POLICY GRADIENTS FOR AVERAGE PATH-DEPENDENT UTILITY IN CONTINUING MDPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning typically represents desired behavior through additive per-step rewards, which can be difficult to design, and may fail to capture preferences that depend on more complicated patterns or entire trajectories. Unlike Multi-Objective RL, Average Reward RL, or General Utility RL, we study average path-dependent utility (APDU), a long-run average objective for continuing tasks in which performance is evaluated using trajectory-level utility. We restrict our attention to ergodic Markov decision processes and parametrized Markovian policies. For this setting, we first consider the case where the utility functional is known. We show that the APDU objective admits a policy-dependent proxy-reward whose expectation under the stationary state–action distribution exactly recovers the original objective. This proxy reward enables us to define a differential action-value function and derive an average-utility policy-gradient theorem. We then address the more challenging (and versatile) setting in which the utility functional is unknown and only trajectory-level utility feedback is observable. We introduce an action-value quantity that can be estimated directly from such feedback, establish its asymptotic relationship with the known utility action-value function, and derive a corresponding policy-gradient theorem. Building on these results, we propose Average Utility Actor–Critic (AUAC), an algorithm for optimizing APDU from trajectory-level feedback. Experiments validate our theoretical findings and demonstrate the effectiveness of AUAC on path-dependent control tasks.

1 INTRODUCTION

Reinforcement learning typically represents desired behaviour through additive per-step rewards. In many continuing applications, however, a meaningful immediate reward is difficult to specify or fails to capture preferences that depend on temporal order or longer trajectory patterns. Consider, for example, an edge router or server with two operating modes, A and B, that must continually decide whether to remain in its current mode or switch. Its operator may wish to penalise a temporal pattern such as ABB, or to keep the long-run mode distribution close to a prescribed target. Such objectives are naturally evaluated at the trajectory level, even when the controller itself observes only the current state and follows a Markov policy (for hardware constraints).

We study this problem in a finite continuing Markov decision process under a stationary, differentiable Markov policy π_θ. Let $\tau_{0:t} = (S_0, A_0, S_1, A_1, \ldots, S_t)$ denote a roll-out trajectory, and let $u(\tau_{0:t})$ be its accumulated utility. The utility may be non-Markovian: it may depend on the path leading up to S_t, rather than only on the current state or state–action pair. Because a continuing process has no natural episode boundary, a finite segment should be evaluated without discarding its preceding history. We therefore define the utility gained over the next T steps from time t by

$$\Delta_t^\tau u(T) := u(\tau_{0:t+T}) - u(\tau_{0:t}). \tag{1}$$

Our objective is the *average path-dependent utility* (APDU) J^u under policy pi_θ

$$J^u(\theta) \equiv \rho_\theta^u := \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_\theta[\Delta_t^\tau u(T)] \overset{notation}{=} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_\theta\left[u(\tau_T)\right], \tag{2}$$

where t is an arbitrary time point in the stationary continuing process. Section 4 gives the formal definition and assumptions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The standard average-reward objective is recovered when utility is additive. Indeed, if $u(\tau_{0:T}) = \sum_{t=1}^{T} r(S_t, A_t)$, then

$$J(\theta) \equiv \rho_\theta = \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_\theta \left[\sum_{t=1}^{T} r(S_t, A_t) \right]. \tag{3}$$

Under the ergodicity assumptions adopted in this paper (see Section 3), the policy induces a stationary state–action occupancy measure μ_θ, and that allows one to express the time-average in Eq. 3 as a weighted average over states:

$$J(\theta) = \sum_{s,a} \mu_\theta(s, a) r(s, a). \tag{4}$$

Thus, the conventional average-reward objective is a linear functional of μ_θ.

We consider the broader class of APDU objectives that admit an occupancy-measure representation

$$J^u(\theta) = \mathcal{F}(\mu_\theta), \tag{5}$$

where $\mathcal{F}$ need not be linear and may be unknown to the learner. Equation 5 is a structural assumption defining the class studied here; it does not follow from ergodicity alone, nor does every history-dependent utility necessarily admit such a representation. This class nevertheless includes distributional objectives and temporal-pattern objectives such as those used in our experiments. Optimising it is challenging because a nonlinear or history-dependent utility generally cannot be represented by a fixed Markovian per-step reward; furthermore in APDU setting, the utility as well as the functional may not be known explicitly.

Contributions. First, when $\mathcal{F}$ is known, we construct a *policy-dependent proxy reward* $r^u_\theta(s, a)$ satisfying

$$J^u(\theta) = \sum_{s,a} \mu_\theta(s, a) r^u_\theta(s, a). \tag{6}$$

The proxy reward has the same policy-gradient direction as the shadow reward of Zhang et al. (2020), while its stationary expectation also recovers the APDU objective itself. It yields a differential action-value function q^u_θ and an average-utility policy-gradient theorem analogous to the standard average-reward result (Sutton et al., 1998).

Second, we address the setting in which $\mathcal{F}$ is unknown and only trajectory-level utility feedback is available. Using increments of the form in Eq. 1, we define a trajectory-based differential action-value function q^τ_θ that can be estimated without knowing the utility functional. We establish its asymptotic relationship with q^u_θ, derive the corresponding policy-gradient theorem, and develop the *Average Utility Actor–Critic* (AUAC) algorithm. Finally, experiments on a two-state MDP and a continuing queue-control problem validate the theory and evaluate the resulting algorithms on distributional, entropy-rate, and temporal-pattern objectives.

2 Related Work

APDU lies at the intersection of average-reward, utility-based, and general-utility reinforcement learning. In this section we briefly mention the most relevant papers. A more detailed and comprehensive comparison is provided in Appendix A.

Average-reward reinforcement learning. Average-reward RL evaluates steady-state performance in continuing MDPs without introducing a discount factor (Puterman, 2014). Existing value- and policy-based methods optimize the long-run average of additive per-step rewards (Schwartz, 1993; Abounadi et al., 2001; Wan et al., 2021; Jin et al., 2024; Wan et al., 2024; Saxena et al., 2023; Powell et al.; Hisaki & Ono, 2024). APDU recovers this objective when $u(\tau_T) = \sum_{t=0}^{T-1} r(s_t, a_t)$, but additionally represents nonlinear preferences that depend on temporal order and trajectory history.

Multi-objective (or utility-based RL). MORL represents preferences by applying a utility or scalarization function to vector-valued rewards (Roijers et al., 2013; Hayes et al., 2022). Most methods apply linear or nonlinear utilities to individual reward vectors($J = \mathbb{E}[\sum \gamma^{t-1} u(r_t)]$), discounted cumulative returns $J = \mathbb{E}[u(\sum \gamma^{t-1} r_t)]$, or long-run average reward vectors $J = u(\rho)$. (Pirotta et al., 2015; Xu et al., 2020; Hayes et al., 2021; Reymond et al., 2023; Röpke et al., 2024; Peng et al., 2025; Cheung, 2019; Siddique et al., 2020; Agarwal & Aggarwal, 2023). These formulations generally assume a predefined per-step reward vector and an explicitly specified scalarization of its aggregate. In contrast, APDU permits feedback on complete trajectories without requiring preferences to be decomposed into fixed-dimensional per-step rewards.

General utilities and shadow rewards. General-utility RL optimizes an explicitly specified functional $J = \mathcal{F}(\mu)$ of a discounted or stationary occupancy measure (Hazan et al., 2019; Tarbouriech & Lazaric, 2019; Zhang et al., 2020; Zahavy et al., 2021; Geist et al., 2022; Barakat et al., 2023). General-utility RL assumes this functional to be known, while our APDU does not. Several methods construct a policy-dependent shadow reward $r_\theta^{\mathrm{sh}}(s,a) = \partial \mathcal{F}(\mu)/\partial \mu(s,a)$, which yields the gradient of the original objective (Zhang et al., 2020; 2021; 2022; Barakat et al., 2023). However, its stationary average generally differs from the objective: $\langle \mu_\theta, r_\theta^{\mathrm{sh}} \rangle \neq \mathcal{F}(\mu_\theta)$. Using a novel construction based on a positively homogeneous functional of degree one (see Appendix B), our proxy reward matches both the policy gradient and the APDU gain,

3 PRELIMINARIES

In this section we briefly recap some of the theoretical underpinnings and formalism needed in this paper. For more details we refer to Puterman (2014); Sutton et al. (1998))

Continuing MDPs and Ergodicity Assumption We consider a finite continuing Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r)$, where $\mathcal{S}$ is a (finite) state space, $\mathcal{A}$ is a (finite) action space, $P(\cdot \mid s,a)$ is the transition kernel on $\mathcal{S}$, and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the (expected) one-step reward function. A stationary stochastic policy is a mapping $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, where $\Delta(\mathcal{A})$ denotes the probability simplex over $\mathcal{A}$; we write $\pi(a \mid s)$ for the probability of choosing action a in state s; A continuing MDP has no terminal state or natural episode boundary, and the interaction is assumed to proceed indefinitely. Throughout this paper, we consider a differentiable parameterized policy family $\Pi_\Theta = \{\pi_\theta : \theta \in \Theta\}$, where $\Theta \subseteq \mathbb{R}^d$ is the parameter space. A trajectory $\tau = \{s_0, a_0, r_1, s_1, \ldots, s_T, r_T\}$ of length T list states, actions and rewards. Given a stationary policy π, the induced Markov chain on $\mathcal{S}$ has transition kernel

$$P_\pi(s' \mid s) \;=\; \sum_{a \in \mathcal{A}} \pi(a \mid s)\, P(s' \mid s, a).$$

We assume that, for every policy in a policy class $\theta \in \Theta$, the Markov chain induced by π_θ is irreducible and aperiodic. Consequently, it admits a unique stationary distribution d_θ and converges to d_θ from any initial state. This **ergodicity assumption** is standard in the policy gradient literature Sutton et al. (1998); Sutton (2000); Konda & Tsitsiklis (1999); Baxter & Bartlett (2001).

Occupancy measure view Under the ergodicity assumption, the long-run *state* and *state-action occupancy measure* are defined as

$$\mu_\pi(s) \;=\; \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \Pr(s_t = s \mid \pi), \quad \text{and} \quad \mu_\pi(s,a) \;=\; \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \Pr(s_t = s,\, a_t = a \mid \pi),$$

and they are linked by: $\mu_\pi(s,a) = \mu_\pi(s)\,\pi(a \mid s)$ and $\mu_\pi(s) = \sum_a \mu_\pi(s,a)$. Hence $\mu_\pi(s,a)$ gives the asymptotic frequency of visiting state s and taking action a when following π. Finally, we use $\widehat{\mu}_\tau$ to denote the **empirical occupancy measure** from one specific trajectory $\widehat{\mu}_\tau(s) = \frac{1}{T}\sum_1^T \mathbb{1}(s_t = s|\tau)$.

Gain, Relative value function and Poisson Equation For the standard continuing MDP (which receives an immediate reward r_t at every step), the fundamental object is no longer a value function

but a scalar—the **long-run average reward per step** under policy π starting from ANY state s:

$$\rho_\pi(s) \;=\; \lim_{T\to\infty} \frac{1}{T}\, \mathbb{E}_\pi \left[\sum_{t=1}^{T} r_t \;\middle|\; s_0 = s \right]. \tag{7}$$

Under the *ergodicity* assumption this limit exists, is finite, and is **independent of the starting state** s. Hence, in that case $\rho_\pi(s) = \rho_\pi$ for all s, and this single scalar summarises the quality of π.

Similar to discounted MDP, the *differential value function* $h_\pi(s)$ (also called the *bias function*) measures how much better or worse each state s is relative to the average reward ρ_π: $h_\pi(s) = \mathbb{E}_\pi[\sum_{t=0}^{\infty}(r_t - \rho_\pi) \mid s_0 = s]$,

Similarly, the *differential q function* is $q_\pi(s,a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty}(r_t - \rho_\pi) \mid s_0 = s, a_0 = a]$, and measures how much better or worse each state-action (s,a) is relative to the average reward ρ_π.

The Poisson equation The analogue of the Bellman equation is the *Poisson equation* linking both value functions:

$$q_\pi(s,a) + \rho_\pi = r(s,a) + \sum_{s'} p(s' \mid s,a)\, h_\pi(s') \qquad \forall\, s,a. \tag{8}$$

Average-Reward Policy Gradient Assuming that the reward $r(s,a)$ is not conditional on the policy π, then under the ergodicity assumption, π_θ induces a unique stationary distribution $\mu_\theta(s)$ and occupancy measure $\mu_\theta(s,a) = \mu_\theta(s)\pi_\theta(a \mid s)$, and the following theorem holds.

Theorem 1 (Average-reward policy-gradient theorem (Sutton, 2000))**.**

$$\begin{aligned} \nabla_\theta \rho(\theta) &= \sum_s \mu_\theta(s) \sum_a \pi_\theta(a \mid s)\, q_\theta(s,a)\, \nabla_\theta \log \pi_\theta(a \mid s) \\ &= \mathbb{E}_{(S,A)\sim\mu_\theta} \left[q_\theta(S,A)\, \nabla_\theta \log \pi_\theta(A \mid S) \right]. \end{aligned} \tag{9}$$

□

Any state-dependent baseline $b(s)$ may be subtracted from $q_\theta(s,a)$ without changing the gradient.

The main contribution in this paper is to show that a similar theorem holds for a more general class of history-dependent utilities u.

4 Average Path-Dependent Utility (APDU)

Recall that our goal is to extend the objective in Eq. 3 to include a more general class of (history-dependent) utilities. Specifically, we move beyond mere immediate rewards, and introduce (possibly non-linear) **utility functions** that assign a utility $u(\tau_T)$ to any path τ. As explained in Section 1, since the process is continuously running, there is no designated starting point, and it is therefore more natural to compute the time-averaged increase of utility as specified in Eq. 1

$$\frac{\Delta_t u(T)}{T} = \frac{u(t_0 : t+T) - u(t_0 : t)}{T}, \tag{10}$$

yielding the following optimisation objective:

$$J^u(\theta) := \lim_{T\to\infty} \mathbb{E}_\theta \left[\frac{u(\tau_{t_0:t}) - u(\tau_{t_0:t+T})}{T} \right] \tag{11}$$

Since the choices of $t_0 < t$ are immaterial (as long as difference is suffiently large to wash out any transient behaviour), we will simplify the notation and just write:

$$J^u(\theta) := \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_\theta[u(\tau_T)], \tag{12}$$

where it is tacitly understood to be shorthand for Eq. (11). In line with the notation introduced above, we will also denote $J^u(\theta)$ as $\rho^u(\theta)$.

5 INTRODUCING PROXY REWARDS

Motivation for the introduction of a proxy reward Recall that we want to extend the Policy Gradient Theorem (PGT, see Eq. 9). In the standard approach to PGT one assumes that the target objective can be expressed as: $J(\theta) = \sum_{s,a} \mu_\theta(s,a)\, r(s,a)$ where $r(s,a)$ represents the expected reward for taking action a in state s. Furthermore, since the reward is issued by the environment, it does not depend on the agent's policy, i.e. it does not depend on θ. As a consequence, the gradient simplifies:

$$\nabla_\theta J(\theta) = \sum_{s,a} \left(\nabla_\theta \mu_\theta(s,a)\right)\, r(s,a) + \sum_{s,a} \mu_\theta(s,a)\, \underbrace{\nabla_\theta r(s,a)}_{=0} = \sum_{s,a} \nabla_\theta \mu_\theta(s,a)\, r(s,a). \tag{13}$$

Subsequently, the Poisson equation is invoked to remove the often intractable gradient $\nabla_\theta \mu_\theta(s,a)$ in the RHS. In the case of the more general utility in the objective Eq. 12 it will still possible to express J as a weighted mean, but now there is no guarantee that the reward will be independent of the policy. Hence, in this more general setting, we are forced to consider a reward that is θ-dependent: $J(\theta) = \sum_{s,a} \mu_\theta(s,a)\, r_\theta(s,a)$. Notice however that for the simplification in Eq.13 to hold, it suffices that $\sum_{s,a} \mu_\theta(s,a)\, \nabla_\theta r_\theta(s,a) = 0$.

This observation explains the rationale underpinning the approach in this paper: given $J^u(\theta)$ where

$$J^u(\theta) := \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{\tau\sim\pi_\theta}[u(\tau_T)] \tag{14}$$

we need to define a **proxy reward** $r^u_\theta(s,a)$ such that the following two conditions hold:

$$\text{Proxy condition \#1 on } r^u\text{:} \qquad \sum_{s,a} \mu_\theta(s,a)\, r^u_\theta(s,a) = J^u(\theta), \tag{15}$$

$$\text{Proxy condition \#2 on } r^u\text{:} \qquad \sum_{s,a} \mu_\theta(s,a)\, \nabla_\theta r^u_\theta(s,a) = 0. \tag{16}$$

It turns out that homogeneous functions (see below) provide a convenient in-road to this problem.

