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

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

A new and interesting direction in Reinforcement Learning (RL) has been the development and use of special Bellman equations to solve for novel problem formulations. For example, 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 reward/worst penalty 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], relevant to fundamental and practical problems in many application domains.

In this work, we advance the existing HJ-RL formulations to a broader class of problems. To date, the HJ-RL Bellman equations are limited to: 1) Reach, wherein the agent seeks to reach a goal (i.e. achieve a reward threshold), 2) Avoid, wherein the agent seeks to avoid an obstacle (i.e. avoid a penalty threshold), and 3) Reach-Avoid, a combination where the agent reaches a goal while avoiding obstacles in the process. 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 corresponding to the Reach-Always-Avoid and Reach-Reach problems.
- We provide theoretical results showing that these novel value functions and their optimal policies can be decomposed into reach, avoid, and reach-avoid value functions.
- We leverage previous work on PPO-based algorithms for learning HJ-RL value functions and their policies to design a novel PPO-based algorithm DO-HJ-PPO for solving the Reach-Always-Avoid and Reach-Avoid problems.
- We demonstrate the performance of both a simpler Q-learning algorithm and DO-HJ-PPO on several experimental environments compared to baselines.

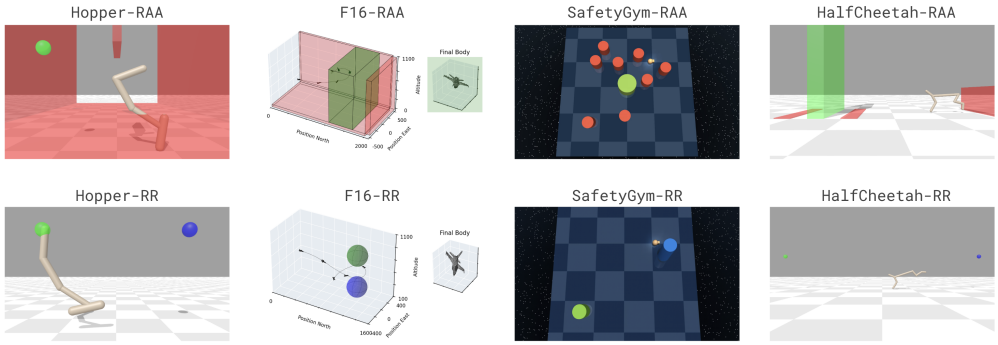

Figure 1: **Depiction of the Reach-Always-Avoid (RAA) and Reach-Reach (RR) Tasks** In the RAA tasks, the zero-level set of the rewards (goals) and penalties (obstacles) are depicted in **green** and **red** respectively, while in the RR problem, the zero-level set of the two rewards (two goals) are depicted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

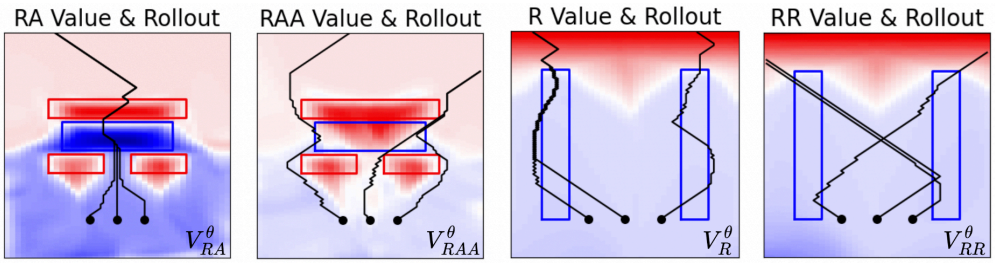

Figure 2: **DDQN Grid-World Demonstration of the RAA & RR Problems** We compare our novel formulations with previous HJ-RL formulations (RA & R) in a simple grid-world problem with DDQN. The zero-level sets of q (*hazards*) are highlighted in **red**, those of r (*goals*) in **blue**, and trajectories 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^π_t, i.e. $s^\pi_{t+1} = f\left(s^\pi_t, \pi\left(s^\pi_t\right)\right)$.

