Explaining RL Agents with Diverse Near-Optimal Alternatives plus Guarantees

Noel Brindise NBRINDI2@ILLINOIS.EDU
Vijeth Hebbar VHEBBAR2@ILLINOIS.EDU
Riya Shah RIYAHS3@ILLINOIS.EDU
Cedric Langbort LANGBORT@ILLINOIS.EDU
Dept. of Aerospace Engineering, University of Illinois Urbana-Champaign, Urbana, USA

Abstract

In this work, we present a new approach to explainable Reinforcement Learning called Diverse Near-Optimal Alternatives (DNA). DNA seeks a set of reasonable "options" for trajectory-planning agents, optimizing policies to produce qualitatively diverse trajectories in Euclidean space. In the spirit of explainability, these distinct policies are used to "explain" an agent's options in terms of available trajectory shapes from which a human user may choose. In particular, DNA applies to value function-based policies on Markov decision processes where agents are limited to continuous trajectories. Here, we describe DNA, which uses reward shaping in local, modified Q-learning problems to solve for distinct policies with guaranteed ϵ-optimality. We show that it successfully returns qualitatively different policies that constitute meaningfully different "options" in simulation, including a brief comparison to related approaches in the stochastic optimization field of Quality Diversity.

Keywords: Explainable Reinforcement Learning, Explainable AI for Planning, Q-Learning, Quality Diversity

1. Introduction

The field of explainable AI, which seeks to explain AI outputs and behavior to human users, is remarkably eclectic. Though commonly associated with the interpretation of neural networks, xAI encompasses many applications in explainable planning (XAIP) as well, where "explanations" describe plans, trajectories, or policies to characterize the intent or behavior of autonomous agents. In this work, we consider the common problem of RL agents on a Markov decision process (MDP). While these agents typically seek one optimal policy, we suggest that alternative policies may be of interest, particularly those with distinct behaviors but similar expected cost to the global optimum: an *explanation via alternatives*.

Take the example of an optimal route plan for a ground vehicle. If the human operator is dissatisfied with this plan (suppose it passes through risky terrain or an unwanted waypoint), alternative options help them to understand which other courses of action are available. Alternatives may also illuminate the flexibility of a plan. An "inflexible" plan may traverse states where there are limited reasonable choices of action, such as when the vehicle is confined to a valley or moving along the edge of a cliff; the latter case has been partly addressed by a notion of "critical states," where different actions have drastically different effects on cost (Huang et al. (2018)). In contrast, a plan may be called "flexible" if distinct, reasonable policies are available, such as when multiple roads lead to the same place in similar time.

In this work, we pursue an explanatory method which searches for diverse, near-optimal policies in a trajectory-planning setting. Given an agent in a particular state, we aim to answer questions such as "what policies/paths can I reasonably take from here?" and "how will the cost compare?" We specifically consider the case of **continuous trajectories in Euclidean space** and introduce "corridors," connected subsets of the state space which originate at an initial state and are allowed to propagate outward. Our novel method applies to reinforcement learning agents operating on a **value function**, as we leverage the optimal policy to assess candidate alternatives. A novel reward-shaping scheme rewards agents for traversing a corridor and penalizes them for leaving its boundaries, incentivizing trajectories to take on distinct shapes. Usefully, the continuity of trajectories means that any corridor with an $\epsilon-$suboptimal policy may be truncated by the optimality principle, allowing for dynamic bounding of the search space. The resulting policy "options" and their trajectory shapes are subject to provable guarantees.

In this paper, Section 2 gives an overview of the state of the art in Explainable Reinforcement Learning and Explainable AI Planning as it relates to our approach. Section 3 provides theoretical background, and Section 4 describes our proposed algorithm. A simple illustrative experiment is discussed in Section 5 in which a Q-learning agent is applied to a tabular environment. We also compare our method to related approaches from the field of Quality Diversity, which we briefly introduce in the next section.

2. Previous Work

Explainable Reinforcement Learning (XRL) and Explainable AI Planning (XAIP) improve human understanding of autonomous system behavior. While these fields have yet to identify unified goals or metrics for "explainability," distinct branches have emerged (Milani et al. (2022), Chakraborti et al. (2020), Vouros (2022)). At the highest level, *interpretable design* approaches (re)construct the RL agent itself to be inherently more explainable, while *post-hoc* methods interpret the existing model. Post-hoc explanation takes many forms. Approaches may track or summarize agent behavior based on trajectory observations (Brindise and Langbort (2023), Movin et al. (2023)), "highlight" important moments (Pierson et al. (2023)), or seek causal relationships between variables (Madumal et al. (2020)). Others assess the influence of trajectories on success probability (Cruz et al. (2019)) or policy shape (Deshmukh et al. (2022)). Explanations also exist for multiagent settings (Heuillet et al. (2022)).

In many approaches, a human initially suggests an "alternative" policy or trajectory. The explanation may then highlight infeasible trajectory segments (Alsheeb and Brandão (2023)) or suggest environment changes to enable the suggestion (Brandão and Setiawan (2022), Finkelstein et al. (2022)). Reward shaping has also been used to incentivize an RL agent to visit desired waypoints (Movin et al. (2023), Beyret et al. (2019)). However, explanations which **seek and offer multiple policy suggestions** are rare in XAIP/XRL. Navigation tools have long provided sets of possible routes, making a case for the usefulness of an options search, but the typical A^* and Dijkstra algorithms used here are limited to the graph-traversal setting. Instead, it is worthwhile to consider a recent branch of stochastic optimization called Quality Diversity (QD).

QD seeks a set of high-performing ("quality") policies which are behaviorally distinct ("diversity"); approaches aim to populate a *behavior* or *feature* space with the best performers, categorizing policies based on where their solutions fit in an "archive" (Chatzilygeroudis et al. (2021); Pugh et al. (2016)). This approach is introduced in the evolutionary algorithm MAP-Elites, which uses a **behavior descriptor** function to classify solutions (Mouret and Clune (2015)). This is generally human-defined to capture features of interest, e.g., landing speed and impact location of a lander, or duration and frequency of propulsive burns in space mission design. In a deterministic environment, each control policy corresponds to a single trajectory and therefore one behavioral category in the solution archive. In the stochastic setting, however, the one-to-one policy-to-trajectory connection disappears. New work in QD has improved optimization efficiency and coverage of the behavior space (Salimans et al. (2017)), but stochastic environments remain a significant stumbling block (Flageat and Cully (2023)).

Our work approaches the particular case of distinct **spatially-continuous trajectories** from a new direction. Our novel method of **reward shaping** defines local optimization problems which encourage trajectories to remain within their distinct corridors in Euclidean space. The corridors represent distinct policy options and come with a set of optimality and safety guarantees to address the uncertainty in trajectories introduced through stochasticity.

3. Background

This work provides an initial proof of concept using a simple RL agent on a **Markov decision process**, where trajectories are Lipschitz continuous with respect to the Manhattan norm (see Asadi et al. (2018)). In general, trajectories need only be continuous in a subset of the state space dimensions, i.e., diversity may be sought for the projections of trajectories onto a subset of spatial dimensions. In our application, value functions are estimated using Deep Q networks (DQN).

N.B.: superscripts denote "named" constants and subscripts denote indices in a sequence, i.e. $(s_0, s_1, s_2) = (s^a, s^b, s^a)$ means that the second state in a sequence was State s^b.

Definition 1 (Markov Decision Process) *A Markov decision process (MDP) is a tuple $\mathcal{M} = \langle S, A, T, R \rangle$, where S is a finite set of discrete states, A is a finite set of actions, $T : S \times A \times S \to \mathbb{R}$ is a stochastic transition function, and $R : S \times A \to \mathbb{R}$ is a reward function.*

We will require that $R(\cdot, \cdot) \geq 0$. Now, to simulate continuous trajectories in Euclidean space, we specifically consider an MDP on a grid, a common setting in gridworld-based RL.

Definition 2 (MDP on a Grid) *An MDP on a grid is an MDP where $S \subset \mathbb{Z}^K$ with neighbors at a point $s \in S$ defined as*

$$\mathcal{N}(s) \triangleq \{s' \in S \mid \|s' - s\|_M = 1\}$$

where $\| \cdot \|_M$ denotes the standard metropolis metric. Additionally, transition function satisfies the property that, for any state $s \in S$, $T(s, a, s') > 0$ for some $a \in A$ only if $s' \in \mathcal{N}(s)$.

