"What are my options?": Explaining RL Agents with Diverse Near-Optimal Alternatives

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

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

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 some optimum: an *explanation via alternatives*.

Consider a route planning task for a ground vehicle. If the human operator is dissatisfied with a given plan (say it passes through risky terrain or an unwanted waypoint), the operator may want to assess potential alternatives before making a decision. Providing a user with alternatives also illuminates plan flexibility. A plan may be "inflexible" if it traverses states where there are limited choices of action, e.g. the vehicle is confined to a valley or moving along the edge of a cliff. This echoes the "critical states" proposed by Huang et al., where available actions have drastically different effects on cost (Huang et al. (2018)). In contrast, a plan may be called "flexible" if multiple reasonable policies are available, such as when different roads lead to the same destination in similar time.

In this work, we pursue an explanatory method for **value-based** Reinforcement Learning agents which we call *Diverse Near-Optimal Alternatives* (DNA). DNA aims to answer questions such as "what policies/paths can I reasonably take from here?" and "how will the cost compare?" Given an agent starts in a particular state, the method partitions the state space into distinct "corridors," each of which will correspond to a possible policy option from that state. It then establishes "local problems," adaptations of the original environment which reward the agent for successfully traversing the corridor. The resulting policies and the trajectories they produce are subject to a set of safety 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) aim to improve human understanding of autonomous system behavior. Though 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 an agent to be inherently more explainable, while *post-hoc* methods interpret the existing model.

Post-hoc explanation alone encompasses a wide variety of approaches. It may describe 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)). It may also assess the influence of trajectories on success probability (Cruz et al. (2019)) or policy shape (Deshmukh et al. (2022)). In many approaches, a human suggests an "alternative" policy or trajectory; the explanation then highlights infeasible segments (Alsheeb and Brandão (2023)) or suggests environment changes to enable the suggestion (Brandão and Setiawan (2022), Finkelstein et al. (2022)). Reward shaping has also been used to drive an agent to desired waypoints (Movin et al. (2023), Beyret et al. (2019)).

However, explanations which **seek and offer multiple policy suggestions** are rare in XAIP/XRL. This type of problem has been addressed e.g. in road navigation, where multiple route suggestions are commonly offered; unfortunately, the Dijkstra-based algorithms in such applications are limited to graph-traversal settings. For more general settings, a recent branch of stochastic optimization called Quality Diversity (QD) shows promise. QD solves for a set of high-performing ("quality") policies which are behaviorally distinct ("diversity") by iteratively populating a *behavior* or *feature* space with the best performers (Mouret and Clune (2015); Chatzilygeroudis et al. (2021); Pugh et al. (2016)). These methods suffer in the stochastic setting, however, as they heavily rely on a one-to-one policy-to-trajectory connection (Flageat and Cully (2023)).

In contrast, our work is tailored to stochastic environments and addresses uncertain dynamics more directly than potential QD-inspired approaches. The local optimization problems defined by DNA encourage trajectories to remain within distinct corridors, leading to provable guarantees

on trajectory shapes. Coupled with the proposed structure of the alternative policies, the reward-shaping scheme of DNA will also provide useful optimality guarantees.

3. Background

This work includes a simple proof of concept 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 $R(\cdot, \cdot) \geq 0$. As our case study will simulate continuous trajectories on a grid, we specifically consider the common setting of gridworld-based RL:

Definition 2 (MDP on a Grid) *An MDP on a grid is an MDP where $S \subset \mathbb{Z}^K$ and the **neighbors** of any $s \in S$ are defined as*

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

where $\|\cdot\|_M$ denotes the Manhattan distance. The 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,

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

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 $s \in S$. Now, 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 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. Often, trajectories in space are described using waypoints, e.g. in aviation; however, 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. This will require some adaptation:

1. **Waypoints $\to$ 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 $\to$ Corridors:** we replace a sequence of waypoints with a *corridor*, an object defined on subsets of S using way-regions. We will alter R to incentivize trajectories to remain inside a corridor and traverse from an initial state to an intermediate goal.

