Dual-Objective Reinforcement Learning with Novel Hamilton-Jacobi-Bellman Formulations

William Sharpless* wsharpless@ucsd.edu

Dylan Hirsch* dhirsch@ucsd.edu

Sander Tonkens stonkens@ucsd.edu

Nikhil Shinde nshinde@ucsd.edu

Sylvia Herbert sherbert@ucsd.edu

Abstract: Hard constraints in reinforcement learning (RL), whether imposed via the reward function or the model architecture, often degrade policy performance. Lagrangian methods offer a way to blend objectives with constraints, but often require intricate reward engineering and parameter tuning. In this work, we extend recent advances that connect Hamilton-Jacobi (HJ) equations with RL to propose two novel value functions for dual-objective satisfaction. Namely, we address: (1) the **Reach-Always-Avoid** problem – of achieving distinct reward and penalty thresholds – and (2) the **Reach-Reach** problem – of achieving thresholds of two distinct rewards. We derive explicit, tractable Bellman forms in this context by decomposing our problem. The RAA and RR problems are fundamentally different from standard sum-of-rewards problems and temporal logic problems, providing a new perspective on constrained decision-making. We leverage our analysis to propose a variation of Proximal Policy Optimization (**DO-HJ-PPO**), which solves these problems. Across a range of tasks for safe-arrival and multi-target achievement, we demonstrate that DO-HJ-PPO out-competes many baselines.

1 Introduction

In a safety-critical scenario, an infinite-horizon accumulation of costs does not properly account for safety violations. Rather, Bellman equations that encode best (or worst) values over time have allowed RL to generalize to these and other relevant problems. These equations, including the Safety Bellman Equation (SBE) [1] and Reach-Avoid Bellman Equation (RABE) [2], are derived from the Hamilton-Jacobi (HJ) perspective of dynamic programming, and directly propagate the best/worst values encountered over time. By focusing on extremal rewards and penalties, rather than their sums, qualitatively distinct behaviors arise that act with respect to the best or worst outcomes in time-optimal fashions [1, 2, 3, 4]. Ultimately, this yields policies with significantly improved performance in target-achievement and obstacle-avoidance tasks over long horizons [3, 4, 5, 6].

In this work, we advance existing HJ-RL formulations to a broader class of problems. HJ-RL Bellman equations are limited to: 1) Reach (R), wherein the agent seeks to reach a goal (i.e. achieve a reward threshold), 2) Avoid (A), wherein the agent seeks to avoid an obstacle (i.e. avoid a penalty threshold), and 3) Reach-Avoid (RA), a combination where the agent reaches a goal while avoiding obstacles on the way. In this light, we extend the HJ-RL Bellman equations to larger problems concerned with dual-satisfaction tasks, namely the Reach-Reach (reach two goals) and Reach-Always-Avoid (continue avoiding hazards after successfully reaching a goal) problems, demonstrated in Figure 1. Our contributions include:

- We introduce novel value functions for the RAA and RR problems.
- We prove these novel value functions and their optimal policies can be decomposed into R, A, and RA value functions.
- We leverage design a novel PPO-based algorithm DO-HJ-PPO for solving the RAA and RR problems.

Workshop on Safe and Robust Robot Learning for Operation in the Real World at CoRL 2025.

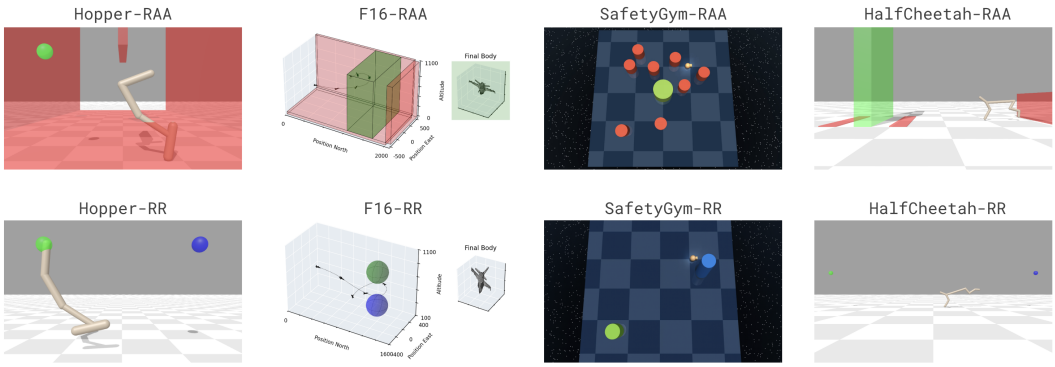

Figure 1: **Depiction of the Reach-Always-Avoid (RAA) and Reach-Reach (RR) Tasks** In the RAA tasks, the zero-level set of the rewards (goals) and penalties (obstacles) are depicted in **green** and **red** respectively, while in the RR problem, the zero-level set of the two rewards (two goals) are depicted in **green** and **blue**.

2 Related Works

This work involves aspects of safety (e.g. hazard avoidance), liveness (e.g. goal reaching), and balancing competing objectives. We summarize the relevant related works here.

Constrained RL and Multi-Objective RL. Constrained Markov decision processes (CMPDs) maximize the expected sum of discounted rewards subject to an expected sum of discounted costs, or an instantaneous safety violation function remaining below a set threshold [7, 8, 9, 10, 11]. CMDPs are an effective way to incorporate state constraints into RL problems, and the efficient and accurate solution of the underlying optimization problem has been extensively researched, first by Lagrangian methods and later by an array of more sophisticated techniques [12, 13, 14, 15, 16, 17, 18, 19, 20]. Multi-objective RL is an approach to designing policies that obtain *Pareto-optimal* expected sums of discounted *vector-valued* rewards [21, 22, 23], including by deep-Q and other deep learning techniques [24, 25, 26]. By contrast, this work explicitly balances rewards and penalties in a way that does not require specifying a Lagrange multiplier or similar hyperparameter.

Hierarchical RL. Hierarchical RL represents a large body of work related to learning how to decompose challenging problems into lower-level tasks, solve these simpler tasks, and recompose them [27, 28]. This has been studied for decades [29, 30], with more recent approaches employing representation learning [31], stochastic deep learning [32], off-policy RL [33], continuous adaptation of the low-level policies [34], and skill-transfer [35]. While this line of work is similar to ours in spirit, most hierarchical RL problems still involve optimizing the expected sum of discounted rewards, which will lead to non-optimal policies in our case, and do not usually involve constraints.

Linear Temporal Logic (LTL), Automatic State Augmentation, and Automatons. Many works have been explored that merge LTL and RL, canonically focused on Non-Markovian Reward Decision Processes (NMRDPs) [36]. Here, the reward gained at each time step may depend on the previous state history. Many of these works convert these NMRDPs to MDPs via state augmentation [36, 37, 38, 39, 40, 41]. Often the augmented states are taken to be products between an ordinary state and an automaton state, where the automaton is used to determine "where" in the LTL specification an agent currently is. Other works using RL for LTL tasks involve MDP verification [42], hybrid systems theory [43], GCRL with complex LTL tasks [44], almost-sure objective satisfaction [45], incorporating (un)timed specifications [46], and using truncated LTL [47]. While the problems we attempt to solve (e.g. reaching multiple goals) can be thought of as specific instantiations of LTL specifications, our approach to solving these problems is fundamentally different from those in this line of work. Our state augmentation and subsequent decomposition of the problem are performed in a specific manner to leverage new HJ-based methods on the subproblems. Through our specific choice of state augmentation, we still prove that we can achieve an optimal policy in theory (and approximately so in practice) despite the non-NMRDP setup.

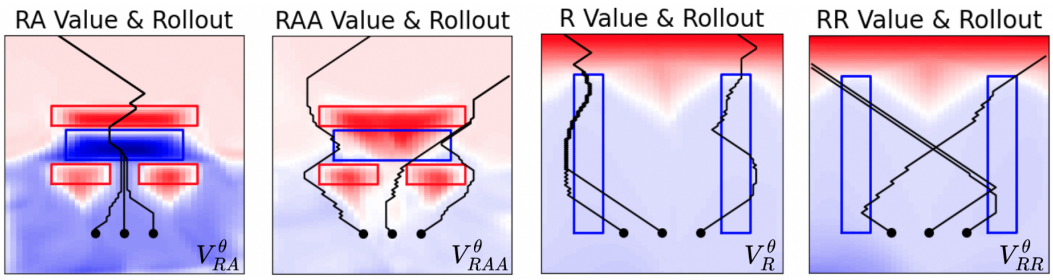

Figure 2: **DDQN Grid-World Demonstration of the RAA & RR Problems** We compare our novel formulations with previous HJ-RL formulations (RA & R) in a simple grid-world problem with DDQN. The zero-level sets of q (*hazards*) are highlighted in **red**, those of r (*goals*) in **blue**, and trajectories in **black** (starting at the dot). In both models, the agents actions are limited to {left, right, straight} and the system flows upwards over time.

Hamilton-Jacobi (HJ) Methods. HJ is a dynamic programming-based framework for solving reach, avoid, and reach-avoid tasks [48, 49]. The value functions used in HJ have the advantage of directly specifying desired behavior, so that a positive value corresponds to task achievement and a negative value corresponds to task failure. Recent works use RL to find corresponding optimal policies by leveraging the unconventional Bellman updates associated with these value functions [4, 50, 2, 1]. We build on these works by extending these advancements to more complex tasks, superficially mirroring the progression from MDPs to NMRDPs in the LTL-RL literature. Additional works merge HJ and RL, but do not concern themselves with such composite tasks [3, 5, 51].

3 Problem Definition

Consider a Markov decision process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, f \rangle$ consisting of finite state and action spaces $\mathcal{S}$ and $\mathcal{A}$, and *unknown* discrete dynamics f that define the deterministic transition $s_{t+1} = f(s_t, a_t)$. Let an agent interact with the MDP by selecting an action with policy $\pi : \mathcal{S} \to \mathcal{A}$ to yield a state trajectory s_t^π, i.e. $s_{t+1}^\pi = f\left(s_t^\pi, \pi\left(s_t^\pi\right)\right)$.

In this work, we consider the **Reach-Always-Avoid** (RAA) and **Reach-Reach** (RR) problems, which both involve the composition of two objectives, which are each specified in terms of the best reward and worst penalty encountered over time. In the RAA problem, let $r, p : \mathcal{S} \to \mathbb{R}$ represent a reward to be maximized and a penalty to be minimized. We will let $q = -p$ for mathematical convenience, but for conceptual ease we recommend the reader think of trying to minimize the largest-over-time penalty p rather than maximize the smallest-over-time q. In the RR problem, let $r_1, r_2 : \mathcal{S} \to \mathbb{R}$ be two distinct rewards to be maximized. The agent's overall objective is to maximize the *worst-case* outcome between the best-over-time reward and worst-over-time penalty (in RAA) and the two best-over-time rewards (in RR), i.e.

$$(\text{RAA}) \begin{cases} \text{maximize (w.r.t. } \pi) & \min\left\{\max_t r(s_t^\pi),\ \min_t q(s_t^\pi)\right\} \\ \text{s.t.} & s_{t+1}^\pi = f\left(s_t^\pi, \pi\left(s_t^\pi\right)\right), \\ & s_0^\pi = s, \end{cases}$$

$$(\text{RR}) \begin{cases} \text{maximize (w.r.t. } \pi) & \min\left\{\max_t r_1(s_t^\pi),\ \max_t r_2(s_t^\pi)\right\} \\ \text{s.t.} & s_{t+1}^\pi = f\left(s_t^\pi, \pi\left(s_t^\pi\right)\right), \\ & s_0^\pi = s. \end{cases}$$

As the problem names suggest, these optimization problems are inspired by (but not limited to) tasks involving goal reaching and hazard avoidance. More specifically, the RAA problem is motivated by a task in which an agent wishes to both reach a goal $\mathcal{G}$ and perennially avoid a hazard $\mathcal{H}$ (even after it reaches the goal). The RR problem is motivated by a task in which an agent wishes to reach two goals, $\mathcal{G}_1$ and $\mathcal{G}_2$, in either order. While these problems are thematically distinct, they are mathematically complementary (differing by a single $\max/\min$ operation), and hence we tackle them together.

The values for any policy in these problems then take the forms V^π_{RAA} and V^π_{RR},

$$V^\pi_{\text{RAA}}(s) = \min\left\{\max_t r(s_t^\pi),\ \min_t q(s_t^\pi)\right\} \quad \text{and} \quad V^\pi_{\text{RR}}(s) = \min\left\{\max_t r_1(s_t^\pi),\ \max_t r_2(s_t^\pi)\right\}.$$

One may observe that these values are fundamentally different from the infinite-sum value commonly employed in RL [52], and do not accrue over the trajectory but, rather, are determined by certain points. Moreover, while each return considers two objectives, these objectives are combined in worst-case fashion to ensure *dual-satisfaction*. Although many of the works discussed in the previous section approach related tasks (e.g. goal reaching and hazard avoidance) via traditional sum-of-discounted-rewards formulations, these novel value functions have a more direct interpretation in the following sense: if r is positive (only) within $\mathcal{G}$ and q is positive (only) inside $\mathcal{H}$, $V^\pi_{\text{RAA}}(s)$ will be positive if and only if the RAA task will be accomplished by the policy π. Similarly if r_1 and r_2 are positive within $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively, $V^\pi_{\text{RR}}(s)$ will be positive if and only if the RR task will be accomplished by the policy π.

4 Reachability and Avoidability in RL

Prior works [1, 2] study the reach V^π_{R}, avoid V^π_{A}, and reach-avoid V^π_{RA} values, respectively defined by

$$V^\pi_{\text{R}}(s) = \max_t r(s^\pi_t), \quad V^\pi_{\text{A}}(s) = \min_t q(s^\pi_t), \quad V^\pi_{\text{RA}}(s) = \max_t \min\left\{ r(s^\pi_t), \max_{\tau \le t} q(s^\pi_\tau) \right\},$$

resulting in the derivation of special Bellman equations [1]. To put these value functions in context, assume the goal $\mathcal{G}$ is the set of states for which $r(s)$ is positive and the hazard $\mathcal{H}$ is the set of states for which $q(s)$ is non-positive. See Figure 2 for a simple grid-world demonstration comparing the RAA and RR values with the previously existing RA and R values. Then V^π_{R}, V^π_{A}, and V^π_{RA} are positive if and only if π causes the agent to eventually reach $\mathcal{G}$, to always avoid $\mathcal{H}$, and to reach $\mathcal{G}$ without hitting $\mathcal{H}$ prior to the reach time, respectively. The Reach-Avoid Bellman Equation (RABE), for example, takes the form [2]

$$V^*_{\text{RA}}(s) = \min\left\{ \max\left\{ \max_{a \in \mathcal{A}} V^*_{\text{RA}}\left(f(s,a)\right), r(s) \right\}, q(s) \right\},$$

and is associated with optimal policy $\pi^*_{\text{RA}}(s)$ (without the need for state augmentation, see Section A in the Supplementary Material). This formulation does not naturally induce a contraction, but may be discounted to induce contraction by defining $V^\gamma_{\text{RA}}(z)$ implicitly via

$$V^\gamma_{\text{RA}}(s) = (1-\gamma)\min\{r(s), q(s)\} + \gamma \min\left\{ \max\left\{ \max_{a \in \mathcal{A}} V^\gamma_{\text{RA}}\left(f(s,a)\right), r(s) \right\}, q(s) \right\},$$

for each $\gamma \in [0, 1)$, as in [2]. A fundamental result (Proposition 3 in [2]) is that

$$\lim_{\gamma \to 1} V^\gamma_{\text{RA}}(s) = V_{\text{RA}}(s).$$

These prior value functions and corresponding Bellman equations have proven powerful for these simple reach/avoid/reach-avoid problem formulations. In this work, we generalize the aforementioned results to the broader class involving V_{RAA} (assure no penalty after the reward threshold is achieved) and V_{RR} (achieve multiple rewards optimally). Through this generalization, we are able to train an agent to accomplish more complex tasks with noteworthy performance.

5 The need for augmenting states with historical information

We here discuss a small but important detail regarding the problem formulation. The value functions we introduce may appear similar to the simpler HJ-RL value functions discussed in the previous section; however, in these new formulations the goal of choosing a policy $\pi : \mathcal{S} \to \mathcal{A}$ is inherently flawed without state augmentation. In considering multiple objectives over an infinite horizon, situations arise in which the optimal action depends on more than the current state, but rather the **history** the trajectory. This complication is not unique to our problem formulation, but also occurs for NMDPs (see the Related Works section). To those unfamiliar with NMDPs, this at first may seem like a paradox as the MDP is by definition Markov, but the problem occurs not due to the state-transition dynamics but the nature of the reward. An example clarifying the issue is shown in Figure 3.

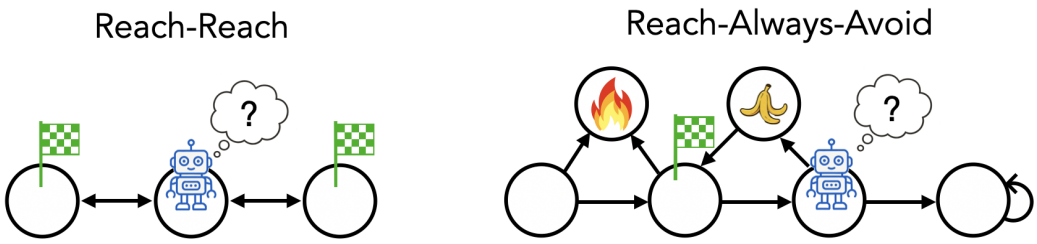

Figure 3: **Examples where a Non-Augmented Policy is Flawed** In both MDPs, consider an agent with no memory. (Left) For a deterministic policy based on the current state, the agent can only achieve one target (RR), as this policy must associate the middle state with either of the two possible actions. (Right) The RAA case is slightly more complex. Assume the robot will make sure to avoid the fire at all costs (which is easily done from the current state). It would also prefer to not encounter the banana peel hazard, but will do so if needed to achieve the target. From its current state the robot cannot determine whether to pursue the target by crossing the banana peel or move to the right. The correct decision depends on state history, specifically on whether the robot has already reached the target state or not (e.g. imagine the initial state is on the target state).