We will also require an optimal value function, necessitating additional definitions:

Definition 3 (Optimal Q Function) *An optimal Q function Q^* is a mapping of state-action pairs $Q : S \times A \to \mathbb{R}$ such that*

$$Q^*(s_t, a_t) = R(s_t, a_t) + \gamma \mathbb{E}[\max_{a'} Q^*(s_{t+1}, a')]. \tag{1}$$

for all $s \in S$, where $s_t \in S$ is the state at time t and $\gamma \in [0, 1]$ is a discount factor. On an MDP with transition function T,

$$\mathbb{E}[\max_{a'} Q^*(s_{t+1}, a')] = \sum_{s' \in S} T(s_t, a_t, s') \max_{a'} Q^*(s', a') \tag{2}$$

The associated value function is then:

Definition 4 (Optimal Value Function) *The optimal value function $V^* : S \to \mathbb{R}$ is defined as*

$$V^*(s) = \max_{a \in A} Q^*(s, a). \tag{3}$$

We define a **policy** as any mapping $\pi : S \to A$. An **optimal policy** π^* must always take the action associated with the largest value of Q^* at s, i.e. $\pi^*(s) = \arg\max_{a \in A} Q^*(s, a)$ for all $s \in S$. For any policy π, we may define a generic value function V^π:

Definition 5 (Value Function for Policy π) *The value function $V^\pi : S \to \mathbb{R}$ is defined for state $s_t \in S$ as*

$$V^\pi(s_t) = R(s_t, \pi(s_t)) + \gamma \sum_{s' \in S} T(s_t, \pi(s_t), s') V^\pi(s'). \tag{4}$$

We may now move into the discussion of our proposed algorithm.

4. Diverse Near-Optimal Alternative Policies via Corridor Search

4.1. Motivation: Diverse Trajectory Shapes through Corridors

We suppose that a human user seeks **policy options** to create **distinct trajectory shapes** from an initial state. We must quantify *distinctness*. For paths in Euclidean space, waypoints are often used to describe trajectories, e.g. in aviation, but it is unclear a priori how to select such arbitrary reference points in a general setting. Moreover, stochasticity means that any policy π produces a **family** of trajectories, so a single policy cannot be associated with a waypoint sequence in a one-to-one manner.

Here, we will aim to optimize and distinguish trajectory families by first establishing an altered concept of waypoints.

1. **Waypoints → Way-regions:** rather than selecting states s individually, we select larger *way-regions* $W \subset S$ to describe and shape trajectories.

2. **Sequences of Waypoints → Corridors:** we replace a sequence of waypoints with a *corridor*, a set of way-regions which forms a connected subset of S. We will alter R outside of a corridor to incentivize trajectories to remain inside and traverse from an initial state to a terminal region.

Here, we use corridors as the quantitative description of a trajectory's shape in space, an analogous assumption to Quality Diversity's hand-selection of "feature" variables and archive granularity.

Thus motivated, we introduce *corridor search*: given a starting state s_i, we systematically check corridors connecting s_i to terminal way-regions S_Ω, optimizing local policies via reward shaping. Here, policies optimized for distinct corridors are considered qualitatively distinct from each other. The result is a set of diverse, near-optimal policy options, providing the human an overview of the choices from s_i and, by extension, the flexibility of the planning situation.

4.2. Definitions

Given an agent on MDP $\mathcal{M}$ at state s_i, we seek alternative policies π which (i) have sufficiently high expected payoff from s_i and (ii) produce diverse families of trajectories. We begin with the discretization of S into *cells*. For the purpose of this paper, a cell will be a square in $\mathbb{R}^2$; in general, our guarantees will hold for cells $c \subseteq S$ taking arbitrary shapes provided they are connected and cover S completely, where the latter is defined below.

Definition 6 (Cell for Grid Case Study) *A cell centered at s' is described by*

$$c(s') = \{s \in S \mid s'[k] - d \leq s[k] \leq s'[k] + d \quad \forall k\} \tag{5}$$

for selected distance d, where $s[k]$ is the k^{th} component of s.

When the entire state space is grouped into cells, it is called *complete discretization:*

Definition 7 (Complete Discretization) *Given a discretization $\mathcal{C}$ of S into cells $c \subseteq S$, S is completely discretized if all states $s \in S$ belong to some cell $c \in \mathcal{C}$.*

A sequence of cells can then be used to construct a *corridor*:

Definition 8 (Corridor) *A corridor of length B is a sequence $C = (c_0, ..., c_B)$ of cells in $\mathcal{C}$ such that any consecutive cells c_b, c_{b+1} in C are adjacent.*

Here, two cells c_i, c_j are *adjacent* if and only if there exist some $s^1 \in c_i$, $s^2 \in c_j$ such that s^1, s^2 are neighbors by Definition 2 and $c_i \neq c_j$. Finally, at the end of a corridor, we place a *terminal edge* $S_\Omega \subset S$. In general, this may be any connected subset of states in c_B such that all states $s \in S_\Omega$ neighbor a state $s \notin c_B$. For our case study, we select one side of c_B:

Definition 9 (Terminal Edge for Grid Case Study) *For corridor $C = (c_0, ..., c_B)$, a terminal edge $S_\Omega \subset S$ is the set of all states contained in an edge of c_B, i.e., for c_B centered at s',*

$$S_\Omega = \{s \in c_B \mid s[k] - s'[k] = \alpha d\} \tag{6}$$

for a selection of k and $\alpha \in \{-1, 1\}$. The set of all terminal edges for corridor C is denoted $\mathcal{E}^C$.

Now we consider the **cost** of potential policies. From a state s_i, we consider a policy as a reasonable choice only if it satisfies the criterion for *ϵ-optimality:*

Definition 10 (ϵ-Optimal Policy) *A policy π with corresponding value function V^π is ϵ-optimal if for a given $\epsilon \geq 0$ it satisfies*

$$V^\pi(s_i) \geq V^*(s_i) - \epsilon. \tag{7}$$

By this definition, any reasonable alternative must have an expectation which is sufficiently close to the global optimum, determined by a user-defined ϵ.

4.3. Algorithm: Methodology and Guarantees

We propose local Q learning problems which incentivize policies to follow corridors. Informally, we seek policies which achieve a sufficient reward even when the system is altered such that (i) if the agent leaves the corridor at an $s \notin S_\Omega$, $R(s) = 0$ and the episode **immediately terminates**, and (ii) if the agent reaches $s \in S_\Omega$, it **receives reward** $V^*(s)$ and terminates. Formally:

Definition 11 (Local Q Problem)

For corridor $C = (c_0, ..., c_B)$ and terminal edge $S_\Omega \in \mathcal{E}^C$ the local Q problem is the problem solving for optimal local policy π_L on MDP $\mathcal{M}_L$ with $S_L = S$, $A_L = A$,

$$T_L(s,a,s') = \begin{cases} T(s,a,s') & s \in S_{in} \backslash S_\Omega \\ \mathbb{1}_{s'=s} & \text{otherwise,} \end{cases}$$

where $\mathbb{1}_E$ is indicator over event E and

$$R_L(s,a) = \begin{cases} (1-\gamma)V^*(s) & s \in S_\Omega \\ R(s,a) & s \in S_{\text{in}} \backslash S_\Omega \\ 0 & \text{otherwise.} \end{cases}$$

where interior states $S_{\text{in}} = \{s \mid s \in c, c \in C\}$.

With π_L defined, we can define a related policy on the global MDP:

Definition 12 (Alternative Policy) *Consider auxiliary state Δ_t for trajectory $\rho = (s_0, ..., s_t)$ and corridor C, where $\Delta_0 = 0$ if $s_0 \notin S_\Omega$ and $s_0 \in c$ for some $c \in C$, $\Delta_0 = 1$ otherwise; and*

$$\Delta_{t+1} = \begin{cases} 1 & s_{t+1} \notin (c \cup S_\Omega) \; \forall c \in C \\ \Delta_t & \text{otherwise.} \end{cases}$$

Then an alternative policy for corridor C, defined on augmented state $\tilde{s} = (s[0], ..., s[k], \Delta)^T$ takes the piecewise form

$$\hat{\pi}(\tilde{s}) = \begin{cases} \pi_L(\cdot) & \Delta_t = 0 \\ \pi^*(\cdot) & \text{otherwise.} \end{cases}$$

Thus, an alternative policy from s_i in a corridor follows a local policy π_L until the trajectory exits the corridor, after which it follows the global optimal policy.

Now, we recall that for an alternative policy $\hat{\pi}$ to be accepted, it must have comparable optimality to π^* in line with (7). This brings us to our first important guarantee.

Theorem 13 (ϵ-Optimality Guarantee) *Let V_L^* be the value function corresponding to the local Q-learning problem in Definition 11. Then we have*

$$V_L^*(s) \leq V^{\hat{\pi}}((s,0)^T)$$

where $(s, \Delta)^T$ denotes $\tilde{s} = (s[0], ..., s[K], \Delta)^T$ where $s \in S_{\text{in}}$ and the inequality holds pointwise.