Homogeneous functions of degree 1 satisfy the proxy conditions Recall that a homogeneous function F (in n variables) of degree p satisfies $F(\lambda x_1, \lambda x_2, \ldots, \lambda x_n) = \lambda^p F(x_1, x_2, \ldots, x_n)$. For such functions, Euler's theorem tells us:

$$\sum_{i=1}^n x_i \frac{\partial F}{\partial x_i} = p\, F(x_1, x_2, \ldots, x_n) = p\, F(x), \qquad \text{or, if } p = 1, \quad \sum_{i=1}^n x_i \frac{\partial F}{\partial x_i} = F(x). \tag{17}$$

Hence, **when the objective $J(\theta)$ can be written as a degree-1 homogeneous functional F of the stationary distribution**, i.e. $J^u(\theta) = F(\mu_\theta)$, then it follows from Euler's theorem (cf. Eq. 17) that:

$$F(\mu_\theta) = \sum_{s,a} \mu_\theta(s,a)\, \frac{\partial F(\mu_\theta)}{\partial \mu_\theta(s,a)}. \tag{18}$$

We therefore arrive at the following conclusion:

Lemma 1. *If $F(\mu_\theta)$ is differentiable and **homogeneous of degree 1**, and we define the proxy reward by:*

$$r^u_\theta(s,a) := \left.\frac{\partial F(\mu)}{\partial \mu(s,a)}\right|_{\mu=\mu_\theta}. \tag{19}$$

then the two proxy conditions (Eqs. 15 and 16) hold. □

In the Appendix B we show how to extend this observation to more general functionals. This construction introduces an additional corrective term (see Eq. 21) that vanishes when F is homogeneous of degree 1. We now first state the relevant assumptions followed by the formal theorem statement.

- **Assumption 1: occupancy-measure representation:** The APDU objective depends on π_θ through its stationary state-action occupancy measure μ_θ. Specifically, there exists a (sufficiently smooth) functional $\mathcal{F}$ defined on the set of feasible occupancy measures such that
$$J^u(\theta) = \mathcal{F}(\mu_\theta). \tag{20}$$
The functional $\mathcal{F}$ may be unknown to the learner;
- **Assumption 2: Ergodicity Assumption** See Preliminaries.

Theorem 2. *Given the above assumption, the following statements for the APDU objective hold:*

1. ***Existence of proxy reward*** *For each state-action (s, a) we can define a **proxy reward** by:*
$$r^u_\theta(s,a) := \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s,a)} + \left(\rho^u_\theta - \sum_{s',a'} \mu(s',a') \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s',a')} \right) \tag{21}$$
which satisfies the proxy conditions:
$$J^u(\theta) = \sum_{s,a} \mu_\theta(s,a)\, r^u_\theta(s,a) \quad \textit{and} \quad \sum_{s,a} \mu_\theta(s,a)\, \nabla_\theta r^u_\theta(s,a) = 0.$$
Notice how the bracketed term in the RHS of Eq. 21
2. ***Existence of q-function associated with proxy reward*** *For the above proxy reward we can now determine the corresponding $q^u_\theta(s,a)$ by invoking the Poisson equation:*
$$q^u_\theta(s,a) = \mathbb{E}[r^u_\theta(s,a) - \rho^u_\theta + \sum_{s'} p(s' \mid s,a)\, h^u_\theta(s')]$$
where $h^u_\theta(s)$ satisfies the analogous Poisson equation.
3. ***PGT for J^u*** *In terms of the above concepts, the following version of the standard PGT holds:*
$$\nabla_\theta J^u(\theta) = \sum_s \mu_\theta(s) \sum_a q^u_\theta(s,a)\, \pi_\theta(a \mid s)\, \nabla_\theta \log \pi_\theta(a \mid s). \tag{22}$$

□

Relation with Shadow Reward has been described in related works Sec. 2.

6 TRAJECTORY-BASED DIFFERENTIAL q^τ-FUNCTION

The construction in Thm. 2 is helpful when the functional F is known explicitly. In many applications, however, the utility is observed only at trajectory level. For example, a human evaluator may assign a single score $u(\tau_T)$ to an entire trajectory, while neither an immediate reward nor the functional F is known. The purpose of this section is therefore to introduce a *trajectory-based differential q-function* that can be estimated from observed trajectories and their associated utilities.

An important issue arises because the utility may depend on events that occur over several time steps. Consequently, artificially cutting a continuing trajectory into independent pieces may destroy relevant information immediately before the cutting point. We avoid this problem by evaluating changes in utility using prefixes of the *same* continuing trajectory. Let $\tau_{0:t} = (S_0, A_0, S_1, A_1, \ldots, S_t)$ denote the prefix of a continuing trajectory up to time t. For a time t inside the continuing stationary process and a horizon $L > 0$, define the incremental utility
$$\Delta u_t(L) := u(\tau_{0:t+L}) - u(\tau_{0:t}). \tag{23}$$
Notice the importance of using two prefixes of the *same* trajectory: we do not restart the utility evaluator at time t. As a consequence, any relevant history before time t remains implicitly available to the utility function. This observation motivates the following definition of **trajectory-based differential q-function**. Specifically, let
$$\rho^u_\theta = J^u(\theta) = \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_\theta \left[u(\tau_T) \right] \tag{24}$$

denote the long-run average utility per time step. We then define the **trajectory-based differential action-value function** by

$$q_\theta^\tau(s,a) := \lim_{T\to\infty} \mathbb{E}_\theta \left[\Delta u_t(T) - T\rho_\theta^u \mid S_t = s,\ A_t = a\right]. \tag{25}$$

Here t is understood to be a time point in the stationary continuing process. This definition is analogous to the standard differential action-value function in an average-reward MDP. It measures the long-run relative advantage or disadvantage associated with taking action a in state s, relative to the average utility rate ρ_θ^u.

Based on asymptotic analysis as $T \to \infty$ (the complete theoretical derivation can be found in the Appendix C) we obtain:

$$q_\theta^\tau(s,a) = q_\theta^u(s,a) + C_\theta + B_\theta(s,a), \tag{26}$$

where

$$B_\theta(s,a) := \lim_{L\to\infty} \mathbb{E}_\theta \left[R_{t,L} \mid S_t = s, A_t = a\right], \tag{27}$$

and the correction

$$C_\theta := \frac{1}{2} \operatorname{tr}\left[H_F(\mu_\theta)\Sigma_\theta\right]$$

represents the interaction between the nonlinearity of F and the stochastic fluctuations of the empirical occupancy measure. (For the precise definition of H_F and Σ_θ, see Appendix C.2).

In particular, if $\Delta u_t(T) = F(\widehat{\mu}_{t,T})\,T$, then $B_\theta(s,a) = 0$ and

$$q_\theta^\tau(s,a) = q_\theta^u(s,a) + C_\theta. \tag{28}$$

(A simple concrete example where $B_\theta(s,a) \neq 0$ is provided in Section C.7.) Because C_θ only depends on the policy, regardless of initial (s,a), it can be viewed as harmless baseline in advantage function. Then we have the corresponding policy gradient theorem:

Theorem 3. *If, in addtion to Assumptions 1 and 2 in Page 5 we also assume that*

$$\sum_s \mu_\theta(s) \sum_a B_\theta(s,a)\,\pi_\theta(a \mid s)\,\nabla_\theta \log \pi_\theta(a \mid s) = 0 \tag{29}$$

then *the following version of PGT holds:*

$$\nabla_\theta J_\theta^u = \sum_s \mu_\theta(s) \sum_a q_\theta^\tau(s,a)\,\pi_\theta(a \mid s)\,\nabla_\theta \log \pi_\theta(a \mid s) \tag{30}$$

□

Remark It is important to point out that there are plenty of interesting cases where Eq. 29 hold. For instance, any utility depending on a finite-memory pattern-count would result in $B_\theta(s,a)$ NOT depending on a and therefore condition Eq. 29 would hold.

7 ALGORITHMS

When the functional $\mathcal{F}(\mu)$ is known, ProxyAC converts it into a policy-dependent per-step reward and applies average-reward actor–critic. When only black-box trajectory utility feedback is available, our Average Utility Actor–Critic (AUAC) learns directly from incremental utility observations.

7.1 PROXYAC WITH A KNOWN FUNCTIONAL

Given an occupancy estimate $\widehat{\mu}$, let $g = \nabla_\mu \mathcal{F}(\widehat{\mu})$. ProxyAC constructs the gain-matched reward

$$r_{\widehat{\mu}}^u(s,a) = g(s,a) + \mathcal{F}(\widehat{\mu}) - \langle \widehat{\mu}, g \rangle. \tag{31}$$

At $\widehat{\mu} = \mu_\theta$, this reward matches both the APDU objective and its policy gradient. We based on Average AC in Chapter 13Sutton et al. (1998) give a algorithm ProxyAC. For comparison, Shadow Reward Eq. 31 and ShadowAC-Biased can be applied in the same ProxyAC framework. Full description and pseudocode is provided in Appendix D.1.

7.2 Average Utility Actor–Critic with unknown Functional

Inspired by Theorem 3, AUAC(Algo. 1 and details in Sec D.2) requires only incremental utility feedback Eq. 25. Here, $\Delta u_{t_i}(L)$ is scalar feedback from the black-box utility function; AUAC need not know the required history length($t_i >= k$ is for sure). The black box may internally use a prefix overlapping the preceding window, whose direct contribution is removed by subtraction. We use contiguous trajectories to preserve history and non-overlapping evaluation windows to reduce dependence.

Algorithm 1 Average Utility Actor–Critic (AUAC)

Require: Black-box incremental utility Δu; batch length $T = NH$; window length H; warm-up W; EMA rate λ; learning rates α, β

1: Initialize policy π_θ, critic q^τ_ϕ, and utility rate ρ^u
2: **for** each iteration **do**
3: Set $\theta_0 \leftarrow \theta$ and run π_{θ_0} for W warm-up steps
4: Collect a connected trajectory of length T under π_{θ_0}
5: Set $t_i = (i-1)H$and query $\Delta u^i \leftarrow \Delta u_{t_i}(L)$ for $i = 1, \ldots, N$
6: Estimate the batch utility rate:

$$\widehat{\rho}^u \leftarrow \frac{\Delta u_0(T)}{T}$$

7: Update $\rho^u \leftarrow (1-\lambda)\rho^u + \lambda\widehat{\rho}^u$
8: Form critic targets $Y_i \leftarrow \Delta u^i - H\rho^u$
9: Update the critic using

$$\mathcal{L}(\phi) = \frac{1}{N}\sum_{i=1}^{N}\left(q^\tau_\phi(s_{t_i}, a_{t_i}) - Y_i\right)^2$$

10: Update the actor:

$$\theta \leftarrow \theta_0 + \frac{\beta}{T}\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(a_t \mid s_t)|_{\theta=\theta_0}\, q^\tau_\phi(s_t, a_t)$$

11: **end for**

8 Experiments

8.1 Two-State MDP

For the 2-state MDP introduced in Section 1, we introduce a 1-parameter policy π_θ (where $0 < \theta < 1$) defined as follows:$\pi_\theta(switch|A) = 1$ and $\pi_\theta(switch|B) = \theta$ and $\pi_\theta(stay|B) = 1-\theta$. We consider three path-dependent objectives: (i) reverse KL divergence, $J^u(\theta) = -\mathrm{KL}(d_{\mathrm{tar}}\|d_\theta)$, with optimum $\theta^\star = 1/3$; (ii) the long-run frequency of the temporal pattern ABB, with optimum $\theta^\star = \sqrt{2}-1$; and (iii) the entropy rate $J^u(\theta) = -\sum_{s,a}\mu_\theta(s,a)\log\frac{\mu_\theta(s,a)}{d_\theta(s)}$, with optimum $\theta^\star = (3-\sqrt{5})/2$. (more details in Appendix E).

Figures 1(a,b,g,h) show that ProxyAC and ShadowAC have nearly identical learning behavior on the KL-divergence and entropy-rate objectives, consistent with their equivalent policy gradients. Both converge to the analytical optima. AUAC approaches the same solutions without access to the analytical utility functional, although its finite trajectory windows introduce a small estimation bias. APDU-Reinforce learns substantially more slowly and exhibits higher variance.

Figures 1(d,e) evaluate the temporally dependent ABB-pattern utility. Because its functional representation is linear, ProxyAC, ShadowAC, and ShadowAC-Biased use the same reward and converge to the analytical optimum. In contrast, ProxyAC-PO, which ,in ProxyAC, applies the raw pattern reward rather than first degree proxy reward, overshoots the optimum because the resulting reward is non-Markovian with respect to the critic input. AUAC reaches the optimum without state augmentation because its trajectory-level targets retain the history required to evaluate the pattern.

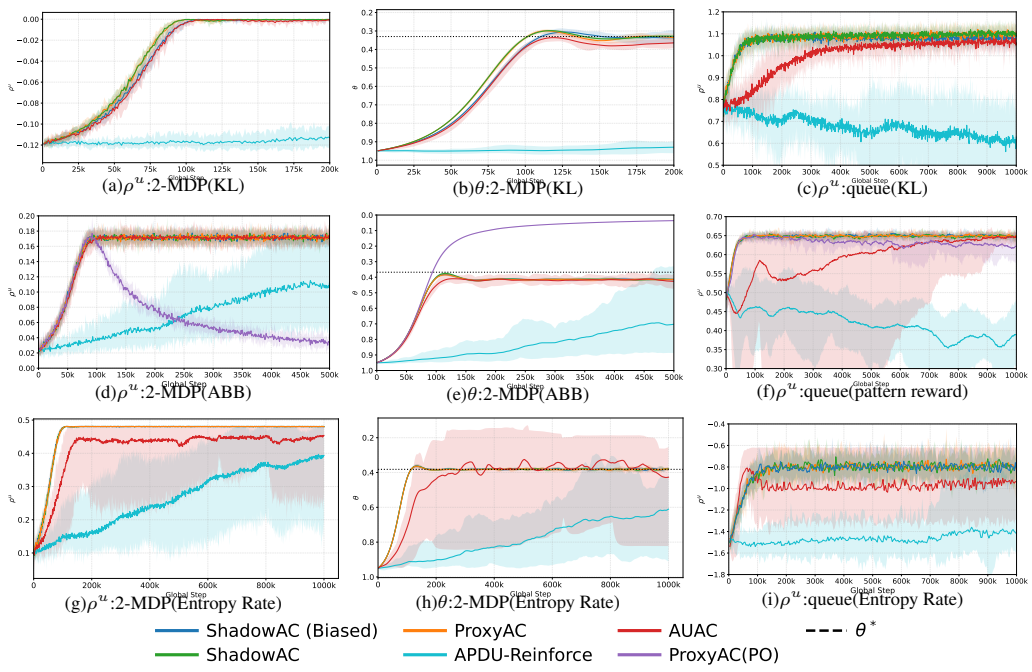


Figure 1: Experimental results across the two-state MDP and neural queue-control task. Rows correspond to KL-divergence objectives (a–c), temporal pattern utilities (d–f), and entropy-rate objectives (g–i). The first two columns show the APDU objective and policy parameter, respectively, on the two-state MDP; the third column shows the APDU objective on queue control.

8.2 Queue Control

In the continuing access-control task of Sutton et al. (1998), each state details the number of available servers and the priority of the arriving customer. The controller accepts or rejects the customer, balancing immediate reward against the value of preserving capacity for future arrivals. We consider three utilities corresponding to the rows of Figure 1: a KL-divergence penalty on the empirical client–action distribution, an entropy-rate penalty on the client–action process, and a temporal penalty on repeated decision patterns. These objectives test distributional, transition-dependent, and local temporal preferences, respectively (more details in Appendix E).

Figures 1(c,f,i) show that using explicit proxy or shadow rewards generally converge faster than AUAC. But AUAC learns effective policies using only trajectory-level utility feedback and remains competitive on the KL-divergence and temporal-pattern objectives. The entropy-rate objective is more difficult: estimating transition frequencies from short trajectory windows produces a noisy critic target, and AUAC can stabilize at a suboptimal policy. APDU-Reinforce is consistently slower and more variable across the three objectives. In addition, similar to ABB in two-state task, ProxyAC(PO) which is purple also is suboptimal like before in Figure 1(f).

9 Conclusion

We introduced average path-dependent utility (APDU), a long-run objective for optimizing trajectory-level preferences in continuing, ergodic MDPs. For known utility functions, we constructed a policy-dependent proxy reward that matches both the APDU objective and its policy gradient, and derived the corresponding differential value function and policy-gradient theorem. For unknown utilities, we developed a trajectory-based value estimator and the Average Utility Actor–Critic (AUAC) algorithm, which requires only observed utility feedback. Experiments demonstrate that AUAC can optimize nonlinear and history-dependent objectives without explicit knowledge of their functional form, although finite trajectory windows can introduce additional variance and bias.

10 AI USE STATEMENT

In this work, we used generative AI tools to assist with verification of hypotheses, theoretical developments, and algorithms developed by the authors, and to assist with the implementation of baseline methods. We have not used generative AI tools to generate the research hypotheses, theoretical contributions, or proposed algorithms, or to interpret the experimental results; the remaining required-disclosure tasks are not applicable to this work.

Additionally, we used generative AI tools to assist with drafting and editing parts of the manuscript, generating initial baseline code, and converting one author-designed handcrafted figure into a digital version in Appendix.Fig 38 . All other figures were generated locally by the authors.

We have reviewed all AI-assisted work. AI-assisted verification of hypotheses, theories, and algorithms was used only as an additional checking mechanism, and all resulting suggestions were independently assessed by the authors. AI-generated baseline code was reviewed line by line by the authors and checked for correctness before being used in the experiments. The AI-assisted figure was based on an author-created design, and its final content was reviewed by the authors for accuracy and consistency with the intended presentation.

We take responsibility for the final content of this work, including text, claims, code, figures, and other artifacts produced with the aid of generative AI.

11 REPRODUCIBILITY STATEMENT

We support reproducibility by stating the APDU setting, policy class, assumptions, and theoretical results explicitly in the main text. Complete proofs and auxiliary derivations are provided in the appendix. The AUAC and ProxyAC are specified in pseudocode. The experimental appendix provides detailed descriptions of the two-state MDP and queue-control environments, utility functions, baselines, model architectures, hyper-parameters, trajectory lengths, and evaluation protocol.