Thus motivated, we will assess possible options via *corridor search*: given a starting state s_i, we create corridors connecting s_i to intermediate goals expressed as "terminal" way-regions S_Ω. We then establish local problems using reward shaping. Training policies on these local problems results in a set of diverse, near-optimal 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 $\hat{\pi}$ which (i) have sufficiently high expected payoff from s_i and (ii) produce diverse families of trajectories. First, taking W as the set of all possible subsets of S, way-regions can be formalized via functions $b : S \to W$ which are defined for each corridor. Some regions are selected to be avoided $w \in W_{bad}$ and others as (intermediate) goal states $w \in W_{good}$. All $s \in S$ may then be organized into sets:

$$S_{out} = \{s \in w \mid w \in W_{bad}\} \tag{5}$$

$$S_\Omega = \{s \in w \mid w \in W_{good}\} \tag{6}$$

$$S_{in} = \{s \in S \mid s \notin (W_{good} \cup W_{bad})\}$$

Formally, a corridor is then:

Definition 6 (Corridor) *Consider a subset of states $S_{in} \subset S$ with initial way-region $w_0 \subset S_{in}$ and final way region(s) $w_\Omega \in W_{good}$. A corridor is the set of states $S_c \subseteq S$ such that $S_c = S_{in} \cup \{s \in w \mid w \in W_{good}\}$.*

For our case study, the partition of S will be accomplished using square *cells.*

Definition 7 (Cell for Grid Case Study) *A cell centered at s' is described by*

$$c(s') = \{s \in S \mid s'[k] - d \leq s[k] \leq s'[k] + d \quad \forall k\} \tag{7}$$

for selected distance d, where $s[k]$ is the coordinate of s along the k^{th} spatial dimension.

For continuous trajectories, a sequence of cells is used to construct a *continuous corridor*:

Definition 8 (Continuous Corridor) *A continuous corridor of length B is a corridor which is expressible as $S_c = \{s \in c \mid c \text{ in } C\}$, where $C = (c_0, ..., c_B)$ is a sequence of cells in $\mathcal{C}$ and any consecutive cells c_b, c_{b+1} in C are adjacent.*

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 region $S_\Omega \subset S$. For our case study, we select one side of c_B, called a *terminal edge*:

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',*

$$S_\Omega = \{s \in c_B \mid s[k] - s'[k] = \alpha d\} \tag{8}$$

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

Now we consider the **cost** of potential policies. From a state s_i and some optimal benchmark policy π^*, a policy π is 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*

$$\frac{V^\pi(s_i)}{V^*(s_i)} \geq \epsilon. \tag{9}$$

By this definition, any reasonable alternative must have an expectation which is sufficiently close to the benchmark 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 with $C = (c_0, ..., c_B)$ 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 $S_L = S$, $A_L = A$,*

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

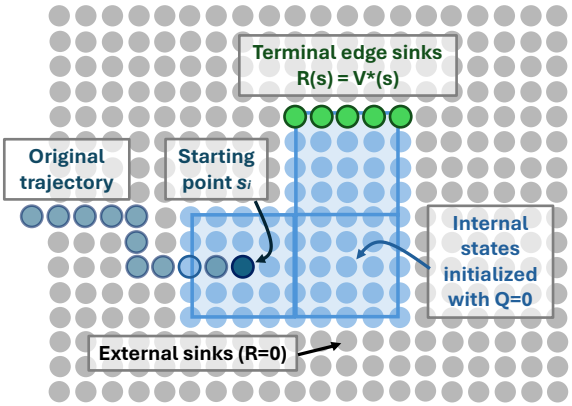

Figure 1: **(Local Q-Learning Illustrated)** Simulated corridor $|C| = 3$ with local MDP reward shaping applied.

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

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

where interior states $S_{in} = \{s \mid s \in (c \setminus S_\Omega), c \in C\}$.

Intuitively, the reward $(1-\gamma)V^*(s)$ is the infinite-trajectory equivalent to $V^*(s)$, as shown in the proof. Now, 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 with 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 $\hat{\pi}$ for the corridor, defined on augmented state $\lambda = (s[0], ..., s[K], \Delta)^T$ takes the piecewise form

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

Intuitively, 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 a benchmark optimal policy π^*. Now, we recall that for an alternative policy $\hat{\pi}$ to be accepted, it must have comparable optimality to π^* in line with (9). 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* $\lambda = (s[0], ..., s[K], \Delta)^T$ *for* $s \in S_{\text{in}}$ *given that* s *has* K *components and the inequality holds pointwise.*

Since our algorithm (Line 19) requires that $V_L^*(s) \geq \epsilon V^*(s)$ for all $s \in S_{in}$, Theorem 13 allows us to claim that $V^{\hat{\pi}}((s, 0)^T) \geq \epsilon V^*(s)$ 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 the full alternative policy $\hat{\pi}$.

Proof for Theorem 13. The full proof is available at https://www.vijethhebbar.com/publication/explainable-q-learning.pdf. As a sketch, the theorem can be proven 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 $\Delta_0 = 0$.
- $\tilde{\mathcal{M}}_\lambda$, where further alterations are made to the transition and reward functions. We show that the value functions for these $\mathcal{M}_\lambda$ and $\tilde{\mathcal{M}}_\lambda$ are identical from any starting state $\lambda_0 = (s_0, 0)^T$.
- $\tilde{\mathcal{M}}_{R_0}$, which is the same as $\tilde{\mathcal{M}}_\lambda$ except for a change to the reward function $\tilde{R}_\lambda$. We show that $\tilde{V}_{R_0}^\pi \leq \tilde{V}_\lambda^\pi$ at all λ.