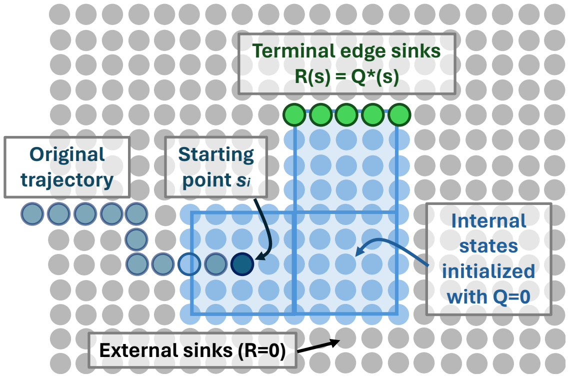

Figure 1: **(Local Q-Learning Illustrated)** Simulated corridor $|C| = 3$ with local MDP reward shaping applied.

Since our algorithm (Line 19) requires that $V_L^*(s) \geq V^*(s) - \epsilon$ for all $s \in S_{in}$, Theorem 13 allows us to claim that $V^{\hat{\pi}}((s,0)^T) \geq V^*(s) - \epsilon$ for all $s \in S_{in}$. Thus, $\epsilon-$optimality of the local policy π_L at points inside the corridor is a sufficient condition for $\epsilon-$optimality of its global alternative policy $\hat{\pi}$. For proof of Theorem 13, see the Appendix.

We can also show that, given knowledge of the global optimal policy and reward function, the probability that a trajectory remains within a corridor can be bounded from below:

Theorem 14 (Bound on Success Probability of Trajectories in Corridors) *For local value function V_L^* on a corridor with terminal side S_Ω, the probability $\mathbb{P}_{success}$ that a trajectory from s_t reaches S_Ω is bounded by*

$$\left(V_L^*(s_t) - \frac{\max(r_{in})}{1-\gamma}\right) \frac{1}{\gamma^\tau \max_{s \in S_\Omega} V^*(s)} \leq \mathbb{P}_{success} \tag{8}$$

where $\max(r_{in}) = \max_{s \in S_{in}} R(s)$, V^* is the value function for the global optimal policy, γ is the discount factor. $\tau \geq 0$ is any lower bound on the number of steps between s_t and $s \in S_\Omega$.

Here, d must be a lower bound for the shortest path from s_t to $s \in S_\Omega$. Most conservatively $\tau \geq 0$; a first-order estimate for the grid is $\tau = \min_{s \in S_\Omega} ||s_t - s||_M$. Proof is included in the Appendix.

4.4. Algorithm: Application and Complexity.

Our algorithm seeks policies by searching over a set of corridors. The size of this set is dependent on corridor length, cell dimensions, and discretization of S. In this demonstration, we apply a discretization which uniformly places square cells centered at distance $d-1$ from each other along each spatial dimension k; Figure 1 shows this in $\mathbb{R}^2$. The resulting $\mathcal{C}$ satisfies completeness by Definition 7. Our example c are n-dimensional (hyper)cubes with one adjacent c' per (hyper)face. Thus each c_b has $2k$ possible terminal edges (and $2k$ possible c_{b+1} to extend the corridor). Then, given starting state $s_i \in c_0$ and state space dimension k, there exist

$$n = \sum_{b=0}^{B} (2k)^{(b+1)} \tag{9}$$

Algorithm 1 ϵ-Optimal Alternative Policy Search

```
Require: Environment $env$, $Q^*$, $\epsilon > 0$, $s_0$, B, policy $\pi$.
 1: Initialize $corridor\_stack = \big[(c_0)\big]$
 2: Initialize $required\_corridors = \big[\big]$
 3: while $corridor\_stack$ is not empty do
 4:   $curr\_corridor \leftarrow$ pop first corridor from stack
 5:   if (length of $curr\_corridor$) $> B$ then
 6:     break
 7:   for each $terminal\_edge$ of $curr\_corridor$ do
 8:     if $\max\{Q^*(s)|s \in terminal\_edge\} < Q^*(s_0) - \epsilon$ then
 9:       continue
10:     Initialize $Q = 0$, Set $Q(s) = Q^*(s)$ if $s \in terminal\_edge$
11:     while $Q$ has not converged do
12:       Initialize $s \in curr\_corridor$
13:       while $s \in curr\_corridor$ and $s \notin terminal\_edge$ do
14:         Take step $s \rightarrow s'$ according to policy $\pi$.
15:         if $s' \notin curr\_corridor$ then
16:           Set reward 0 for step
17:         Perform Q-learning update
18:         Set $s = s'$
19:     if $Q(s_0) > Q^*(s_0) - \epsilon$ then
20:       Create cell $c'$ from $terminal\_edge$ away from $curr\_corridor$.
21:       Obtain $next\_corridor$ by appending $c'$ to $curr\_corridor$.
22:       if $c' \in curr\_corridor$ then
23:         Ignore $terminal\_edge$ when exploring terminal edges (line 7) of $next\_corridor$
24:       Push $next\_corridor$ to $corridor\_stack$
25:       if length of $next\_corridor = B$ then
26:         Push $next\_corridor$ to $required\_corridor$
27: return $required\_corridor$
```

potential local Q problems. However, continuity may mitigate complexity; note that, if the policy corresponding to $(c_0, ..., c_\beta)$ is not $\epsilon-$optimal for any terminal edge of c_β (Line 19), there can be no $\epsilon-$optimal policy for $(c_0, ..., c_{\beta+1})$; thus, the search for $C = (c_0, ..., c_\beta, ...)$ may be truncated, eliminating $\sum_{b=0}^{B-\beta}(2k)^{b+1}$ local problems. We also eliminate local problems where $c_0 \cup ... \cup c_B$ and S_Ω are identical to a previous problem (Line 22), since this represents the same local MDP and is thus redundant. In all, complexity will depend on the quantity of $\epsilon-$optimal options, such that maximum complexity is achieved only if every corridor allows for an $\epsilon-$optimal policy.

In this work, we will simulate $k = 2$, resulting in $n = \sum_{b=0}^{B} 4^b$ corridors to consider. In particular, we use Algorithm 1 on a simulation environment, discussed next.

5. Experimental Results

The results of Algorithm 1 for the Frozen Lake environment of OpenAI Gym are highlighted in Figures 2 and 3. A link to a Github repository will be included pending acceptance of the paper.

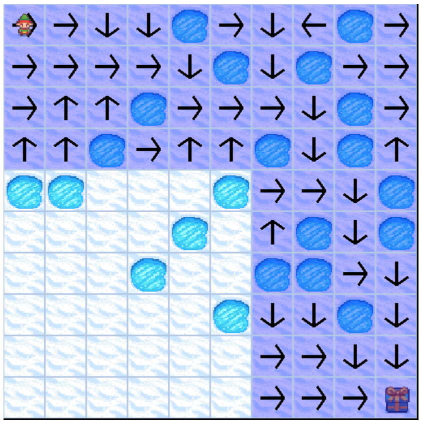

Figure 2: **(Corridor 1 with $\epsilon = 0.99$ sub-optimality.)** Arrows indicate the local policy actions.

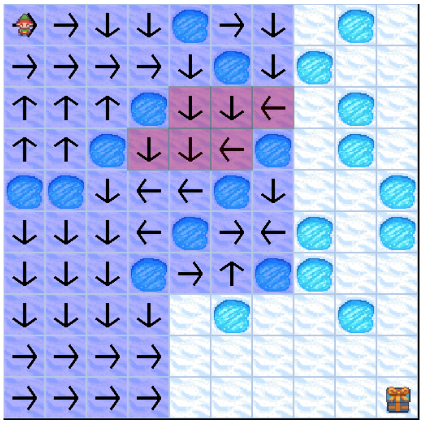

Figure 3: **(Corridor 2 with $\epsilon = 0.9$ sub-optimality.)** Differences from previous corridor are highlighted.

The task of the agent (an 'elf') is to begin from the *initial state* at the top-left corner $(y, x) = (0, 0)$ and reach the *goal state* at the bottom-right corner $(9, 9)$ of the domain. *Holes* are absorbing states with reward 0, and slipping is possible via a stochastic transition function. There are four actions $\{a_N, a_S, a_E, a_W\}$ corresponding to steps in cardinal directions $y - 1, y + 1, x + 1, x - 1$; each action transitions a step in the intended direction with probability 0.9 and left or right each with probability 0.05. The results shown used tabular Q-learning to estimate the value function; other methods such as Deep Q-Network (DQN) have been subsequently implemented to enhance scalability. Fundamentally, the algorithm is compatible with any method that can provide estimates of a value function at each state.

We begin by specifying $\epsilon = 0.99$, so that all solutions are required to have $V_L^*((0, 0))$ very near the global optimum. This results in only one option, shown in Figure 2. Here, the corridor is shown in blue with corresponding local policy π_L (arrows). Reducing epsilon to $\epsilon = 0.90$ yields a second corridor option (Figure 3). This second corridor terminates short of the goal (note that the agent can still *reach* the goal upon exiting the corridor's terminal cell). Briefly interpreting the two policies, we find that the states highlighted red in Figure 3 differ from Figure 2, presumably resulting in more trajectories that proceed to the bottom left as opposed to the top right.