Code availability. The source code, configuration files, and scripts required to reproduce the experiments will be made publicly available upon acceptance.

REFERENCES

Jinane Abounadi, Dimitrib Bertsekas, and Vivek S Borkar. Learning algorithms for markov decision processes with average cost. *SIAM Journal on Control and Optimization*, 40(3):681–698, 2001.

Mridul Agarwal and Vaneet Aggarwal. Reinforcement learning for joint optimization of multiple rewards. *Journal of Machine Learning Research*, 24(49):1–41, 2023.

Anas Barakat, Ilyas Fatkhullin, and Niao He. Reinforcement learning with general utilities: Simpler variance reduction and large state-action space. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1753–1800. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/barakat23a.html.

Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *journal of artificial intelligence research*, 15:319–350, 2001.

David Blackwell. Discrete dynamic programming. *The Annals of Mathematical Statistics*, pp. 719–726, 1962.

Wang Chi Cheung. Regret minimization for reinforcement learning with vectorial feedback and complex objectives. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/a02ffd91ece5e7efeb46db8f10a74059-Paper.pdf.

Matthieu Geist, Julien Pérolat, Mathieu Laurière, Romuald Elie, Sarah Perrin, Oliver Bachem, Rémi Munos, and Olivier Pietquin. Concave utility reinforcement learning: The mean-field game viewpoint. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, pp. 489–497, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

Julien Grand-Clément and Marek Petrik. Reducing blackwell and average optimality to discounted mdps via the blackwell discount factor. *Advances in Neural Information Processing Systems*, 36: 52628–52647, 2023.

Conor F Hayes, Mathieu Reymond, Diederik M Roijers, Enda Howley, and Patrick Mannion. Distributional monte carlo tree search for risk-aware and multi-objective reinforcement learning. In *Proceedings of the 20th international conference on autonomous agents and multiagent systems*, pp. 1530–1532, 2021.

Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2681–2691. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hazan19a.html.

Yukinari Hisaki and Isao Ono. Rvi-sac: average reward off-policy deep reinforcement learning. *arXiv preprint arXiv:2408.01972*, 2024.

Ying Jin, Ramki Gummadi, Zhengyuan Zhou, and Jose Blanchet. Feasible q-learning for average reward reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1630–1638. PMLR, 2024.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Gao Peng, Eric Pauwels, and Hendrik Baier. Multi-objective utility actor critic with utility critic for nonlinear utility function. In *Eighteenth European Workshop on Reinforcement Learning*, 2025.

Matteo Pirotta, Simone Parisi, and Marcello Restelli. Multi-objective reinforcement learning with continuous pareto frontier approximation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

William Powell, Jeongyeol Kwon, Qiaomin Xie, and Hanbaek Lyu. Offline actor-critic for average reward mdps. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Mathieu Reymond, Conor F Hayes, Denis Steckelmacher, Diederik M Roijers, and Ann Nowé. Actor-critic multi-objective reinforcement learning for non-linear utility functions. *Autonomous Agents and Multi-Agent Systems*, 37(2):23, 2023.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

Willem Röpke, Diederik M Roijers, Ann Nowé, Roxana Rădulescu, and Hendrik Baier. Model-based reinforcement learning in multi-objective environments with a distributional critic. In *Multi-objective Decision Making Workshop at ECAI 2024*, 2024.

Naman Saxena, Subhojyoti Khastagir, Shishir Kolathaya, and Shalabh Bhatnagar. Off-policy average reward actor-critic with deterministic policy search. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 30130–30203. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/saxena23a.html.

Anton Schwartz. A reinforcement learning method for maximizing undiscounted rewards. In *ICML*, volume 93, pp. 298–305, 1993.

Umer Siddique, Paul Weng, and Matthieu Zimmer. Learning fair policies in multi-objective (deep) reinforcement learning with average and discounted rewards. In *International Conference on Machine Learning*, pp. 8905–8915. PMLR, 2020.

Richard Sutton. Policy gradient methods for reinforcement learning with function approxiamation. *Advances in neural...*, 2000.

Richard S Sutton, Andrew G Barto, and Andrew Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Jean Tarbouriech and Alessandro Lazaric. Active exploration in markov decision processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 974–982. PMLR, 2019.

Peter Vamplew, Cameron Foale, Conor F. Hayes, Patrick Mannion, Enda Howley, Richard Dazeley, Scott Johnson, Johan Källström, Gabriel Ramos, Roxana Radulescu, Willem Röpke, and Diederik M. Roijers. Utility-based reinforcement learning: Unifying single-objective and multi-objective reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, pp. 2717–2721, Richland, SC, 2024. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.

Yi Wan, Abhishek Naik, and Richard S Sutton. Learning and planning in average-reward markov decision processes. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10653–10662. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/wan21a.html.

Yi Wan, Huizhen Yu, and Richard S Sutton. On convergence of average-reward q-learning in weakly communicating markov decision processes. *arXiv preprint arXiv:2408.16262*, 2024.

Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International conference on machine learning*, pp. 10607–10616. PMLR, 2020.

Tom Zahavy, Brendan O'Donoghue, Guillaume Desjardins, and Satinder Singh. Reward is enough for convex mdps. *Advances in Neural Information Processing Systems*, 34:25746–25759, 2021.

Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. *Advances in Neural Information Processing Systems*, 33:4572–4583, 2020.

Junyu Zhang, Chengzhuo Ni, Csaba Szepesvari, Mengdi Wang, et al. On the convergence and sample efficiency of variance-reduced policy gradient method. *Advances in Neural Information Processing Systems*, 34:2228–2240, 2021.

Junyu Zhang, Amrit Singh Bedi, Mengdi Wang, and Alec Koppel. Multi-agent reinforcement learning with general utilities via decentralized shadow reward actor-critic. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9031–9039, 2022.

A Appendix: Related Work (detailed)

In this paper, we investigate the problem: in continuing setting, try to optimise the policy for the objective function which is the average utility of a trajectory. This setting is novel in RL and naturally extended from several related RL sub areas.

MORLRoijers et al. (2013); Hayes et al. (2022). One inspiration for our setting comes from multi-objective reinforcement learning (MORL). Standard MORL assumes a vector-valued environmental reward ($\mathbf{r}(s,a) \in \mathbb{R}^m$) and commonly adopts a utility-based paradigm, in which a utility function ($u : \mathbb{R}^m \to \mathbb{R}$) represents the utility that a user derives from the different reward dimensions Vamplew et al. (2024). A substantial part of the MORL literature considers linear utility functions or nonlinear utility functions over (discounted) accrued reward for example Peng et al. (2025); Reymond et al. (2023); Röpke et al. (2024); Xu et al. (2020); Pirotta et al. (2015); Hayes et al. (2021).

Only a relatively small body of work combines nonlinear utility with an average-reward criterion. Regret Minimization with Vectorial Feedback and Complex Objectives Cheung (2019) optimises a concave function of empirical time-average vector outcomes and shows that non-stationary policies may be necessary. Learning Fair Policies Siddique et al. (2020) uses the nonlinear Generalised Gini welfare function and analyses both average and discounted vector rewards, although its deep algorithms use discounted optimisation to approximate the average-reward objective. Reinforcement Learning for Joint Optimization of Multiple Rewards Agarwal & Aggarwal (2023) directly formulates a concave utility of the vector of long-run average rewards and provides an average-reward model-based method, while its scalable model-free policy-gradient implementation uses finite-horizon normalised discounted returns.

- These approaches assume that the environment supplies a **predefined per-step reward vector and that the user utility is an explicitly specified function of its long-run aggregate.** In contrast, APDU does not require human preferences to be decomposed into a fixed-dimensional per-step reward vector or an explicitly known scalarisation function. It instead permits utility feedback on complete trajectories, thereby capturing path-dependent preferences that cannot generally be recovered from only the cumulative or average values of a reward vector.

Average-reward MDPs Puterman (2014)have long been studied in dynamic programming and reinforcement learning. For finite MDPs, Blackwell optimality Blackwell (1962) guarantees a stationary policy that is optimal for all discount factors sufficiently close to one and is also average-reward optimal. The Blackwell discount factor Grand-Clément & Petrik (2023) characterizes this threshold, but it depends on the unknown MDP and is therefore unavailable to the learner. Directly optimizing average reward consequently avoids selecting an artificial discount factor and better matches the steady-state objective of continuing tasks.

The theory and algorithms for average-reward reinforcement learning continue to receive substantial attention. Early value-based approaches include R-learning Schwartz (1993) and relative-value-iteration Q-learning Abounadi et al. (2001). More recent work introduces Differential Q-learning Wan et al. (2021), which provides convergent off-policy learning and planning methods without requiring a reference state, and feasible average-reward Q-learning Jin et al. (2024), which dynamically adjusts a discounted formulation to approximate the average-reward objective without requiring unknown problem parameters. Average-reward Q-learning for weakly communicating MDPs Wan et al. (2024) further extends the convergence theory of value-based methods to a broader class of MDPs. Policy-based approaches include Average Reward Off-Policy Deep Deterministic Policy Gradient (ARO-DDPG) Saxena et al. (2023) and Offline Actor–Critic for Average-Reward MDPs Powell et al.. RVI-SAC Hisaki & Ono (2024) further extends Soft Actor–Critic to an entropy-regularised average-reward objective. Of course, there are many more papers about average-reward algorithms or especially application research; we only list the important, highly related and general results here. To note, all the settings here can be represented by our APDU setting with knowing utility funciton, but none of them considers utility function which can be aligned with human preference.

General Utility RL/Convex RL conclude that any RL problem(discounted, average, pure exploration etc..) can be recast into the $\mathcal{F}(\mu)$ form; they usually make assumptions to allow optimisation to be applicable.

Early objective-specific examples include maximum-entropy exploration Hazan et al. (2019) and active exploration Tarbouriech & Lazaric (2019). More generally, several multi-objective formulations optimise a utility of a vector of discounted or long-run average rewards rather than the full occupancy measure. These include joint optimisation of multiple rewards Agarwal & Aggarwal (2023), generalised-Gini social welfare Siddique et al. (2020), and concave functions of long-run vector outcomes Cheung (2019); their objectives are defined as scalarized accrued vector rewards $u(\lim \frac{1}{T} \sum r_t)$, while their model-free policy-gradient implementations typically use discounted approximations.

More general formulations directly optimise an explicitly given functional $\mathcal{F}$ of an occupancy measure. Variational Policy Gradient Zhang et al. (2020) and its variance-reduced extension Zhang et al. (2021) use the unnormalised discounted state–action occupancy. Reinforcement Learning with General Utilities Barakat et al. (2023) likewise considers a smooth function of the discounted state–action occupancy. Reward Is Enough for Convex MDPs Zahavy et al. (2021) covers both the normalised discounted state–action occupancy and the average stationary state–action occupancy. Concave Utility Reinforcement Learning Geist et al. (2022) instead uses the normalised discounted state–action occupancy and interprets the resulting problem through a mean-field-game formulation. These optimisation-based formulations provide useful tools for theoretical analysis, but commonly require smoothness, convexity or concavity, occupancy estimation, and repeated solution of linearised MDPs.

Shadow rewards is a common method applied in General Utility RL that we also take. Variational Policy Gradient (VPG) Zhang et al. (2020) introduced a Fenchel-dual construction in which the dual variable acts as a policy-dependent 'pseudo reward", and established a variational policy-gradient theorem that characterises the exact gradient of a general utility of the discounted state–action occupancy measure through a stochastic saddle-point problem. Its Monte Carlo implementation is accompanied by a finite-sample gradient-error bound, rather than an unbiasedness guarantee, because estimating the saddle-point solution and truncating infinite-horizon trajectories introduce approximation error. Independently, Reward Is Enough for Convex MDPs Zahavy et al. (2021) used Fenchel duality to reformulate a convex occupancy-measure objective as a two-player game between a policy player and a cost player, whose dual output acts as a non-stationary reward; this framework covers both discounted and average stationary occupancy measures, but does not derive a policy-gradient theorem. Subsequent work develops the policy-gradient branch in several directions: TSIVR-PG Zhang et al. (2021) improves sample efficiency through truncated variance reduction, Decentralized Shadow Reward Actor–Critic (DSAC) Zhang et al. (2022) extends the construction to cooperative multi-agent learning and explicitly calls the utility derivative a 'shadow reward", and the general-utility policy-gradient method of Barakat et al. (2023) provides simpler variance reduction and extensions to large state–action spaces. Writing the relevant occupancy measure uniformly as μ, these methods all construct a linear reward signal from the component-wise derivative $r_F(s,a) = \partial F/\partial \mu(s,a)$, evaluated at the occupancy measure induced by the current policy. Up to a sign change when F is minimised, we collectively refer to reward signals of this form as *shadow rewards*. The policy-gradient developments above nevertheless remain formulated in terms of discounted state–action occupancy measures. Using the chain rule (Zhang et al., 2020) or duality (Zahavy et al., 2021), arriving at $r_\theta^{\mathrm{sh}}(s,a) = \partial \mathcal{F}(\mu)/\partial \mu(s,a)|_{\mu=\mu_\theta}$. This construction satisfies the gradient-matching requirement in Condition 2 and yields the correct policy gradient. However, its stationary average generally does not recover the APDU objective, i.e., $\sum_{s,a} \mu_\theta(s,a) r_\theta^{\mathrm{sh}}(s,a) \neq \mathcal{F}(\mu_\theta)$. Thus, the resulting average-reward objective need not coincide with the original APDU objective. Our proxy reward can be interpreted as augmenting these constructions with a gain-matching correction; we establish this result through a novel construction based on a positively homogeneous functional of degree one (see Appendix B). This construction also provides a complementary perspective on proxy rewards beyond their conventional optimization-based interpretation.

B APPENDIX:THEORY ON PROXYREWARD

Let us briefly recall the considerations leading up to Thm. 2.

- We are focussing on parametrised policies π_θ where θ varies in some m-dimensional parameter space Θ. Each of these policies give rise to a stationary distribution μ_θ on the state-actions.
- Let us assume that there are n states (or state-actions). Therefore each stationary distribution $\mu = (\mu_1, \ldots, \mu_n)$, belongs to the probability simplex

$$\Delta^{n-1} = \left\{ \mu_i \geq 0, \quad \sum_{i=1}^{n} \mu_i = 1 \right\}. \tag{32}$$

- Furthermore, assume that the objective can be written as a functional of the stationary distribution,

$$J(\theta) = F(\mu_1, \ldots, \mu_n). \tag{33}$$

  This assumption makes sense as we are interested in optimising objectives that are computable based on the observations of (long) sample paths.
- The function F can be any function defined on the simplex and need not be homogeneous. We can therefore not invoke Euler's theorem on F to define the proxy reward. However, we can always extend F to an homogeneous function $\widetilde{F}$ that coincides with F on the probability simplex. This extension $\widetilde{F}$ will then give rise to the correct proxy reward. This is taken up and explained in more detail in the sections below.

B.1 HOMOGENEOUS EXTENSIONS ARE ALWAYS POSSIBLE

Since we need to compute derivatives, the restriction to the probability simplex introduces complications: we are only allowed to consider changes that *"don't leave the simplex"* (also see Fig. 2). A convenient way to handle this problem is to extend F to the ambient space in which the simplex lives, and project unconstrained gradient back onto the simplex. Therefore, the question arises whether one can construct a **degree-one homogeneous functional $\widetilde{F}$ on the positive cone** $K_+^n = \{x \in \mathbb{R}^n \mid x \neq 0,\ x_i \geq 0\}$, that agrees with F on the probability simplex Δ^{n-1}. The answer is yes. Specifically, for any (non-zero) vector in the positive cone, define:

$$S(x) := \sum_{i=1}^{n} x_i. \tag{34}$$

We then define $\mu(x)$ to be the unique point where the ray emanating to x intersects the probability simplex Δ^{n-1} (also see Fig. 2):

$$\mu(x) := \frac{x}{S(x)} \tag{35}$$

Now, in view of Eq. 35 , it is straightforward to define the extension $\widetilde{F}$:

$$\widetilde{F}(x) := S(x)\, F(\mu(x)) = S(x)\, F\left(\frac{x}{S(x)}\right) \qquad (\forall x \in K_+^n). \tag{36}$$

By definition $\widetilde{F}$ is degree-1 homogeneous and coincides with F on Δ^{n-1}, i.e.

$$\widetilde{F}(\mu) = F(\mu) \qquad \text{for all } \mu \in \Delta^{n-1}. \tag{37}$$

There is an interesting further observation. Suppose $\widetilde{F}$ is required to be degree-one homogeneous and to agree with F on the probability simplex, then $\widetilde{F}$ is uniquely defined because the value along a ray is determined by the value of the simplex-point where the ray "punctures" the simplex.