5.1 Augmentation of the RAA Problem

We consider an augmentation of the MDP defined by $\overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \mathcal{A}, f \rangle$ consisting of augmented states $\overline{\mathcal{S}} = \mathcal{S} \times \mathcal{Y} \times \mathcal{Z}$ and the same actions $\mathcal{A}$. For any initial state s, let the augmented states be initialized as $y = r(s)$ and $z = q(s)$, and let the transition of $\overline{\mathcal{M}}$ be defined by

$$s^{\bar{\pi}}_{t+1} = f\left(s^{\bar{\pi}}_t, \bar{\pi}\left(s^{\bar{\pi}}_t, y^{\bar{\pi}}_t, z^{\bar{\pi}}_t\right)\right); \quad y^{\bar{\pi}}_{t+1} = \max\left\{r\left(s^{\bar{\pi}}_{t+1}\right), y^{\bar{\pi}}_t\right\}; \quad z^{\bar{\pi}}_{t+1} = \min\left\{q\left(s^{\bar{\pi}}_{t+1}\right), z^{\bar{\pi}}_t\right\},$$

such that y_t and z_t track the best reward and worst penalty up to any point. Hence, the policy for $\overline{\mathcal{M}}$ given by $\bar{\pi} : \overline{\mathcal{S}} \to \mathcal{A}$ may now consider information regarding the history of the trajectory.

By definition, the RAA value for $\overline{\mathcal{M}}$,

$$V^{\bar{\pi}}_{\text{RAA}}(s) = \min\left\{\max_t r(s^{\bar{\pi}}_t), \min_t q(s^{\bar{\pi}}_t)\right\},$$

is equivalent to that of $\mathcal{M}$ except that it allows for a policy $\bar{\pi}$ which has access to historical information. We seek to find $\bar{\pi}$ that maximizes this value.

5.2 Augmentation of the RR Problem

For the Reach-Reach problem, we augment the system similarly, except that z_t is updated using a $\max$ operation instead of a $\min$:

$$s^{\bar{\pi}}_{t+1} = f\left(s^{\bar{\pi}}_t, \bar{\pi}\left(s^{\bar{\pi}}_t, y^{\bar{\pi}}_t, z^{\bar{\pi}}_t\right)\right); \quad y^{\bar{\pi}}_{t+1} = \max\left\{r_1\left(s^{\bar{\pi}}_{t+1}\right), y^{\bar{\pi}}_t\right\}; \quad z^{\bar{\pi}}_{t+1} = \max\left\{r_2\left(s^{\bar{\pi}}_{t+1}\right), z^{\bar{\pi}}_t\right\}.$$

Again, by definition,

$$V^{\bar{\pi}}_{\text{RR}}(s) = \min\left\{\max_t r_1(s^{\bar{\pi}}_t), \max_t r_2(s^{\bar{\pi}}_t)\right\}.$$

The RR problem is again to find an augmented policy $\bar{\pi}$ which maximizes this value.

6 Optimal Policies for RAA and RR by Value Decomposition

We now discuss our first theoretical contributions. We refer the reader to the supplementary material for the proofs of the theorems.

6.1 Decomposition of RAA into avoid and reach-avoid problems

Our main theoretical result for the RAA problem shows that we can solve this problem by first solving the avoid problem corresponding to the penalty $q(s)$ to obtain the optimal value function $V^*_{\text{A}}(s)$ and then solving a reach-avoid problem with the negated penalty function $q(s)$ and a modified reward function $r_{\text{RAA}}(s)$.

Theorem 1. *For all initial states $s \in \mathcal{S}$,*

$$\max_{\bar{\pi}} V^{\bar{\pi}}_{\text{RAA}}(s) = \max_{\pi} \max_{t} \min \left\{ r_{\text{RAA}}\left(s^{\pi}_{t}\right), \max_{\tau \leq t} q\left(s^{\pi}_{\tau}\right) \right\}, \tag{1}$$

where $r_{\text{RAA}}(s) := \min\left\{r(s), V^{*}_{\text{A}}(s)\right\}$, *with*

$$V^{*}_{\text{A}}(s) := \max_{\pi} \min_{t} q\left(s^{\pi}_{t}\right).$$

This decomposition is significant, as methods customized to solving avoid and reach-avoid problems were recently explored in [1, 2, 4, 50], allowing us to effectively solve the optimization problem defining $V^{*}_{\text{A}}(s)$ as well as the optimization problem that defines the right-hand-side of 1.

Corollary 1. *The value function* $V^{*}_{\text{RAA}}(s) := \max_{\bar{\pi}} V^{\bar{\pi}}_{\text{RAA}}(s)$ *satisfies the Bellman equation*

$$V^{*}_{\text{RAA}}(s) = \min \left\{ \max \left\{ \max_{a \in \mathcal{A}} V^{*}_{\text{RAA}}\left(f(s,a)\right), r_{\text{RAA}}(s) \right\}, q(s) \right\}.$$

6.2 Decomposition of the RR problem into three reach problems

Our main result for the RR problem shows that we can solve this problem by first solving two reach problems corresponding to the rewards $r_1(s)$ and $r_2(s)$ to obtain reach value functions $V^{*}_{\text{R1}}(s)$ and $V^{*}_{\text{R2}}(s)$, respectively. We then solve a third reach problem with a modified reward $r_{\text{RR}}(s)$.

Theorem 2. *For all initial states $s \in \mathcal{S}$,*

$$\max_{\bar{\pi}} V^{\bar{\pi}}_{\text{RR}}(s) = \max_{\pi} \max_{t} r_{\text{RR}}\left(s^{\pi}_{t}\right), \tag{2}$$

where $r_{\text{RR}}(s) := \min\left\{\max\left\{r_1(s), V^{*}_{\text{R2}}(s)\right\}, \max\left\{r_2(s), V^{*}_{\text{R1}}(s)\right\}\right\}$, *with*

$$V^{*}_{\text{R1}}(s) := \max_{\pi} \max_{t} r_1\left(s^{\pi}_{t}\right), \quad V^{*}_{\text{R2}}(s) := \max_{\pi} \max_{t} r_2\left(s^{\pi}_{t}\right).$$

Corollary 2. *The value function* $V^{*}_{\text{RR}}(s) := \max_{\bar{\pi}} V^{\bar{\pi}}_{\text{RR}}(s)$ *satisfies the Bellman equation*

$$V^{*}_{\text{RR}}(s) = \max \left\{ \max_{a \in \mathcal{A}} V^{*}_{\text{RR}}\left(f(s,a)\right), r_{\text{RR}}(s) \right\}.$$

6.3 Optimality of the augmented problems

We previously motivated the choice to consider an augmented MDP $\overline{\mathcal{M}}$ over the original MDP in the context of the RAA and RR problems. In this section, we justify our particular choice of augmentation. Indeed, the following theoretical result shows that further augmenting the states with additional historical information cannot improve performance under the optimal policy.

Theorem 3. *Let $s \in \mathcal{S}$. Then*

$$\max_{\pi} V^{\pi}_{\text{RAA}}(s) \leq \max_{\bar{\pi}} V^{\bar{\pi}}_{\text{RAA}}(s) = \max_{a_0, a_1, \ldots} \min \left\{ \max_{t} r(s_t), \min_{t} q(s_t) \right\},$$

and

$$\max_{\pi} V^{\pi}_{\text{RR}}(s) \leq \max_{\bar{\pi}} V^{\bar{\pi}}_{\text{RR}}(s) = \max_{a_0, a_1, \ldots} \min \left\{ \max_{t} r_1(s_t), \max_{t} r_2(s_t) \right\}$$

where $s_{t+1} = f(s_t, a_t)$ *and* $s_0 = s$.

The terms on the right of the lines above reflect the best possible sequence of actions to solve the RAA or RR problem, and the theorem states that the optimal augmented policy achieves that value, represented by the middle terms.

7 DO-HJ-PPO: Solving RAA and RR with RL

In the previous sections, we demonstrated that the RAA and RR problems can be solved through decomposition of the values into formulations amenable to existing RL methods. In this section, we propose relaxations to the RR and RAA theory and devise a custom variant of Proximal Policy Optimization, **DO-HJ-PPO**, to solve this broader class of problems, and demonstrate its performance.

7.1 Stochastic Reach-Avoid Bellman Equation

In this section we proceed by closely following [4], modifying the Stochastic Reachability Bellman Equation (SRBE) into a Stochastic Reach-Avoid Bellman Equation (SRABE) to allow for stochasticity. Using Theorems 1 and 2, the SRBE and SRABE offer the necessary tools for designing a PPO variant for solving the RR and RAA problems.

We define $\tilde{V}^{\pi}_{\text{RAA}}$ to be the solution to the following Bellman equation:

$$\tilde{V}^{\pi}_{\text{RAA}}(s) = \mathbb{E}_{a\sim\pi}\left[\min\left\{\max\left\{\tilde{V}^{\pi}_{\text{RAA}}\left(f(s,a)\right), r_{\text{RAA}}(s)\right\}, q(s)\right\}\right] \tag{SRABE}$$

The corresponding action-value function is

$$\tilde{Q}^{\pi}_{\text{RAA}}(s,a) = \min\left\{\max\left\{\tilde{V}^{\pi}_{\text{RAA}}\left(f(s,a)\right), r_{\text{RAA}}(s)\right\}, q(s)\right\}.$$

We define a modification of the dynamics f involving an absorbing state s_∞ as follows:

$$f'(s,a) = \begin{cases} f(s,a) & q\left(f(s,a)\right) < \tilde{V}^{\pi}_{\text{RAA}}(s) < r_{\text{RAA}}\left(f(s,a)\right), \\ s_\infty & \text{otherwise.} \end{cases}$$

We then have the following proposition:

Proposition 1. *For each $s \in \mathcal{S}$ and every $\theta \in \mathbb{R}^{n_p}$, we have*

$$\nabla_\theta \tilde{V}^{\pi_\theta}_{\text{RAA}}(s) \propto \mathbb{E}_{s'\sim d'_\pi(s), a\sim\pi_\theta}\left[\tilde{Q}^{\pi_\theta}_{\text{RAA}}(s',a)\nabla_\theta \ln \pi_\theta(a|s')\right],$$

where $d'_\pi(s)$ is the stationary distribution of the Markov Chain with transition function

$$P(s'|s) = \sum_{a\in\mathcal{A}} \pi(a|s)\left[f'(s,\pi(a|s)) = s'\right],$$

with the bracketed term equal to 1 if the proposition inside is true and 0 otherwise.

Following [2], we then define the discounted value and action-value functions with $\gamma \in [0,1)$ which may be found in the Supplemental.

7.2 Algorithm

Briefly, we propose co-learning the decomposed values to provide a unified algorithm and associated implementation, named **DO-HJ-PPO**. This includes "smarter" environment resets as well as bootstrapping for efficient computation. Crucially, the decomposed values are used in the definition of the special targets r_{RR} and r_{RAA} defined in Theorems 1 and 2, which then yield the RAA and RR values. See the Supplementary Material for details.

8 Experiments

8.1 DDQN Demonstration

We begin by demonstrating the utility of our theoretical results (Theorems 1 and 2) through a simple 2D grid-world experiment using Double Deep Q-Networks (DDQN) (Figure 2). In this environment, the agent can move left, right, or remain stationary, while drifting upward at a constant rate. Throughout, reward regions are shown in blue and penalty regions in red. In the RA scenario, trajectories successfully avoid the obstacle but may terminate in regions from which future collisions are inevitable, as there is no incentive to consider what happens after reaching the minimum reward threshold. In contrast, under the RAA formulation, where the objective involves maximizing cumulative reward while accounting for future penalties (as per Theorem 1), the agent learns to reach the target while remaining in safe regions thereafter. On the right, we consider a similar environment without obstacles but with two distinct targets. Here, the Reach-Reach (RR) formulation induces trajectories that visit both targets, unlike simple reach tasks in which the agent halts after reaching a single goal. These qualitative results highlight the behavioral distinctions induced by the RAA and RR objectives compared to their simpler counterparts.

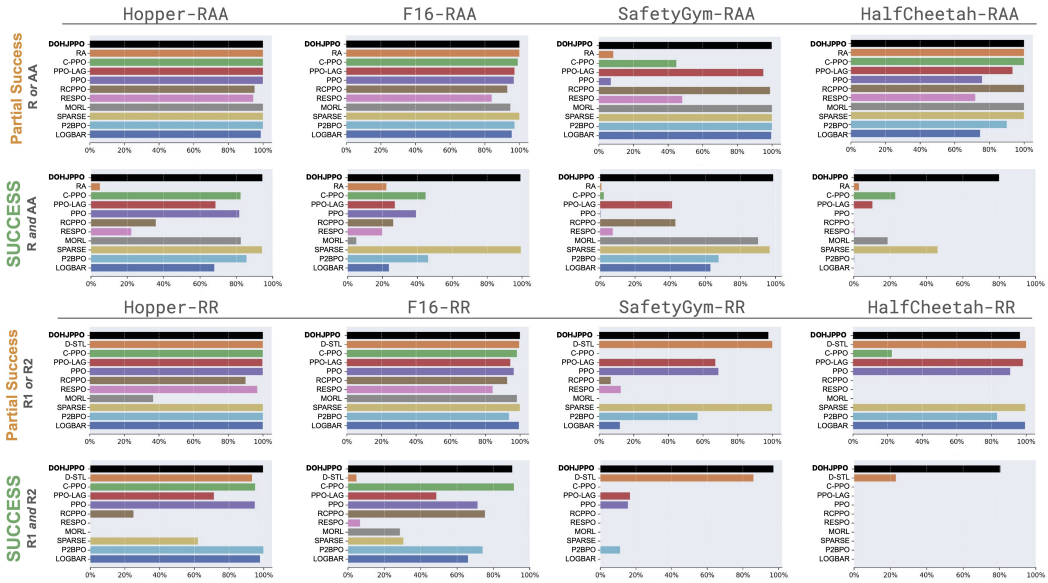

Figure 4: **Success (→) and Partial Success (→) in RAA and RR Tasks for DO-HJ-PPO and Baselines** We evaluate our method **DO-HJ-PPO in black** against relevant baselines over 1,000 trajectories (with random initial conditions) for both the Reach-Avoid-Avoid (RAA) and Reach-Reach (RR) problems in the Hopper, F16, SafetyGym and HalfCheetah environments. In the first and third row, the partial success percentage of each algorithm is given, defined by the number of trajectories to achieve either of the two objectives (reaching or always-avoiding in the RAA, reaching either in the RR). In the second and fourth rows, success percentage is given, defined by the number of trajectories to achieve both objectives.

8.2 Continuous Control Tasks with DO-HJ-PPO

To evaluate the method under more complex and less structured conditions, we extend our analysis to continuous control settings qith on-policy methods. Specifically, we consider RAA and RR tasks in the Hopper, F16, SafetyGym, and HalfCheetah environments, depicted in 1. In the RAA tasks, the penalty function generally characterizes regions of states where the agent, certain body parts, is intended to avoid, while in both RAA and RR tasks, the reward characterizes regions of states where the agent is intended to reach (in any order). We include several relevant baselines, detailed in the supplemental.

Empirically, we find that our method performs at the top-level, achieving first or second place among all tasks and environments (Figure 4). In fact, as the multi-target (RR) or safe-achievement (RAA) tasks become more complex (e.g. the SafetyGym or HalfCheetah tasks), our algorithm increasingly dominates the 10 state-of-the-art baselines with success percentages. Note, that almost all algorithms can achieve partial success at a high rate in each dual-objective task, highlighting the difficulty of mixed or competing objectives, particularly with discounted-sum rewards. Moreover, it is the sole performant algorithm in both dual-objective tasks, and displays the fastest achievement times in both RAA and RR tasks for complicated and simple tasks (see Supplemental).

9 Conclusion

In this work, we introduced two novel Bellman formulations for new problems (RAA and RR) which generalize those considered in several recent publications. We prove decomposition results for these problems that allow us to break them into simpler Bellman problems, which can then be composed to obtain the value functions and corresponding optimal policies. We use these results to design a PPO-based algorithm for practical solution of RAA and RR. More broadly, this work provides a road-map to extend the range of Bellman formulations that can be solved, via decomposing higher-level problems into lower-level ones.

References

[1] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8550–8556. IEEE, 2019.

[2] K.-C. Hsu, V. Rubies-Royo, C. J. Tomlin, and J. F. Fisac. Safety and liveness guarantees through reach-avoid reinforcement learning. In *Proceedings of Robotics: Science and Systems*, Held Virtually, July 2021. doi:10.15607/RSS.2021.XVII.077.

[3] M. Ganai, C. Hirayama, Y.-C. Chang, and S. Gao. Learning stabilization control from observations by learning lyapunov-like proxy models. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[4] O. So, C. Ge, and C. Fan. Solving minimum-cost reach avoid using reinforcement learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=jzngdJQ2lY.

[5] D. Yu, H. Ma, S. Li, and J. Chen. Reachability constrained reinforcement learning. In *International Conference on Machine Learning*, pages 25636–25655. PMLR, 2022.