Given only the local policies, it remains unclear how consistently trajectories will follow either corridor to its terminus. Returning to our explanatory story and "options," a human might prefer the second, less optimal policy e.g. if it avoids more of the (dangerous) holes and achieves the goal with higher probability, a matter of "safety." This is the purpose of **Theorem** 14, which provides a bound on the probability that the agent remains within C until reaching S_Ω. Calculated values for this bound, as well as experimental estimates of the probability, are given under the **DNA** entry in Table 1.

For the bound calculation on Corridor 1, the shortest-path bound was taken to be $\tau = 15$, the Manhattan distance between $(0, 0)$ and the nearest state in terminal edge $\{(9, 6), (9, 7), (9, 8), (9, 9)\}$. Similarly, $\tau = 9$ is the Manhattan distance from $(0, 0)$ to the edge $\{(6, 3), (7, 3), (8, 3), (9, 3)\}$ for Corridor 2. Both bounds are clearly conservative in comparison to the experimental estimates, where the agent successfully traversed the corridor much more frequently than the lower bound on expectation.

5.1. Comparison to Quality Diversity

A basic Quality Diversity example was implemented for this setting using the python QD toolbox provided by pyribs (Tjanaka et al. (2023b)). The general optimization problem considered in **Quality Diversity** relies on an objective function with behavioral descriptor $\mathbf{b}_\theta$ and fitness value f_θ (*how* the problem is solved and *how well* it is solved, respectively). Defining a feature space $\mathcal{B}$ based on the possible behavior assignments of $\mathbf{b}$, QD seeks to solve

$$\forall \mathbf{b} \in \mathcal{B} \quad \theta^* = \arg\max_\theta f_\theta \quad \text{s.t.} \quad \mathbf{b} = \mathbf{b}_\theta \tag{10}$$

where parameters θ describe the control policy. In our comparison, a behavior classifier $\mathbf{b}(\rho)$ was defined to assign the appropriate corridor of length b, $1 \leq b \leq B$ to each trajectory from s_i. The resulting behavioral archive contained slots for 5^{B-1} corridor configurations, where corridors were defined relatively from the origin point. The fitness value f was based on the discounted reward from the MDP environment.

In this section, we present two results obtained via the Covariance Matrix Adaptation MAP-Annealing (CMA-MAE) algorithm variants proposed in Tjanaka et al. (2023a). Specifically, we test separable CMA (CMA-Sep) for 3000 iterations and Limited Memory Matrix Adaptation (LM-MA) for 6000 iterations. As shown in Table 1, LM-MA was the more successful of the two, identifying a policy which achieved 38% consistency for Corridor 2. Neither algorithm recovered reliable policies for Corridor 1.

	Corridor 1	**Corridor 2**
DNA	48.6% (bound: 34.1%)	72.6% (bound: 26.2%)
CMA-Sep	1.0%	2.0%
LM-MA	-	38.2%

Table 1: Experimental probability that trajectory safely reaches S_Ω $(n = 500)$

6. Conclusion and Future Work

In this proof-of-concept example, our corridor search algorithm for continuous trajectories produced qualitatively distinct policies, successfully identifying “alternative options” from a state of interest on an MDP. The proposed local reward shaping problems satisfy a set of optimality and safety guarantees. Moreover, the method provides an interesting alternative to the evolutionary methods of Quality Diversity, optimizing local problems independently rather than via policy sampling and variation; this leads to more robust handling of stochasticity in experiment.

As this paper is conceptual in nature, future work is necessary to explore the applications of the method in experiment. This may include a study of the effect of parameter adjustments, including cell size, spacing, and dimension, on the returned policies, as well as efficiency considerations on higher-dimensional environments.

References

Khalid Alsheeb and Martim Brandão. Towards explainable road navigation systems. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 16–22. IEEE, 2023.

Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 264–273. PMLR, 2018.

Benjamin Beyret, Ali Shafti, and A. Aldo Faisal. Dot-to-dot: Explainable hierarchical reinforcement learning for robotic manipulation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5014–5019, 2019. doi: 10.1109/IROS40897.2019.8968488.

Martim Brandão and Yonathan Setiawan. 'why not this mapf plan instead?'contrastive map-based explanations for optimal mapf. In *ICAPS 2022 Workshop on Explainable AI Planning*, 2022.

Noel Brindise and Cedric Langbort. Pointwise-in-time explanation for linear temporal logic rules. *arXiv preprint arXiv:2306.13956*, 2023.

Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. The emerging landscape of explainable automated planning and decision making. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4803–4811. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/669. URL https://doi.org/10.24963/ijcai.2020/669. Survey track.

Konstantinos Chatzilygeroudis, Antoine Cully, Vassilis Vassiliades, and Jean-Baptiste Mouret. Quality-diversity optimization: a novel branch of stochastic optimization. In *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*, pages 109–135. Springer, 2021.

Francisco Cruz, Richard Dazeley, and Peter Vamplew. Memory-based explainable reinforcement learning. In *AI 2019: Advances in Artificial Intelligence: 32nd Australasian Joint Conference, Adelaide, SA, Australia, December 2–5, 2019, Proceedings 32*, pages 66–77. Springer, 2019.

Shripad Vilasrao Deshmukh, Arpan Dasgupta, Chirag Agarwal, Nan Jiang, Balaji Krishnamurthy, Georgios Theocharous, and Jayakumar Subramanian. Trajectory-based explainability framework for offline rl. In *3rd Offline RL Workshop: Offline RL as a"Launchpad"*, 2022.

Mira Finkelstein, Nitsan levy, Lucy Liu, Yoav Kolumbus, David C Parkes, Jeffrey S Rosenschein, and Sarah Keren. Explainable reinforcement learning via model transforms. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 34039–34051. Curran Associates, Inc., 2022.

Manon Flageat and Antoine Cully. Uncertain quality-diversity: Evaluation methodology and new methods for quality-diversity in uncertain domains, 2023. URL https://arxiv.org/abs/2302.00463.

Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. Collective explainable ai: Explaining cooperative strategies and agent contribution in multiagent reinforcement learning with shapley values. *IEEE Computational Intelligence Magazine*, 17(1):59–71, 2022.

Sandy H Huang, Kush Bhatia, Pieter Abbeel, and Anca D Dragan. Establishing appropriate trust via critical states. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 3929–3936. IEEE, 2018.

Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2493–2500, 2020.

Stephanie Milani, Nicholay Topin, Manuela Veloso, and Fei Fang. A survey of explainable reinforcement learning. *arXiv preprint arXiv:2202.08434*, 2022.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Maria Movin, Guilherme Dinis Junior, Jaakko Hollmén, and Panagiotis Papapetrou. Explaining black box reinforcement learning agents through counterfactual policies. In *International Symposium on Intelligent Data Analysis*, pages 314–326. Springer, 2023.

Britt Davis Pierson, Dustin Arendt, John Miller, and Matthew E Taylor. Comparing explanations in rl. *Neural Computing and Applications*, pages 1–12, 2023.

Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3, 2016. ISSN 2296-9144. doi: 10.3389/frobt.2016.00040. URL https://www.frontiersin.org/articles/10.3389/frobt.2016.00040.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning, 2017. URL https://arxiv.org/abs/1703.03864.

Bryon Tjanaka, Matthew C. Fontaine, David H. Lee, Aniruddha Kalkar, and Stefanos Nikolaidis. Training diverse high-dimensional controllers by scaling covariance matrix adaptation map-annealing, 2023a.

Bryon Tjanaka, Matthew C Fontaine, David H Lee, Yulun Zhang, Nivedit Reddy Balam, Nathaniel Dennler, Sujay S Garlanka, Nikitas Dimitri Klapsis, and Stefanos Nikolaidis. pyribs: A barebones python library for quality diversity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 220–229, 2023b.

George A Vouros. Explainable deep reinforcement learning: state of the art and challenges. *ACM Computing Surveys*, 55(5):1–39, 2022.

Appendix A. Proof for Theorem 13

We will prove the following:

Let V_L^* be the value function corresponding to the local Q-learning problem in Definition 11. Then we have

$$V_L^*(s) \leq V^{\hat{\pi}}((s,0)^T)$$

where λ denotes $(s[0], ..., s[K], \Delta)^T$ with $s \in S_{\text{in}}$ and the inequality holds pointwise.

Outline: We will prove the theorem by defining and comparing several MDPs as follows:

- $\mathcal{M}$, the unaltered finite-horizon MDP of the original environment, where $\mathcal{M} = \langle S, A, T, R, \gamma \rangle$
- $\mathcal{M}_\lambda$, defined on state space Λ of augmented states $\lambda = (s, \Delta)^T$. Importantly, the projection of $\mathcal{M}_\lambda$ onto S yields the same value function as $\mathcal{M}$ from any starting state with $\Delta_0 = 0$ (Lemma 15).
- $\tilde{\mathcal{M}_\lambda}$, where further alterations are made to the transition and reward functions. We will show that the value functions for these $\mathcal{M}_\lambda$ and $\tilde{\mathcal{M}_\lambda}$ are identical from any starting state $\lambda_0 = (s_0, 0)^T$ (Lemma 16).
- $\tilde{\mathcal{M}_{R_0}}$, which is the same as $\tilde{\mathcal{M}_\lambda}$ except for a change to the reward function $\tilde{R}_\lambda$. We will show that $\tilde{V}_{R_0}^\pi \leq \tilde{V}_\lambda^\pi$ at all λ (Lemma 17).