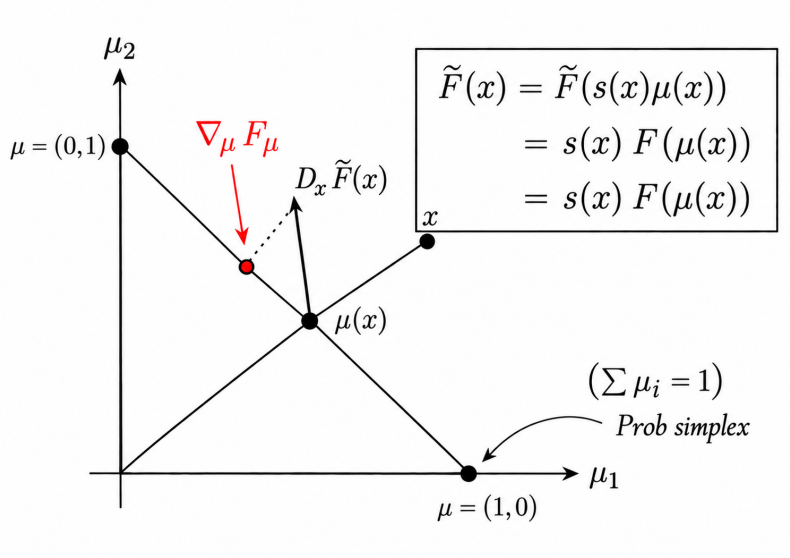


Figure 2: Clarifying the difference between the gradient of the extended function $\nabla_x \widetilde{F}$ and the gradient along the simplex $\nabla_\mu F$. For a two-state MDP, all the possible stationary distributions μ are on the line segment (2-dim simplex) connecting (1,0) and (0,1). Any point on the simplex is denoted as μ as it corresponds to a distribution on the 2-states state space. On this simplex, a function $F(\mu)$ is defined which we then want to extend to any point x in the positive cone. We extend F to the homogeneous $\widetilde{F}$ in any point x in the positive cone by referring back to the point $\mu(x)$ (see main text for details).

B.2 Defining proxy rewards

Suppose now that $J(\theta) = F(\mu_\theta)$ where μ_θ is a stationary probability distribution (over n states or state-actions). Now, denote the corresponding unique degree-1 homogeneous extension on the positive cone as:

$$\widetilde{F}(x) := S(x)\, F(\mu(x)) = S(x)\, F\left(\frac{x}{S(x)}\right) \tag{38}$$

For any μ_θ on the probability simplex,

$$J(\theta) = F(\mu_\theta) = \widetilde{F}(\mu_\theta), \tag{39}$$

and because $\widetilde{F}$ is homogeneous of degree one, Euler's theorem tells us:

$$\widetilde{F}(x) = \sum_i x_i \frac{\partial \widetilde{F}(x)}{\partial x_i}. \tag{40}$$

We now **define the proxy reward** (in state i) as:

$$\boxed{\widetilde{r}_i(\theta) := \left.\frac{\partial \widetilde{F}}{\partial x_i}\right|_{x=\mu_\theta}} \tag{41}$$

Plugging this into Euler equality Eq. 40 (evaluated at $x = \mu_\theta$) then yields:

$$\widetilde{F}(\mu_\theta) \equiv F(\mu_\theta) \equiv J(\theta) = \sum_{i=1}^{n} \mu_\theta(i)\, \widetilde{r}_i(\theta) \tag{42}$$

where the RHS-equality is exactly proxy-condition 1 (Eq. 15).

Re-expressing proxy reward in terms of derivatives with respect to μ Notice that definition of the proxy reward in Eq. 41 is in terms of derivatives on the positive cone (i.e. $\partial/\partial x_i$). Such derivatives allow one to "move away from the probability simplex". However, for the PGT we are interested in the gradient with respect to θ. Varying θ induces variations of the stationary distribution μ_θ but this results in **changes within the probability simplex.** The chain rule tells us that for $J(\theta) = F(\mu_\theta)$:

$$\frac{\partial J}{\partial \theta} = \sum_i \frac{\partial F}{\partial \mu_i} \frac{\partial \mu_i}{\partial \theta}$$

The RHS decomposes the change due to θ in two steps:

$$\text{change in } \theta \quad \longrightarrow \quad \text{change in } \mu \quad \longrightarrow \quad \text{change in } J$$

For that reason we re-express the derivative in terms of the stationary distribution μ. The starting point is the definition Eq. 38: Taking partial derivatives yields:

$$\begin{aligned}
\frac{\partial \widetilde{F}}{\partial x_i} &= \frac{\partial}{\partial x_i}\Big[S(x)\, F(\mu(x))\Big] && (43)\\
&= F(\mu(x)) + S(x) \left[\sum_{j=1}^n \frac{\partial F}{\partial \mu_j} \frac{\partial \mu_j}{\partial x_i}\right] && (44)\\
&= F(\mu(x)) + S(x) \left[\sum_{j=1}^n \frac{\partial F}{\partial \mu_j} \left(\frac{1}{S(x)}(\delta_{ij} - \frac{x_j}{S(x)})\right)\right] && (45)\\
&= F(\mu(x)) + \frac{\partial F}{\partial \mu_i} - \frac{1}{S(x)} \sum_{j=1}^n x_j \frac{\partial F}{\partial \mu_j} && (46)
\end{aligned}$$

Finally we evaluate Eq. 46 at $x = \mu_\theta$ for which $S(x) = 1$ to obtain the following expression for the proxy reward:

$$\boxed{\widetilde{r}_i(\theta) := \left.\frac{\partial \widetilde{F}}{\partial x_i}\right|_{x=\mu_\theta} = F(\mu_\theta) + \left.\frac{\partial F}{\partial \mu_i}\right|_{\mu=\mu_\theta} - \sum_{j=1}^n \mu_j(\theta) \left.\frac{\partial F}{\partial \mu_j}\right|_{\mu=\mu_\theta}} \tag{47}$$

B.3 Reformulating proxy reward for state-action occupancy measures

Recap of the proxy reward and proxy-conditions For the sake of completeness and clarity we recast the proxy reward definition (Eq. 47) in terms of the stationary distribution $\mu_\theta(s,a)$ over state-actions:

$$r^u_\theta(s,a) := \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s,a)} + \left(\rho^u_\theta - \sum_{s',a'} \mu(s',a') \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s',a')}\right). \tag{48}$$

Notice that the term between brackets vanishes for a degree-1 homogeneous function. So this 2nd term can be seen as a correction for non-homogeneity. In order to check the two proxy conditions, we introduce an auxiliary functional functional $\overline{J}^u(\theta)$ that is obtained by plugging in this proxy reward:

$$\overline{J}^u(\theta) := \sum_{s,a} \mu_\theta(s,a)\, r^u_\theta(s,a)$$

Interestingly, we make the following two observations:

- **Proxy condition 1** Working out the above equation, we have

$$\overline{J}(\theta) = \sum_{s,a} \mu_\theta(s,a) \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s,a)} + \left(\rho^u - \sum_{s',a'} \mu(s',a') \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s',a')}\right) \sum_{s,a} \mu_\theta(s,a) \tag{49}$$

Since $\sum_{s,a} \mu_\theta(s,a) = 1$, it follows that $\overline{J}(\theta) = \rho^u(\theta) = J^u_\theta$. Hence, we have confirmed:

$$J^u(\theta) = \sum_{s,a} \mu_\theta(s,a)\, r^u_\theta(s,a)$$

- **Proxy condition 2:**

$$\begin{aligned}
\sum \mu(s,a) \nabla_\theta r^u(s,a) &= \sum_{s,a} \mu(s,a)\, \nabla_\theta \left(\frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s,a)} + J(\theta) - \sum_{s',a'} \mu(s',a') \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s',a')}) \right) \\
&= \sum_{s,a} \mu(s,a) \Big[\nabla_\theta \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s,a)} + \underbrace{\left(\nabla_\theta J - \sum_{s,a} \frac{\partial \mu(s,a)}{\partial \theta} \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s,a)} \right)}_{=0 \text{ (chain rule)}} \\
&\qquad - \sum_{s,a} \mu(s,a) \nabla_\theta \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s,a)} \Big] \\
&= \sum \mu(s,a) \nabla_\theta \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s,a)} - \sum_{s,a} \mu(s,a) \left[\sum_{s',a'} \mu(s',a') \nabla_\theta \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s',a')}) \right] \\
&= \sum_{s,a} \mu_\theta(s,a) \nabla_\theta \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s,a)} - \left[\sum_{s',a'} \mu(s',a') \nabla_\theta \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s',a')}) \right] \\
&= 0, \qquad \checkmark
\end{aligned}$$

since $\sum_{s,a} \mu(s,a) = 1$

- In addition, the so-called non-homogeneity correction

$$g(\theta) := \rho^u_\theta - \sum_{s,a} \mu(s,a) \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s,a)},$$

only depends on θ, and not on (s,a) (as we average over all states). Then the gradient of J can be expanded using the chain rule

$$\nabla_\theta J = \sum_{s,a} \frac{\partial \mathcal{F}(\mu_\theta)}{\partial \mu(s,a)} \nabla_\theta \mu(s,a) = \sum_{s,a} [r^u_\theta + g(\theta)] \nabla_\theta \mu_\theta(s,a) = \nabla_\theta \overline{J}, \tag{50}$$

since

$$g(\theta) \sum_{s,a} \nabla_\theta \mu_\theta(s,a) = g(\theta)\, \nabla(1) = 0.$$

What's next? Previous work Zhang et al. (2020) put forward the pseudo reward from the chain rule as well as Zahavy et al. (2021) cost from duality space, they all use the form $\frac{F}{\mu(s,a)}$ as a frozen reward which does not influence the Proxy condition 2; theirs as well as other shadow reward method(see the related works) assure the Policy Gradient Estimator coincide but not the relative q(or v) function as well as the objective function.

C Appendix: Introducing trajector- based q^τ

Recall the definition of the trajectory-based q-function:

$$q^\tau_\theta(s,a) := \lim_{T \to \infty} \mathbb{E}_\theta \left[\Delta u_t(T) - T \rho^u_\theta \mid S_t = s,\, A_t = a \right]. \tag{51}$$

Estimating $\widehat{q}^{\tau}_{\theta}$ from a single trajectory The above definition immediately suggests an empirical estimator. For a sufficiently large but finite L, define

$$Q_t^{(L)} := \Delta u_t(L) - L\widehat{\rho}^u_\theta, \tag{52}$$

where $\widehat{\rho}^u_\theta$ is an estimate of the average utility rate. We can then estimate

$$q^{\tau}_{\theta}(s,a) \approx \mathbb{E}_\theta \left[Q_t^{(L)} \mid S_t = s, A_t = a \right]. \tag{53}$$

In a finite state-action space, this conditional expectation may simply be estimated by averaging:

$$\widehat{q}^{\tau}_{\theta,L}(s,a) = \frac{1}{N_{s,a}} \sum_{\substack{t: \\ (S_t,A_t)=(s,a)}} \left[u(\tau_{0:t+L}) - u(\tau_{0:t}) - L\widehat{\rho}^u_\theta \right]. \tag{54}$$

More generally, regression or function approximation can be used to estimate the conditional expectation as a function of (s,a).

An attractive feature of this construction is that the estimator itself does not need to know how many previous time steps are relevant to the utility. The utility evaluator receives the complete prehistory through the trajectory prefixes and therefore performs the required bookkeeping implicitly.

C.1 The appropriate occupancy measure

We now connect the trajectory-based quantity $\widehat{q}^{\tau}_{\theta}$ to the proxy-reward differential q-function introduced in Section 5. The relevant occupancy measure need not necessarily be the ordinary one-step state distribution. If the utility depends on finite temporal patterns, an augmented state representation may be required. Let X_t denote an appropriate finite-dimensional representation of the process such that:

1. (X_t) is stationary and ergodic under π_θ;
2. its stationary distribution is denoted by μ_θ;
3. the long-run objective can be expressed as
$$\rho^u_\theta = F(\mu_\theta). \tag{55}$$

The variable X_t may represent the original state, a state-action pair, or an augmented state containing a finite amount of previous history. The important point is that this representation is needed for the *theoretical analysis*; it **need not be known explicitly when estimating $\widehat{q}^{\tau}_{\theta}$ through Eq. 53.**

Consider the **empirical occupancy measure** over the interval $[t, t+L)$, generated by the roll-out path τ:

$$\widehat{\mu}_{t,L}(x) := \frac{1}{L} \sum_{k=t}^{t+L-1} \mathbf{1}\{X_k = x\}. \tag{56}$$

Under ergodicity,

$$\widehat{\mu}_{t,L} \longrightarrow \mu_\theta \qquad \text{as } L \to \infty. \tag{57}$$

Define

$$\delta_{t,L} := \widehat{\mu}_{t,L} - \mu_\theta. \tag{58}$$

For the following analysis we assume that the incremental path utility admits an asymptotic representation

$$\Delta u_t(L) = F(\widehat{\mu}_{t,L}) \cdot L + R_{t,L}, \tag{59}$$

where $R_{t,L}$ is a residual finite-window term. In the most favourable case,

$$R_{t,L} = 0, \tag{60}$$

so that the incremental utility is represented exactly by the empirical occupancy functional. More generally we assume that $R_{t,L} = O_p(1)$ and that the relevant conditional limits exist. Notice that Eq. 59 is stronger than merely requiring

$$J^u(\theta) = F(\mu_\theta). \tag{61}$$

The latter concerns only the long-run average, whereas Eq. 59 controls the finite-window fluctuations around that average.

C.2 TAYLOR EXPANSION AROUND THE STATIONARY OCCUPANCY MEASURE

Assume that F is twice continuously differentiable in a neighbourhood of μ_θ. Expanding around μ_θ gives

$$F(\widehat{\mu}_{t,L}) = F(\mu_\theta) + \nabla F(\mu_\theta)^\top \delta_{t,L} + \frac{1}{2}\delta_{t,L}^\top H_F(\mu_\theta)\delta_{t,L} + o_p\left(\|\delta_{t,L}\|^2\right), \tag{62}$$

where $H_F(\mu_\theta) := \nabla^2 F(\mu_\theta)$ is the Hessian of F. Subtracting $F(\mu_\theta) = \rho_\theta^u$ and multiplying by L yields

$$L\left[F(\widehat{\mu}_{t,L}) - F(\mu_\theta)\right] = L\nabla F(\mu_\theta)^\top \delta_{t,L} + \frac{L}{2}\delta_{t,L}^\top H_F(\mu_\theta)\delta_{t,L} + o_p(1). \tag{63}$$

Under the usual Markov-chain central limit scaling,

$$\delta_{t,L} = O_p(L^{-1/2}). \tag{64}$$

Consequently,

- the first-order term is $O_p(\sqrt{L})$ as a random variable;
- the second-order term is $O_p(1)$;
- Taylor terms of order three and higher vanish asymptotically after multiplication by L.

C.3 THE FIRST-ORDER TERM AND THE PROXY-REWARD q-FUNCTION

Let us abbreviate the componentes of the gradient as:

$$g_i := \left.\frac{\partial F}{\partial \mu_i}\right|_{\mu=\mu_\theta}. \tag{65}$$

The first-order term can be written as

$$L\nabla F(\mu_\theta)^\top \delta_{t,L} = L\sum_i g_i\left(\widehat{\mu}_{t,L,i} - \mu_{\theta,i}\right) = \sum_{k=t}^{t+L-1} g(X_k) - L\sum_i \mu_{\theta,i} g_i. \tag{66}$$

Now recall the proxy reward introduced previously. It has the form

$$r_{\theta,i}^u = g_i + \underbrace{\rho_\theta^u - \sum_j \mu_{\theta,j} g_j}_{=c_\theta} = g_i + c_\theta. \tag{67}$$

Therefore

$$r_\theta^u(X_k) - \rho_\theta^u = g(X_k) - \sum_i \mu_{\theta,i} g_i. \tag{68}$$

Substitution into Eq. 66 gives the exact identity

$$\boxed{L\nabla F(\mu_\theta)^\top \delta_{t,L} = \sum_{k=t}^{t+L-1}\left[r_\theta^u(X_k) - \rho_\theta^u\right].} \tag{69}$$

Thus the first-order Taylor term is precisely the centred cumulative proxy reward.

Conditioning on the observed state-action pair $(S_t, A_t) = (s, a)$ and letting $L \to \infty$ gives

$$\boxed{\lim_{L\to\infty} \mathbb{E}_\theta\left[L\nabla F(\mu_\theta)^\top \delta_{t,L} \,\middle|\, S_t = s, A_t = a\right] = q_\theta^u(s,a),} \tag{70}$$

where, if X_t contains more information than S_t, the notation $q_\theta^u(s,a)$ denotes the corresponding stationary-history aggregation of the augmented-state proxy q-function.

Equation 70 provides the fundamental connection between the proxy-reward construction and the trajectory-based q-function: the proxy-reward q-function is the conditional limiting effect of the *first-order perturbation of the empirical occupancy measure*.

C.4 THE SECOND-ORDER CURVATURE CORRECTION

Consider now the second-order term

$$\frac{L}{2}\delta_{t,L}^{\top} H_F(\mu_\theta)\delta_{t,L}. \tag{71}$$

Assume that the stationary process satisfies a Markov-chain central limit theorem,

$$\sqrt{L}\,(\widehat{\mu}_{t,L} - \mu_\theta) \Rightarrow \mathcal{N}(0, \Sigma_\theta), \tag{72}$$

where Σ_θ denotes the asymptotic covariance matrix of the empirical occupancy measure. Then

$$L\,\mathbb{E}\left[\delta_{t,L}\delta_{t,L}^{\top}\right] \longrightarrow \Sigma_\theta. \tag{73}$$

Using the cyclic permutation property of traces,

$$x^{\top}Ax = \mathrm{tr}(xAx^{\top}) = \mathrm{tr}(Axx^{\top}), \tag{74}$$

we obtain

$$\mathbb{E}\left[\frac{L}{2}\delta_{t,L}^{\top} H_F(\mu_\theta)\delta_{t,L}\right] \longrightarrow \frac{1}{2}\,\mathrm{tr}\left[H_F(\mu_\theta)\Sigma_\theta\right]. \tag{75}$$

We therefore define

$$\boxed{C_\theta := \frac{1}{2}\,\mathrm{tr}\left[H_F(\mu_\theta)\Sigma_\theta\right].} \tag{76}$$

The correction C_θ represents the interaction between the nonlinearity of F and the stochastic fluctuations of the empirical occupancy measure.