[6] D. Yu, W. Zou, Y. Yang, H. Ma, S. E. Li, J. Duan, and J. Chen. Safe model-based reinforcement learning with an uncertainty-aware reachability certificate. *arXiv preprint arXiv:2210.07553*, 2022.

[7] E. Altman. *Constrained Markov decision processes: Stochastic modeling*. Routledge, Boca Raton, 13 Dec. 2021.

[8] J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. *ICML*, abs/1705.10528:22–31, 30 May 2017.

[9] A. Wachi and Y. Sui. Safe reinforcement learning in constrained Markov decision processes. *ICML*, 119:9797–9806, 12 July 2020.

[10] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, and A. Knoll. A review of safe reinforcement learning: Methods, theories, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12): 11216–11235, Dec. 2024.

[11] Y. Chow, A. Tamar, S. Mannor, and M. Pavone. Risk-sensitive and robust decision-making: A CVaR optimization approach. *Neural Inf Process Syst*, abs/1506.02188, 6 June 2015.

[12] A. Stooke, J. Achiam, and P. Abbeel. Responsive safety in reinforcement learning by PID lagrangian methods. *ICML*, 119:9133–9143, 8 July 2020.

[13] T. Li, Z. Guan, S. Zou, T. Xu, Y. Liang, and G. Lan. Faster algorithm and sharper analysis for constrained Markov decision process. *Oper. Res. Lett.*, 54(107107):107107, May 2024.

[14] Y. Chen, J. Dong, and Z. Wang. A primal-dual approach to constrained Markov decision processes. *arXiv [math.OC]*, 26 Jan. 2021.

[15] S. Miryoosefi and C. Jin. A simple reward-free approach to constrained reinforcement learning. *ICML*, abs/2107.05216:15666–15698, 12 July 2021.

[16] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge. Projection-based constrained policy optimization. *arXiv [cs.LG]*, 7 Oct. 2020.

[17] D. Ding, K. Zhang, T. Başar, and M. Jovanović. Natural policy gradient primal-dual method for constrained Markov decision processes. *Neural Inf Process Syst*, 33:8378–8390, 2020.

[18] C. Tessler, D. J. Mankowitz, and S. Mannor. Reward constrained policy optimization. *arXiv [cs.LG]*, 28 May 2018.

[19] A. Gattami, Q. Bai, and V. Aggarwal. Reinforcement learning for constrained Markov decision processes. *AISTATS*, 130:2656–2664, 2021.

[20] H. Satija, P. Amortila, and J. Pineau. Constrained Markov decision processes via backward value functions. *ICML*, 119:8502–8511, 12 July 2020.

[21] M. A. Wiering, M. Withagen, and M. M. Drugan. Model-based multi-objective reinforcement learning. In *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 1–6. IEEE, Dec. 2014.

[22] M. K. Van and A. Nowé. Multi-objective reinforcement learning using sets of Pareto dominating policies. *The Journal of Machine Learning Research*, 15(1):3483–3512, 2014.

[23] X.-Q. Cai, P. Zhang, L. Zhao, J. Bian, M. Sugiyama, and A. Llorens. Distributional Pareto-optimal multi-objective reinforcement learning. *Neural Inf Process Syst*, 36:15593–15613, 2023.

[24] H. Mossalam, Y. M. Assael, D. M. Roijers, and S. Whiteson. Multi-objective deep reinforcement learning. *arXiv [cs.AI]*, 9 Oct. 2016.

[25] A. Abels, D. Roijers, T. Lenaerts, A. Nowé, and D. Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 11–20. PMLR, 2019.

[26] R. Yang, X. Sun, and K. Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Advances in Neural Information Processing Systems*. proceedings.neurips.cc, 2019.

[27] S. Pateria, B. Subagdja, A.-H. Tan, and C. Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Comput. Surv.*, 54(5):1–35, 30 June 2022.

[28] A. G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dyn. Syst.: Theory Appl.*, 13(4):341–379, 2003.

[29] T. G. Dietterich. The MAXQ method for hierarchical reinforcement learning. *ICML*, pages 118–126, 24 July 1998.

[30] T. G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *J. Artif. Intell. Res.*, cs.LG/9905014, 21 May 1999.

[31] O. Nachum, S. Gu, H. Lee, and S. Levine. Near-optimal representation learning for hierarchical reinforcement learning. *arXiv [cs.AI]*, 2 Oct. 2018.

[32] C. Florensa, Y. Duan, and P. Abbeel. Stochastic neural networks for hierarchical reinforcement learning. *arXiv [cs.AI]*, 10 Apr. 2017.

[33] O. Nachum, S. Gu, H. Lee, and S. Levine. Data-efficient hierarchical reinforcement learning. *Neural Inf Process Syst*, 31:3307–3317, 21 May 2018.

[34] A. C. Li, C. Florensa, I. Clavera, and P. Abbeel. Sub-policy adaptation for hierarchical reinforcement learning. *arXiv [cs.LG]*, 13 June 2019.

[35] A. H. Qureshi, J. J. Johnson, Y. Qin, T. Henderson, B. Boots, and M. C. Yip. Composing task-agnostic policies with deep reinforcement learning. *arXiv [cs.LG]*, 25 May 2019.

[36] F. Bacchus, C. Boutilier, and A. J. Grove. Rewarding behaviors. In *Proceedings of the National Conference on Artificial Intelligence.*, pages 1160–1167. cs.toronto.edu, 4 Aug. 1996.

[37] F. Bacchus, C. Boutilier, and A. Grove. Structured solution methods for non-Markovian decision processes. In *AAAI/IAAI*, pages 112–117, 1997.

[38] S. Thiebaux, C. Gretton, J. Slaney, D. Price, and F. Kabanza. Decision-theoretic planning with non-Markovian rewards. *J. Artif. Intell. Res.*, 25:17–74, 29 Jan. 2006.

[39] A. Camacho, O. Chen, S. Sanner, and S. McIlraith. Non-Markovian rewards expressed in LTL: Guiding search via reward shaping. *Proceedings of the International Symposium on Combinatorial Search*, 8(1):159–160, 1 Sept. 2021.

[40] R. T. Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith. Using reward machines for high-level task specification and decomposition in reinforcement learning. *ICML*, 80:2112–2121, 3 July 2018.

[41] A. Camacho, R. Toro Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith. LTL and beyond: Formal languages for reward function specification in reinforcement learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 6065–6073, California, 1 Aug. 2019. International Joint Conferences on Artificial Intelligence Organization.

[42] T. Brázdil, K. Chatterjee, M. Chmelík, V. Forejt, J. Křetínský, M. Kwiatkowska, D. Parker, and M. Ujma. Verification of Markov decision processes using learning algorithms. *arXiv [cs.LO]*, 10 Feb. 2014.

[43] M. H. Cohen, Z. Serlin, K. Leahy, and C. Belta. Temporal logic guided safe model-based reinforcement learning: A hybrid systems approach. *Nonlinear Anal. Hybrid Syst.*, 47(101295): 101295, Feb. 2023.

[44] W. Qiu, W. Mao, and H. Zhu. Instructing goal-conditioned reinforcement learning agents with temporal logic objectives. *Neural Inf Process Syst*, 36:39147–39175, 2023.

[45] D. Sadigh, E. S. Kim, S. Coogan, S. S. Sastry, and S. A. Seshia. A learning based approach to control synthesis of Markov decision processes for linear temporal logic specifications. In *53rd IEEE Conference on Decision and Control*, pages 1091–1096. IEEE, Dec. 2014.

[46] N. Hamilton, P. K. Robinette, and T. T. Johnson. Training agents to satisfy timed and untimed signal temporal logic specifications with reinforcement learning. In *Software Engineering and Formal Methods*, Lecture notes in computer science, pages 190–206. Springer International Publishing, Cham, 2022.

[47] X. Li, C.-I. Vasile, and C. Belta. Reinforcement learning with temporal logic rewards. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3834–3839. IEEE, Sept. 2017.

[48] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control*, 50 (7):947–957, 2005.

[49] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry. Reach-avoid problems with time-varying dynamics, targets and constraints. In *Hybrid Systems: Computation and Control*. ACM, 2015.

[50] O. So and C. Fan. Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning. *arXiv [cs.RO]*, 23 May 2023.

[51] K. Zhu, F. Lan, W. Zhao, and T. Zhang. Safe multi-agent reinforcement learning via approximate hamilton-jacobi reachability. *J. Intell. Robot. Syst.*, 111(1), 30 Dec. 2024.

[52] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

[53] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

[54] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[55] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016. URL https://arxiv.org/abs/1606.01540.

Supplementary Material

Contents

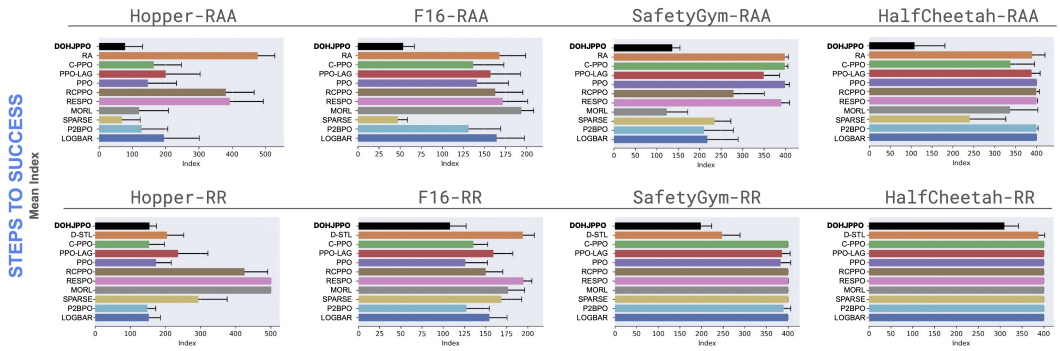

Figure 5: **Steps to Success (←) in RAA and RR Tasks for DO-HJ-PPO and Baselines** For the same 1000 trajectories in Figure 4, we quantify here the number of steps until achievement of both tasks: reaching without crash afterward in the RAA, reaching both goal in the RR. DO-HJ-PPO is not only competitive but consistently achieves the dual-objective problems in the fewest number of steps.

Mean Steps to Success for DO-HJ-PPO

Here, we show the mean steps to success for each of the RAA and RR tasks included in the work. **DO-HJ-PPO** proves to be among the top three fastest always and frequently appears as the first to achieve dual-objective success on average. This underscores the ability of the algorithm to pick the target (and policy) which will allow it to safely accomplish the entire task.

Proof Notation

Throughout the theoretical sections of this supplement, we use the following notation.

We let $\mathbb{N} = \{0, 1, \dots\}$ be the set of whole numbers.

We let $\mathbb{A}$ be the set of maps from $\mathbb{N}$ to $\mathcal{A}$. In other words, $\mathbb{A}$ is the set of sequences of actions the agent can choose. Given $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{A}$, and $\tau \in \mathbb{N}$, we let $[\mathbf{a}_1, \mathbf{a}_2]_\tau$ be the element of $\mathbb{A}$ for which

$$[\mathbf{a}_1, \mathbf{a}_2]_\tau(t) = \begin{cases} \mathbf{a}_1(t) & t < \tau, \\ \mathbf{a}_2(t-\tau) & t \geq \tau. \end{cases}$$

Similarly, given $a \in \mathcal{A}$ and $\mathbf{a} \in \mathbb{A}$, we let $[a, \mathbf{a}]$ be the element of $\mathbb{A}$ for which

$$[a, \mathbf{a}](t) = \begin{cases} a & t = 0, \\ \mathbf{a}(t-1) & t \geq 1. \end{cases}$$

Additionally, given $\mathbf{a} \in \mathbb{A}$ and $\tau \in \mathbb{N}$, we let $\mathbf{a}|_\tau$ be the element of $\mathbb{A}$ for which

$$\mathbf{a}|_\tau(t) = \mathbf{a}(t+\tau) \quad \forall t \in \mathbb{N}.$$

The $[\cdot, \cdot]_\tau$ operation corresponds to concatenating two action sequences (using only the 0^{th} to $(\tau-1)^{\text{st}}$ elements of the first sequence), the $[\cdot, \cdot]$ operation corresponds to prepending an action to an action sequence, and the $\cdot|_\tau$ operation corresponds to removing the 0^{th} to $(\tau-1)^{\text{st}}$ elements of an action sequence.

We let Π be the set of policies $\pi : \mathcal{S} \to \mathcal{A}$. Given $s \in \mathcal{S}$ and $\pi \in \Pi$, we let $\xi_s^\pi : \mathbb{N} \to \mathcal{S}$ be the solution of the evolution equation

$$\xi_s^\pi(t+1) = f\left(\xi_s^\pi(t), \pi\left(\xi_s^\pi(t)\right)\right)$$

for which $\xi_s^\pi(0) = s$. In other words, $\xi_s^\pi(\cdot)$ is the state trajectory over time when the agent begins at state s and follows policy π.

We will also "overload" this trajectory notation for signals rather than policies: given $\mathbf{a} \in \mathbb{A}$, we let $\xi_s^{\mathbf{a}} : \mathbb{N} \to \mathcal{S}$ be the solution of the evolution equation

$$\xi_s^{\mathbf{a}}(t+1) = f\left(\xi_s^{\mathbf{a}}(t), \mathbf{a}(t)\right)$$

for which $\xi_s^{\mathbf{a}}(0) = s$. In other words, $\xi_s^{\mathbf{a}}(\cdot)$ is the state trajectory over time when the agent begins at state s and follows action sequence $\mathbf{a}$.

A Proof of RAA Main Theorem

We first define the value functions, $V^*_{\mathrm{A}}, \tilde{V}^*_{\mathrm{RA}}, V^*_{\mathrm{RAA}} : \mathcal{S} \to \mathbb{R}$ by

$$\begin{aligned}
V^*_{\mathrm{A}}(s) &= \max_{\pi \in \Pi} \min_{\tau \in \mathbb{N}} q\left(\xi^\pi_s(\tau)\right), \\
\tilde{V}^*_{\mathrm{RA}}(s) &= \max_{\pi \in \Pi} \max_{\tau \in \mathbb{N}} \min \left\{ r_{\mathrm{RAA}}\left(\xi^\pi_s(\tau)\right), \min_{\kappa \leq \tau} q\left(\xi^\pi_s(\kappa)\right) \right\}, \\
V^*_{\mathrm{RAA}}(s) &= \max_{\pi \in \Pi} \min \left\{ \max_{\tau \in \mathbb{N}} r\left(\xi^\pi_s(\tau)\right), \min_{\kappa \in \mathbb{N}} q\left(\xi^\pi_s(\kappa)\right) \right\},
\end{aligned}$$

where r_{RAA} is as in Theorem 1.

We next define the value functions, $v^*_{\mathrm{A}}, \tilde{v}^*_{\mathrm{RA}}, v^*_{\mathrm{RAA}} : \mathcal{S} \to \mathbb{R}$, which maximize over action sequences rather than policies:

$$\begin{aligned}
v^*_{\mathrm{A}}(s) &= \max_{\mathbf{a} \in \mathbb{A}} \min_{\tau \in \mathbb{N}} q\left(\xi^{\mathbf{a}}_s(\tau)\right), \\
\tilde{v}^*_{\mathrm{RA}}(s) &= \max_{\mathbf{a} \in \mathbb{A}} \max_{\tau \in \mathbb{N}} \min \left\{ r_{\mathrm{RAA}}\left(\xi^{\mathbf{a}}_s(\tau)\right), \min_{\kappa \leq \tau} q\left(\xi^{\mathbf{a}}_s(\kappa)\right) \right\}, \\
v^*_{\mathrm{RAA}}(s) &= \max_{\mathbf{a} \in \mathbb{A}} \min \left\{ \max_{\tau \in \mathbb{N}} r\left(\xi^{\mathbf{a}}_s(\tau)\right), \min_{\kappa \in \mathbb{N}} q\left(\xi^{\mathbf{a}}_s(\kappa)\right) \right\},
\end{aligned}$$

Observe that for each $s \in \mathcal{S}$,

$$v^*_{\mathrm{A}}(s) \geq V^*_{\mathrm{A}}(s), \quad \tilde{v}^*_{\mathrm{RA}}(s) \geq \tilde{V}^*_{\mathrm{RA}}(s), \quad v^*_{\mathrm{RAA}}(s) \geq V^*_{\mathrm{RAA}}(s).$$

We now prove a series of lemmas that will be useful in the proof of the main theorem.

Lemma 1. *There is a $\pi \in \Pi$ such that*

$$v^*_{\mathrm{A}}(s) = \min_{\tau \in \mathbb{N}} q\left(\xi^\pi_s(\tau)\right)$$

for all $s \in \mathcal{S}$.