The final theorem then shows that

$$\tilde{V}_{R_0}^\pi((s_0, 0)) \leq V^\pi(s_0).$$

MDP 1: $\mathcal{M}_\lambda$. Consider the MDP $\mathcal{M} = \langle S, A, T, R, \gamma \rangle$. We create a new MDP, $\mathcal{M}_\lambda$, by augmenting the state with an additional variable for "switching" purposes. This variable, Δ, yields the full state $\lambda = (s, \Delta)^T$ (with the set of all augmented states notated Λ).

We assign some subset $S_\Delta \subset S$. The variable Δ will track whether the agent has visited any $s \in S_\Delta$ via the function

$$\Delta' = f_\Delta(\lambda, s'), \quad f_\Delta(\lambda, s') := \begin{cases} 1 & s' \in S_\Delta \text{ or } \Delta = 1 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where s', Δ' denote the values of s, Δ at the next time step. Intuitively, $\Delta = 1$ if and only if the current state belongs to S_Δ or a previous state has visited it.

Aside: Existence of t_Δ and Uniqueness of ρ_λ. Consider augmented trajectory $\rho_\lambda = (\lambda_0, \lambda_1, ...)$ with $\lambda_0 = (s_0, \Delta_0)^T$ where $\Delta_0 = 0$. Firstly, we claim that if

$$\lambda_{t+1} = (s_{t+1}, \Delta_{t+1})^T, \quad \Delta_{t+1} = f_\Delta(\lambda_t, s_{t+1}) \tag{12}$$

for all $t \geq 0$, there exists some t_Δ such that, for all $t \geq 0$,

$$\Delta_t = \begin{cases} 0 & t < t_\Delta \\ 1 & \text{otherwise} \end{cases} \tag{13}$$

or $\Delta_t = 0$ everywhere. Secondly, we claim that, for a given $\rho = (s_0, s_1, ...)$, there is one unique ρ_λ which satisfies (11) on all t.

To prove the first claim, we will first show that f_Δ must be monotonically increasing over $\{0, 1\}$, which is possible by induction.

Show $f_\Delta(\lambda_0, s_1) \leq f_\Delta(\lambda_1, s_2)$: Consider the following cases.

- $s_1 \notin S_\Delta$. By (11), $f_\Delta((s_0, 0)^T, s_1) = 0$. As $f_\Delta : \Lambda \times S \to \{0, 1\}$, clearly $0 \leq f_\Delta(\lambda_1, s_2)$ regardless of s_2.

- $s_1 \in S_\Delta$. By (11), $f_\Delta((s_0, 0)^T, s_1) = 1$ and thus $\lambda_1 = (s_1, 1)^T$. Then $f_\Delta(\lambda_1, s_2)^T = 1$ and $f_\Delta((s_0, 0)^T, s_1) \leq f_\Delta(\lambda_1, s_2)$.

Prove $f_\Delta(\lambda_t, s_{t+1}) \leq f_\Delta(\lambda_{t+1}, s_{t+2})$:

- $\Delta_t = 0$: Our result follows from the logic for λ_0 above, with indices $0, 1, 2$ replaced by $t, t+1, t+2$.

- $\Delta_t = 1$: From (11), we see that

$$f_\Delta((s_t, 1)^T, s_{t+1}) = 1 \quad \Rightarrow \quad \lambda_{t+1} = (s_{t+1}, 1)^T \tag{14}$$

$$\Rightarrow f_\Delta(\lambda_{t+1}, s_{t+2}) = 1 \tag{15}$$

  and thus $f_\Delta(\lambda_t, s_{t+1}) \leq f_\Delta(\lambda_{t+1}, s_{t+2})$.

Therefore, $f_\Delta(\lambda_t, s_{t+1}) \leq f_\Delta(\lambda_{t+1}, s_{t+2})$ for all $t \geq 0$ when $\Delta_0 = 0$. To show the existence of t_Δ defined above, simply select

$$t_\Delta = \min(\{t \mid \Delta_t = 1\}) \tag{16}$$

If the set is not empty, we have the minimum t' such that $\Delta_t = 0$ for all $t < t'$ by definition; by monotonicity, $\Delta_t = 1$ for all $t \geq t_\Delta$. If the set is empty, this means that there exists no t such that $\Delta_t = 1$. As f_Δ maps to $\{0, 1\}$, all Δ_t must therefore equal 0, satisfying (13) and therefore our claim. (QED 1)

The second claim follows from our first claim. Any ρ_λ which is not everywhere 0 must admit a t_Δ as above (or violate monotonicity). Since t strictly increases in any sequence, (16) must have a unique solution. Therefore, any ρ will have exactly one corresponding ρ_λ satisfying (11) at all t. (QED 2)

We now return to the definition of $\mathcal{M}_\lambda$. We enforce transitions according to f_Δ using the **augmented transition function**

$$T_\lambda(\lambda, a, \lambda') = \begin{cases} T(s, a, s') & \Delta' = f_\Delta(\lambda, s') \\ 0 & \text{otherwise} \end{cases}. \tag{17}$$

The **augmented reward function** remains the same:

$$R_\lambda(\lambda, a) = R(s, a). \tag{18}$$

Now we will consider the value function for trajectories on $\mathcal{M}_\lambda$, comparing it to $\mathcal{M}$ in Lemma 15.

Lemma 15 *Given $\mathcal{M}$ and $\mathcal{M}_\lambda$ as defined above,*

$$V^\pi(s_t) = V^\pi_\lambda((s_t, \Delta_t)^T) \quad \textit{if } \Delta_t = 0. \tag{19}$$

for a policy $\pi : \Lambda \to A$.

Proof of Lemma 15: For an infinite horizon MDP with reward function $\mathcal{R}$, states $s \in \mathcal{S}$, and infinite trajectories $\rho = (s_t, s_{t+1}, ...)$, the generic value function $V^\pi : \mathcal{S} \to \mathbb{R}$ for policy $\pi : \mathcal{S} \to A$ can be expressed as

$$V^\pi(s_t) = \mathbb{E}_\pi[\sum_{k=t}^{\infty} \gamma^{k-t} \mathcal{R}(s_k, \pi(s_k))]. \tag{20}$$

Denote the probability of $\rho = (s^a, s^b ...)$ under policy π as $p_{\rho|\pi}$. Let $\rho^i[k]$ denote the k^{th} state in the i^{th} trajectory ρ^i and let a^i_k be the policy action $\pi(\rho^i[k])$. Then (63) can be rewritten for $\mathcal{M} = \langle S, A, T, R, \gamma \rangle$:

$$V^\pi(s_t) = p_{\rho^1|\pi} \sum_{k=t}^{\infty} \gamma^{k-t} R(\rho^1[k], a^1_k) + p_{\rho^2|\pi} \sum_{k=t}^{\infty} \gamma^{k-t} R(\rho^2[k], a^2_k) + ... \tag{21}$$

where

$$p_{\rho|\pi} = \prod_{k=0}^{\infty} T(\rho[k], a_k, \rho[k+1]). \tag{22}$$

For $\mathcal{M}_\lambda$, we consider augmented trajectories ρ_λ of $\lambda_t \in \Lambda$; these are generated according to T_λ as in (17). For augmented trajectory $\rho_\lambda = ((s^a, \Delta^a)^T, (s^b, \Delta^b)^T, ...)$ corresponding to $\rho = (s^a, s^b, ...)$, we then have

$$p_{\rho_\lambda|\pi} = \prod_{k=0}^{\infty} T(\rho[k], a_k, \rho[k+1]) = p_{\rho|\pi} \tag{23}$$

if ρ_λ is the unique trajectory with $\Delta_0 = 0$ and (11) true everywhere, and $p_{\rho_\lambda|\pi} = 0$ otherwise. Therefore,

$$V^\pi_\lambda((s_t, 0)^T) = p_{\rho^1|\pi} \sum_{k=t}^{\infty} \gamma^{k-t} R(\rho^1[k], a^1_k) + p_{\rho^2|\pi} \sum_{k=t}^{\infty} \gamma^{k-t} R(\rho^2[k], a^2_k) + ... \tag{24}$$

which is exactly the same as (21). (QED)

MDP 2: $\tilde{\mathcal{M}}_\lambda$. We will now construct an altered MDP in which the transition and reward functions behave differently. For the transition function,

$$\tilde{T}_\lambda(\lambda, a, \lambda') = \begin{cases} T_\lambda(\lambda, a, \lambda') & \Delta = 0 \\ 1 & \Delta = 1, \; s = s', \Delta' = f_\Delta(\lambda, s') \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

This new $\tilde{T}_\lambda$ simply makes the set S_Δ into absorbing states. The new reward function is

$$\tilde{R}_\lambda(\lambda, a) = \begin{cases} R(s, a) & \Delta = 0 \\ (1-\gamma) V^\pi_\lambda(\lambda) & \text{otherwise} \end{cases} \tag{26}$$

This reward assigns a discounted value function of $\mathcal{M}_\lambda$ as a reward upon reaching any absorbing state $s \in S_\Delta$.