For a linear functional, $H_F = 0$, and consequently $C_\theta = 0$. To leading order, C_θ depends on the stationary process and on θ, but not on the state-action pair (s, a) on which we condition.

C.5 RESIDUAL TERMS AND THE BOUNDARY ISSUE

Combining Eqs. 59 and 63, we obtain

$$\Delta u_t(L) - L\rho_\theta^u = L\nabla F(\mu_\theta)^{\top}\delta_{t,L} + \frac{L}{2}\delta_{t,L}^{\top} H_F(\mu_\theta)\delta_{t,L} + R_{t,L} + o_p(1). \tag{77}$$

Taking conditional expectations and letting $L \to \infty$ gives

$$\boxed{\widehat{q}_\theta^{\tau}(s,a) = q_\theta^u(s,a) + C_\theta + B_\theta(s,a),} \tag{78}$$

where

$$B_\theta(s,a) := \lim_{L\to\infty} \mathbb{E}_\theta\left[R_{t,L} \mid S_t = s, A_t = a\right], \tag{79}$$

provided this limit exists.

It is important to distinguish this residual term from an *artificial boundary effect* created by cutting a continuing trajectory and evaluating the utility on the isolated piece. The latter problem is avoided by construction in Eq. 23: since both utility evaluations are performed on prefixes of the same trajectory, the prehistory before time t is retained.

The remaining quantity $B_\theta(s,a)$ in Eq. 78 measures any residual mismatch between the actual incremental path utility and the empirical-occupancy representation in Eq. 59. In particular, if

$$\Delta u_t(L) = LF(\widehat{\mu}_{t,L}) \tag{80}$$

exactly, then

$$B_\theta(s,a) = 0 \tag{81}$$

and

$$\boxed{\widehat{q}_\theta^{\tau}(s,a) = q_\theta^u(s,a) + C_\theta.} \tag{82}$$

A simple concrete example where $B_\theta(s,a) \neq 0$ is provided in Section C.7.

C.6 Policy Gradient Theorem using the trajectory-based q-function

From Section 5, the proxy-reward q-function satisfies

$$\nabla_\theta J^u(\theta) = \sum_s \mu_\theta(s) \sum_a q^u_\theta(s,a)\pi_\theta(a \mid s)\nabla_\theta \log \pi_\theta(a \mid s). \tag{83}$$

Suppose first that

$$\widehat{q}^\tau_\theta(s,a) = q^u_\theta(s,a) + C_\theta, \tag{84}$$

where C_θ is independent of (s,a). For every state s,

$$\sum_a \pi_\theta(a \mid s) C_\theta \nabla_\theta \log \pi_\theta(a \mid s) = C_\theta \nabla_\theta \sum_a \pi_\theta(a \mid s) = C_\theta \nabla_\theta 1 = 0.$$

Consequently the common curvature correction disappears from the policy-gradient expression and we end up with the standard form of PGT for $\widehat{q}^u_\theta$:

$$\boxed{\nabla_\theta J^u(\theta) = \sum_s \mu_\theta(s) \sum_a \widehat{q}^\tau_\theta(s,a)\pi_\theta(a \mid s)\nabla_\theta \log \pi_\theta(a \mid s).} \tag{85}$$

More generally, Eq. 85 continues to hold whenever the residual correction $B_\theta(s,a)$ makes no contribution to the score-function expectation, i.e. whenever

$$\sum_s \mu_\theta(s) \sum_a B_\theta(s,a)\pi_\theta(a \mid s)\nabla_\theta \log \pi_\theta(a \mid s) = 0. \tag{86}$$

A sufficient condition is that $B_\theta(s,a)$ is independent of the action a for each fixed state s. In particular, this condition holds when $B_\theta(s,a) = 0$.

Thus, under the stated regularity and residual conditions, the trajectory-based quantity q^τ_θ can replace the proxy-reward differential q-function q^u_θ in the Policy Gradient Theorem.

C.7 A simple example with a non-zero residual term $B_\theta(s,a)$

A natural example of a non-zero residual term arises in a genuinely continuing process. Consider the two-state Markov process Sec 8.1.

$$P_\theta = \begin{pmatrix} 0 & 1 \\ \theta & 1-\theta \end{pmatrix}, \tag{87}$$

so that the transition from A to B is deterministic, whereas from state B the process switches to A with probability θ. The stationary distribution is

$$\mu_\theta(A) = \frac{\theta}{1+\theta}, \qquad \mu_\theta(B) = \frac{1}{1+\theta}. \tag{88}$$

Now define the path utility as the number of completed $B \to A$ transitions. Such a transition can, for example, be interpreted as the completion of a job, service cycle, or production cycle in a continuing system.

The long-run average utility is therefore

$$J^u(\theta) = \Pr_\theta(S_t = B, S_{t+1} = A) = \mu_\theta(B)\,\theta = \frac{\theta}{1+\theta}. \tag{89}$$

Hence the corresponding stationary occupancy functional can be written as $F(\mu_\theta) = \mu_\theta(A)$. Notice that this might seem odd at first as the utility is based on *transitions ($B \to A$)*, but this is tightly linked to the stationary distribution in A. We will return to this below.

Now consider an arbitrary observation time t in the stationary continuing process and a finite horizon L. The incremental utility over the observation window is

$$\Delta u_t(L) = \sum_{k=t}^{t+L-1} \mathbf{1}\{S_k = B, S_{k+1} = A\} = \sum_{k=t+1}^{t+L} \mathbf{1}\{S_k = A\}, \tag{90}$$

(because the only possible transition into A is $B \to A$). On the other hand, define the empirical state occupancy measure over the interval $[t, t+L)$ by

$$\widehat{\mu}_{t,L}(A) := \frac{1}{L} \sum_{k=t}^{t+L-1} \mathbf{1}\{S_k = A\}. \tag{91}$$

Since $F(\widehat{\mu}_{t,L}) = \widehat{\mu}_{t,L}(A)$, we obtain

$$LF(\widehat{\mu}_{t,L}) = \sum_{k=t}^{t+L-1} \mathbf{1}\{S_k = A\}. \tag{92}$$

Comparing this expression with the incremental path utility gives

$$\Delta u_t(L) - LF(\widehat{\mu}_{t,L}) = \sum_{k=t+1}^{t+L} \mathbf{1}\{S_k = A\} - \sum_{k=t}^{t+L-1} \mathbf{1}\{S_k = A\} = \mathbf{1}\{S_{t+L} = A\} - \mathbf{1}\{S_t = A\}. \tag{93}$$

Hence the residual term is exactly

$$\boxed{R_{t,L} = \mathbf{1}\{S_{t+L} = A\} - \mathbf{1}\{S_t = A\}.} \tag{94}$$

This residual is of order $O(1)$. Consequently,

$$\frac{R_{t,L}}{L} \longrightarrow 0, \tag{95}$$

so it has no effect on the long-run average utility. However, it need not disappear from the differential trajectory-based q-function, which itself is an $O(1)$ quantity. Indeed, conditioning on the state-action pair at the beginning of the window gives

$$B_\theta(s,a) = \lim_{L\to\infty} \mathbb{E}_\theta \left[R_{t,L} \mid S_t = s, A_t = a \right]. \tag{96}$$

By ergodicity,

$$\mathbb{E}_\theta \left[\mathbf{1}\{S_{t+L} = A\} \mid S_t = s, A_t = a \right] \longrightarrow \mu_\theta(A). \tag{97}$$

Therefore

$$\boxed{B_\theta(s,a) = \mu_\theta(A) - \mathbf{1}\{s = A\}.} \tag{98}$$

Explicitly,

$$B_\theta(A,a) = \frac{\theta}{1+\theta} - 1 = -\frac{1}{1+\theta}, \qquad \text{whereas} \qquad B_\theta(B,a) = \frac{\theta}{1+\theta}. \tag{99}$$

Thus, $B_\theta(s,a) \neq 0$ in this simple continuing-process example.

The interpretation is straightforward. In the infinite stationary limit, the number of transitions into state A and the number of visits to state A have the same average rate. Over a finite observation window, however, the corresponding counts may differ because of what happens at the two ends of the window. Indeed,

$$\boxed{\text{number of entries into } A - \text{number of visits to } A = \mathbf{1}\{S_{t+L} = A\} - \mathbf{1}\{S_t = A\}.} \tag{100}$$

The term "boundary" therefore does not refer to a genuine beginning or end of the underlying process. The process itself continues indefinitely. The boundaries are merely the arbitrary observation times t and $t+L$ of a finite window extracted from the continuing process.

Finally, notice that in this example $B_\theta(s,a) = B_\theta(s)$ does not depend on the action a. Therefore its contribution to the policy-gradient expression vanishes:

$$\sum_a \pi_\theta(a \mid s) B_\theta(s) \nabla_\theta \log \pi_\theta(a \mid s) = B_\theta(s) \sum_a \nabla_\theta \pi_\theta(a \mid s) \tag{101}$$

$$= B_\theta(s) \nabla_\theta \sum_a \pi_\theta(a \mid s) \tag{102}$$

$$= 0. \tag{103}$$

Hence this example illustrates the important conclusion that as long as B_θ does NOT depend on the action a, the Policy Gradient Theorem is NOT affected.

What matters is whether the residual contains an action-dependent component whose score-function expectation does not vanish.

C.8 Summary and Statement

The main object introduced in this section is the stationary trajectory-based differential action value

$$q_\theta^\tau(s,a) = \lim_{T\to\infty} \mathbb{E}_\theta \left[u(\tau_{0:t+T}) - u(\tau_{0:t}) - \rho_\theta^u T \,|\, S_t = s, A_t = a \right]. \tag{104}$$

This definition retains the prehistory of the continuing trajectory and therefore avoids the artificial boundary bias that can arise when a history-dependent utility is evaluated on isolated subsequences. For empirical estimation, we use

$$q_\theta^\tau(s,a) \approx \mathbb{E}_\theta \left[\Delta u_t(L) - L\rho_\theta^u \mid S_t = s, A_t = a \right]. \tag{105}$$

When the incremental utility admits an empirical-occupancy representation

$$\Delta u_t(L) = LF(\widehat{\mu}_{t,L}) + R_{t,L}, \tag{106}$$

a Taylor expansion around the stationary occupancy measure provides the decomposition

$$q_\theta^\tau(s,a) = q_\theta^u(s,a) + C_\theta + B_\theta(s,a). \tag{107}$$

The first-order Taylor term generates the proxy-reward differential q-function,

$$\text{first-order term} \longrightarrow q_\theta^u(s,a), \tag{108}$$

while the second-order term generates the curvature correction

$$C_\theta = \frac{1}{2} \operatorname{tr} \left[H_F(\mu_\theta) \Sigma_\theta \right]. \tag{109}$$

The curvature correction is independent of the state-action pair and therefore disappears from the Policy Gradient Theorem.

If the remaining residual term $B_\theta(s,a)$ either vanishes or has zero score-function contribution, then

$$\nabla_\theta J^u(\theta) = \sum_s \mu_\theta(s) \sum_a \widehat{q}_\theta^u(s,a) \pi_\theta(a \mid s) \nabla_\theta \log \pi_\theta(a \mid s). \tag{110}$$

Consequently, the gradient of the long-run path-dependent utility can be estimated from trajectory-level utility observations without requiring the explicit proxy reward or even explicit knowledge of the temporal memory length of the utility.

Theorem 4. *Under the same assumptions as the ones for Thm. 2 we can make the following conclusions:*

- *The trajectory-dependent q_θ^τ is linked to the proxy-q^u through the following equation:*

$$\boxed{q_\theta^\tau(s,a) = q_\theta^u(s,a) + C_\theta + B_\theta(s,a),} \tag{111}$$

  where C_θ is independent of (s,a), and

$$B_\theta(s,a) := \lim_{L\to\infty} \mathbb{E}_\theta \left[R_{t,L} \mid S_t = s, A_t = a \right], \tag{112}$$

  provided this limit exists.

- *If the score contribution of B_θ vanishes, i.e.*

$$\sum_s \mu_\theta(s) \sum_a B_\theta(s,a) \pi_\theta(a \mid s) \nabla_\theta \log \pi_\theta(a \mid s) = 0 \tag{113}$$

  *then the **trajectory-based version of PGT holds**:*

$$\boxed{\nabla_\theta J^u(\theta) = \sum_s \mu_\theta(s) \sum_a \widehat{q}_\theta^u(s,a)\, \pi_\theta(a \mid s)\, \nabla_\theta \log \pi_\theta(a \mid s).} \tag{114}$$

- *A **sufficient condition for eq. 113, and hence PGT (114), to hold** is that $B_\theta(s,a)$ is independent of a (for every state s):*

$$B_\theta(s,a) = B_\theta(s) \qquad (\textit{for all } s).$$

□

C.9 LINKING THE RESIDUAL TERM TO PROPERTIES OF THE UTILITY FUNCTION

The residual term

$$B_\theta(s,a) = \lim_{L\to\infty} \mathbb{E}_\theta \left[R_{t,L} \mid S_t = s, A_t = a \right] \tag{115}$$

can be related directly to the structural properties of the path utility $u(\tau)$. The key question is how closely the path utility can be represented as an additive functional of an appropriate stationary process. Suppose that, on a possibly augmented Markov process Z_t, the incremental utility admits the decomposition

$$\Delta u_t(L) = \sum_{k=t}^{t+L-1} g(Z_k, A_k) + \Phi(Z_{t+L}) - \Phi(Z_t) + o(1). \tag{116}$$

The first term in eq. 116 is extensive: it grows linearly with the window length L. The remaining term

$$\Phi(Z_{t+L}) - \Phi(Z_t) \tag{117}$$

is a bounded boundary, or coboundary, contribution. In this case the residual takes the form

$$R_{t,L} = \Phi(Z_{t+L}) - \Phi(Z_t) + o(1). \tag{118}$$

Under ergodicity, the process forgets its starting condition, so

$$\mathbb{E}_\theta \left[\Phi(Z_{t+L}) \mid S_t = s, A_t = a \right] \longrightarrow \mathbb{E}_{\mu_\theta} \left[\Phi(Z) \right] \qquad \text{as } L \to \infty. \tag{119}$$

Therefore,

$$B_\theta(s,a) = \mathbb{E}_{\mu_\theta} \left[\Phi(Z) \right] - \mathbb{E}_\theta \left[\Phi(Z_t) \mid S_t = s, A_t = a \right]. \tag{120}$$

Thus the residual is determined by the finite-memory information present at the observation boundary.

Effect of a Markovian policy. Suppose that the policy is Markovian,

$$A_t \sim \pi_\theta(\cdot \mid S_t), \tag{121}$$

while Z_t contains the relevant history up to the current state S_t. Then, conditional on the current state S_t, the action A_t should be independent of Z_t (i.e. no information from $Z_t|S_t$ is supposed to seep into the action):

$$A_t \perp Z_t \mid S_t. \tag{122}$$

Hence, if the boundary term depends only on the pre-action augmented state Z_t,

$$\mathbb{E}_\theta \left[\Phi(Z_t) \mid S_t = s, A_t = a \right] = \mathbb{E}_\theta \left[\Phi(Z_t) \mid S_t = s \right]. \tag{123}$$

Substitution into Eq. 120 gives

$$\boxed{B_\theta(s,a) = B_\theta(s).} \tag{124}$$

Consequently,

$$\sum_s \mu_\theta(s) \sum_a B_\theta(s,a) \pi_\theta(a \mid s) \nabla_\theta \log \pi_\theta(a \mid s) = 0. \tag{125}$$

This suggests the following useful implication:

$$\boxed{\begin{array}{c} \text{utility is asymptotically additive} \\ + \\ \text{the residual is a boundary function of past/current state} \\ + \\ \text{the policy is Markovian} \\ \Downarrow \\ B_\theta(s,a) = B_\theta(s) \\ \Downarrow \\ \text{the residual does not affect the Policy Gradient Theorem.} \end{array}} \tag{126}$$

Exactly additive utilities The simplest case is an ordinary additive utility

$$u(\tau_T) = \sum_{t=0}^{T-1} r(S_t, A_t), \tag{127}$$

and when the appropriate state-action occupancy representation is used, we have $R_{t,L} = 0$. Therefore,

$$\boxed{B_\theta(s, a) = 0.} \tag{128}$$

Finite-memory pattern-count utilities (reward machines) Consider, for example, a utility that counts occurrences of the pattern ABB.

$$u(\tau_T) = N_{ABB}(\tau_T). \tag{129}$$

By augmenting the state space sufficiently, for instance using $Z_t = (S_{t-1}, S_t)$, the pattern-count utility becomes additive on the augmented process. Indeed, a new ABB occurrence is completed whenever $Z_t = (A, B)$ and the next state is $S_{t+1} = B$. Hence the path utility can be represented as a sum of local rewards on the augmented process, up to finite-window boundary terms associated with patterns that are partially present at the ends of an observation window. Such residual terms are naturally functions of the augmented boundary state.

If the action at time t is selected only from the current state S_t, then conditioning on A_t in addition to S_t does not introduce any additional information, hence:

$$\Pr_\theta(Z_t = z \mid S_t = s, A_t = a) = \Pr_\theta(Z_t = z \mid S_t = s). \tag{130}$$

Therefore the corresponding limiting boundary correction is expected to depend on the state but not on the action:

$$B_\theta(s, a) = B_\theta(s). \tag{131}$$

This provides an important interpretation for finite-memory non-Markovian utilities: they may generate non-zero residual terms, but under a Markovian policy these boundary residuals are often harmless for the policy gradient.