Proof. Choose $\pi \in \Pi$ such that

$$\pi(s) \in \arg\max_{a \in \mathcal{A}} v^*_{\mathrm{A}}\left(f(s,a)\right) \quad \forall s \in \mathcal{S}.$$

Fix $s \in \mathcal{S}$. Note that for each $\tau \in \mathbb{N}$,

$$\begin{aligned}
v^*_{\mathrm{A}}\left(\xi^\pi_s(\tau+1)\right) &= v^*_{\mathrm{A}}\left(f\left(\xi^\pi_s(\tau), \pi\left(\xi^\pi_s(\tau)\right)\right)\right) \\
&= \max_{a \in \mathcal{A}} v^*_{\mathrm{A}}\left(f\left(\xi^\pi_s(\tau), a\right)\right) \\
&= \max_{a \in \mathcal{A}} \max_{\mathbf{a} \in \mathbb{A}} \min_{\kappa \in \mathbb{N}} q\left(\xi^{\mathbf{a}}_{f(\xi^\pi_s(\tau), a)}(\kappa)\right) \\
&= \max_{a \in \mathcal{A}} \max_{\mathbf{a} \in \mathbb{A}} \min_{\kappa \in \mathbb{N}} q\left(\xi^{[a,\mathbf{a}]}_{\xi^\pi_s(\tau)}(\kappa+1)\right) \\
&= \max_{\mathbf{a} \in \mathbb{A}} \min_{\kappa \in \mathbb{N}} q\left(\xi^{\mathbf{a}}_{\xi^\pi_s(\tau)}(\kappa+1)\right) \\
&\geq \max_{\mathbf{a} \in \mathbb{A}} \min_{\kappa \in \mathbb{N}} q\left(\xi^{\mathbf{a}}_{\xi^\pi_s(\tau)}(\kappa)\right) \\
&\geq v^*_{\mathrm{A}}\left(\xi^\pi_s(\tau)\right).
\end{aligned}$$

It follows by induction that $v^*_{\mathrm{A}}\left(\xi^\pi_s(\tau)\right) \geq v^*_{\mathrm{A}}\left(\xi^\pi_s(0)\right)$ for all $\tau \in \mathbb{N}$, so that

$$v^*_{\mathrm{A}}(s) \geq \min_{\tau \in \mathbb{N}} q\left(\xi^\pi_s(\tau)\right) \geq \min_{\tau \in \mathbb{N}} v^*_{\mathrm{A}}\left(\xi^\pi_s(\tau)\right) = v^*_{\mathrm{A}}\left(\xi^\pi_s(0)\right) = v^*_{\mathrm{A}}(s).$$

□

Corollary 3. *For all $s \in \mathcal{S}$, we have $V^*_{\mathrm{A}}(s) = v^*_{\mathrm{A}}(s)$.*

Lemma 2. *There is a $\pi \in \Pi$ such that*

$$\tilde{v}^*_{\mathrm{RA}}(s) = \max_{\tau\in\mathbb{N}} \min\left\{ r_{\mathrm{RAA}}\left(\xi^\pi_s(\tau)\right), \min_{\kappa\le\tau} q\left(\xi^\pi_s(\kappa)\right)\right\}$$

for all $s \in \mathcal{S}$.

Proof. First, let us note that in this proof we will use the standard conventions that

$$\max \varnothing = -\infty \quad \text{and} \quad \min \varnothing = +\infty.$$

We next introduce some notation. First, for convenience, we set $v^* = \tilde{v}^*_{\mathrm{RA}}$ and $V^* = \tilde{V}^*_{\mathrm{RA}}$. Given $s \in \mathcal{S}$ and $\mathbf{a} \in \mathbb{A}$, we write

$$v^{\mathbf{a}}(s) = \max_{\tau\in\mathbb{N}} \min\left\{ r_{\mathrm{RAA}}\left(\xi^{\mathbf{a}}_s(\tau)\right), \min_{\kappa\le\tau} q\left(\xi^{\mathbf{a}}_s(\kappa)\right)\right\}.$$

Similarly, given $s \in \mathcal{S}$ and $\pi \in \Pi$, we write

$$V^\pi(s) = \max_{\tau\in\mathbb{N}} \min\left\{ r_{\mathrm{RAA}}\left(\xi^\pi_s(\tau)\right), \min_{\kappa\le\tau} q\left(\xi^\pi_s(\kappa)\right)\right\}.$$

Then

$$V^*(s) = \max_{\pi\in\Pi}\max_{\tau\in\mathbb{N}} \min\left\{ r_{\mathrm{RAA}}\left(\xi^\pi_s(\tau)\right), \min_{\kappa\le\tau} q\left(\xi^\pi_s(\kappa)\right)\right\} = \max_{\pi\in\Pi} V^\pi(s),$$

and

$$v^*(s) = \max_{\mathbf{a}\in\mathbb{A}}\max_{\tau\in\mathbb{N}} \min\left\{ r_{\mathrm{RAA}}\left(\xi^{\mathbf{a}}_s(\tau)\right), \min_{\kappa\le\tau} q\left(\xi^{\mathbf{a}}_s(\kappa)\right)\right\} = \max_{\mathbf{a}\in\mathbb{A}} v^{\mathbf{a}}(s).$$

It is immediate that $v^*(s) \ge V^*(s)$ for each $s \in \mathcal{S}$, so it suffices to show the reverse inequality. Toward this end, it suffices to show that there is a $\pi \in \Pi$ for which $V^\pi(s) = v^*(s)$ for each $s \in \mathcal{S}$. Indeed, in this case, $V^*(s) \ge V^\pi(s) = v^*(s)$.

We now construct the desired policy π. Let $\alpha_0 = +\infty$, $S_0 = \varnothing$, and $v^*_0 : \mathcal{S} \to \mathbb{R}\cup\{-\infty\}, s \mapsto -\infty$. We recursively define $\alpha_t \in \mathbb{R}$, $S_t \subseteq \mathcal{S}$, and $v^*_t : \mathcal{S} \to \mathbb{R}\cup\{-\infty\}$ for $t = 1, 2, \dots$ by

$$\alpha_{t+1} = \max_{s\in\mathcal{S}\setminus S_t} \min\left\{ \max\left\{ r_{\mathrm{RAA}}(s), \max_{a\in\mathcal{A}} v^*_t\left(f(s,a)\right)\right\}, q(s)\right\}, \tag{3}$$

$$S_{t+1} = S_t \cup \left\{ s \in \mathcal{S}\setminus S_t \;\middle|\; \min\left\{ \max\left\{ r_{\mathrm{RAA}}(s), \max_{a\in\mathcal{A}} v^*_t\left(f(s,a)\right)\right\}, q(s)\right\} = \alpha_{t+1}\right\}, \tag{4}$$

$$v^*_{t+1}(s) = \begin{cases} v^*_t(s) & s \in S_t, \\ \alpha_{t+1} & s \in S_{t+1}\setminus S_t, \\ -\infty & s \in \mathcal{S}\setminus S_{t+1}. \end{cases} \tag{5}$$

From (4) it follows that

$$S_0 \subseteq S_1 \subseteq S_2 \subseteq \dots, \tag{6}$$

which together with (3) shows that

$$\alpha_0 \ge \alpha_1 \ge \alpha_2 \ge \dots. \tag{7}$$

Also, whenever $\mathcal{S}\setminus S_t$ is non-empty, the set being appended to S_t in (4) is non-empty so

$$\bigcup_{t=0}^{\infty} S_t = \mathcal{S}. \tag{8}$$

For each $s \in \mathcal{S}$, let $\sigma(s)$ be the smallest $t \in \mathbb{N}$ for which $s \in S_t$. We choose the policy $\pi \in \Pi$ of interest by insisting

$$\pi(s) \in \arg\max_{a\in\mathcal{A}} v^*_{\sigma(s)-1}(f(s,a)) \quad \forall s \in \mathcal{S}. \tag{9}$$

In the remainder of the proof, we show that $V^\pi(s) = v^*(s)$ for each $s \in \mathcal{S}$ by induction. Let $n \in \mathbb{N}$ and suppose the following induction assumptions hold:

$$V^\pi(s) = v^*(s) = v_n^*(s) \geq \alpha_n \quad \forall s \in S_n, \tag{10}$$

$$v^*(s') \leq \alpha_n \quad \forall s' \in \mathcal{S} \setminus S_n. \tag{11}$$

Note that the above hold trivially when $n = 0$ since $S_0 = \varnothing$ and $\alpha_0 = +\infty$. Fix some particular $y \in S_{n+1}$ and some $z \in \mathcal{S} \setminus S_{n+1}$. We must show that

$$V^\pi(y) = v^*(y) = v_{n+1}^*(y) \geq \alpha_{n+1}, \tag{12}$$

$$v^*(z) \leq \alpha_{n+1}. \tag{13}$$

In this case, induction then shows that $V^\pi(s) = v^*(s)$ for all $s \in \cup_{n=0}^{\infty} S_t$. Since this union is equal to $\mathcal{S}$ by (8), the desired result then follows.

To show (12)-(13), we first demonstrate the following three claims.

1. Let $x \in \mathcal{S}$ and $w \in \mathcal{A}$ be such that $f(x, w) \in S_n$ and $q(x) \geq \alpha_{n+1}$. We claim $x \in S_{n+1}$.

   We can assume $x \notin S_n$, for otherwise the claim follows immediately from (6). Since $f(x, w) \in S_n$, we have $v_n^*(f(x, w)) \geq \alpha_n$ by (10). Thus

   $$\begin{aligned}\alpha_{n+1} &\geq \min\left\{\max\left\{r_{\text{RAA}}(x), \max_{a \in \mathcal{A}} v_n^*(f(x, a))\right\}, q(x)\right\} \\ &\geq \min\left\{\max\{r_{\text{RAA}}(x), \alpha_n\}, \alpha_{n+1}\right\} \\ &= \alpha_{n+1},\end{aligned}$$

   where the first inequality follows from (3), and the equality follows from (7). Thus

   $$\alpha_{n+1} = \min\left\{\max\left\{r_{\text{RAA}}(x), \max_{a \in \mathcal{A}} v_n^*(f(x, a))\right\}, q(x)\right\},$$

   so the claim follows from (4).

2. Let $x \in S_{n+1} \setminus S_n$ and $w \in \mathcal{A}$ be such that $f(x, w) \in S_n$. We claim that

   $$V^\pi(x) = v^*(x) = \alpha_{n+1}. \tag{14}$$

   To show this claim, we will make use of the dynamic programming principle

   $$v^{\mathbf{a}}(s) = \min\left\{\max\left\{r_{\text{RAA}}(s), v^{\mathbf{a}|_1}(f(s, \mathbf{a}(0)))\right\}, q(s)\right\}, \quad \forall s \in \mathcal{S}, \mathbf{a} \in \mathbb{A},$$

   from which it follows that

   $$V^\pi(s) = \min\left\{\max\left\{r_{\text{RAA}}(s), V^\pi(f(s, \pi(s)))\right\}, q(s)\right\}, \quad \forall s \in \mathcal{S}, \tag{15}$$

   and

   $$v^*(s) = \min\left\{\max\left\{r_{\text{RAA}}(s), \max_{a \in \mathcal{A}} v^*(f(s, a))\right\}, q(s)\right\}, \quad \forall s \in \mathcal{S}. \tag{16}$$

   Since $x \in S_{n+1} \setminus S_n$, then $\sigma(x) = n + 1$ by definition of σ, so $\pi(x) \in \arg\max_{a \in \mathcal{A}} v_n^*(f(x, a))$ by (9). Thus

   $$v_n^*(f(x, \pi(x))) = \max_{a \in \mathcal{A}} v_n^*(f(x, a)). \tag{17}$$

   But then

   $$v_n^*(f(x, \pi(x))) \geq v_n^*(f(x, w)) \geq \alpha_n \geq \alpha_{n+1} > -\infty,$$

   where the second inequality comes from (10), the third comes from (7), and the final inequality comes from (3) ($\mathcal{S} \setminus S_n$ is non-empty because $x \in \mathcal{S} \setminus S_n$). Thus $f(x, \pi(x)) \in S_n$ by (5). It then follows from (10) that

   $$V^\pi(f(x, \pi(x))) = v^*(f(x, \pi(x))) = v_n^*(f(x, \pi(x))). \tag{18}$$

Now, observe that for all $s \in S_n$ and $s' \in \mathcal{S} \setminus S_n$,

$$v^*(s) = v_n^*(s) \geq \alpha_n \geq v^*(s') \geq -\infty = v_n^*(s'), \tag{19}$$

where the first equality and inequality are from (10), the second inequality is from (11), and the final equality is from (5). Moreover, $f(x,a) \in S_n$ for at least one a (in particular $a = w$). Letting $\mathcal{A}' = \{a \in \mathcal{A} \mid f(x,a) \in S_n\}$, it follows from (19) that

$$\max_{a \in \mathcal{A}} v^*\left(f(x,a)\right) = \max_{a \in \mathcal{A}'} v^*\left(f(x,a)\right) = \max_{a \in \mathcal{A}'} v_n^*\left(f(x,a)\right) = \max_{a \in \mathcal{A}} v_n^*\left(f(x,a)\right). \tag{20}$$

From (17)-(20) we have

$$V^\pi\left(f(x,\pi(x))\right) = \max_{a \in \mathcal{A}} v^*\left(f(x,a)\right) = \max_{a \in \mathcal{A}} v_n^*\left(f(x,a)\right). \tag{21}$$

Now observe that

$$\begin{aligned}
V^\pi(x) &= \min\left\{\max\left\{r_{\text{RAA}}(x), V^\pi\left(f(x,\pi(x))\right)\right\}, q(x)\right\}, \\
v^*(x) &= \min\left\{\max\left\{r_{\text{RAA}}(x), \max_{a \in \mathcal{A}} v^*\left(f(x,a)\right)\right\}, q(x)\right\}, \\
\alpha_{n+1} &= \min\left\{\max\left\{r_{\text{RAA}}(x), \max_{a \in \mathcal{A}} v_n^*\left(f(x,a)\right)\right\}, q(x)\right\},
\end{aligned}$$

where the first equation is from (15), the second is from (16), and the third is from (4). But then (14) follows from the above equations together with (21).

3. Let $x \in \mathcal{S} \setminus S_n$. We claim that $v^*(x) \leq \alpha_{n+1}$. Suppose otherwise. Then we can choose $\mathbf{a} \in \mathbb{A}$ and $\tau \in \mathbb{N}$ such that

$$\min\left\{r_{\text{RAA}}\left(\xi_x^{\mathbf{a}}(\tau)\right), \min_{\kappa \leq \tau} q\left(\xi_x^{\mathbf{a}}(\kappa)\right)\right\} > \alpha_{n+1}. \tag{22}$$

It follows that $\xi_x^{\mathbf{a}}(\tau) \in S_n$, for otherwise

$$\alpha_{n+1} \geq \min\left\{r_{\text{RAA}}(\xi_x^{\mathbf{a}}(\tau)), q(\xi_x^{\mathbf{a}}(\tau))\right\}$$

by (3), creating a contradiction.

So $x \notin S_n$ and $\xi_x^{\mathbf{a}}(\tau) \in S_n$, indicating that there is some $\theta \in \{0, \ldots, \tau - 1\}$ such that $\xi_x^{\mathbf{a}}(\theta) \notin S_n$ and $f\left(\xi_x^{\mathbf{a}}(\theta), \mathbf{a}(\theta)\right) = \xi_x^{\mathbf{a}}(\theta+1) \in S_n$. Moreover, $q\left(\xi_x^{\mathbf{a}}(\theta)\right) > \alpha_{n+1}$ by (22). It follows from claim 1 that $\xi_x^{\mathbf{a}}(\theta) \in S_{n+1}$.

But then it follows from claim 2 that $v^*\left(\xi_x^{\mathbf{a}}(\theta)\right) = \alpha_{n+1}$. However,

$$\begin{aligned}
v^*\left(\xi_x^{\mathbf{a}}(\theta)\right) &\geq \min\left\{r_{\text{RAA}}\left(\xi_{\xi_x^{\mathbf{a}}(\theta)}^{\mathbf{a}}(\tau-\theta)\right), \min_{\kappa \leq \tau - \theta} q\left(\xi_{\xi_x^{\mathbf{a}}(\theta)}^{\mathbf{a}}(\kappa)\right)\right\} \\
&= \min\left\{r_{\text{RAA}}\left(\xi_x^{\mathbf{a}}(\tau-\theta+\theta)\right), \min_{\kappa \leq \tau - \theta} q\left(\xi_x^{\mathbf{a}}(\kappa+\theta)\right)\right\} \\
&= \min\left\{r_{\text{RAA}}\left(\xi_x^{\mathbf{a}}(\tau)\right), \min_{\kappa \in \{\theta, \theta+1, \ldots, \tau\}} q\left(\xi_x^{\mathbf{a}}(\kappa)\right)\right\} \\
&> \alpha_{n+1},
\end{aligned}$$

giving the desired contradiction.

Having established these claims, we return to proving (12) and (13) hold. In fact, (13) follows immediately from claim 3, so we actually only need to show (12).

If $y \in S_n$, then from (5) and (10), we have that $V^\pi(y) = v^*(y) = v_n^*(y) = v_{n+1}^*(y)$, and from (7) and (10), we also have that $v_n^*(y) \geq \alpha_n \geq \alpha_{n+1}$. Together these establish (12) when $y \in S_n$.

So suppose $y \in S_{n+1} \setminus S_n$. First, observe that $v_{n+1}^*(y) = \alpha_{n+1}$ by (5). There are now two possibilities. If there is some $a \in \mathcal{A}$ for which $f(y,a) \in S_n$, then (12) follows from claim 2. If

instead, $f(y,a) \notin S_n$ for each $a \in \mathcal{A}$, then $\max_{a\in\mathcal{A}} v_n^*(f(y,a)) = -\infty$ by (5) (or if $n = 0$ by definition of v_0^*). Thus $\alpha_{n+1} = \min\{r_{\text{RAA}}(y), q(y)\}$ by (4), so

$$v^*(y) \geq V^\pi(y) \geq \min\{r_{\text{RAA}}(y), q(y)\} = \alpha_{n+1} \geq v^*(y),$$

where the final inequality follows from claim 3. This completes the proof. □

Corollary 4. *For all $s \in \mathcal{S}$, we have $\tilde{V}^*_{\text{RA}}(s) = \tilde{v}^*_{\text{RA}}(s)$.*

Lemma 3. *Let $F : \mathbb{A} \times \mathbb{N} \to \mathbb{R}$. Then*

$$\sup_{\mathbf{a}\in\mathbb{A}} \sup_{\tau\in\mathbb{N}} \sup_{\mathbf{a}'\in\mathbb{A}'} F\left([\mathbf{a},\mathbf{a}']_\tau, \tau\right) = \sup_{\mathbf{a}\in\mathbb{A}} \sup_{\tau\in\mathbb{N}} F\left(\mathbf{a},\tau\right). \tag{23}$$

Proof. We proceed by showing both inequalities corresponding to (23) hold.