The final theorem then gives $\tilde{V}_{R_0}^\pi((s_0, 0)^T) \leq V^\pi(s_0)$.

We can also show that, given knowledge of the benchmark 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*

$$\left(V_L^*(s_t) - \frac{\max(r_{in})}{1-\gamma}\right) \frac{1}{\gamma^\tau \max_{s \in S_\Omega} V^*(s)} \leq \mathbb{P}_{success} \tag{10}$$

where $\max(r_{in}) = \max_{s \in S_{in}} R(s)$*,* V^* *is the value function for a benchmark optimal policy and* 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, τ 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 $\tau = \min_{s \in S_\Omega} ||s_t - s||_M$.

4.4. Algorithm: Application and Complexity.

Our algorithm seeks policies by searching over a set of corridors. For the cell-based example here, the number of corridors to check depends on corridor length and cell dimensions. In this demonstration, S_c is created by uniformly placing square cells centered at distance $d-1$ from each other along each spatial dimension k; Figure 1 shows this in $\mathbb{R}^2$. Each cell c_b in a corridor 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

$$n = \sum_{b=0}^{B} (2k)^{(b+1)} \tag{11}$$

Algorithm 1 ϵ-Optimal Alternative Policy Search

Require: Environment env, Q^*, $\epsilon > 0$, s_0, B, policy π.

1: Initialize $corridor\_stack = [(c_0)]$
2: Initialize $required\_corridors = []$
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** $\max\{V^*(s) | s \in terminal\_edge\} < \epsilon V^*(s_0)$ **then**
9: **continue**
10: Initialize $Q = 0$, Set $Q(s, \cdot) = Q^*(s, \cdot)$ 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 π.
15: **if** $s' \notin curr\_corridor$ **then**
16: Set reward 0 for step
17: Perform Q-learning update
18: Set $s = s'$
19: **if** $V(s_0) \geq \epsilon V^*(s_0)$ **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, 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 $(c_0, ..., c_{\beta+1})$ by the optimality principle; 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. Our example will have $k = 2$, resulting in $n = \sum_{b=0}^{B} 4^b$ corridors to consider.

5. Experimental Results

The results of Algorithm 1 are now demonstrated for the Frozen Lake environment of OpenAI Gym. (See Github: n-brindise/div-near-opt-alternatives.) The agent begins at an *initial state* in the top-left corner $(y, x) = (0, 0)$ and attempts to reach the *goal state* at the bottom-right corner $(9, 9)$. *Holes* in the frozen lake are absorbing states with reward 0. There are four actions $\{a_N, a_S, a_E, a_W\}$

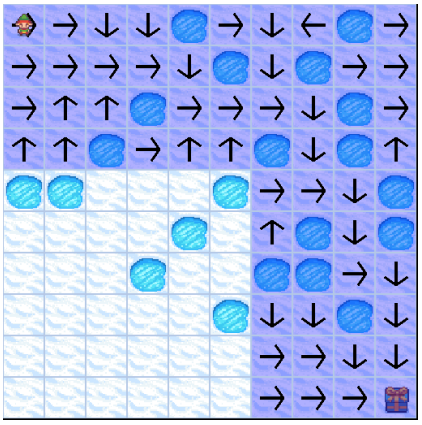

Figure 2: **(Corridor 1 with $\epsilon = 0.99$ sub-optimality.)** Arrows indicate the local policy actions.

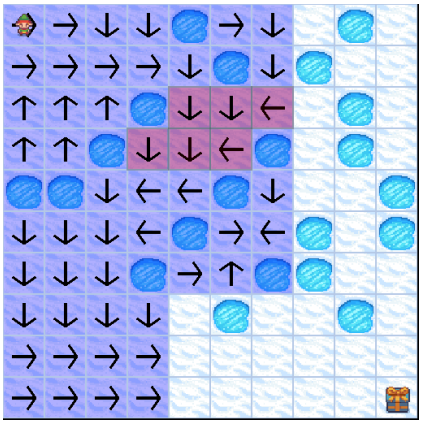

Figure 3: **(Corridor 2 with $\epsilon = 0.9$ sub-optimality.)** Differences from previous corridor are highlighted.

corresponding to steps in cardinal directions $y-1, y+1, x+1, x-1$; the environment dynamics allow for "slipping" via a stochastic transition function. In this application, each action transitions a step in the intended direction with probability 0.9 and left or right of the intended direction each with 0.05. This example used tabular Q-learning; fundamentally, any method may be used which estimates a value function, such as a Deep Q-Network (DQN).