When can an action-dependent residual arise? The situation changes if the finite-window residual itself couples the historical boundary context with the current action. Suppose, schematically, that

$$R_{t,L} = \Psi(Z_t, A_t) + \Phi(Z_{t+L}) + o(1). \tag{132}$$

Then, again $\lim_{L\to\infty} \Phi(Z_{t+L}) = \mathbb{E}_{\mu_\theta} \Phi(Z)$ and therefore:

$$\begin{aligned} B_\theta(s, a) &= \lim_{L\to\infty} \mathbb{E}_\theta [R_{t,L} \mid S_t = s, A_t = a] \\ &= \mathbb{E}_{\mu_\theta} [\Phi(Z)] + \mathbb{E}_\theta [\Psi(Z_t, a) \mid S_t = s]. \end{aligned} \tag{133}$$

Because of the second term, there is now no reason for $B_\theta(s, a)$ to be independent of the action. Hence we obtain the distinction

$$\boxed{\begin{array}{ll} \text{residual depends only on historical boundary context} & \Longrightarrow B_\theta(s, a) = B_\theta(s), \\ \text{residual couples historical context with current action} & \Longrightarrow B_\theta(s, a) \text{ may depend on } a. \end{array}} \tag{134}$$

Genuinely non-local utilities. A different situation arises for utilities that cannot be represented, even after finite augmentation, as an extensive additive term plus bounded boundary corrections. Examples include utilities involving running extrema, such as

$$u(\tau_T) = \max_{t\le T} X_t, \tag{135}$$

drawdown-type utilities such as

$$u(\tau_T) = -\max_{t\le T} (M_t - X_t), \tag{136}$$

or utilities depending on quantities such as the longest run observed so far.

For such utilities, the relevant memory variable may itself evolve over time and may not admit a finite-dimensional stationary augmentation. In that case the residual need not have the simple boundary form

$$\Phi(Z_{t+L}) - \Phi(Z_t), \tag{137}$$

and an action-dependent $B_\theta(s, a)$ may arise more naturally.

A useful classification. The preceding observations suggest the following classification:

Exactly additive	$R_{t,L} = 0$	$\Longrightarrow B_\theta = 0,$
Additive plus boundary/coboundary	$R_{t,L} = \Phi(Z_{t+L}) - \Phi(Z_t)$	$\Longrightarrow B_\theta(s, a)$ is typically state-only,
Genuinely non-local utility	$R_{t,L}$ has a more complex form	$\Longrightarrow B_\theta(s, a)$ may be action-dependent.

This gives a structural interpretation to the residual condition appearing in Theorem 2.

For an important class of path-dependent utilities, it follows naturally from the combination of asymptotic additivity, bounded boundary corrections, and a Markovian policy. This observation suggests the following additional result:

Theorem 5. *Suppose the incremental utility admits the representation*

$$\Delta u_t(L) = \sum_{k=t}^{t+L-1} g(Z_k, A_k) + \Phi(Z_{t+L}) - \Phi(Z_t) + o(1), \tag{138}$$

where Z_t is determined by the trajectory history up to the current state S_t, and suppose that the policy is Markovian,

$$A_t \sim \pi_\theta(\cdot \mid S_t). \tag{139}$$

Then

$$B_\theta(s, a) = B_\theta(s), \tag{140}$$

and consequently

$$\boxed{\sum_s \mu_\theta(s) \sum_a B_\theta(s, a)\pi_\theta(a \mid s)\nabla_\theta \log \pi_\theta(a \mid s) = 0.} \tag{141}$$

Hence the residual term does not affect the trajectory-based Policy Gradient Theorem.

D Detailed Algorithmic Development

This appendix provides the complete algorithmic development of the methods summarized in Section 7. We distinguish between two settings. When the occupation-measure functional $\mathcal{F}$ is known, ProxyAC constructs an equivalent per-step reward. When $\mathcal{F}$ is unknown and only trajectory-level utility feedback is available, AUAC directly estimates a trajectory action-value function from incremental utility observations.

D.1 ProxyAC for a Known Utility Functional

Let

$$\mu_\theta(s, a) = d_\theta(s)\pi_\theta(a \mid s)$$

denote the stationary state–action occupation measure induced by π_θ. Under Assumption 1, the APDU objective can be written as

$$J^u(\theta) = \mathcal{F}(\mu_\theta) = \rho^u(\theta).$$

Suppose that $\mathcal{F}$ is differentiable and define its functional gradient at μ_θ by

$$g_\theta(s, a) := \left.\frac{\partial \mathcal{F}(\mu)}{\partial \mu(s, a)}\right|_{\mu=\mu_\theta}. \tag{142}$$

A natural shadow reward is $g_\theta(s,a)$. Although this reward produces the desired policy-gradient direction, its stationary average is generally

$$\sum_{s,a} \mu_\theta(s,a) g_\theta(s,a) \neq \mathcal{F}(\mu_\theta).$$

Consequently, using $\rho^u(\theta)$ to center this reward gives an incorrect differential Bellman equation. We address this issue by adding a state–action-independent correction.

For a reference occupation measure μ, define the gain-matched proxy reward

$$r^u_\mu(s,a) = g_\mu(s,a) + \mathcal{F}(\mu) - \langle \mu, g_\mu \rangle , \tag{143}$$

where

$$g_\mu = \nabla_\mu \mathcal{F}(\mu), \qquad \langle \mu, g_\mu \rangle = \sum_{s,a} \mu(s,a) g_\mu(s,a).$$

D.1.1 Differential actor–critic update

For a frozen proxy reward r^u_μ, the differential value function satisfies the usual average-reward Bellman equation. Given a value approximation V_ϕ, the temporal-difference error is

$$\delta_t = r^u_\mu(s_t,a_t) - \widehat{\rho}^{\,u} + V_\phi(s_{t+1}) - V_\phi(s_t), \tag{144}$$

where

$$\widehat{\rho}^{\,u} = \mathcal{F}(\widehat{\mu})$$

is the estimated APDU gain. The critic and actor updates are

$$\phi \leftarrow \phi + \alpha\, \delta_t \nabla_\phi V_\phi(s_t), \tag{145}$$

$$\theta \leftarrow \theta + \beta\, \delta_t \nabla_\theta \log \pi_\theta(a_t \mid s_t), \tag{146}$$

respectively.

D.1.2 Shadow-reward baselines

ShadowAC uses

$$r^{\mathrm{sh}}_\mu(s,a) = g_\mu(s,a)$$

without the gain-matching constant. Because adding a constant does not change a policy gradient, ShadowAC and ProxyAC have the same exact policy-gradient direction when μ_θ and g_θ are known. However, ShadowAC must center its critic using its own stationary gain,

$$\rho^{\mathrm{sh}}(\theta) = \langle \mu_\theta, g_\theta \rangle .$$

ShadowAC-Biased instead centers the shadow reward using $\rho^u(\theta) = \mathcal{F}(\mu_\theta)$. Its TD error is therefore

$$\delta^{\mathrm{biased}}_t = g_\theta(s_t,a_t) - \rho^u(\theta) + V_\phi(s_{t+1}) - V_\phi(s_t).$$

Unless

$$\mathcal{F}(\mu_\theta) = \langle \mu_\theta, g_\theta \rangle ,$$

this update does not correspond to the differential Bellman equation of the shadow reward.

D.2 AUAC with Black-Box Utility Feedback

We next consider the setting in which neither $\mathcal{F}$ nor its functional gradient is available. The learner instead receives scalar utility feedback from a black-box evaluator.

For a time t and horizon H, recall the incremental utility

$$\Delta u_t(H) = u(\tau_{0:t+H}) - u(\tau_{0:t}), \tag{147}$$

where both terms are evaluated on prefixes of the same connected trajectory. Thus, the segment beginning at t is evaluated with its preceding history, and the utility evaluator is not implicitly restarted at the beginning of the segment.

Algorithm 2 ProxyAC

Require: Known functional $\mathcal{F}$; rollout length B; actor and critic learning rates β, α; occupation-measure update rate η_μ

Initialize policy π_θ, critic V_ϕ, and occupation estimate $\widehat{\mu}$
for each iteration **do**
  Freeze the behavior policy: $\theta_0 \leftarrow \theta$
  Collect B transitions using π_{θ_0}
  Form the batch occupation estimate

$$\overline{\mu}(s,a) = \frac{1}{B}\sum_{t=0}^{B-1} \mathbb{I}\{s_t = s, a_t = a\}$$

  Update

$$\widehat{\mu} \leftarrow (1-\eta_\mu)\widehat{\mu} + \eta_\mu \overline{\mu}$$

  Compute

$$g \leftarrow \nabla_\mu \mathcal{F}(\widehat{\mu}), \qquad \widehat{\rho}^{\,u} \leftarrow \mathcal{F}(\widehat{\mu})$$

  Construct

$$r^u(s,a) \leftarrow g(s,a) + \widehat{\rho}^{\,u} - \langle \widehat{\mu}, g \rangle$$

  for $t = 0, \ldots, B-1$ **do**
    Compute

$$\delta_t = r^u(s_t, a_t) - \widehat{\rho}^{\,u} + V_\phi(s_{t+1}) - V_\phi(s_t)$$

    Update the critic using δ_t
  end for
  Update the actor using

$$\theta \leftarrow \theta_0 + \frac{\beta}{B}\sum_{t=0}^{B-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)|_{\theta=\theta_0}\, \delta_t$$

end for

D.2.1 INCREMENTAL TRAJECTORY ACTION-VALUE

We define the finite-horizon trajectory action-value by

$$q^\tau_{\theta,H}(s,a) := \mathbb{E}_{\pi_\theta}\left[\Delta u_t(H) - H\rho^u(\theta) \mid S_t = s, A_t = a\right]. \tag{148}$$

The expectation includes the future trajectory and the stationary prehistory preceding (S_t, A_t). The prehistory is relevant to the utility evaluator but need not be observed by the learner.

When the centered limit exists, define

$$q^\tau_\theta(s,a) := \lim_{H\to\infty} q^\tau_{\theta,H}(s,a). \tag{149}$$

Under the regularity assumptions of the policy-gradient theorem,

$$\nabla_\theta J^u(\theta) = \mathbb{E}_{(S,A)\sim\mu_\theta}\left[\nabla_\theta \log \pi_\theta(A \mid S) q^\tau_\theta(S,A)\right]. \tag{150}$$

The action-value function is identifiable only up to a state–action-independent constant. Indeed, for any C_θ,

$$\mathbb{E}_{\mu_\theta}\left[\nabla_\theta \log \pi_\theta(A \mid S) C_\theta\right] = 0.$$

Consequently, an estimated trajectory action-value and the differential action-value induced by the proxy reward may differ by a constant while producing the same policy gradient.

Equation equation 148 replaces the restarted-window definition

$$\mathbb{E}[u(\tau_{t:t+H}) - H\rho^u \mid S_t = s, A_t = a].$$

The restarted definition discards utility terms that cross the beginning of the window and is therefore unsuitable for general history-dependent utilities.

D.2.2 UTILITY-RATE ESTIMATION

Consider a connected trajectory beginning after a warm-up period at time b. Divide the following $T = NH$ transitions into N non-overlapping windows with starting times

$$t_i = b + (i-1)H, \qquad i = 1, \dots, N.$$

For each window, query

$$\Delta u_i = \Delta u_{t_i}(H).$$

Because adjacent increments telescope,

$$\sum_{i=1}^{N} \Delta u_i = u(\tau_{0:b+T}) - u(\tau_{0:b}) = \Delta u_b(T). \tag{151}$$

A batch estimate of the utility rate is therefore

$$\widehat{\rho}^{\,u}_{\mathrm{batch}} = \frac{\Delta u_b(T)}{T} = \frac{1}{NH} \sum_{i=1}^{N} \Delta u_i. \tag{152}$$

We reduce its variance using an exponential moving average,

$$\overline{\rho}^{\,u} \leftarrow (1-\lambda)\overline{\rho}^{\,u} + \lambda \widehat{\rho}^{\,u}_{\mathrm{batch}}. \tag{153}$$

The Monte Carlo target for the ith window is

$$Y_i = \Delta u_i - H\overline{\rho}^{\,u}. \tag{154}$$

The critic is trained by minimizing

$$\mathcal{L}_q(\phi) = \frac{1}{N} \sum_{i=1}^{N} \left[q^{\tau}_{\phi}(s_{t_i}, a_{t_i}) - Y_i \right]^2. \tag{155}$$

D.2.3 ACTOR UPDATE

Once the critic has been fitted, AUAC uses

$$\widehat{g}_{\mathrm{AUAC}} = \frac{1}{T} \sum_{t=b}^{b+T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t) q^{\tau}_{\phi}(s_t, a_t) \tag{156}$$

as its policy-gradient estimator. Although critic targets are constructed at window starting points, stationarity allows the learned critic to be evaluated at every state–action pair in the batch. In practice, the window offset may also be randomized between iterations so that different transitions appear as window starting points.

Non-overlapping windows are used to reduce computation and excessive dependence between neighboring critic targets. They are not required by the theoretical definition of q^{τ}_{θ}.

D.3 NONLINEAR PREFIX FUNCTIONALS

For a nonlinear functional, it is important to distinguish an empirical rate from a cumulative utility. Suppose $\widehat{\mathcal{F}}_t$ is computed from the empirical occupation measure of a length-t prefix. The cumulative path utility is

$$u(\tau_{0:t}) = t\,\widehat{\mathcal{F}}_t, \tag{157}$$

and its increment is

$$\Delta u_t(H) = (t+H)\widehat{\mathcal{F}}_{t+H} - t\widehat{\mathcal{F}}_t. \tag{158}$$

Using

$$\widehat{\mathcal{F}}_{t+H} - \widehat{\mathcal{F}}_t$$

alone would compare two rates rather than the accrued utilities required by the APDU definition.

For a finite-memory utility with window length k, such as an *ABB* pattern count, we retain the evaluator's history and choose $W \geq k$. For a global nonlinear utility such as empirical KL divergence, the empirical accumulator may instead be initialized after the Markov chain burn-in and that point labeled $t = 0$; if an existing prefix is retained, a sufficiently long evaluation horizon is used to make its finite-prefix effect negligible.

Algorithm 3 Average Utility Actor–Critic (AUAC)

Require: Incremental utility evaluator Δu; warm-up length W; window length H; number of windows N; $T = NH$; EMA rate λ; critic and actor learning rates α, β

Initialize policy π_θ, critic q_ϕ^τ, and utility rate $\overline{\rho}^{\,u}$
for each iteration **do**
  Freeze the behavior policy: $\theta_0 \leftarrow \theta$
  Start a connected trajectory and run π_{θ_0} for W warm-up steps
  Set $b \leftarrow W$ and continue the same trajectory for T steps
  for $i = 1, \dots, N$ **do**
    Set $t_i \leftarrow b + (i-1)H$
    Query

$$\Delta u_i \leftarrow u(\tau_{0:t_i+H}) - u(\tau_{0:t_i})$$

  end for
  Estimate

$$\widehat{\rho}^{\,u}_{\text{batch}} \leftarrow \frac{1}{NH} \sum_{i=1}^{N} \Delta u_i$$

  Update

$$\overline{\rho}^{\,u} \leftarrow (1-\lambda)\overline{\rho}^{\,u} + \lambda \widehat{\rho}^{\,u}_{\text{batch}}$$

  for $i = 1, \dots, N$ **do**
    Set

$$Y_i \leftarrow \Delta u_i - H\overline{\rho}^{\,u}$$

  end for
  Update the critic by minimizing

$$\mathcal{L}_q(\phi) = \frac{1}{N} \sum_{i=1}^{N} \left[q_\phi^\tau(s_{t_i}, a_{t_i}) - Y_i \right]^2$$

  Update the actor:

$$\theta \leftarrow \theta_0 + \frac{\beta}{T} \sum_{t=b}^{b+T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)|_{\theta=\theta_0} \, q_\phi^\tau(s_t, a_t)$$

end for

D.4 Trajectory-Level REINFORCE Baseline

APDU-Reinforce uses the same incremental utility observations but does not learn an action-value critic. For window i, define

$$Y_i = \Delta u_i - H\overline{\rho}^{\,u}.$$

A likelihood-ratio gradient estimator is

$$\widehat{g}_{\text{RF}} = \frac{1}{NH} \sum_{i=1}^{N} Y_i \sum_{j=0}^{H-1} \nabla_\theta \log \pi_\theta \left(a_{t_i+j} \mid s_{t_i+j} \right). \tag{159}$$

The centering term $H\overline{\rho}^{\,u}$ acts as a baseline and does not change the expected likelihood-ratio gradient. In contrast to AUAC, however, the same window-level return is multiplied by every score term in the window, resulting in substantially higher variance.

E Experiments

To verify our results by experinments, we implement following algorithms:

1. **AUAC**, The algo proposed in this paper, see section 3

2. **ProxyAC** AC with proxy reward (Eq.2 and update critic with Poisson Equation $q^u(s,a) = r^u(s,a) - \rho + v(s')$),

3. **ShadowAC** AC with shadow reward: (D.1.2 without correction item in proxy reward and update critic with Poisson Equation $q^u(s,a) = r^u(s,a) - \rho + v(s')$ where ρ is the mean r^u.),

4. **ShadowAC-biased** AC with shadow reward and biased (D.1.2 without correction item in proxy reward and update critic with Poisson Equation $q^u(s,a) = r^u(s,a) - \rho^u + v(s')$ where ρ^u is the average path dependent utility),

5. **APDU-Reinforce**D.4 (extended from MOJPG, we use trajectory utility as the policy evaluation).