($\geq$) Given any $\mathbf{a} \in \mathbb{A}$ and $\tau \in \mathbb{N}$, we have $\sup_{\mathbf{a}'\in\mathbb{A}'} F\left([\mathbf{a},\mathbf{a}']_\tau,\tau\right) \geq F\left(\mathbf{a},\tau\right)$. Taking the suprema over $\mathbf{a} \in \mathbb{A}$ and $\tau \in \mathbb{N}$ on both sides of this inequality gives the desired result.

($\leq$) Given any $\mathbf{a} \in \mathbb{A}$ and $\tau \in \mathbb{N}$, we have

$$\sup_{\mathbf{a}'\in\mathbb{A}'} F\left([\mathbf{a},\mathbf{a}']_\tau,\tau\right) \leq \sup_{\mathbf{a}''\in\mathbb{A}} F\left(\mathbf{a}'',\tau\right),$$

so that the result follows from taking the suprema over $\mathbf{a} \in \mathbb{A}$ and $\tau \in \mathbb{N}$ on both sides of this inequality.

□

Lemma 4. *For each $s \in \mathcal{S}$,*

$$v^*_{\text{RAA}}(s) = \tilde{v}^*_{\text{RA}}(s).$$

Proof. For each $s \in \mathcal{S}$, we have

$$\begin{aligned}
\tilde{v}^*_{\text{RA}}(s) &= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \min\left\{ r_{\text{RAA}}\left(\xi_s^{\mathbf{a}}(\tau)\right), \min_{\kappa\leq\tau} q\left(\xi_s^{\mathbf{a}}(\kappa)\right)\right\} && (24)\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \min\left\{ r\left(\xi_s^{\mathbf{a}}(\tau)\right), v^*_{\text{A}}\left(\xi_s^{\mathbf{a}}(\tau)\right), \min_{\kappa\leq\tau} q\left(\xi_s^{\mathbf{a}}(\kappa)\right)\right\} && (25)\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \min\left\{ r\left(\xi_s^{\mathbf{a}}(\tau)\right), \max_{\mathbf{a}'\in\mathbb{A}} \min_{\kappa'\in\mathbb{N}} q\left(\xi^{\mathbf{a}'}_{\xi_s^{\mathbf{a}}(\tau)}(\kappa')\right), \min_{\kappa\leq\tau} q\left(\xi_s^{\mathbf{a}}(\kappa)\right)\right\}\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \min\left\{ r\left(\xi_s^{\mathbf{a}}(\tau)\right), \max_{\mathbf{a}'\in\mathbb{A}} \min_{\kappa'\in\mathbb{N}} q\left(\xi_s^{[\mathbf{a},\mathbf{a}']_\tau}(\tau+\kappa')\right), \min_{\kappa\leq\tau} q\left(\xi_s^{\mathbf{a}}(\kappa)\right)\right\}\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \max_{\mathbf{a}'\in\mathbb{A}} \min\left\{ r\left(\xi_s^{\mathbf{a}}(\tau)\right), \min_{\kappa'\in\mathbb{N}} q\left(\xi_s^{[\mathbf{a},\mathbf{a}']_\tau}(\tau+\kappa')\right), \min_{\kappa\leq\tau} q\left(\xi_s^{\mathbf{a}}(\kappa)\right)\right\}\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \max_{\mathbf{a}'\in\mathbb{A}} \min\left\{ r\left(\xi_s^{[\mathbf{a},\mathbf{a}']_\tau}(\tau)\right), \min_{\kappa'\in\mathbb{N}} q\left(\xi_s^{[\mathbf{a},\mathbf{a}']_\tau}(\tau+\kappa')\right), \min_{\kappa\leq\tau} q\left(\xi_s^{[\mathbf{a},\mathbf{a}']_\tau}(\kappa)\right)\right\} && (26)\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \min\left\{ r\left(\xi_s^{\mathbf{a}}(\tau)\right), \min_{\kappa'\in\mathbb{N}} q\left(\xi_s^{\mathbf{a}}(\tau+\kappa')\right), \min_{\kappa\leq\tau} q\left(\xi_s^{\mathbf{a}}(\kappa)\right)\right\} && (27)\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \min\left\{ r\left(\xi_s^{\mathbf{a}}(\tau)\right), \min_{\kappa\in\mathbb{N}} q\left(\xi_s^{\mathbf{a}}(\kappa)\right)\right\}\\
&= \max_{\mathbf{a}\in\mathbb{A}} \min\left\{ \max_{\tau\in\mathbb{N}} r\left(\xi_s^{\mathbf{a}}(\tau)\right), \min_{\kappa\in\mathbb{N}} q\left(\xi_s^{\mathbf{a}}(\kappa)\right)\right\}\\
&= v^*_{\text{RAA}}(s),
\end{aligned}$$

where the equality between (24) and (25) follows from Corollary 3, and where the equality between (26) and (27) follows from Lemma 3. □

Before the next lemma, we need to introduce two last pieces of notation. First, we let $\overline{\Pi}$ be the set of augmented policies $\bar{\pi} : \mathcal{S} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{A}$, where

$$\mathcal{Y} = \{r(s) \mid s \in \mathcal{S}\} \quad \text{and} \quad \mathcal{Z} = \{q(s) \mid s \in \mathcal{S}\}.$$

Next, given $s \in \mathcal{S}$, $y \in \mathcal{Y}$, $z \in \mathcal{Z}$, and $\bar{\pi} \in \overline{\Pi}$, we let $\bar{\xi}_s^{\bar{\pi}} : \mathbb{N} \to \mathcal{S}$, $\bar{\eta}_s^{\bar{\pi}} : \mathbb{N} \to \mathcal{Y}$, and $\bar{\zeta}_s^{\bar{\pi}} : \mathbb{N} \to \mathcal{Z}$, be the solution of the evolution

$$\begin{aligned}
\bar{\xi}_s^{\bar{\pi}}(t+1) &= f\left(\bar{\xi}_s^{\bar{\pi}}(t), \bar{\pi}\left(\bar{\xi}_s^{\bar{\pi}}(t), \bar{\eta}_s^{\bar{\pi}}(t), \bar{\zeta}_s^{\bar{\pi}}(t)\right)\right), \\
\bar{\eta}_s^{\bar{\pi}}(t+1) &= \max\left\{r\left(\bar{\xi}_s^{\bar{\pi}}(t+1)\right), \bar{\eta}_s^{\bar{\pi}}(t)\right\}, \\
\bar{\zeta}_s^{\bar{\pi}}(t+1) &= \min\left\{q\left(\bar{\xi}_s^{\bar{\pi}}(t+1)\right), \bar{\zeta}_s^{\bar{\pi}}(t)\right\},
\end{aligned}$$

for which $\bar{\xi}_s^{\bar{\pi}}(0) = s$, $\bar{\eta}_s^{\bar{\pi}}(0) = r(s)$, and $\bar{\zeta}_s^{\bar{\pi}}(0) = q(s)$.

Lemma 5. *There is a $\bar{\pi} \in \overline{\Pi}$ such that*

$$v^*_{\text{RAA}}(s) = \min\left\{\max_{\tau \in \mathbb{N}} r\left(\bar{\xi}_s^{\bar{\pi}}(\tau)\right), \min_{\tau \in \mathbb{N}} q\left(\bar{\xi}_s^{\bar{\pi}}(\tau)\right)\right\} \tag{28}$$

for all $s \in \mathcal{S}$.

Proof. By Lemmas 1 and 2 together with Corollary 3, we can choose $\pi, \theta \in \Pi$ such that

$$\begin{aligned}
\tilde{v}^*_{\text{RA}}(s) &= \max_{\tau \in \mathbb{N}} \min\left\{r\left(\xi_s^{\pi}(\tau)\right), v^*_{\text{A}}\left(\xi_s^{\pi}(\tau)\right), \min_{\kappa \le \tau} q\left(\xi_s^{\pi}(\kappa)\right)\right\} \quad \forall s \in \mathcal{S}, \\
v^*_{\text{A}}(s) &= \min_{\tau \in \mathbb{N}} q\left(\xi_s^{\theta}(\tau)\right) \quad \forall s \in \mathcal{S}.
\end{aligned}$$

We introduce some useful notation we will use throughout the rest of the proof. For each $s \in \mathcal{S}$, let $[s]^+ = f(s, \pi(s))$, $[y]_s^+ = \max\{y, r\left([s]^+\right)\}$, $[z]_s^+ = \min\{z, q\left([s]^+\right)\}$.

We define an augmented policy $\bar{\pi} \in \overline{\Pi}$ by

$$\bar{\pi}(s, y, z) = \begin{cases} \pi(s) & \min\{[y]_s^+, [z]_s^+, v^*_{\text{A}}([s]^+)\} \ge \min\{y, z, v^*_{\text{A}}(s)\}, \\ \theta(s) & \text{otherwise.} \end{cases}$$

Now fix some $s \in \mathcal{S}$. For all $t \in \mathbb{N}$, set $\bar{x}_t = \bar{\xi}_s^{\bar{\pi}}(t)$, $\bar{y}_t = \bar{\eta}_s^{\bar{\pi}}(t) = \max_{\tau \le t} r(\bar{x}_\tau)$, and $\bar{z}_t = \bar{\zeta}_s^{\bar{\pi}}(t) = \min_{\tau \le t} q(\bar{x}_\tau)$, and also set $x_t^\circ = \xi_s^{\pi}(t)$, $y_t^\circ = \max_{\tau \le t} r(x_\tau^\circ)$, and $z_t^\circ = \min_{\tau \le t} q(x_\tau^\circ)$.

First, assume that t is such that $\min\{[\bar{y}_t]_{\bar{x}_t}^+, [\bar{z}_t]_{\bar{x}_t}^+, v^*_{\text{A}}([\bar{x}_t]^+)\} < \min\{\bar{y}_t, \bar{z}_t, v^*_{\text{A}}(\bar{x}_t)\}$. In this case, $\bar{\pi}(\bar{x}_t, \bar{y}_t, \bar{z}_t) = \theta(\bar{x}_t)$, so that

$$\min\{\bar{z}_t, v^*_{\text{A}}(\bar{x}_t)\} = \min\{\bar{z}_{t+1}, v^*_{\text{A}}(\bar{x}_{t+1})\}$$

by our choice of θ. Since $\bar{y}_t$ is non-decreasing in t, thus have

$$\min\{\bar{y}_t, \bar{z}_t, v^*_{\text{A}}(\bar{x}_t)\} \le \min\{\bar{y}_{t+1}, \bar{z}_{t+1}, v^*_{\text{A}}(\bar{x}_{t+1})\}.$$

Next, assume that t is such that $\min\{[\bar{y}_t]_{\bar{x}_t}^+, [\bar{z}_t]_{\bar{x}_t}^+, v^*_{\text{A}}([\bar{x}_t]^+)\} \ge \min\{\bar{y}_t, \bar{z}_t, v^*_{\text{A}}(\bar{x}_t)\}$. In this case, we have that $\bar{\pi}(\bar{x}_t, \bar{y}_t, \bar{z}_t) = \pi(\bar{x}_t)$, so

$$\min\{\bar{y}_t, \bar{z}_t, v^*_{\text{A}}(\bar{x}_t)\} \le \min\{[\bar{y}_t]_{\bar{x}_t}^+, [\bar{z}_t]_{\bar{x}_t}^+, v^*_{\text{A}}([\bar{x}_t]^+)\} = \min\{\bar{y}_{t+1}, \bar{z}_{t+1}, v^*_{\text{A}}(\bar{x}_{t+1})\}.$$

It thus follows from these two cases that $\min\{\bar{y}_t, \bar{z}_t, v^*_{\text{A}}(\bar{x}_t)\}$ is non-decreasing in t. Let

$$T = \min\left\{t \in \mathbb{N} \mid \min\{[\bar{y}_t]_{\bar{x}_t}^+, [\bar{z}_t]_{\bar{x}_t}^+, v^*_{\text{A}}([\bar{x}_t]^+)\} < \min\{\bar{y}_t, \bar{z}_t, v^*_{\text{A}}(\bar{x}_t)\}\right\}.$$

There are again two cases:

($T < \infty$) In this case, $\bar{\pi}(\bar{x}_t, \bar{y}_t, \bar{z}_t) = \pi(\bar{x}_t)$ for $t < T$. Then $\bar{x}_t = x_t^\circ$, $\bar{y}_t = y_t^\circ$, and $\bar{z}_t = z_t^\circ$ for all $t \leq T$. It follows that $[\bar{x}_t]^+ = x_{t+1}^\circ$, $[\bar{y}_t]^+_{\bar{x}_t} = y_{t+1}^\circ$, and $[\bar{z}_t]^+_{\bar{x}_t} = z_{t+1}^\circ$ for all $t \leq T$. Thus by definition of T,

$$\min\left\{y_{t+1}^\circ, z_{t+1}^\circ, v_{\mathrm{A}}^*\left(x_{t+1}^\circ\right)\right\} \geq \min\left\{y_t^\circ, z_t^\circ, v_{\mathrm{A}}^*\left(x_t^\circ\right)\right\} \quad \forall t < T.$$

and

$$\min\left\{y_{T+1}^\circ, z_{T+1}^\circ, v_{\mathrm{A}}^*\left(x_{T+1}^\circ\right)\right\} < \min\left\{y_T^\circ, z_T^\circ, v_{\mathrm{A}}^*\left(x_T^\circ\right)\right\}.$$

But since y_t° is non-decreasing and $\min\{z_t^\circ, v_{\mathrm{A}}^*(x_t^\circ)\}$ is non-increasing in t, it follows that $\min\{y_t^\circ, z_t^\circ, v_{\mathrm{A}}^*(x_t^\circ)\}$ must achieve its maximal value at the smallest t for which it strictly decreases from t to $t+1$, i.e.

$$\begin{aligned}
\min\left\{\bar{y}_T, \bar{z}_T, v_{\mathrm{A}}^*\left(\bar{x}_T\right)\right\} &= \min\left\{y_T^\circ, z_T^\circ, v_{\mathrm{A}}^*\left(x_T^\circ\right)\right\} \\
&= \max_{t\in\mathbb{N}} \min\left\{y_t^\circ, z_t^\circ, v_{\mathrm{A}}^*\left(x_t^\circ\right)\right\} \\
&\geq \max_{t\in\mathbb{N}} \min\left\{r\left(x_t^\circ\right), z_t^\circ, v_{\mathrm{A}}^*\left(x_t^\circ\right)\right\} \\
&= \tilde{v}_{\mathrm{RA}}^*(s).
\end{aligned}$$

where the final equality follows from our choice of π. Since $\min\{\bar{y}_t, \bar{z}_t, v_{\mathrm{A}}^*(\bar{x}_t)\}$ is non-decreasing in t, then

$$\min\{\bar{y}_t, \bar{z}_t\} \geq \min\{\bar{y}_t, \bar{z}_t, v_{\mathrm{A}}^*(\bar{x}_t)\} \geq \min\{\bar{y}_T, \bar{z}_T, v_{\mathrm{A}}^*(\bar{x}_T)\} = \tilde{v}_{\mathrm{RA}}^*(s) \quad \forall t \geq T.$$

Thus

$$v_{\mathrm{RAA}}^*(s) \geq \min\left\{\max_{t\in\mathbb{N}} r\left(\bar{x}_t\right), \min_{t\in\mathbb{N}} q\left(\bar{x}_t\right)\right\} = \lim_{t\to\infty} \min\{\bar{y}_t, \bar{z}_t\} \geq \tilde{v}_{\mathrm{RA}}^*(s) = v_{\mathrm{RAA}}^*(s),$$

where the final equality follows from Lemma (4). Thus the proof is complete in this case.

($T = \infty$) In this case, $\bar{\pi}(\bar{x}_t, \bar{y}_t, \bar{z}_t) = \pi(\bar{x}_t)$ for all $t \in \mathbb{N}$. Then $\bar{x}_t = x_t^\circ$, $\bar{y}_t = y_t^\circ$, and $\bar{z}_t = z_t^\circ$ for all $t \in \mathbb{N}$. Also $[\bar{x}_t]^+ = x_{t+1}^\circ$, $[\bar{y}_t]^+_{\bar{x}_t} = y_{t+1}^\circ$, and $[\bar{z}_t]^+_{\bar{x}_t} = z_{t+1}^\circ$ for all $t \in \mathbb{N}$. Thus by definition of T,

$$\min\left\{y_{t+1}^\circ, z_{t+1}^\circ, v_{\mathrm{A}}^*\left(x_{t+1}^\circ\right)\right\} \geq \min\left\{y_t^\circ, z_t^\circ, v_{\mathrm{A}}^*\left(x_t^\circ\right)\right\} \quad \forall t \in \mathbb{N}.$$

Let $T' \in \arg\max_{t\in\mathbb{N}} \min\left\{y_t^\circ, z_t^\circ, v_{\mathrm{A}}^*\left(x_t^\circ\right)\right\}$. Then

$$\begin{aligned}
\min\left\{\bar{y}_{T'}, \bar{z}_{T'}, v_{\mathrm{A}}^*\left(\bar{x}_{T'}\right)\right\} &= \min\left\{y_{T'}^\circ, z_{T'}^\circ, v_{\mathrm{A}}^*\left(x_{T'}^\circ\right)\right\} \\
&= \max_{t\in\mathbb{N}} \min\left\{y_t^\circ, z_t^\circ, v_{\mathrm{A}}^*\left(x_t^\circ\right)\right\} \\
&\geq \max_{t\in\mathbb{N}} \min\left\{r\left(x_t^\circ\right), z_t^\circ, v_{\mathrm{A}}^*\left(x_t^\circ\right)\right\} \\
&= \tilde{v}_{\mathrm{RA}}^*(s).
\end{aligned}$$

The rest of the proof the follows the same as the previous case with T replaced by T'.