Select experimental results for corridors with 5 cells ($B = 4$) are highlighted in Figures 2 and 3. Towards explanation, suppose a user asks what policy options exist from $(0, 0)$. We first specify $\epsilon = 0.99$, so that all solutions are required to have $V_L^*((0, 0))$ very near the benchmark. This yields one option, shown in Figure 2. Here, the corridor is highlighted in blue, with corresponding local policy π_L shown by arrows; a left arrow indicates action a_W, up arrow a_N, and so on.

Now, supposing lower performance is acceptable, take $\epsilon = 0.90$. This yields a second option (Figure 3). This corridor terminates short of the goal (note that the agent can still *reach* the goal upon exiting the corridor's terminal cell, corresponding to the piecewise policy switch from $\hat{\pi} = \pi_L$ to $\hat{\pi} = \pi^*$). The states highlighted red in Figure 3 have different optimal actions from Figure 2, presumably resulting in more trajectories that proceed to the bottom left as opposed to the top right.

In the interest of safety, we may also assess how consistently any policy will follow a corridor to its terminus, i.e., the agent remains within S_c until reaching S_Ω. In our example, a user might prefer a more costly policy if it avoids the (dangerous) holes with higher probability as a matter of "safety." **Theorem** 14 yields the bounds in the **DNA** entry in Table 1. Note that τ for Corridor 1 is the Manhattan distance between $(0, 0)$ and the nearest state in terminal edge $\{(9, 6), (9, 7), (9, 8), (9, 9)\}$. Similarly for Corridor 2, $\tau = 9$ is the Manhattan distance from $(0, 0)$ to the edge $\{(6, 3), (7, 3), (8, 3), (9, 3)\}$. As reflected in the table, these bounds are indeed much lower than the experimentally-determined expectation for successful traversal.

5.1. Comparison to Quality Diversity

A basic Quality Diversity (QD) example was implemented for comparison with the DNA experimental setting. This was done using the Python QD toolbox pyribs (Tjanaka et al. (2023b)). QD

methods rely on a descriptor function $\mathbf{b} : U \to \mathcal{B}$, where $\mathcal{B}$ is a behavior space and U is an observation of the trajectory space. In our case, $\mathbf{b}(\rho)$ identifies the corridor of length i, $1 \leq i \leq B$ corresponding to each trajectory. The function has the form $\mathbf{b} : \rho \to \{0, ..., 4\}^B$, and the process to map ρ to $\mathcal{C}$ can be summarized as follows: (1) partition S into cells by the same scheme as in DNA, (2) take $c_0 = c$ such that $s_0 \in c$, (3) let $c' = c_0$, and (4) for each s_t in $\rho = (s_0, ..., s_t, ...)$ and current c': if $s_t \notin c'$, set $c_i \leftarrow c$ s.t. $s_t \in c$ and set $c' \leftarrow c_i$. Increment i.

Given this mapping, the corridor descriptor is simply some $(b_1, ..., b_B)$ where $b_i = 0$ if cell c_i does not exist, $b_i = 1$ when cell c_{i-1} is below c_i, $b_i = 2$ when c_{i-1} is right of c_i, and so on. Accordingly, a feature space can be defined on $\{0, ..., 4\}^B$, allowing for 5^{B-1} corridor configurations (when corridors are defined relatively from one origin point). Now, to capture policy costs, a fitness value f_θ is assigned as

$$f_\theta(\rho) = \sum_{t=0} \gamma^t R(s_t) \tag{12}$$

where $\rho = (s_0, ...)$. Then the full QD problem seeks to solve

$$\forall \mathbf{b} \in \mathcal{B} \quad \theta^* = \arg\max_\theta f_\theta \quad \text{s.t.} \quad \mathbf{b} = \mathbf{b}_\theta \tag{13}$$

for each $\mathbf{b}$, where $\mathbf{b}$ is a point in $\mathcal{B}$. In this path-planning domain, the parameters θ correspond to control policy π_θ.

We briefly test whether QD identifies the corridors found by DNA, taking the Covariance Matrix Adaptation MAP-Annealing (CMA-MAE) variants proposed in Tjanaka et al. (2023a). Results are shown in Table 1 for separable CMA (CMA-Sep) over 3000 iterations and Limited Memory Matrix Adaptation (LM-MA) over 6000 iterations. LM-MA is the more successful of the two, identifying a policy which achieved 38% consistency for Corridor 2. However, neither algorithm recovers 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.

Acknowledgments

This research was funded in part by a National Defense Science and Engineering Graduate Fellowship and an Army Educational Outreach Program fellowship. The authors would also like to thank Andres Posada-Moreno for his helpful input.

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.

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.

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 bare-bones 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.