We will finally consider the value function and prove 16. **For the remainder of the proof,** we will allow ρ to refer to trajectories over the augmented state space, i.e. $\rho = (\lambda_0, \lambda_1, ...)$.

Lemma 16 *Given $\mathcal{M}_\lambda$ and $\tilde{\mathcal{M}}_\lambda$ as defined above,*

$$V_\lambda^\pi(\lambda_0) = \tilde{V}_\lambda^\pi(\lambda_0) \tag{27}$$

From (63) we may write a function similar to (24):

$$\tilde{V}_\lambda^\pi(\lambda_t) = p_{\rho^1|\pi} \sum_{k=t}^{\infty} \gamma^{k-t} \tilde{R}_\lambda(\rho^1[k], a_k^1) + p_{\rho^M|\pi} \sum_{k=t}^{\infty} \gamma^{k-t} \tilde{R}_\lambda(\rho^2[k], a_k^2) + ... \tag{28}$$

If a t_Δ exists as defined previously, we may divide any trajectory originating at some $(s, 0)$ on $\mathcal{M}_\lambda$ or $\tilde{\mathcal{M}}_\lambda$ into two parts: a "prefix," where $t < t_\Delta$ and $\Delta = 0$; and a "suffix," where $t \geq t_\Delta$ and $\Delta = 1$. We then have

$$\rho =((s_0, 0)^T, ..., (s_{t_\Delta - 1}, 0)^T, (s_{t_\Delta}, 1)^T, ...). \tag{29}$$

If no such t_Δ exists, the prefix extends to infinity and the suffix has length 0.

Reward for Prefixes: We will first consider **all prefixes,** which have the form $(s_0, 0)^T, (s_1, 0)^T, ...$ From (18), the reward for each state $\lambda = (s, \Delta)$ in $\mathcal{M}_\lambda$ given policy π is simply

$$R(s, \pi(\lambda)). \tag{30}$$

For $\tilde{\mathcal{M}}_\lambda$, the reward $\tilde{R}_\lambda$ from (26) is dependent on Δ. Given that $\Delta = 0$ on the entire prefix,

$$\tilde{R}_\lambda = R_\lambda = R(s, \pi(\lambda)). \tag{31}$$

Probability of Prefixes: From (25), the transition function for $\tilde{\mathcal{M}}_\lambda$ is identical to $\mathcal{M}_\lambda$ when $\Delta = 0$. Therefore, the probability of a given trajectory ρ on both $\mathcal{M}_\lambda$ and $\tilde{\mathcal{M}}_\lambda$ is

$$p_{\rho|\pi} = \prod_{k=0}^{t_\Delta - 1} T_\lambda(\lambda_k, \pi(\lambda_k), \lambda_{k+1}) * \prod_{k=t_\Delta}^{\infty} \mathcal{T}_{suf}(\lambda_k, \pi(\lambda_k), \lambda_{k+1}) \tag{32}$$

if t_Δ exists, where $\mathcal{T}_{suf}$ remains unknown. When t_Δ does not exist,

$$p_{\rho|\pi} = \prod_{k=0}^{\infty} T_\lambda(\lambda_k, \pi(\lambda_k), \lambda_{k+1}). \tag{33}$$

For shorthand, we may rewrite this

$$p_{\rho|\pi} = p_{\rho|\pi}^{pre} * p_{\rho|\pi}^{suf} \tag{34}$$

where $p_{\rho|\pi}^{suf} = 1$ in the latter case.

Reward for Suffixes: We now move to the suffixes. From (18), the reward for $\mathcal{M}_\lambda$ given policy π is once again

$$R(s, \pi(\lambda)) \tag{35}$$

for each λ_t, $t \geq t_\Delta$. For $\tilde{\mathcal{M}}_\lambda$, (26) when $\Delta_t = 1$ gives

$$\tilde{R}_\lambda = (1 - \gamma) V^\pi_\lambda(\lambda_t) \tag{36}$$

Probabilities for Suffixes: In general, we have

$$p^{suf}_{\rho|\pi} = \prod_{k=t_\Delta}^{\infty} \mathcal{T}_{suf}(\lambda_k, \pi(\lambda_k), \lambda_{k+1})$$

For $\mathcal{M}_\lambda$, we simply have $\mathcal{T} = T_\lambda$ from (17). For $\tilde{\mathcal{M}}_\lambda$, we have $\Delta = 1$; thus, from (25), only transitions with $s_t = s_{t_\Delta}$ have nonzero probability. Then

$$p^{suf}_{\rho|\pi} = \mathbb{1}(s_t, s_{t_\Delta}) \tag{37}$$

where $\mathbb{1} : S \times S \to \{0, 1\}$ is an indicator function with $\mathbb{1}(s_i, s_j) = 1$ when $s_i = s_j$ and 0 otherwise.

Putting the trajectories together: We finally have expressions for $p_{\rho|\pi}$ for both MDPs. For $\mathcal{M}_\lambda$, we see that $\mathcal{T}$ is identical for prefix and suffix, so

$$p_{\rho|\pi} = \prod_{k=0}^{\infty} T_\lambda(\lambda_k, \pi(\lambda_k), \lambda_{k+1}). \tag{38}$$

For $\tilde{\mathcal{M}}_\lambda$,

$$\tilde{p}_{\rho|\pi} = \prod_{k=0}^{t_\Delta - 1} T_\lambda(\lambda_k, \pi(\lambda_k), \lambda_{k+1}) \prod_{k=t_\Delta}^{\infty} \mathbb{1}(s_k, s_{t_\Delta}) \tag{39}$$

when t_Δ exists and $\tilde{p}_{\rho|\pi} = p_{\rho|\pi}$ (as in (38)) otherwise. Now we recall the value functions for $\mathcal{M}_\lambda$ and $\tilde{\mathcal{M}}_\lambda$, given that trajectories start with $\lambda_0 = (s_0, 0)^T$. On the prefix, we have shown that $R_\lambda((s, \Delta)^T, a) = \tilde{R}_\lambda((s, \Delta)^T, a) = R(s, a)$. Thus, each term of V^π_λ as in (24) becomes

$$p_{\rho^i|\pi}\left(\sum_{k=0}^{\infty} \gamma^k R(\rho^i[k], a^i_k)\right). \tag{40}$$

Now we consider the summands of $\tilde{V}^\pi_\lambda$. For $\tilde{V}^\pi_\lambda$, any ρ^i where no t_Δ exists clearly result in terms identical to (40). For all other ρ^i with nonzero probability (see (37)),

$$p_{\rho^i|\pi}\left(\sum_{k=0}^{t_\Delta - 1} \gamma^k R(\rho^i[k], a^i_k) + \sum_{k=t_\Delta}^{\infty} \gamma^k (1 - \gamma) V^\pi_\lambda(\rho^i[t_\Delta])\right) \tag{41}$$

Of these, we may **group** the terms with identical prefixes. Clearly, identical prefixes will have the same value for t_Δ; for each such set of ρ^i, we then have a set-specific constant

$$R^{pre} = \sum_{k=0}^{t_\Delta - 1} \gamma^k R(\rho^i[k]) \tag{42}$$

and can write the sum of terms in the value function $\tilde{V}_\lambda^\pi$ for all ρ^i in the set as

$$p_{\rho^1|\pi}\Bigg(R^{pre} + \sum_{k=t_\Delta}^{\infty} \gamma^k (1-\gamma) V_\lambda^\pi(\rho^1[t_\Delta])\Bigg) + p_{\rho^2|\pi}\Bigg(R^{pre} + \sum_{k=t_\Delta}^{\infty} \gamma^k (1-\gamma) V_\lambda^\pi(\rho^2[t_\Delta])\Bigg) + ... \tag{43}$$

By (34), we may split all $p_{\rho^i|\pi}$ into $p^{pre} * p^{suf}$. As all prefixes are identical in our grouping, p^{pre} is the same for each term; p^{suf} will vary. Thus, with some relaxing of notation, we have

$$p^{pre}p_1^{suf}\Bigg(R^{pre} + \sum_{k=t_\Delta}^{\infty} \gamma^k (1-\gamma) V_\lambda^\pi(\rho^1[t_\Delta])\Bigg) + p^{pre}p_2^{suf}\Bigg(R^{pre} + \sum_{k=t_\Delta}^{\infty} ...\Bigg) + ... \tag{44}$$