□

Corollary 5. *For all $s \in \mathcal{S}$, we have $V_{\mathrm{RAA}}^*(s) = v_{\mathrm{RAA}}^*(s)$.*

Proof of Theorem 1. Theorem 1 is now a direct consequence of the previous corollary together with Corollary 4 and Lemma 4. □

B Proof of RR Main Theorem

We first define the value functions, $V^*_{\text{R1}}, V^*_{\text{R2}}, \tilde{V}^*_{\text{R}}, V^*_{\text{RR}} : \mathcal{S} \to \mathbb{R}$ by

$$\begin{aligned}
V^*_{\text{R1}}(s) &= \max_{\pi\in\Pi} \max_{\tau\in\mathbb{N}} r_1\left(\xi^\pi_s(\tau)\right), \\
V^*_{\text{R2}}(s) &= \max_{\pi\in\Pi} \max_{\tau\in\mathbb{N}} r_2\left(\xi^\pi_s(\tau)\right), \\
\tilde{V}^*_{\text{R}}(s) &= \max_{\pi\in\Pi} \max_{\tau\in\mathbb{N}} r_{\text{RR}}\left(\xi^\pi_s(\tau)\right), \\
V^*_{\text{RR}}(s) &= \max_{\pi\in\Pi} \min\left\{ \max_{\tau\in\mathbb{N}} r_1\left(\xi^\pi_s(\tau)\right), \max_{\tau\in\mathbb{N}} r_2\left(\xi^\pi_s(\tau)\right) \right\}.
\end{aligned}$$

We next define the value functions, $v^*_{\text{R1}}, v^*_{\text{R2}}, \tilde{v}^*_{\text{R}}, v^*_{\text{RR}} : \mathcal{S} \to \mathbb{R}$, which maximize over action sequences rather than policies:

$$\begin{aligned}
v^*_{\text{R1}}(s) &= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} r_1\left(\xi^\mathbf{a}_s(\tau)\right), \\
v^*_{\text{R2}}(s) &= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} r_2\left(\xi^\mathbf{a}_s(\tau)\right), \\
\tilde{v}^*_{\text{R}}(s) &= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} r_{\text{RR}}\left(\xi^\mathbf{a}_s(\tau)\right), \\
v^*_{\text{RR}}(s) &= \max_{\mathbf{a}\in\mathbb{A}} \min\left\{ \max_{\tau\in\mathbb{N}} r_1\left(\xi^\mathbf{a}_s(\tau)\right), \max_{\tau\in\mathbb{N}} r_2\left(\xi^\mathbf{a}_s(\tau)\right) \right\},
\end{aligned}$$

where r_{RR} is as in Theorem 2. Observe that for each $s \in \mathcal{S}$,

$$v^*_{\text{R1}}(s) \geq V^*_{\text{R1}}(s), \quad v^*_{\text{R2}}(s) \geq V^*_{\text{R2}}(s), \quad \tilde{v}^*_{\text{R}}(s) \geq \tilde{V}^*_{\text{R}}(s), \quad v^*_{\text{RR}}(s) \geq V^*_{\text{RR}}(s).$$

We now prove a series of lemmas that will be useful in the proof of the main theorem.

Lemma 6. *There are $\pi_1, \pi_2 \in \Pi$ such that*

$$v^*_{\text{R1}}(s) = \max_{\tau\in\mathbb{N}} r_1\left(\xi^{\pi_1}_s(\tau)\right) \text{ and } v^*_{\text{R2}}(s) = \max_{\tau\in\mathbb{N}} r_2\left(\xi^{\pi_2}_s(\tau)\right)$$

for all $s \in \mathcal{S}$.

Proof. We will just prove the result for $v^*_{\text{R1}}(s)$ since the other result follows identically. For each $s \in \mathcal{S}$, let τ_s be the smallest element of $\mathbb{N}$ for which

$$\max_{\mathbf{a}\in\mathbb{A}} r_1\left(\xi^\mathbf{a}_s(\tau_s)\right) = v^*_{\text{R1}}(s).$$

Moreover, for each $s \in \mathcal{S}$, let $\mathbf{a}_s$ be such that

$$r_1\left(\xi^{\mathbf{a}_s}_s(\tau_s)\right) = v^*_{\text{R1}}(s).$$

Let $\pi_1 \in \Pi$ be given by $\pi_1(s) = \mathbf{a}_s(0)$. It suffices to show that

$$r_1\left(\xi^{\pi_1}_s(\tau_s)\right) = v^*_{\text{R1}}(s) \tag{29}$$

for all $s \in \mathcal{S}$, for in this case, we have

$$v^*_{\text{R1}}(s) \geq \max_{\tau\in\mathbb{N}} r_1\left(\xi^{\pi_1}_s(\tau)\right) \geq r_1\left(\xi^{\pi_1}_s(\tau_s)\right) = v^*_{\text{R1}}(s) \quad \forall s \in \mathcal{S}.$$

We show (29) holds for each $s \in \mathcal{S}$ by induction on τ_s. First, suppose that $s \in \mathcal{S}$ is such that $\tau_s = 0$. Then

$$r_1\left(\xi^{\pi_1}_s(\tau_s)\right) = r_1(s) = r_1\left(\xi^{\mathbf{a}_s}_s(\tau_s)\right) = v^*_{\text{R1}}(s).$$

For the induction step, let $n \in \mathbb{N}$ and suppose that

$$r_1\left(\xi^{\pi_1}_s(\tau_s)\right) = v^*_{\text{R1}}(s) \quad \forall s \in \mathcal{S} \text{ such that } \tau_s \leq n.$$

Now fix some $x \in \mathcal{S}$ such that $\tau_x = n+1$. Notice that

$$\begin{aligned} v^*_{\mathrm{R1}}(x) &\geq v^*_{\mathrm{R1}}\left(f\left(x, \pi_1(x)\right)\right) \\ &\geq \max_{\mathbf{a}\in\mathbb{A}} r_1\left(\xi^{\mathbf{a}}_{f(x,\pi_1(x))}(n)\right) \\ &\geq r_1\left(\xi^{\mathbf{a}_x|_1}_{f(x,\pi_1(x))}(n)\right) \\ &= r_1\left(\xi^{[\pi_1(x),\mathbf{a}_x|_1]}_{x}(n+1)\right) \\ &= r_1\left(\xi^{\mathbf{a}_x}_{x}(\tau_x)\right) \\ &= v^*_{\mathrm{R1}}(x), \end{aligned}$$

so that $v^*_{\mathrm{R1}}\left(f\left(x, \pi_1(x)\right)\right) = v^*_{\mathrm{R1}}(x)$ and $\tau_{f(x,\pi_1(x))} \leq n$. It suffices to show

$$\tau_{f(x,\pi_1(x))} = n, \tag{30}$$

for then, by the induction assumption, we have

$$r_1\left(\xi^{\pi_1}_x(\tau_x)\right) = r_1\left(\xi^{\pi_1}_{f(x,\pi_1(x))}(n)\right) = v^*_{\mathrm{R1}}\left(f\left(x, \pi_1(x)\right)\right) = v^*_{\mathrm{R1}}(x).$$

To show (30), assume instead that

$$\tau_{f(x,\pi_1(x))} < n.$$

But

$$\begin{aligned} v^*_{\mathrm{R1}}(x) &\geq \max_{\mathbf{a}\in\mathbb{A}} r_1\left(\xi^{\mathbf{a}}_x\left(\tau_{f(x,\pi_1(x))}+1\right)\right) \\ &\geq r_1\left(\xi^{\left[\pi_1(x),\mathbf{a}_{f(x,\pi_1(x))}\right]}_x\left(\tau_{f(x,\pi_1(x))}+1\right)\right) \\ &= r_1\left(\xi^{\mathbf{a}_{f(x,\pi_1(x))}}_{f(x,\pi_1(x))}\left(\tau_{f(x,\pi_1(x))}\right)\right) \\ &= v^*_{\mathrm{R1}}\left(f\left(x, \pi_1(x)\right)\right) \\ &= v^*_{\mathrm{R1}}(x), \end{aligned}$$

so that

$$v^*_{\mathrm{R1}}(x) = \max_{\mathbf{a}\in\mathbb{A}} r_1\left(\xi^{\mathbf{a}}_x\left(\tau_{f(x,\pi_1(x))}+1\right)\right)$$

and thus

$$\tau_x \leq \tau_{f(x,\pi_1(x))} + 1 < n+1,$$

giving our desired contradiction. □

Corollary 6. *For all $s \in \mathcal{S}$, we have $V^*_{\mathrm{R1}}(s) = v^*_{\mathrm{R1}}(s)$ and $V^*_{\mathrm{R2}}(s) = v^*_{\mathrm{R2}}(s)$.*

Lemma 7. *There is a $\pi \in \Pi$ such that*

$$\tilde{v}^*_{\mathrm{R}}(s) = \max_{\tau\in\mathbb{N}} r_{\mathrm{RR}}\left(\xi^{\pi}_s(\tau)\right).$$

for all $s \in \mathcal{S}$.

Proof. This lemma follows by precisely the same proof as the previous lemma, with r_1, v^*_{R1}, and π_1 replaced with r_{RR}, $\tilde{v}^*_{\mathrm{R}}$, and π respectively. □

Corollary 7. *For all $s \in \mathcal{S}$, we have $\tilde{V}^*_{\mathrm{R}}(s) = \tilde{v}^*_{\mathrm{R}}(s)$.*

Lemma 8. *Let $\zeta_1 : \mathbb{N} \to \mathbb{R}$ and $\zeta_2 : \mathbb{N} \to \mathbb{R}$. Then*

$$\begin{aligned} &\sup_{\tau\in\mathbb{N}} \max\left\{\min\left\{\zeta_1(\tau), \sup_{\tau'\in\mathbb{N}} \zeta_2(\tau+\tau')\right\}, \min\left\{\sup_{\tau'\in\mathbb{N}} \zeta_1(\tau+\tau'), \zeta_2(\tau)\right\}\right\} \\ &\quad = \min\left\{\sup_{\tau\in\mathbb{N}} \zeta_1(\tau), \sup_{\tau\in\mathbb{N}} \zeta_2(\tau)\right\}. \end{aligned}$$

Proof. We proceed by showing both inequalities corresponding to the above equality hold.

($\leq$) Observe that

$$\begin{aligned}
&\sup_{\tau\in\mathbb{N}} \max\left\{\min\left\{\zeta_1(\tau), \sup_{\tau'\in\mathbb{N}} \zeta_2(\tau+\tau')\right\}, \min\left\{\sup_{\tau'\in\mathbb{N}} \zeta_1(\tau+\tau'), \zeta_2(\tau)\right\}\right\} \\
&\quad \leq \max\left\{\min\left\{\sup_{\tau\in\mathbb{N}} \zeta_1(\tau), \sup_{\tau\in\mathbb{N}}\sup_{\tau'\in\mathbb{N}} \zeta_2(\tau+\tau')\right\}, \min\left\{\sup_{\tau\in\mathbb{N}}\sup_{\tau'\in\mathbb{N}} \zeta_1(\tau+\tau'), \sup_{\tau\in\mathbb{N}} \zeta_2(\tau)\right\}\right\} \\
&\quad = \min\left\{\sup_{\tau\in\mathbb{N}} \zeta_1(\tau), \sup_{\tau\in\mathbb{N}} \zeta_2(\tau)\right\}
\end{aligned}$$

($\geq$) Fix $\varepsilon > 0$. Choose $\tau_1, \tau_2 \in \mathbb{N}$ such that $\zeta_1(\tau_1) \geq \sup_{\tau\in\mathbb{N}} \zeta_1(\tau) - \varepsilon$ and $\zeta_2(\tau_2) \geq \sup_{\tau\in\mathbb{N}} \zeta_2(\tau) - \varepsilon$. Without loss of generality, we can assume $\tau_1 \leq \tau_2$. Then

$$\begin{aligned}
&\sup_{\tau\in\mathbb{N}} \max\left\{\min\left\{\zeta_1(\tau), \sup_{\tau'\in\mathbb{N}} \zeta_2(\tau+\tau')\right\}, \min\left\{\sup_{\tau'\in\mathbb{N}} \zeta_1(\tau+\tau'), \zeta_2(\tau)\right\}\right\} \\
&\quad \geq \sup_{\tau\in\mathbb{N}} \min\left\{\zeta_1(\tau), \sup_{\tau'\in\mathbb{N}} \zeta_2(\tau+\tau')\right\} \\
&\quad \geq \min\left\{\zeta_1(\tau_1), \sup_{\tau'\in\mathbb{N}} \zeta_2(\tau_1+\tau')\right\} \\
&\quad \geq \min\left\{\zeta_1(\tau_1), \zeta_2(\tau_2)\right\} \\
&\quad \geq \min\left\{\sup_{\tau\in\mathbb{N}} \zeta_1(\tau) - \varepsilon, \sup_{\tau\in\mathbb{N}} \zeta_2(\tau) - \varepsilon\right\} \\
&\quad = \min\left\{\sup_{\tau\in\mathbb{N}} \zeta_1(\tau), \sup_{\tau\in\mathbb{N}} \zeta_2(\tau)\right\} - \varepsilon.
\end{aligned}$$

But since $\varepsilon > 0$ was arbitrary, the desired inequality follows.

□

Lemma 9. *For each $s \in \mathcal{S}$,*

$$\tilde{v}^*_{\mathrm{R}}(s) = v^*_{\mathrm{RR}}(s).$$

Proof. For each $s \in \mathcal{S}$,

$$\begin{aligned}
\tilde{v}^*_{\mathrm{R}}(s) &= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} r_{\mathrm{RR}}\left(\xi^{\mathbf{a}}_s(\tau)\right) && (31)\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \max\left\{\min\left\{r_1\left(\xi^{\mathbf{a}}_s(\tau)\right), v^*_{\mathrm{R2}}\left(\xi^{\mathbf{a}}_s(\tau)\right)\right\}, \min\left\{v^*_{\mathrm{R1}}\left(\xi^{\mathbf{a}}_s(\tau)\right), r_2\left(\xi^{\mathbf{a}}_s(\tau)\right)\right\}\right\} && (32)\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \max\left\{\min\left\{r_1\left(\xi^{\mathbf{a}}_s(\tau)\right), \max_{\mathbf{a}'\in\mathbb{A}} \max_{\tau'\in\mathbb{N}} r_2\left(\xi^{\mathbf{a}'}_{\xi^{\mathbf{a}}_s(\tau)}(\tau')\right)\right\},\right.\\
&\qquad\qquad \left.\min\left\{\max_{\mathbf{a}'\in\mathbb{A}} \max_{\tau'\in\mathbb{N}} r_1\left(\xi^{\mathbf{a}'}_{\xi^{\mathbf{a}}_s(\tau)}(\tau')\right), r_2\left(\xi^{\mathbf{a}}_s(\tau)\right)\right\}\right\}\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \max\left\{\min\left\{r_1\left(\xi^{\mathbf{a}}_s(\tau)\right), \max_{\mathbf{a}'\in\mathbb{A}} \max_{\tau'\in\mathbb{N}} r_2\left(\xi^{[\mathbf{a},\mathbf{a}']_\tau}_s(\tau+\tau')\right)\right\},\right.\\
&\qquad\qquad \left.\min\left\{\max_{\mathbf{a}'\in\mathbb{A}} \max_{\tau'\in\mathbb{N}} r_1\left(\xi^{[\mathbf{a},\mathbf{a}']_\tau}_s(\tau+\tau')\right), r_2\left(\xi^{\mathbf{a}}_s(\tau)\right)\right\}\right\}\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \max_{\mathbf{a}'\in\mathbb{A}} \max\left\{\min\left\{r_1\left(\xi^{\mathbf{a}}_s(\tau)\right), \max_{\tau'\in\mathbb{N}} r_2\left(\xi^{[\mathbf{a},\mathbf{a}']_\tau}_s(\tau+\tau')\right)\right\},\right.\\
&\qquad\qquad \left.\min\left\{\max_{\tau'\in\mathbb{N}} r_1\left(\xi^{[\mathbf{a},\mathbf{a}']_\tau}_s(\tau+\tau')\right), r_2\left(\xi^{\mathbf{a}}_s(\tau)\right)\right\}\right\}\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \max_{\mathbf{a}'\in\mathbb{A}} \max\left\{\min\left\{r_1\left(\xi^{[\mathbf{a},\mathbf{a}']_\tau}_s(\tau)\right), \max_{\tau'\in\mathbb{N}} r_2\left(\xi^{[\mathbf{a},\mathbf{a}']_\tau}_s(\tau+\tau')\right)\right\},\right.\\
&\qquad\qquad \left.\min\left\{\max_{\tau'\in\mathbb{N}} r_1\left(\xi^{[\mathbf{a},\mathbf{a}']_\tau}_s(\tau+\tau')\right), r_2\left(\xi^{[\mathbf{a},\mathbf{a}']_\tau}_s(\tau)\right)\right\}\right\} && (33)\\
&= \max_{\mathbf{a}\in\mathbb{A}} \max_{\tau\in\mathbb{N}} \max\left\{\min\left\{r_1\left(\xi^{\mathbf{a}}_s(\tau)\right), \max_{\tau'\in\mathbb{N}} r_2\left(\xi^{\mathbf{a}}_s(\tau+\tau')\right)\right\},\right.\\
&\qquad\qquad \left.\min\left\{\max_{\tau'\in\mathbb{N}} r_1\left(\xi^{\mathbf{a}}_s(\tau+\tau')\right), r_2\left(\xi^{\mathbf{a}}_s(\tau)\right)\right\}\right\} && (34)\\
&= \max_{\mathbf{a}\in\mathbb{A}} \min\left\{\max_{\tau\in\mathbb{N}} r_1\left(\xi^{\mathbf{a}}_s(\tau)\right), \max_{\tau\in\mathbb{N}} r_2\left(\xi^{\mathbf{a}}_s(\tau)\right)\right\} && (35)\\
&= v^*_{\mathrm{RR}}(s),
\end{aligned}$$

where the equality between 31 and 32 follows from Corollary 6, the equality between 33 and 34 follows from Lemma 3, and the equality between 34 and 35 follows from Lemma 8. □

