PINN (KAN): 3 Chebyshev KAN layers with hidden width 161 and degree 5.
- Deep BSDE: two MLPs (gradient network and Y_0 network) with 3 hidden layers of width 164 (tanh).

D.2. PPO objective and implementation details

Anchored reward shaping and $\gamma = 1$. We define the per-step reward

$$r_k \;=\; \mathbf{1}[t_k < T] \cdot D(t_0, t_k) \cdot \ell(u_k) \cdot \Delta t, \tag{36}$$

and add the terminal reward at the final step:

$$r_{M-1} \;\leftarrow\; r_{M-1} + D(t_0, T)\, g(X_T). \tag{37}$$

Because discounting is already included via $D(t_0, \cdot)$ in r_k, PPO uses an undiscounted return with $\gamma = 1$. This removes ambiguity about how non-exponential discounting interacts with RL discount factors.

GAE and targets. Let $V_\phi(s_k)$ denote the value prediction. We compute generalized advantage estimates with $\gamma = 1$:

$$\delta_k = r_k + V_\phi(s_{k+1}) - V_\phi(s_k), \tag{38}$$

$$A_k = \delta_k + \lambda A_{k+1}, \qquad \lambda = 0.95, \tag{39}$$

and the value target is $\hat{R}_k = V_\phi(s_k) + A_k$. We normalize advantages within the collected batch.

PPO surrogate. The policy is Gaussian, $\pi_\theta(u|s) = \mathcal{N}(\mu_\theta(s), \mathrm{diag}(\sigma_\theta^2))$ with learnable $\log \sigma$. The clipped objective is

$$L^{\text{clip}}(\theta) = \mathbb{E}\Big[\min\big(\rho_k(\theta)A_k,\ \mathrm{clip}(\rho_k(\theta), 1-\epsilon, 1+\epsilon)A_k\big)\Big], \quad \rho_k(\theta) = \frac{\pi_\theta(u_k|s_k)}{\pi_{\theta_{\text{old}}}(u_k|s_k)}, \tag{40}$$

with $\epsilon = 0.2$. The total loss is

$$\mathcal{L}(\theta, \phi) = -L^{\text{clip}}(\theta) + c_v\, \mathbb{E}\big[(V_\phi(s_k) - \hat{R}_k)^2\big] - c_e\, \mathbb{E}\big[\mathcal{H}(\pi_\theta(\cdot|s_k))\big], \tag{41}$$

with $c_v = 0.5$ and $c_e = 0.0$.

Evaluation policy. At evaluation time, PPO uses the deterministic mean action $u = \mu_\theta(s)$ (no sampling), evaluated on the same (t, x) grid as other methods.