Distributing the p_i^{suf}, this becomes

$$p^{pre}\Bigg(R^{pre}\sum_{i=1}^{I} p_i^{suf} + \sum_{i=1}^{I} p_i^{suf} \sum_{k=t_\Delta}^{\infty} \gamma^k (1-\gamma) V_\lambda^\pi(\rho^i[t_\Delta])\Bigg) \tag{45}$$

for a set with I trajectories. First, we note that the full set of p^{suf} must cover every possible suffix from λ_n and therefore

$$\sum_{i=1}^{I} p_i^{suf} = 1 \tag{46}$$

Then, distributing p^{pre},

$$=p^{pre}R^{pre} + \sum_{i=1}^{I} p^{pre}p_i^{suf} \sum_{k=t_\Delta}^{\infty} \gamma^k (1-\gamma) V_\lambda^\pi(\rho^i[t_\Delta]) \tag{47}$$

$$=p^{pre}R^{pre} + \sum_{i=1}^{I} p_{\rho^i|\pi} \sum_{k=t_\Delta}^{\infty} \gamma^k (1-\gamma) V_\lambda^\pi(\rho^i[t_\Delta]) \tag{48}$$

This may be further simplified using the sum of an infinite geometric series. Then

$$\sum_{k=t_\Delta}^{\infty} \gamma^k (1-\gamma) V_\lambda^\pi(\rho^i[t_\Delta]) = \sum_{k=t_\Delta}^{\infty} \gamma^k V_\lambda^\pi(\rho^i[t_\Delta]) - \sum_{k=t_\Delta+1}^{\infty} \gamma^k V_\lambda^\pi(\rho^i[t_\Delta]) \tag{49}$$

$$\begin{aligned}
&= \Bigg(\frac{V_\lambda^\pi(\rho^i[t_\Delta])}{1-\gamma} - \sum_{k=0}^{t_\Delta-1} \gamma^k V_\lambda^\pi(\rho^i[t_\Delta])\Bigg) - \Bigg(\frac{V_\lambda^\pi(\rho^i[t_\Delta])}{1-\gamma} - \sum_{k=0}^{t_\Delta} \gamma^k V_\lambda^\pi(\rho^i[t_\Delta])\Bigg) \\
&= -\sum_{k=0}^{t_\Delta-1} \gamma^k V_\lambda^\pi(\rho^i[t_\Delta]) + \sum_{k=0}^{t_\Delta} \gamma^k V_\lambda^\pi(\rho^i[t_\Delta]) \\
&= \gamma^{t_\Delta} V_\lambda^\pi(\rho^i[t_\Delta])
\end{aligned} \tag{50}$$

Then we can rewrite (52):

$$p^{pre}R^{pre} + \gamma^{t_\Delta}\sum_{i=1}^{I} p_{\rho^i|\pi} V_\lambda^\pi(\rho^i[t_\Delta]) \tag{51}$$

We have already remarked that, for all ρ with zero-length suffix (t_Δ nonexistent), $\mathcal{M}_\lambda$ and $\tilde{\mathcal{M}}_\lambda$ have identical terms in V. We must still compare these remaining terms of $\tilde{\mathcal{M}}_\lambda$ (as in (51)) to the terms for corresponding ρ in $\mathcal{M}_\lambda$. We may split these terms for $\mathcal{M}$ into the same like-prefixed sets and rewrite them as we did for $\tilde{M}_\lambda$. Then, each set can be expressed by

$$\sum_{i=1}^{I} p_{\rho^i|\pi}\left(\sum_{k=0}^{\infty}\gamma^k R(\rho^i[k], a_k^i)\right) = p^{pre}R^{pre} + \sum_{i=1}^{I} p_{\rho^i|\pi}\sum_{k=t_\Delta}^{\infty}\gamma^k R(\rho^i[k], a_k^i) \tag{52}$$

Recalling the definition of V_λ^π (24), we notice something quite familiar:

$$=p^{pre}R^{pre} + \sum_{i=1}^{I} p^{pre}p_i^{suf}\left(\gamma^{t_\Delta}\sum_{k=t_\Delta}^{\infty}\gamma^{k-t_\Delta}R(\rho^i[k], a_k^i)\right) \tag{53}$$

$$=p^{pre}R^{pre} + \gamma^{t_\Delta}\sum_{i=1}^{I} V_\lambda^\pi(\rho^i[t_\Delta]). \tag{54}$$

which is exactly the same as (51). Having now compared all like-prefixed groupings of summands for both V_λ^π and $\tilde{V}_\lambda^\pi$, we may conclude that

$$V_\lambda^\pi(\lambda_0) = \tilde{V}_\lambda^\pi(\lambda_0) \tag{55}$$

for $\lambda_0 = (s_0, 0)^T$, proving Lemma 16. (QED)

Final MDP: $\tilde{\mathcal{M}}_{R_0}$. We will finally establish an MDP which is identical to $\tilde{\mathcal{M}}_\lambda$ except for a change in the reward function. We define $S_\Omega \subset S_\Delta$, where S_Ω corresponds to the "terminal edges" in our algorithm. The reward function becomes

$$\tilde{R}_{R_0}(\lambda, a) = \begin{cases} R(s,a) & \Delta = 0 \\ (1-\gamma)V_\lambda^\pi(\lambda_t) & \Delta = 1,\ s \in S_\Omega \\ 0 & \text{otherwise} \end{cases} \tag{56}$$

In words, sink states $s \in S_\Delta$ which are not also in S_Ω now receive reward 0. This brings us to our final lemma.

Lemma 17 $\tilde{V}_{R_0}^\pi(s_0, 0) \le \tilde{V}_\lambda^\pi(s_0, 0)$.

As neither reward nor transition function change for the case $\Delta = 0$, terms of the value function $\tilde{V}_{R_0}^\pi$ with zero-length suffix are identical to $\tilde{V}_\lambda^\pi$. For the rest, the contribution of each individual trajectory is

$$p_{\rho^i|\pi}\sum_{k=t_\Delta}^{\infty}\gamma^k \tilde{R}_{R_0}(\rho^i[k], a_k^i). \tag{57}$$

We may compare each such contribution to the corresponding contribution by the same trajectory in $\tilde{V}_\lambda^\pi$:

$$\begin{aligned} \tilde{V}_{R_0}^\pi :&\quad p_{\rho^i|\pi}(\gamma^{t_\Delta}\tilde{R}_{R_0}(\rho^i[t_\Delta], a^i_{t_\Delta}) + \gamma^{t_\Delta+1}\tilde{R}_{R_0}(\rho^i[t_\Delta+1], a^i_{t_\Delta+1}) + ...) \\ \tilde{V}_\lambda^\pi :&\quad p_{\rho^i|\pi}(\gamma^{t_\Delta}\tilde{R}_\lambda(\rho^i[t_\Delta], a^i_{t_\Delta}) \ \ + \gamma^{t_\Delta+1}\tilde{R}_\lambda(\rho^i[t_\Delta+1], a^i_{t_\Delta+1}) + ...) \end{aligned} \tag{58}$$

Examining $\tilde{R}_{R_0}$ and $\tilde{R}_\lambda$, we readily observe that for all $\lambda = (s, \Delta)^T$, the pointwise inequality

$$\tilde{R}_{R_0}(\lambda, \cdot) \le \tilde{R}_\lambda(\lambda, \cdot) \tag{59}$$

holds. Therefore, for all pairs of corresponding terms as in (58),

$$\gamma^t \tilde{R}_{R_0}(\rho^i[t], a^i_t) \le \gamma^t \tilde{R}_\lambda(\rho^i[t], a^i_t) \tag{60}$$

and thus

$$p_{\rho^i|\pi}\sum_{k=t_\Delta}^{\infty}\gamma^k \tilde{R}_{R_0}(\rho^i[k], a^i_k) \le p_{\rho^i|\pi}\sum_{k=t_\Delta}^{\infty}\gamma^k \tilde{R}_\lambda(\rho^i[k], a^i_k) \tag{61}$$

for all such trajectories. Then the full summation for $\tilde{V}_\lambda^\pi$ and $\tilde{V}_{R_0}^\pi$ brings us to

$$\tilde{V}_{R_0}^\pi(\lambda_0) \le \tilde{V}_\lambda^\pi(\lambda_0) \tag{62}$$

for all $\lambda_0 = (s_0, 0)^T$, and the lemma holds. (QED) We may now prove our theorem.