Before the next lemma, we need to introduce two last pieces of notation. First, we let $\overline{\Pi}$ be the set of augmented policies $\bar{\pi} : \mathcal{S} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{A}$, as in the previous section, but where

$$\mathcal{Y} = \{r_1(s) \mid s \in \mathcal{S}\} \quad \text{and} \quad \mathcal{Z} = \{r_2(s) \mid s \in \mathcal{S}\}.$$

Next, given $s \in \mathcal{S}$, $y \in \mathcal{Y}$, $z \in \mathcal{Z}$, and $\bar{\pi} \in \overline{\Pi}$, we let $\bar{\xi}^{\bar{\pi}}_s : \mathbb{N} \to \mathcal{S}$, $\bar{\eta}^{\bar{\pi}}_s : \mathbb{N} \to \mathcal{Y}$, and $\bar{\zeta}^{\bar{\pi}}_s : \mathbb{N} \to \mathcal{Z}$, be the solution of the evolution

$$\begin{aligned}
\bar{\xi}^{\bar{\pi}}_s(t+1) &= f\left(\bar{\xi}^{\bar{\pi}}_s(t), \bar{\pi}\left(\bar{\xi}^{\bar{\pi}}_s(t), \bar{\eta}^{\bar{\pi}}_s(t), \bar{\zeta}^{\bar{\pi}}_s(t)\right)\right),\\
\bar{\eta}^{\bar{\pi}}_s(t+1) &= \max\left\{r_1\left(\bar{\xi}^{\bar{\pi}}_s(t+1)\right), \bar{\eta}^{\bar{\pi}}_s(t)\right\},\\
\bar{\zeta}^{\bar{\pi}}_s(t+1) &= \max\left\{r_2\left(\bar{\xi}^{\bar{\pi}}_s(t+1)\right), \bar{\zeta}^{\bar{\pi}}_s(t)\right\},
\end{aligned}$$

for which $\bar{\xi}^{\bar{\pi}}_s(0) = s$, $\bar{\eta}^{\bar{\pi}}_s(0) = r_1(s)$, and $\bar{\zeta}^{\bar{\pi}}_s(0) = r_2(s)$.

Lemma 10. *There is a $\bar{\pi} \in \overline{\Pi}$ such that*

$$v^*_{\mathrm{RR}}(s) = \min\left\{\max_{\tau\in\mathbb{N}} r_1\left(\bar{\xi}^{\bar{\pi}}_s(\tau)\right), \max_{\tau\in\mathbb{N}} r_2\left(\bar{\xi}^{\bar{\pi}}_s(\tau)\right)\right\}$$

for all $s \in \mathcal{S}$.

Proof. By Lemmas 6 and 7 together with Corollary 6, we can choose $\pi, \theta_1, \theta_2 \in \Pi$ such that

$$v^*_{\text{R1}}(s) = \max_{\tau\in\mathbb{N}} r_1\left(\xi_s^{\theta_1}(\tau)\right) \quad \forall s \in \mathcal{S},$$
$$v^*_{\text{R2}}(s) = \max_{\tau\in\mathbb{N}} r_2\left(\xi_s^{\theta_2}(\tau)\right) \quad \forall s \in \mathcal{S},$$
$$\tilde{v}^*_{\text{R}}(s) = \max_{\tau\in\mathbb{N}} \max\left\{\min\left\{r_1\left(\xi_s^{\pi}(\tau)\right), v^*_{\text{R2}}\left(\xi_s^{\pi}(\tau)\right)\right\}, \min\left\{r_2\left(\xi_s^{\pi}(\tau)\right), v^*_{\text{R1}}\left(\xi_s^{\pi}(\tau)\right)\right\}\right\} \quad \forall s \in \mathcal{S}.$$

Define $\bar{\pi} \in \overline{\Pi}$ by

$$\bar{\pi}(s,y,z) = \begin{cases} \pi(s) & \max\{y,z\} < \tilde{v}^*_{\text{R}}(s) \\ \theta_1(s) & \max\{y,z\} \geq \tilde{v}^*_{\text{R}}(s) \text{ and } y \leq z, \\ \theta_2(s) & \max\{y,z\} \geq \tilde{v}^*_{\text{R}}(s) \text{ and } y > z. \end{cases}$$

Now fix some $s \in \mathcal{S}$. For all $t \in \mathbb{N}$, set $\bar{x}_t = \bar{\xi}^{\bar{\pi}}_s(t)$, $\bar{y}_t = \bar{\eta}^{\bar{\pi}}_s(t) = \max_{\tau\leq t} r_1(\bar{x}_\tau)$, and $\bar{z}_t = \bar{\zeta}^{\bar{\pi}}_s(t) = \max_{\tau\leq t} r_2(\bar{x}_\tau)$, and also set $x^\circ_t = \xi^\pi_s(t)$. It suffices to show

$$v^*_{\text{RR}}(s) \leq \min\left\{\max_{\tau\in\mathbb{N}} r_1\left(\bar{x}_\tau\right), \max_{\tau\in\mathbb{N}} r_2\left(\bar{x}_\tau\right)\right\}, \tag{36}$$

since the reverse inequality is immediate. We proceed in three steps.

1. We claim there exists a $t \in \mathbb{N}$ such that $\max\{r_1(\bar{x}_t), r_2(\bar{x}_t)\} \geq \tilde{v}^*_{\text{R}}(\bar{x}_t)$.

   Suppose otherwise. Then $\bar{\pi}(\bar{x}_t, \bar{y}_t, \bar{z}_t) = \pi(\bar{x}_t)$ so that $\bar{x}_t = x^\circ_t$ for all $t \in \mathbb{N}$. Thus

   $$\begin{aligned}\max_{t\in\mathbb{N}} \max\{r_1(\bar{x}_t), r_2(\bar{x}_t)\} &< \max_{t\in\mathbb{N}} \tilde{v}^*_{\text{R}}(\bar{x}_t) \\ &= \tilde{v}^*_{\text{R}}(s) \\ &= \max_{\tau\in\mathbb{N}} \max\left\{\min\left\{r_1\left(x^\circ_\tau\right), v^*_{\text{R2}}\left(x^\circ_\tau\right)\right\}, \min\left\{r_2\left(x^\circ_\tau\right), v^*_{\text{R1}}\left(x^\circ_\tau\right)\right\}\right\} \\ &= \max_{\tau\in\mathbb{N}} \max\left\{\min\left\{r_1\left(\bar{x}_\tau\right), v^*_{\text{R2}}\left(\bar{x}_\tau\right)\right\}, \min\left\{r_2\left(\bar{x}_\tau\right), v^*_{\text{R1}}\left(\bar{x}_\tau\right)\right\}\right\} \\ &\leq \max_{\tau\in\mathbb{N}} \max\left\{r_1(\bar{x}_\tau), r_2(\bar{x}_\tau)\right\},\end{aligned}$$

   providing the desired contradiction.

2. Let T be the smallest element of $\mathbb{N}$ for which

   $$\max\left\{r_1(\bar{x}_T), r_2(\bar{x}_T)\right\} \geq v^*_{\text{R}}(\bar{x}_T),$$

   which must exist by the previous step, and let T' be the smallest element of $\mathbb{N}$ for which

   $$\max\left\{\min\left\{r_1\left(x^\circ_{T'}\right), v^*_{\text{R2}}\left(x^\circ_{T'}\right)\right\}, \min\left\{r_2\left(x^\circ_{T'}\right), v^*_{\text{R1}}\left(x^\circ_{T'}\right)\right\}\right\} = \tilde{v}^*_{\text{R}}(s),$$

   which must exist by our choice of π. We claim $T' \geq T$.

   Suppose otherwise. Since $\bar{x}_t = x^\circ_t$ for all $t \leq T$, then in particular $\bar{x}_{T'} = x^\circ_{T'}$, so that

   $$\max\left\{\min\left\{r_1\left(\bar{x}_{T'}\right), v^*_{\text{R2}}\left(\bar{x}_{T'}\right)\right\}, \min\left\{r_2\left(\bar{x}_{T'}\right), v^*_{\text{R1}}\left(\bar{x}_{T'}\right)\right\}\right\} = \tilde{v}^*_{\text{R}}(s).$$

   But then

   $$\max\{r_1(\bar{x}_{T'}), r_2(\bar{x}_{T'})\} \geq \tilde{v}^*_{\text{R}}(s) \geq \tilde{v}^*_{\text{R}}(\bar{x}_{T'}).$$

   By our choice of T, we then have $T \leq T'$, creating a contradiction.

3. It follows from the previous step that

   $$\tilde{v}^*_{\text{R}}(\bar{x}_T) = \tilde{v}^*_{\text{R}}(x^\circ_T) = \tilde{v}^*_{\text{R}}(s).$$

   By our choice of T, there are two cases: $r_1(\bar{x}_T) \geq \tilde{v}^*_{\text{R}}(\bar{x}_T)$ and $r_2(\bar{x}_T) \geq \tilde{v}^*_{\text{R}}(\bar{x}_T)$. We assume the first case and prove the desired result, with case two following identically. To reach a contradiction, assume

   $$r_2(\bar{x}_t) < \tilde{v}^*_{\text{R}}(\bar{x}_T) \quad \forall t \in \mathbb{N}.$$

But then $\bar{\pi}(\bar{x}_t, \bar{y}_t, \bar{z}_t) = \theta_2(\bar{x}_t)$ for all $t \geq T$, so $v^*_{\text{R2}}(\bar{x}_T) = \max_{t \geq T} r_2(\bar{x}_t) < \tilde{v}^*_{\text{R}}(\bar{x}_T) \leq \tilde{v}^*_{\text{R}}(s)$. Thus $r_2(x^\circ_{T'}) \leq v^*_{\text{R2}}(x^\circ_{T'}) \leq v^*_{\text{R2}}(x^\circ_T) = v^*_{\text{R2}}(\bar{x}_T) < \tilde{v}^*_{\text{R}}(s)$. It follows that

$$\max\left\{\min\left\{r_1\left(x^\circ_{T'}\right), v^*_{\text{R2}}\left(x^\circ_{T'}\right)\right\}, \min\left\{r_2\left(x^\circ_{T'}\right), v^*_{\text{R1}}\left(x^\circ_{T'}\right)\right\}\right\} < \tilde{v}^*_{\text{R}}(s),$$

contradicting our choice of T'.

Thus $r_2(\bar{x}_t) \geq \tilde{v}^*_{\text{R}}(\bar{x}_T) = \tilde{v}^*_{\text{R}}(s)$ for some $t \in \mathbb{N}$ and also $r_1(\bar{x}_T) \geq \tilde{v}^*_{\text{R}}(\bar{x}_T) = \tilde{v}^*_{\text{R}}(s)$, so that (36) must hold by Lemma 9.

□

Corollary 8. *For all $s \in \mathcal{S}$, we have $V^*_{\text{RR}}(s, r_1(s), r_2(s)) = v^*_{\text{RR}}(s)$.*

Proof of Theorem 2. The proof of this theorem immediately follows from the previous corollary together with Corollary 7 and Lemma 9. □

C Proof of Optimality Theorem

Proof of Theorem 3. The inequalities in both lines of the theorem follow from the fact that for each $\pi \in \Pi$, we can define a corresponding augmented policy $\bar{\pi} \in \overline{\Pi}$ by

$$\bar{\pi}(s, y, z) = \pi(s) \quad \forall s \in \mathcal{S}, y \in \mathcal{Y}, z \in \mathcal{Z},$$

in which case $V^\pi_{\text{RAA}}(s) = V^{\bar{\pi}}_{\text{RAA}}(s)$ and $V^\pi_{\text{RR}}(s) = V^{\bar{\pi}}_{\text{RR}}(s)$ for each $s \in \mathcal{S}$. Note that in general, we cannot define a corresponding policy for each augmented policy, so the reverse inequality does not generally hold (see Figure 3 for intuition regarding this fact).

The equalities in both lines of the theorem are simply restatements of Lemma 5 and Lemma 9. □

D The SRABE and its Policy Gradient

Proof of Proposition 1. We here closely follow the proof of Theorem 3 in [4], which itself modifies the proofs of the Policy Gradient Theorems in Chapter 13.2 and 13.6 [52]. We only make the minimal modifications required to adapt the PPO algorithm developed previously for the SRBE to on for the SRABE.

$$\begin{aligned}
\nabla_\theta \tilde{V}^{\pi_\theta}_{\text{RAA}}(s) =& \nabla_\theta \left(\sum_{a \in \mathcal{A}} \pi_\theta(a|s) \tilde{Q}^{\pi_\theta}_{\text{RAA}}(s,a) \right) \\
=& \sum_{a \in \mathcal{A}} \Big(\nabla_\theta \pi_\theta(a|s) \tilde{Q}^{\pi_\theta}_{\text{RAA}}(s,a) \\
& \qquad + \pi_\theta(a|s) \nabla_\theta \min\left\{ \max\left\{ \tilde{V}^\pi_{\text{RAA}}\left(f(s,a)\right), r_{\text{RAA}}(s) \right\}, q(s) \right\} \Big) \\
=& \sum_{a \in \mathcal{A}} \Big(\nabla_\theta \pi_\theta(a|s) \tilde{Q}^{\pi_\theta}_{\text{RAA}}(s,a) \\
& \qquad + \pi_\theta(a|s) \left[q(s) < \tilde{V}_{\text{RAA}}\left(f(s,a)\right) < r_{\text{RAA}}(s) \right] \nabla_\theta \tilde{V}^\pi_{\text{RAA}}\left(f(s,a)\right) \Big) \qquad (37) \\
=& \sum_{s' \in \mathcal{S}} \left[\left(\sum_{k=0}^\infty \Pr(s \to s', k, \pi) \right) \sum_{a \in \mathcal{A}} \nabla_\theta \pi_\theta(a|s') \tilde{Q}^{\pi_\theta}_{\text{RAA}}(s',a) \right] \qquad (38) \\
=& \sum_{s' \in \mathcal{S}} \left[\left(\sum_{k=0}^\infty \Pr(s \to s', k, \pi) \right) \sum_{a \in \mathcal{A}} \pi_\theta(a|s') \frac{\nabla_\theta \pi_\theta(a|s')}{\pi_\theta(a|s')} \tilde{Q}^{\pi_\theta}_{\text{RAA}}(s',a) \right] \\
=& \sum_{s' \in \mathcal{S}} \left[\left(\sum_{k=0}^\infty \Pr(s \to s', k, \pi) \right) \mathbb{E}_{a \sim \pi_\theta(s')} \left[\nabla_\theta \ln \pi_\theta(a|s') \tilde{Q}^{\pi_\theta}_{\text{RAA}}(s',a) \right] \right] \\
\propto& \mathbb{E}_{s' \sim d'_\pi(s)} \mathbb{E}_{a \sim \pi_\theta(s')} \left[\nabla_\theta \ln \pi_\theta(a|s') \tilde{Q}^{\pi_\theta}_{\text{RAA}}(s',a) \right],
\end{aligned}$$

where the equality between (37) and (38) comes from rolling out the term $\nabla_\theta \tilde{V}^\pi_{\text{RAA}}(f(s,a))$ (see Chapter 13.2 in [52] for details), and where $\Pr(s \to s', k, \pi)$ is the probability that under the policy π, the system is in state s' at time k given that it is in state s at time 0. □

Note, Proposition 1 is vital to updating the actor in Algorithm 1.

E The DO-HJ-PPO Algorithm

In this section, we outline the details of our Actor-Critic algorithm DO-HJ-PPO beyond the details given in Algorithm 1.

Algorithm 1 : DO-HJ-PPO (Actor-Critic)

Require: Composed and Decomposed Actor parameters θ and θ_i, Composed and Decomposed Critic parameters ω and ω_i, GAE λ, learning rate β_k and discount factor γ. Let B^γ amd B^γ_i represent the Bellman update and decomposed Bellman update for the users choice of problem (RR or RAA).

Define *Composed* Actor and Critic $\tilde{Q}$
Define *Decomposed* Actor(s) and Critic(s) $\tilde{Q}_i$
for $k = 0, 1, \ldots$ **do**
  for $t = 0$ to $T - 1$ **do**
    Sample trajectories for $\tau_t : \{\hat{s}_t, a_t, \hat{s}_{t+1}\}$
    Define $\tilde{\ell}(s_t)$ with Decomposed Critics $\tilde{Q}_i(s_t)$ (Theorems 1 & 2)
    Composed Critic update:

$$\omega \leftarrow \omega - \beta_k \nabla_\omega \tilde{Q}(\tau_t) \cdot \left(\tilde{Q}(\tau_t) - B^\gamma[\tilde{Q}, \tilde{r}](\tau_t)\right)$$

    Compute Bellman-GAE A^λ_{HJ} with B^γ
    (Standard) update Composed Actor
    Decomposed Critic update(s):

$$\omega \leftarrow \omega - \beta_k \nabla_\omega \tilde{Q}_i(\tau_t) \cdot \left(\tilde{Q}_i(\tau_t) - B^\gamma_i[\tilde{Q}_i](\tau_t)\right)$$

    Compute Bellman-GAE A^λ_i with B^γ_i
    (Standard) update Decomposed Actor(s)
  end for
end for
return parameter θ, ω

In Algorithm 1, the Bellman update $B^\gamma[\tilde{Q}, \tilde{r}]$ differs for the RAA task and RR task, and the $B^\gamma_i[\tilde{Q}]$ differs between the reach, avoid, and reach-avoid tasks. These Bellman updates are explicitly specified in the Supplementary Material.