To make the comparison fair, without addiontal explanation, each algorithm have the similar framework, run a same length trajectory then update once; the other settings like policy and critic are the same. For those algorihthms require proxy(or shadow reward), we take a tabulor distribution for μ and use autodiff in pytorch to compute the first order deriviate. In addition, because $\rho^u \neq \rho$ in ShadowAC-biased, this version is biased and easily ignored in practical.

E.1 Minimising the KL-divergence to a target distribution

The experiments conducted in this section (see Fig 3) are based on the simple 2-state use case in Section. 8.1 where the policy is trying to minimise the KL-divergence to a target distribution. Notice that since we are trying to drive the utility functional $J(\theta) = \rho(\theta)$ measuring the KL-divergence (between the actual and target distribution) to zero, we recast this problem as maximising $-J(\theta) = -\rho(\theta)$ to agree with the optimisation convention in this paper.

Except the APDU-Reinforce, all algorithms converge to the optimal average path-dependent utility as wanted; in the parameter θ space, similarly, the four successful algorithms converge to the optimal $\theta^* = \frac{1}{3}$.

We notice, APDU-Reinforce converge very slowly afer several millions steps. The behind reason is that for fairness and approximating statrionary distribution, all the algorithms update once after a long trajectory with the same length. Also the APDU-Reinforce shows a larger variance.

ShadowAC and our ProxyAC shows the equivalent performance in both fingures. This verify our arguments these two proxy reward both can give unbiased policy gradient estimator.

ShadowAC(biased) and AUAC are slightly worse than ProxyAC and ShadowAC. As we discuss before, $\rho^u \neq \rho$ will introduce bias in critic update; this difference is very small in this case but exist. Our AUAC is proved to solve the problem without knowing the explicit utility or functional; but $\hat{q^u}$ is a relative limit when time goes to infinity; and without using any kind of r per step, Monte Carlo sampling the trajectory utility is required; this procedure require us to sample multiple trajectories as long as possible. The bias originate from the trajectories count and the length. In this experinment, we take a trajectory(length is 1000) as all other algorithms, then cut it into 50 small trajectories(each length is 20).

Algorithm Setup. Each update consumes a batch of $\texttt{horizon}T = 1000$ collected steps, split into $N = 10$ non-overlapping windows, so the stride equals the window length $H = \lfloor \text{horizon}/N \rfloor$). The critic is a multilayer perceptron with hidden layers $[50, 50, 50]$, fitted with 5 full-batch regression sweeps per update at learning rate $3e{-}3$; the actor is a single-parameter policy initialised at $\theta_0 = 0.95$ and trained at learning rate $3e{-}2$. Both are optimised by Adam.

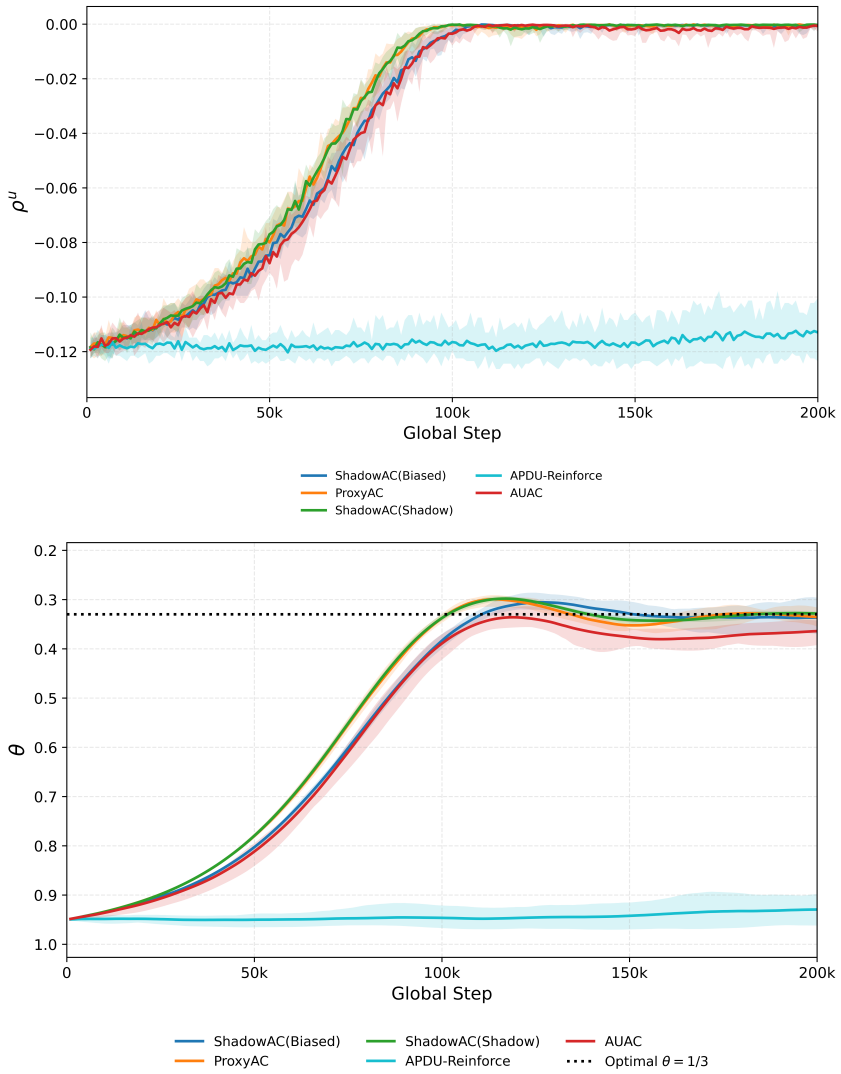


Figure 3: ABB Game with KL divergence as the objective.

E.2 Two-State MDP with ABB-Pattern Utility

We consider the two-state MDP with utility given by the long-run frequency of the temporal pattern *ABB*. The analytically optimal policy parameter is $\theta^\star = \sqrt{2} - 1 \approx 0.414$. Figure 4 presents the training results. Temporal-pattern objectives are common in applications and provide a simple example of preferences that cannot be represented directly by the current state alone.

For ProxyAC and ShadowAC, we use the Markovian proxy rewards derived from the explicit functional through $\partial F/\partial \mu^1$. Since F is linear in this example, the gain-matching correction vanishes, and ProxyAC, ShadowAC, and ShadowAC-Biased use the same per-step reward and baseline. Accordingly, the three methods exhibit nearly identical learning behavior and converge to the analytical optimum. This result supports both the proxy construction and its equivalence to the shadow reward for linear functionals.

As an ablation, we implement ProxyAC-PO, which uses the raw *ABB* pattern indicator as its per-step reward while conditioning both the actor and critic only on the current state. This reward depends on the preceding states and is therefore non-Markovian with respect to the critic input. Consequently, the standard one-step Poisson equation on the original state space does not apply. As shown by the purple curve, ProxyAC-PO overshoots $\theta^\star$ and converges to a suboptimal policy.

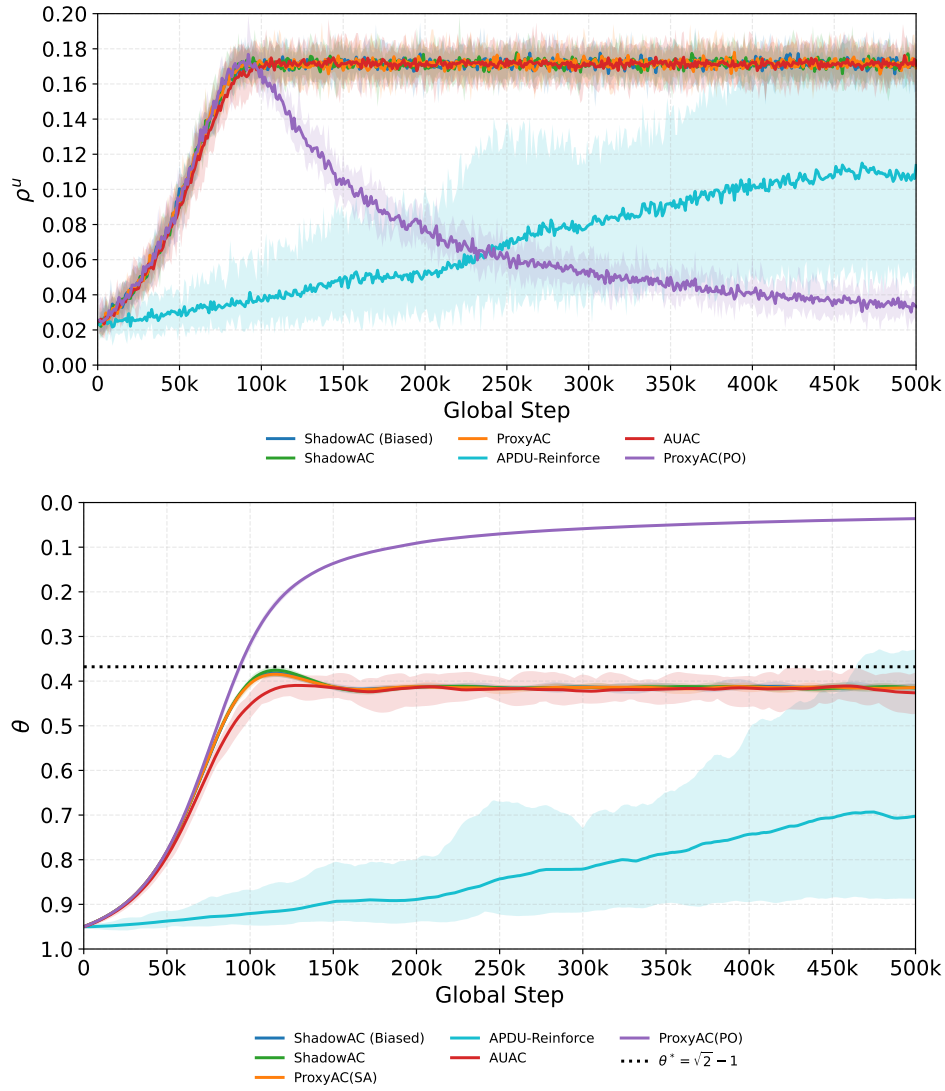


Figure 4: ABB Game with average ABB count as the objecive.

In contrast, the derived proxy reward restores a Markovian reward representation on the original state–action space, without augmenting the policy input with trajectory history.

AUAC also conditions its actor and critic on the original state, yet converges to the optimal policy. Unlike the one-step TD ablation, AUAC uses Monte Carlo targets based on incremental trajectory utility, which retain the history required to evaluate the *ABB* pattern. It therefore avoids the misspecified one-step Poisson equation. APDU-Reinforce uses the same trajectory-level feedback but learns much more slowly because of its higher-variance Monte Carlo updates.

Algorithm Setup. Parameters stay the same as what they are in the before KL examples. For the ABB game, every time the algorithm call the black box $u(\tau)$ that u will automatically use hidden prefix history to compute the $\Delta_1 u(T)$.

E.3 Simple Policy θ in ABB Game 3: Entropy Rate

There is another utility function Entropy Rate as objective: $J^u = -\sum_{s,a} \mu(s,a) \log \frac{\mu(s,a)}{d(s)}$; the optimal policy $\theta^* = (3 - \sqrt{5})/2 \approx 0.382$ The Figure 5 shows the result. In the figures, it can be

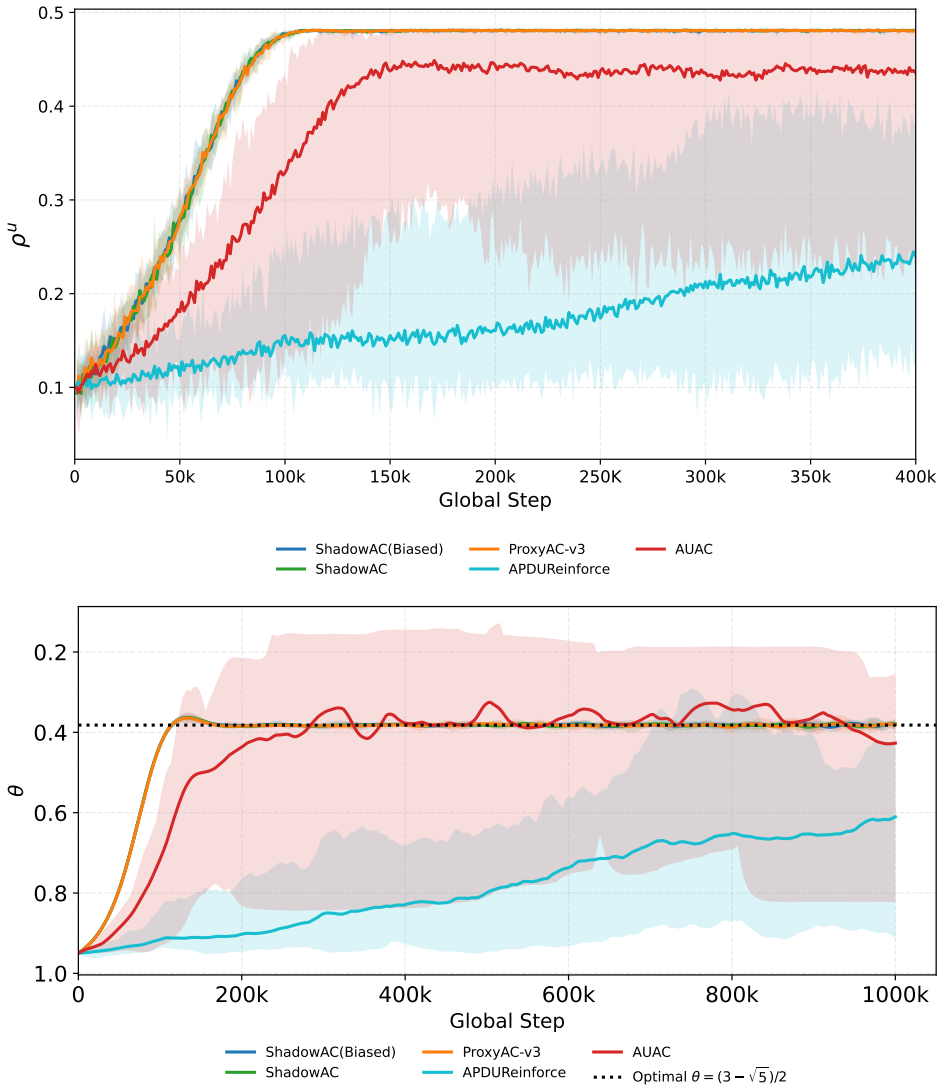


Figure 5: ABB Game with Entropy Rate as the objecive.

observed: APDUReinforce still learns slow; the three AC method with proxy reward converge in similar pace and converge to the optimal θ^* ; with splitting a trajectory(length 1000) into 10 shorter ones, AUAC converge to a slightly biased result. The bias of AUAC comes from the limited splitting number and not long enough trajectory. Without neccesity to know the utility function, we argue that the AUAC result is a reasonable choice with limited resources.

Algorithm Setup. They stay the same as what they are in the before KL examples.

E.4 A FURTHER EXAMPLE IN QUEUE CONTROL

In this section, we implement our algorithms with neural network; in addition we process a harder problem with stochastic transition and larger space, Queue Control.

Queue Control We evaluate on the access-control queuing task of Sutton and Barto Sutton et al. (1998), in a two-server, vector-reward formulation. At each step a single customer stands at the head of the queue with a priority drawn i.i.d. as high with probability $\alpha = 0.3$ and low otherwise, and the controller either admits or rejects them. The state is $s = (free_server, r)$ with $free_server \in$

$[0, 10]$ the number of idle servers and $r \in \{1, 2\}$ the priority of the waiting customer, giving a 22-state chain, and the action is $a \in \{reject(0), accept(1)\}$. An admission with $F > 0$ occupies one server and yields the vector reward $r_t = \big[\mathbb{1}\{\text{low accepted}\}, \mathbb{1}\{\text{high accepted}\}\big]$; each busy server then releases independently with probability $p_{\text{free}} = 0.1$ per step, so one admission commits capacity for a geometric duration of mean $1/p_{\text{free}} = 10$ steps, while rejection is costless (there is no waiting line and no holding cost). The task is continuing and average-reward, and the trade-off is purely one of reserving capacity for high-priority arrivals. Episodes are truncated at 5000 steps for training and 2000 for evaluation, and after each reset (which starts from $free_server = 10$) the state is advanced by 200 steps without collecting data so that trajectories begin near the stationary distribution μ_π.

Algorithm Setup We run 10^6 environment steps with a collection horizon of 1000 split into windows of $H = 100$. The actor is a three-layer MLP softmax policy: the observation is affinely rescaled to $[-1, 1]$ coordinatewise from the bounds of the observation space, passed through three hidden layers of 64 units with $\tanh$ activations and orthogonal initialisation, and mapped to two logits followed by a softmax. The forced rejection at zero free servers is imposed as a hard mask, $\pi(reject \mid free_server = 0) = 1$, so that an admitted action always corresponds to an actual admission; the accept logit of the output layer is initialised with a bias of 2.0, which starts the policy near accept-all. The actor is trained by Adam at learning rate 3×10^{-4} since a flexible policy can saturate to determinism and stall its own gradient. The critic is the action-conditioned network , an MLP on the concatenation of the observation with a one-hot encoding of the action, with three hidden layers of 50 units, $\tanh$ activations and orthogonal initialisation, trained by Adam at learning rate 10^{-3} with 5 sweeps over each collected batch and used as a baseline in the actor update. The per-step utility rate ρ^u is tracked by an exponential moving average with step size 0.1. Training episodes are truncated at 5000 steps and evaluation episodes at 2000, with 200 uncollected warm-up steps after each reset and eight evaluation episodes every 10^5 steps.