Theorem 13 ($\tilde{V}_{R_0}^\pi((s_0, 0)) \le V^\pi(s_0)$)
From Lemmas 15, 16, and 17, we have that

$$V^\pi(s_0) = V_\lambda^\pi(s_0, 0) = \tilde{V}_\lambda^\pi(s_0, 0) \ge \tilde{V}_{R_0}^\pi(s_0, 0)$$

for all $s_0 \in S$. Thus

$$V^\pi(s_0) \ge \tilde{V}_{R_0}^\pi(s_0, 0)$$

at all $s_0 \in S$. (QED)

Appendix B. Proof for Theorem 14

Outline: To prove the theorem, we examine the expected reward for two categories of trajectories ρ over S:

- ρ which reach a terminal state $s \in S_\Omega$ without leaving the corridor ("successes")
- ρ which either leave the corridor before reaching S_Ω or remain inside the corridor forever ("failures")

We will show that the probability of success $\mathbb{P}_{success}$ can be bounded from below by calculating the minimum necessary proportion of successful trajectories to achieve the expectation $V_L^*(s_t)$.

Let $\mathcal{M}_L = \langle S, A, T_L, R_L, \gamma \rangle$ be our local MDP and let $V^* : S \to \mathbb{R}$ be value function corresponding to some policy $\pi^* : S \to A$. The generic value function $V^\pi : S \to \mathbb{R}$ for any policy $\pi : S \to A$ is

$$V^\pi(s_t) = \mathbb{E}_\pi[\sum_{k=t}^{\infty} \gamma^{k-t} R_L(s_{t+k}, \pi(s_{t+k}))]. \tag{63}$$

Now, for any trajectory ρ originating at some s_t, let $\rho[k]$ denote state s_{t+k} in the trajectory and $a_k = \pi(\rho[k])$. Denote the probability of ρ occurring from s_t under policy π as $p_{\rho|\pi}$. Then 63 can be rewritten as a sum over the possible trajectories from s_t, as follows:

$$V^\pi(s_t) = p_{\rho_1|\pi} \sum_{k=t}^{\infty} \gamma^{k-t} R_L(\rho_1[k], a_k^1) + p_{\rho_2|\pi} \sum_{k=t}^{\infty} \gamma^{k-t} R_L(\rho_2[k], a_k^2) + ... \tag{64}$$

We are interested in the **probability that our trajectories remain within the corridor**. Thus, we separate these terms into two groups: one in which all trajectories terminate at the desired terminal edge (*success*), and one in which they either remain in the corridor forever or exit at a non-terminal boundary (*failure*). We may then group ρ such that all $\rho^i \in I$ are *successes* and all $\rho^j \in J$ are *failures*. Then

$$\mathbb{P}_{success} = \sum_{\rho^i \in I}^{\infty} p_{\rho^i|\pi} \quad \text{and} \quad \mathbb{P}_{fail} = \sum_{\rho^j \in J}^{\infty} p_{\rho^j|\pi} \tag{65}$$

We will now examine the trajectories in each case. Let $s_{in} \in S_{in}$ denote any (non-terminal) state inside of the corridor and $s_\Omega \in S_\Omega$ denote any state in the terminal edge. Finally, $s_{out} \in S_{out}$ will be any state outside of the corridor, such that $S_{out} = S \setminus (S_{in} \cup S_\Omega)$. Given $\rho = (s_0, s_1, ...)$ is a **success,** there must exist some

$$t_\Omega = \min(\{t \mid s_t \in S_\Omega\}). \tag{66}$$

Denote the particular s reached at t_Ω by s_Ω. From the reward function R_L, we then have

$$\sum_{k=0}^{\infty} \gamma^k R_L(\rho[k], a_k) = \sum_{k=0}^{t_\Omega - 1} \gamma^k R(s_{in}, a_k) + \sum_{k=t_\Omega}^{\infty} \gamma^k (1-\gamma) V^*(\rho[k]). \tag{67}$$

In words, all states preceding the terminal state must belong to S_{in}; otherwise, (66) would not hold (or ρ would not be a success). Now, we have by the transition function that all $s \in S_\Omega$ are absorbing, and thus $\rho[k] = \rho[t_\Omega] = s_\Omega$ for all $k \geq t_\Omega$. We see then that the last term can be simplified using the convergence of geometric series:

$$\sum_{k=0}^{\infty} \gamma^k R_L(\rho[k], a_k) = \left(\sum_{k=0}^{t_\Omega - 1} \gamma^k R(s_{in}, a_k) \right) + \gamma^{t_\Omega} V^*(s_\Omega). \tag{68}$$

Letting $\max(r_{in}) := \max_{s \in S_{in}} R(s, \pi(s))$ and $\max(r_\Omega) := \max_{s \in S_\Omega} V^*(s)$, we can establish an upper bound on the discounted reward for any successful trajectory:

$$\sum_{k=0}^{\infty} \gamma^k R(\rho[k], a_k) \leq \sum_{k=0}^{t_\Omega - 1} \gamma^k \max(r_{in}) + \gamma^{t_\Omega} \max(r_\Omega). \tag{69}$$

Moreoever, the value $\max(r_{in})$ is constrained to be a nonnegative constant and $0 \leq \gamma < 1$, meaning that

$$\sum_{k=0}^{t-1} \gamma^k \max(r_{in}) \leq \sum_{k=0}^{t} \gamma^k \max(r_{in}) \tag{70}$$

for all t. Again by geometric series,

$$\sum_{k=0}^{t_\Omega - 1} \gamma^k \max(r_{in}) < \frac{\max(r_{in})}{1-\gamma} \tag{71}$$

for any t_Ω.

Since t_Ω may be as small as 0, the term $\gamma^{t_\Omega} \max(r_\Omega)$ is not as easy to bound without additional knowledge of the dimensions of the corridor and t_Ω. If a shortest path is known such that the distance between s_t and S_Ω cannot be traversed in less than d steps,

$$\gamma^{t_\Omega} \max(r_\Omega) \leq \gamma^d \max(r_\Omega) \tag{72}$$

where $d = 0$ if no such minimum step count is known. Applying these bounds to the full set of successful trajectories yields

$$p_{\rho_1|\pi} \sum_{k=0}^{\infty} \gamma^k R_L(\rho_1[k], a_k^1) + p_{\rho_2|\pi} \sum_{k=0}^{\infty} \gamma^k R_L(\rho_2[k], a_k^2) + ... \tag{73}$$

$$\leq \mathbb{P}_{success}(\frac{\max(r_{in})}{1-\gamma} + \gamma^d \max(r_\Omega)). \tag{74}$$

Now we consider **"failing" trajectories**. These may remain within S_{in} forever, i.e.

$$\sum_{k=0}^{\infty} \gamma^k R_L(\rho[k], a_k) = \sum_{k=0}^{\infty} \gamma^k R(s_{in}, a_k), \tag{75}$$

In this case, the reward may be bounded with the same geometric series:

$$\sum_{k=0}^{\infty} \gamma^k R_L(\rho[k], a_k) \leq \frac{\max(r_{in})}{1-\gamma} \tag{76}$$

Alternatively, failed trajectories may exit the corridor at some non-terminal state, i.e.

$$\sum_{k=0}^{\infty} \gamma^k R_L(\rho[k], a_k) = \sum_{k=0}^{T-1} \gamma^k R(s_{in}, a_k) + \sum_{k=T}^{\infty} \gamma^k R(s_{out}, a_k). \tag{77}$$

given the first $s \notin S_{in}$ occurs at time T. In this case, the reward may be once again bounded similarly to successful trajectories, using $s_{out} \in S_{out}$ absorbing but noting this time that $R_L(s_{out}, \cdot) = 0$. Then

$$\sum_{k=0}^{\infty} \gamma^k R_L(\rho[k], a_k) \leq \frac{\max(r_{in})}{1-\gamma} + 0 \tag{78}$$

Thus, returning to 64 and noting that $\mathbb{P}_{fail} = 1 - \mathbb{P}_{success}$, we have

$$\begin{aligned} V^\pi(s_t) \le \mathbb{P}_{success}(\frac{\max(r_{in})}{1-\gamma} + \gamma^d \max(r_\Omega)) \\ +(1 - \mathbb{P}_{success})(\frac{\max(r_{in})}{1-\gamma}) \end{aligned} \tag{79}$$

Simplifying,

$$V^\pi(s_t) \le \mathbb{P}_{success}(\gamma^d \max(r_\Omega)) + \frac{\max(r_{in})}{1-\gamma} \tag{80}$$

As given in the local problem, we may replace $\max(r_\Omega)$ with $\max_{s\in S_\Omega} V^*(s)$, where V^* is the optimal value function for the global policy on the original MDP. This yields the bound

$$(V^\pi(s_t) - \frac{\max(r_{in})}{1-\gamma})\frac{1}{\gamma^d \max_{s\in S_\Omega} V^*(s)} \le \mathbb{P}_{success} \tag{81}$$

In our specific case where $\max(r_{in}) \le 1$, this becomes

$$(V^\pi(s_t) - \frac{1}{1-\gamma})\frac{1}{\gamma^d \max_{s\in S_\Omega} V^*(s)} \le \mathbb{P}_{success} \tag{82}$$

Thus, for any corridor, we have established a lower bound on the probability that a trajectory successfully remains within the corridor until reaching the terminal edge. (QED)