E.1 DO-HJ-PPO Stochastic Relaxation

Per the assumptions made in the relaxation, the discounted contractions for the RAA (and similarly RR) take the following form,

$$\tilde{V}^{\gamma,\pi}_{\text{RAA}}(s) = (1-\gamma)\min\{r_{\text{RAA}}(s), q(s)\} + \gamma \mathbb{E}_{a\sim\pi}\left[\min\left\{\max\left\{\tilde{V}^{\gamma,\pi}_{\text{RAA}}(f(s,a)), r_{\text{RAA}}(s)\right\}, q(s)\right\}\right].$$

$$\tilde{Q}^{\gamma,\pi}_{\text{RAA}}(s,a) = (1-\gamma)\min\{r_{\text{RAA}}(s), q(s)\} + \gamma \min\left\{\max\left\{\tilde{V}^{\gamma,\pi}_{\text{RAA}}(f(s,a)), r_{\text{RAA}}(s)\right\}, q(s)\right\}.$$

The PPO advantage function is then given by $\tilde{A}^\pi_{\text{RAA}} = \tilde{Q}_{\text{RAA}} - \tilde{V}_{\text{RAA}}$ [53].

E.2 The special Bellman updates and the corresponding GAEs

Akin to previous HJ-RL policy algorithms, namely RCPO [6], RESPO [3] and RCPPO [4], DO-HJ-PPO fundamentally depends on the discounted HJ Bellman updates [1]. To solve the RAA and RR problems with the special rewards defined in Theorems 1 & 2, DO-HJ-PPO utilizes the Reach, Avoid and Reach-Avoid Bellman updates, given by

$$B_R^\gamma[Q \mid r](s,a) = (1-\gamma)r(s) + \gamma \max\{r(s), Q(s,a)\}, \tag{39}$$

$$B_A^\gamma[Q \mid q](s,a) = (1-\gamma)q(s) + \gamma \min\{q(s), Q(s,a)\}, \tag{40}$$

$$B_{RA}^\gamma[Q \mid r,q](s,a) = (1-\gamma)\min\{r(s), q(s)\} + \gamma \min\{q(s), \max\{r(s), Q(s,a)\}\}. \tag{41}$$

To improve our algorithm, we incorporate the Generalized Advantage Estimate corresponding to these Bellman equations in the updates of the Actors. As outlined in Section A of [4], the GAE may be defined with a reduction function corresponding to the appropriate Bellman function which will be applied over a trajectory roll-out. We generalize the Reach GAE definition given in [4] to propose a Reach-Avoid GAE (the Avoid GAE is simply the flip of the Reach GAE) as all will be used in DO-HJ-PPO algorithm for either RAA or RR problems. Consider a reduction function $\phi_{RA}^{(n)} : \mathbb{R}^n \to \mathbb{R}$, defined by

$$\phi_{RA}^{(n)}(x_1, x_2, x_3, \ldots, x_{2n+1}) = \phi_{RA}^{(1)}(x_1, x_2, \phi_{RA}^{(n-1)}(x_3, \ldots, x_{2n+1})), \tag{42}$$

$$\phi_{RA}^{(1)}(x,y,z) = (1-\gamma)\min\{x,y\} + \gamma \min\{y, \max\{x,z\}\}. \tag{43}$$

The k-step Reach-Avoid Bellman advantage $A_{RA}^{\pi(k)}$ is then given by,

$$A_{RA}^{(k)}(s) = \phi_{RA}^{(n)}\left(r(s_t), q(s_t), \ldots, r(s_{t+k-1}), q(s_{t+k-1}), V^(s_{t+k})\right) - V^(s_{t+k}). \tag{44}$$

We may then define the Reach-Avoid GAE A_{RA}^λ as the λ-weighted sum over the advantage functions

$$A_{RA}^\lambda(s) = \frac{1}{1-\lambda}\sum_{k=1}^{\infty} \lambda^k A_{RA}^{(k)}(s) \tag{45}$$

which may be approximated over any finite trajectory sample. See [4] for further details.

E.3 Modifications from standard PPO

To address the RAA and RR problems, DO-HJ-PPO introduces several key modifications to the standard PPO framework [53]:

Additional actor and critic networks are introduced to represent the decomposed objectives. Rather than learning the decomposed objectives separately from the composed objective, DO-HJ-PPO optimizes all objectives simultaneously. This design choice is motivated by two primary factors: (i) simplicity and minor computational speed-up, and (ii) coupling between the decomposed and composed objectives during learning.

The decomposed trajectories are initialized using states sampled from the composed trajectory, we refer to as *coupled resets*.
While it is possible to estimate the decomposed objectives independently—i.e., prior to solving the composed task—this approach might lead to inaccurate or irrelevant value estimates in on-policy settings. For example, in the RAA problem, the decomposed objective may prioritize avoiding penalties, while the composed task requires reaching a reward region without incurring penalties. In such a case, a decomposed policy trained in isolation might converge to an optimal strategy within a reward-irrelevant region, misaligned with the overall task. Empirically, we observe that omitting coupled resets causes DO-HJ-PPO to perform no better than standard baselines such as CPPO, whereas their inclusion significantly improves performance.

The special RAA and RR rewards are defined using the decomposed critic values and updated using their corresponding Bellman equations.

This procedure is directly derived from our theoretical results (Theorems 1 and 2), which establish the validity of using modified rewards within the respective RA and R Bellman frameworks. These rewards are used to compute the composed critic target as well as the actor's GAE. In Algorithm 1, this process is reflected in the critic and actor updates corresponding to the composed objective.

F DDQN Demonstration

As described in the paper, we demonstrate the novel RAA and RR problems in a 2D Q-learning problem where the value function may be observed easily. We juxtapose these solitons with those of the previously studied RA and R problems which consider more simple objectives. To solve all values, we employ the standard Double-Deep Q learning approach (DDQN) [54] with only the special Bellman updates.

F.1 Grid-World Environment

The environment is taken from [2] and consists of two dimensions, $s = (x, y)$, and three actions, $a \in \{\text{left}, \text{straight}, \text{right}\}$, which allow the agent to maneuver through the space. The deterministic dynamics of the environment are defined by constant upward flow such that,

$$f((x_i, y_i), a_i) = \begin{cases} (x_{i-1}, y_{i+1}) & a_i = \text{left} \\ (x_i, y_{i+1}) & a_i = \text{straight} \\ (x_{i+1}, y_{i+1}) & a_i = \text{right} \end{cases} \tag{46}$$

and if the agent reaches the boundary of the space, defined by $x \geq |2|$, $y \leq -2$ and $y \geq 10$, the trajectory is terminated. The 2D space is divided into 80×120 cells which the agent traverses through.

In the RA and RAA experiments, the reward function r is defined as the negative signed-distance function to a box with dimensions $(x_c, y_c, w, h) = (0, 4.5, 2, 1.5)$, and thus is negative iff the agent is outside of the box. The penalty function q is defined as the minimum of three (positive) signed distance functions for boxes defined at $(x_c, y_c, w, h) = (\pm 0.75, 3, 1, 1)$ and $(x_c, y_c, w, h) = (0, 6, 2.5, 1)$, and thus is positive iff the agent is outside of all boxes.

In the R and RR experiments, one or two rewards are used. In the R experiment, the reward function r is defined as the maximum of two negative signed-distance function of boxes with dimensions $(x_c, y_c, w, h) = (\pm 1.25, 0, 0.5, 2)$, and thus is negative iff the agent is outside of both boxes. In the RR experiment, the rewards r_1 and r_2 are defined as the negative signed distance functions of the same two boxes independently, and thus are positive if the agent is in one box or the other respectively.

F.2 DDQN Details

As per our theoretical results in Theorems 1 and 2, we may now perform DDQN to solve the RAA and RR problems with solely the previously studied Bellman updates for the RA [2] and R problems [1]. We compare these solutions with those corresponding to the RA and R problems *without* the special RAA and RR targets, and hence solve the previously posed problems. For all experiments, we employ the same adapted algorithm as in [2], with no modification of the hyper-parameters given in Table 1.

G Baselines

In both RAA and RR problems, we employ Constrained PPO (CPPO) [8] as the major baseline as it can handle secondary objectives which are reformulated as constraints. The algorithm was not designed to minimize its constraints necessarily but may do so in attempting to satisfy them. As a novel direction in RL, few algorithms have been designed to optimize max/min accumulated costs and thus CPPO serves as the best proxy. Below we also include a naively decomposed STL algorithm to offer some insight into direct approaches to optimizing the max/min accumulated reward.

Table 1: Hyperparameters for DDQN Grid World

DDQN hyperparameters	**Values**
Network Architecture	MLP
Numbers of Hidden Layers	2
Units per Hidden Layer	100, 20
Hidden Layer Activation Function	tanh
Optimizer	Adam
Discount factor γ	0.9999
Learning rate	1e-3
Replay Buffer Size	1e5 transitions
Replay Batch Size	100
Train-Collect Interval	10
Max Updates	4e6

G.1 CPPO Baselines

Although CPPO formulations do not directly consider dual-objective optimization, the secondary objective in RAA (avoid penalty) or overall objective in RR (reach both rewards) may be transformed into constraints to be satisfied of a surrogate problem. For the RAA problem, this may be defined as

$$\max_\pi \mathbb{E}_\pi \left[\sum_t^\infty \gamma^t \max_{t' \leq t} r(s_{t'}^\pi) \right] \quad \text{s.t.} \quad \min_t q(s_t^\pi) \geq 0. \tag{47}$$

For the RR problem, one might propose that the fairest comparison would be to formulate the surrogate problem in the same fashion, with achievement of both costs as a constraint, such that

$$\max_\pi \mathbb{E}_\pi \left[\sum_t^\infty \gamma^t \min \left\{ \max_{t' \leq t} r_1(s_{t'}^\pi), \max_{t' \leq t} r_2(s_{t'}^\pi) \right\} \right] \quad \text{s.t.} \quad \min \left\{ \max_t r_1(s_t^\pi), \max_t r_2(s_t^\pi) \right\} \geq 0, \tag{48}$$

which we define as variant 1 (CPPO-v1). Empirically, however, we found this formulation to be the poorest by far, perhaps due to the abundance of the non-smooth combinations. We thus also compare with more naive formulations which relax the outer minimizations to summation in the reward

$$\max_\pi \mathbb{E}_\pi \left[\sum_t^\infty \gamma^t \max_{t' \leq t} r_1(s_{t'}^\pi) + \max_{t' \leq t} r_2(s_{t'}^\pi) \right] \quad \text{s.t.} \quad \min \left\{ \max_t r_1(s_t^\pi), \max_t r_2(s_t^\pi) \right\} \geq 0, \tag{49}$$

which we define as variant 2 (CPPOv2), and additionally, in the constraint

$$\max_\pi \mathbb{E}_\pi \left[\sum_t^\infty \gamma^t \max_{t' \leq t} r_1(s_{t'}^\pi) + \max_{t' \leq t} r_2(s_{t'}^\pi) \right] \quad \text{s.t.} \quad \max_t r_1(s_t^\pi) + \max_t r_2(s_t^\pi) \geq 0, \tag{50}$$

which we define as variant 3 (CPPOv3). This last approach, although naive and seemingly unfair, vastly outperforms the other variants in the RR problem.

G.2 STL Baselines

In contrast with constrained optimization, one might also incorporate the STL methods, which in the current context simply decompose and optimize the independent objectives. For the RAA problem, the standard RA solution serves as a trivial STL baseline since we may attempt to continuously attempt to reach the solution while avoiding the obstacle. In the RR case, we define a decomposed STL baseline (DSTL) which naively solves both R problems, and selects the one with lower value to achieve first.

H Details of RAA & RR Experiments: `Hopper`

The Hopper environment is taken from Gym [55] and [4]. In both RAA and RR problems, we define rewards and penalties based on the position of the Hopper head, which we denote as (x, y) in this section.

In the RAA task, the reward is defined as

$$r(x, y) = \sqrt{\|x - 2\| + |y - 1.4\|} - 0.1 \tag{51}$$

to incentive the Hopper to reach its head to the position at $(x, y) = (2, 1.4)$. The penalty q is defined as the minimum of signed distance functions to a ceiling obstacle at $(1, 0)$, wall obstacles at $x > 2$ and $x < 0$ and a floor obstacle at $y < 0.5$. In order to safely arrive at high reward (and always avoid the obstacles), the Hopper thus must pass under the ceiling and not dive or fall over in the achievement of the target, as is the natural behavior.

In the RR task, the first reward is defined again as

$$r_1(x, y) = \sqrt{\|x - 2\| + |y - 1.4\|} - 0.1 \tag{52}$$

to incentive the Hopper to reach its head to the position at $(x, y) = (2, 1.4)$, and the second reward as

$$r_2(x, y) = \sqrt{\|x - 0\| + |y - 1.4\|} - 0.1 \tag{53}$$

to incentive the Hopper to reach its head to the position at $(x, y) = (0, 1.4)$. In order to achieve both rewards, the Hopper must thus hop both forwards and backwards without crashing or diving.

In all experiments, the Hopper is initialized in the default standing posture at a random $x \in [0, 2]$ so as to learn a position-agnostic policy. The DO-HJ-PPO parameters used to train these problems can be found in Table 2.

Table 2: Hyperparameters for Hopper Learning

Hyperparameters for DO-HJ-PPO	**Values**
Network Architecture	MLP
Units per Hidden Layer	256
Numbers of Hidden Layers	2
Hidden Layer Activation Function	tanh
Entropy coefficient	Linear Decay 1e-2 $\rightarrow$ 0
Optimizer	Adam
Discount factor γ	Linear Anneal 0.995 $\rightarrow$ 0.999
GAE lambda parameter	0.95
Clip Ratio	0.2
Actor Learning rate	Linear Decay 3e-4 $\rightarrow$ 0
Reward/Cost Critic Learning rate	Linear Decay 3e-4 $\rightarrow$ 0
Add'l Hyperparameters for CPPO	
K_P	1
K_I	1e-4
K_D	1

I Details of RAA & RR Experiments: `F16`

The F16 environment is taken from [4], including a F16 fighter jet with a 26 dimensional observation. The jet is limited to a flight corridor with up to 2000 relative position north (x_{PN}), 1200 relative altitude (x_H), and ± 500 relative position east (x_{PE}).

In the RAA task, the reward is defined as

$$r(x, y) = \frac{1}{5}|x_{PN} - 1500| - 50 \tag{54}$$

to incentivize the F16 to fly through the geofence defined by the vertical slice at 1500 relative position north. The penalty q is defined as the minimum of signed distance functions to geofence (wall) obstacles at $x_{PN} > 2000$ and $|x_{PE}| > 500$ and a floor obstacle at $x_H < 0$. In order to safely arrive at high reward (and always avoid the obstacles), the F16 thus must fly through the target geofence and then evade crashing into the wall directly in front of it.

In the RR task, the rewards are defined as

$$r_1(x_{PN}, x_H) = \frac{1}{5}\sqrt{\|x_{PN} - 1250\| + |y - 850\|} - 30 \tag{55}$$

and

$$r_2(x_{PN}, x_H) = \frac{1}{5}\sqrt{\|x_{PN} - 1250\| + |y - 350\|} - 30 \tag{56}$$

to incentive the F16 to reach both low and high-altitude horizontal cylinders. In order to achieve both rewards, the F16 must thus aggressively pitch, roll and yaw between the two targets.

In all experiments, the F16 is initialized with position $x_{PN} \in [250, 750], x_H \in [300, 900], x_{PE} \in [-250, 250]$ and velocity in $v \in [200, 450]$. Additionally, the roll, pitch, and yaw are initialized with $\pm\pi/16$ to simulate a variety of approaches to the flight corridor. Further details can be found in [4]. The DO-HJ-PPO parameters used to train these problems can be found in Table 3.

Table 3: Hyperparameters for F16 Learning

Hyperparameters for DO-HJ-PPO	**Values**
Network Architecture	MLP
Units per Hidden Layer	256
Numbers of Hidden Layers	2
Hidden Layer Activation Function	tanh
Entropy coefficient	Linear Decay 1e-2 $\rightarrow$ 0
Optimizer	Adam
Discount factor γ	Linear Anneal 0.995 $\rightarrow$ 0.999
GAE lambda parameter	0.95
Clip Ratio	0.2
Actor Learning rate	Linear Decay 1e-3 $\rightarrow$ 0
Reward/Cost Critic Learning rate	Linear Decay 1e-3 $\rightarrow$ 0
Add'l Hyperparameters for CPPO	
K_P	1
K_I	1e-4
K_D	1

J Broader Impacts

This paper touches on advancing fundamental methods for Reinforcement Learning. In particular, this work falls into the class of methods designed for Safe Reinforcement Learning. Methods in this class are primarily intended to prevent undesirable behaviors in virtual or cyber-physical systems, such as preventing crashes involving self-driving vehicles or potentially even unacceptable speech among chatbots. It is an unfortunate truth that safe learning methods can be repurposed for unintended use cases, such as to prevent a malicious agent from being captured, but the authors do not foresee the balance of potential beneficial and malicious applications of this method to be any greater than other typical methods in Safe Reinforcement Learning.

K Acknowledgments

This section has been redacted for the purpose of anonymous review.