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.

$$(\text{RAA}) \begin{cases} \text{maximize (w.r.t. } \pi) & \min\left\{\max_t r(s^\pi_t),\ \min_t q(s^\pi_t)\right\} \\ \text{s.t.} & s^\pi_{t+1} = f\left(s^\pi_t, \pi\left(s^\pi_t\right)\right), \\ & s^\pi_0 = s, \end{cases}$$

$$(\text{RR}) \begin{cases} \text{maximize (w.r.t. } \pi) & \min\left\{\max_t r_1(s^\pi_t),\ \max_t r_2(s^\pi_t)\right\} \\ \text{s.t.} & s^\pi_{t+1} = f\left(s^\pi_t, \pi\left(s^\pi_t\right)\right), \\ & s^\pi_0 = s. \end{cases}$$

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},

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

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.

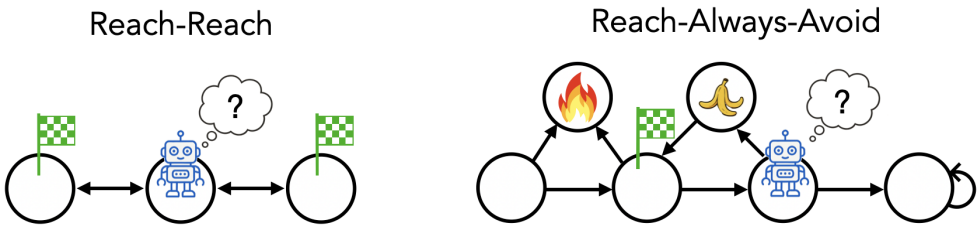

Figure 3: **Examples where a Non-Augmented Policy is Flawed** In both MDPs, consider an agent with no memory. (Left) For a deterministic policy based on the current state, the agent can only achieve one target (RR), as this policy must associate the middle state with either of the two possible actions. (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 $s \in \mathcal{S}$,*

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

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

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

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 $s \in \mathcal{S}$,*

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

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*

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

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.

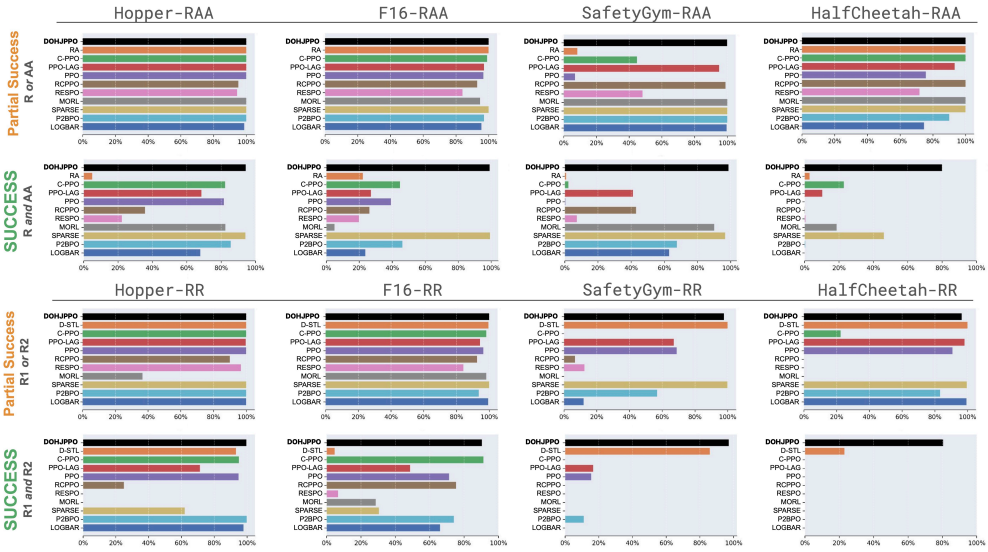

Figure 4: **Success ($\rightarrow$) and Partial Success ($\rightarrow$) in RAA and RR Tasks for DO-HJ-PPO and Baselines** We evaluate our method **DO-HJ-PPO in black** against relevant baselines over 1,000 trajectories (with random initial conditions) for both the Reach-Avoid-Avoid (RAA) and Reach-Reach (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.