KL divergence The second objective keeps the environment reward and adds a penalty on how far the controller's admission behaviour drifts from a prescribed profile:

$$u(\tau) = \sum_{t=0}^{H-1} r_t - \lambda HKL(d||\hat{\mu}(c, a)), \quad \lambda = 2.$$

The first term is the accrued vector reward scalarised by w, so an admitted high-priority client is worth twice a low-priority one. The second term acts on the empirical *client–action* distribution of the window: writing $c(s_t) \in \{\text{low}, \text{high}\}$ for the priority of the client facing the controller at step t and a_t for its decision, $\hat{\mu}$ is the histogram of the four cells $(c(s_t), a_t)$ over the H acting states of τ, normalised by H. The target(generated randomly) is

$$d = (0.145,\ 0.555,\ 0.017,\ 0.283)$$

over $((1, Reject), (1, Accept), (2, Reject), (2, Accept))$.

The KL is taken with the target in the first argument, $\mathrm{KL}(d\|\hat{\mu})$, so every cell of d with positive mass must be covered by the trajectory, and the objective is a coverage requirement rather than a mode-seeking one. Since $\hat{\mu}$ can be exactly zero in a finite window, the empirical occupancy is floored at $0.5/H$ inside the logarithm; the resulting penalty for an unobserved cell scales with d_k, so the rare cell costs only about 2λ per window in which it is missed. Both terms are accrued over the window rather than averaged.

The result(Fig 6) is similar to what happen in 2 state MDP. AUAC for not knowing the utility function, it converges slower than those with per step proxy reward and slightly biased. In addition, in this harder problem, the bias from shadowAC(biased) is very small and can be hard to tell.

Entropy Rate The objective replaces the divergence penalty with one on the *predictability* of the controller's behaviour:

$$u(\tau) \ = \ \sum_{t=0}^{H-1} r_t - \lambda H\hat{h}(\tau),, \quad \lambda = 2.$$

The reward term is unchanged. The penalty acts on the same four-cell process as the previous objective, $X_t = \big(c(s_t), a_t\big)$ with cell index $2\mathbb{1}[c = \text{high}] + a$, but now measures the entropy rate of

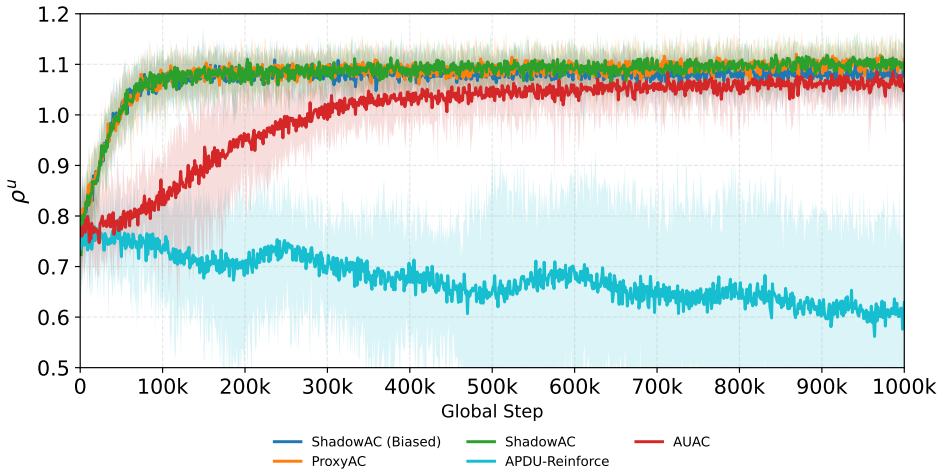


Figure 6: Queue Control tasks. Path dependent utility $u(\tau) = \sum r_t + \lambda TKL(\mu(free_server, a||d_t))$. Both actor and critic are implemented by Neural Network(MLP).

that process rather than its marginal occupancy. We use the plug-in bigram estimator

$$\hat{h}(\tau) = \hat{H}\big(X_{t+1} \mid X_t\big) = H(\hat{\mu}_2) - H(\hat{\mu}_1)$$

in nats, where $\hat{\mu}_2$ is the empirical distribution of consecutive pairs (X_t, X_{t+1}) for $t = 0, \ldots, H-2$ and $\hat{\mu}_1$ is the empirical distribution of the first elements of those same pairs.

We use the same algorithms setup from the KL divergence case, except: AUAC has learning rate for actor($3e{-}3$) and critic($1e-3$); $T = 4000$ and $n_{split} = 5$ in AUAC; other algorihtms have learning rate for actor($3e{-}4$) and critic($1e-3$).

The result(Fig 7) is still similar to the 2 state MDP result. Except for the REINFORCE one, they all converge; still those(ProxyAC, ShadowAC and ShadowAC(biased)) slightly outperforms the AUAC. But without knowing the explicit form of u, the small bias can be accepted. We notice that AUAC here pass by the better policy and is stucked to a suboptimal one. The reason behind should be the critic update mechanism on ProxyAC etc and AUAC are different. The entropy rate part has high variance in limited time window(horizons for windows and window counts); then the critic(updated per trajectory) is limited without enough samples; this leads to the scenario that even in the better policy, the critic cannot offer precise enough estimation.

reward pattern

$$u(\tau) = \sum r_t + \lambda * E[\sum r(s_{t-2}, a_{t-2}, s_{t-1}, a_{t-1}, s_t)]$$

Because with 22 states, the combination of possible time windows can be large. We use a simple rule here to penalty the behavior pattern. Sill $c(s_t)$ is the client type at time t and state s_t; we only give penalty -1 when $c(s_{t-2}) = c(s_{t-1})$ and $a_{t-2} = a_{t-1}$; the scale factor $\lambda = 1$. We use the same algorithms setup from the KL divergence case. For accetping two type of clients, $r = 0.25$ or $r = 2$ seprately. For convenience, we will penalty $r = -1$ if the consecutive rejection happens.

We use the same algorithms setup , except: learning rate for actor and critic are both $1e{-}3$ for every algorithms.

The result(Fig 8) is still similar to the 2 state MDP result. Except for the REINFORCE one, they all converge; still those(ProxyAC, ShadowAC and ShadowAC(biased)) slightly outperforms the AUAC(small bias). Also we notice the variance from the AUAC at begining is large but decrease soon. The purple line (ProxyAC(PO) use non Markovian reward) is suboptimal.

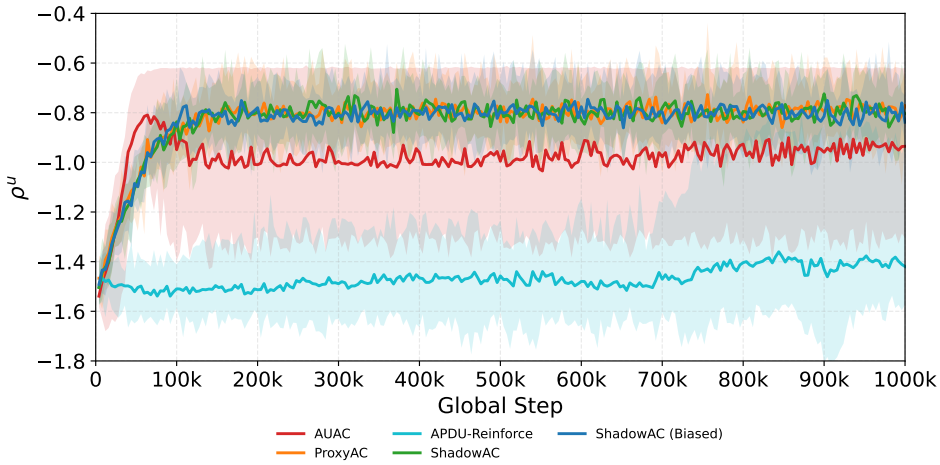


Figure 7: Queue Control tasks. Path dependent utility $u(\tau) = \sum r_t + \lambda T \sum_{s,a} \mu(s,a) \log \frac{\mu(s,a)}{d(s)}$. Both actor and critic are implemented by Neural Network(MLP).

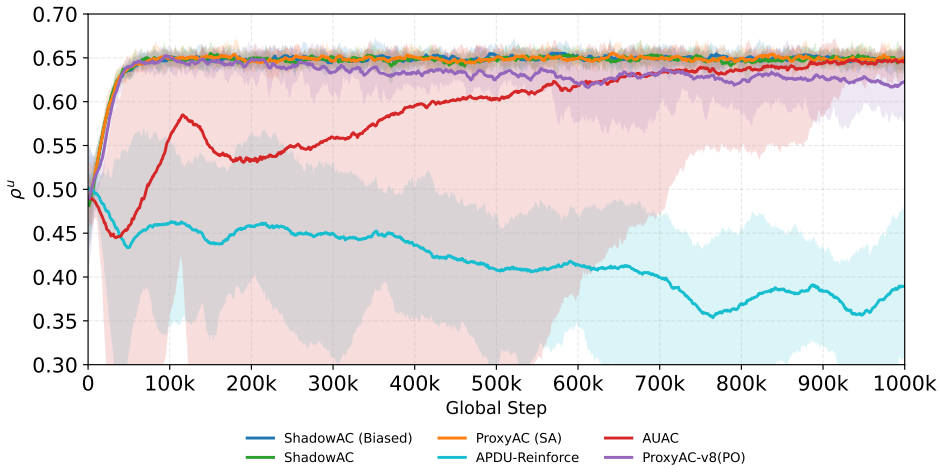


Figure 8: Queue Control tasks(reward pattern).

E.5 VERIFY THE RELATIONSHIP AMONG q^u AND q^τ

In this section, we enumerate the policy then empirically compute the q^u(with r^u) as well as the the $q^{\hat{u}}$ we introduce. It could be seen that once policy is fixed, the difference between them are a fixed value conditioned on policy for any action-state pair.

In the Table 1and Table 2 shows both q^u with (proxy reward) and $q^t au$ when different policy θ is given; for both, we give exact value by analysis before and give estimation with sufficient and long enough trajectories. It proves our theorem that for any given θ, the difference between q^u and $q^{\hat{u}}$ is a fixed value.

Table 1: **Entropy Rate Case** Monte-Carlo verification of $q^\tau(s,a) = q^u + C_\theta$ in the two-state example. Both estimators are computed on the *same* trajectories, so the shared $O_p(\sqrt{T})$ linear fluctuation cancels in the per-trajectory difference; this is why $\hat{C}$ is resolved an order of magnitude more precisely than either column it is formed from. Errors are one standard error; $T = 800$, $N = 4 \times 10^5$ trajectories per pair.

		$q^\tau(s,a)$		$q^u(s,a)$		
θ	(s,a)	simulated	exact	simulated	exact	$\hat{C}$ (paired)
0.2	$(A, switch)$	-0.3443 ± 0.0159	-0.347502	-0.8464 ± 0.0159	-0.847502	-0.50209 ± 0.00112
	$(B, switch)$	$+0.8600 \pm 0.0158$	$+0.844934$	$+0.3580 \pm 0.0157$	$+0.344934$	-0.50204 ± 0.00112
	$(B, stay)$	-0.1452 ± 0.0158	-0.124358	-0.6443 ± 0.0159	-0.624358	-0.49909 ± 0.00112
$\theta^\star$	$(A, switch)$	-0.3479 ± 0.0003	-0.348208	-0.8499 ± 0.0012	-0.848208	-0.50204 ± 0.00112
	$(B, switch)$	$+0.1328 \pm 0.0003$	$+0.133004$	-0.3698 ± 0.0012	-0.366996	-0.50259 ± 0.00112
	$(B, stay)$	$+0.1331 \pm 0.0003$	$+0.133004$	-0.3669 ± 0.0012	-0.366996	-0.49997 ± 0.00112
0.6	$(A, switch)$	-0.2615 ± 0.0143	-0.262895	-0.7627 ± 0.0144	-0.762895	-0.50118 ± 0.00112
	$(B, switch)$	-0.1777 ± 0.0143	-0.172702	-0.6805 ± 0.0144	-0.672702	-0.50281 ± 0.00113
	$(B, stay)$	$+0.6502 \pm 0.0143$	$+0.653396$	$+0.1509 \pm 0.0143$	$+0.153396$	-0.49928 ± 0.00112

Table 2: Two-state ABB toy with the *reverse* KL objective $J(\theta) = \mathrm{KL}(d_t \| d_\theta)$, $d_t = (\frac{1}{4}, \frac{3}{4})$, $d_\theta = (\frac{\theta}{1+\theta}, \frac{1}{1+\theta})$. Monte-Carlo estimates ($N = 10^5$ trajectories, horizon $T = 10^3$, window $s_1, \ldots, s_T$) against the closed forms $\bar{q}(s,a) = h\big((s+a) \bmod 2\big)$, $h = g/(1+\theta)$, $g(s) = 1 - d_t(s)/d(s)$, and $C_\pi = \frac{1}{2}\,\mathrm{tr}\big(\nabla^2 F(d)\,\Sigma\big) = \frac{(1-\theta)(1+3\theta^2)}{8\theta(1+\theta)}$. Standard errors in parentheses. $\hat{C}_\pi$ is evaluated on the *same* trajectories as $\hat{q}^u$ and $\hat{\bar{q}}$, so the leading $O(\sqrt{T})$ fluctuation cancels and its error is one to two orders of magnitude smaller.

		q^τ		q^u		$C_\pi = q^\tau - q^u$	
θ	(s,a)	$\hat{q}^\tau$	exact	$\hat{q}^u$	exact	$\hat{C}_\pi$	exact
0.1	(A,sw)	1.2214 (0.0506)	1.2125	0.1632 (0.0499)	0.1591	1.0582 (0.0048)	1.0534
	(B,sw)	−0.5761 (0.0498)	−0.5375	−1.6243 (0.0499)	−1.5909	1.0482 (0.0047)	1.0534
	(B,st)	1.2221 (0.0508)	1.2125	0.1563 (0.0501)	0.1591	1.0657 (0.0048)	1.0534
	(A,st)	−0.5489 (0.0500)	−0.5375	−1.6064 (0.0501)	−1.5909	1.0575 (0.0047)	1.0534
0.2	(A,sw)	0.5314 (0.0186)	0.5500	0.0610 (0.0183)	0.0833	0.4704 (0.0021)	0.4667
	(B,sw)	0.0437 (0.0183)	0.0500	−0.4244 (0.0183)	−0.4167	0.4681 (0.0021)	0.4667
	(B,st)	0.5606 (0.0185)	0.5500	0.0918 (0.0182)	0.0833	0.4687 (0.0021)	0.4667
	(A,st)	0.0516 (0.0183)	0.0500	−0.4182 (0.0183)	−0.4167	0.4698 (0.0021)	0.4667
$\theta^* = \frac{1}{3}$	(A,sw)	0.2532 (0.0011)	0.2500	0.0000 (0.0000)	0.0000	0.2532 (0.0011)	0.2500
	(B,sw)	0.2505 (0.0011)	0.2500	0.0000 (0.0000)	0.0000	0.2505 (0.0011)	0.2500
	(B,st)	0.2505 (0.0011)	0.2500	0.0000 (0.0000)	0.0000	0.2505 (0.0011)	0.2500
	(A,st)	0.2510 (0.0011)	0.2500	0.0000 (0.0000)	0.0000	0.2510 (0.0011)	0.2500
0.5	(A,sw)	0.0633 (0.0102)	0.0625	−0.0837 (0.0102)	−0.0833	0.1470 (0.0007)	0.1458
	(B,sw)	0.3152 (0.0102)	0.3125	0.1694 (0.0102)	0.1667	0.1458 (0.0007)	0.1458
	(B,st)	0.0651 (0.0102)	0.0625	−0.0809 (0.0102)	−0.0833	0.1461 (0.0007)	0.1458
	(A,st)	0.3037 (0.0102)	0.3125	0.1575 (0.0102)	0.1667	0.1462 (0.0007)	0.1458
0.7	(A,sw)	−0.0975 (0.0138)	−0.0839	−0.1760 (0.0139)	−0.1618	0.0785 (0.0004)	0.0778
	(B,sw)	0.2927 (0.0138)	0.3089	0.2150 (0.0138)	0.2311	0.0777 (0.0003)	0.0778
	(B,st)	−0.0764 (0.0138)	−0.0839	−0.1542 (0.0138)	−0.1618	0.0777 (0.0003)	0.0778
	(A,st)	0.3086 (0.0138)	0.3089	0.2305 (0.0138)	0.2311	0.0781 (0.0003)	0.0778
0.9	(A,sw)	−0.2113 (0.0103)	−0.1986	−0.2366 (0.0103)	−0.2237	0.0253 (0.0001)	0.0251
	(B,sw)	0.2597 (0.0103)	0.2736	0.2345 (0.0103)	0.2485	0.0252 (0.0001)	0.0251
	(B,st)	−0.2006 (0.0103)	−0.1986	−0.2259 (0.0103)	−0.2237	0.0253 (0.0001)	0.0251
	(A,st)	0.2720 (0.0103)	0.2736	0.2466 (0.0103)	0.2485	0.0254 (0.0001)	0.0251