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

Noel Brindise NBRINDI2@ILLINOIS.EDU
Vijeth Hebbar VHEBBAR2@ILLINOIS.EDU
Riya Shah TBD
Cédric 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 (XRL) called Diverse Near-Optimal Alternatives (DNA). DNA is inspired by the field of Quality Diversity (QD), which seeks a set of (locally) optimal but behaviorally diverse solutions in planning and optimization problems. Similarly, DNA seeks a set of reasonable policy “options” for Q-learning agents which result in qualitatively diverse paths over a Markov decision process. These distinct alternatives can then be presented to human users to explain the shape and flexibility of policies, providing them with policy options which may be chosen according to their preferences. This is accomplished in part via a novel method of solving local, modified Q-learning problems to find distinct policies with guaranteed ϵ-optimality. In this work, we establish acceptability criteria for these alternative policies and propose a search algorithm which adapts and extends concepts from QD. We then show that DNA successfully returns a set of qualitatively different policies that constitutes meaningfully different “options” in simulation.

Keywords: Explainable Reinforcement Learning (xRL), Explainable AI for Planning (XAIP), Q-Learning, Quality Diversity (QD)

1. Introduction

The field of explainable AI offers remarkably diverse interpretations of human-friendly “explanation” for system outputs and behavior. Though commonly associated with the interpretation of neural networks, xAI has taken on a variety of approaches and settings, resulting in the niche of explainable planning (XAIP). Generally, XAIP aims to describe plans, trajectories, or policies to characterize the intent or behavior of autonomous agents. This work considers reinforcement learning on a Markov decision process (MDP), specifically leveraging value functions of Q-Learning agents. While these agents typically seek one, globally optimal policy, we observe that additional policies with similar expected cost may exist. Not all such policies are necessarily interesting, since differing policies may still produce qualitatively similar trajectories. However, in the case that some near-optimal policies produce trajectories which are sufficiently different from the optimal policy, they may constitute interesting alternative “options.”

This scenario opens an avenue for *explanation via alternatives*. Suppose that a human is presented with the optimal policy and corresponding trajectory. Firstly, if the human is dissatisfied with the plan (for instance, if it passes through risky terrain or an unwanted waypoint), it would be useful to provide the human with options taking other routes. Secondly, the “flexibility” of a presented plan is unclear a priori. Given a trajectory, some states along the way may leave only one reasonable choice of action. For example, the agent may be confined to a valley where most motion is restricted. If the agent is at the edge of a cliff, any off-nominal action leads to a costly tumble. Indeed, existing explanatory methods seek out such "critical states"Huang et al. (2018). In contrast, the plan may be considered "flexible" from some state if multiple policies are available which offer distinct and near-optimal trajectories.

In this work, we pursue an explanatory method which searches for diverse, near-optimal alternatives, aiming to answer questions such as "what policies/paths can I reasonably take from here?" and "how will the cost compare?" Our method assesses the expected cost of alternative policies by incorporating the known (or estimated) Q values of the optimal policy. To ensure that policies produce qualitatively distinct trajectories, we introduce a notion of "corridors," which may be viewed as an adaptation and extension of methods from Quality Diversity; we will describe Quality Diversity and outline the relevant areas of xAI next.

2. Previous Work

Explainable Reinforcement Learning (XRL) and Explainable AI for Path Planning (XAIP), both of which overlap considerably with Explainable Artificial Intelligence (XAI), have produced many diagnostic and control-design approaches to improve human-agent understanding. While these fields have yet to identify unified goals or metrics for "explainability," distinct branches have appeared with shared techniques and criteria Milani et al. (2022), Chakraborti et al. (2020), Vouros (2022). At the highest level, these include *interpretable design* and *post-hoc* methods.

[Riya's contributions right around here]

Designing Interpretable Agents. Many approaches (re)construct the RL framework or algorithm itself to be interpretable, or inherently more explainable. These range from minor environment modifications to significant learning algorithm alterations. Reward decomposition, for example, separates the reward into 'types,' and the weighted values are used to show which categories of reward contributed more or less to the optimal policy Juozapaitis et al. (2019). In deep RL, deep models have been used to train more transparent models such as decision trees, replacing inscrutable algorithms with relatively simple decision rules Engelhardt et al. (2023).

Post-hoc Explanation Methods. Other approaches apply explanatory frameworks to RL agents post-hoc. For the black-box setting, only state/action information must be supplied to the explainer Brindise and Langbort (2023), Movin et al. (2023); such methods are highly adaptable, but they cannot reason directly from the agent's policy and are thus purely descriptive. Another family of post-hoc diagnostic assesses the relative influence of inputs on the policy's trajectories or success. Here, inputs may be actions or state-action pairs; for example, identifying interesting moments in agent runs based on rewards earned Pierson et al. (2023) or analyzing causal relationships between variables of interest for model-free RL Madumal et al. (2020). Others build memory-based models which estimate the influence of (partial) trajectories on success probability Cruz et al. (2019) or policy shape Deshmukh et al. (2022). In the multiagent setting, Shapley values have been used to estimate the proportional contributions of each agent across many episodes Heuillet et al. (2022).

Spotlight: Alternative Trajectories and Quality Diversity. Many works in XAIP seek alternative trajectories or domain changes which can "explain" a policy by comparing it to hypothetical alternatives. Users may suggest a specific trajectory or waypoint(s) for analysis Movin et al. (2023) Beyret et al. (2019). Explanation may then address infeasibility (Alsheeb and Brandão (2023)) or contextualize the suggestion, e.g. in a hypothetical environment where the user suggestion would be optimal Brandão and Setiawan (2022); Finkelstein et al. (2022).

To identify or generate acceptable alternatives, the field of **Quality Diversity** (QD) becomes relevant. QD has developed a variety of algorithms to search for a set of near-optimal ("quality"), qualitatively-distinct ("diverse") solutions. In the path planning context, classic QD seeks to find a set of policies

[Quality Diversity careful review] Macé et al. (2023) says: The Quality Diversity Transformer. Uses MAP-Elites Low Spread (ME-LS) which "constrains the set of solutions to those that are the most consistent in the behavior space". Second, Quality-Diversity Transformer (QDT). Current issue with QD is policies don't produce consistent behaviors. They introduce a policy "spread" metric to measure consistency of policies in BD space; propose MAP-Elites Low Spread.

Other works have attempted to find average fitness over multiple episodes for each policy (MAP-Elites-sampling).

Define spread of K trajectories as the mean distance between all pairs of behavior descriptors to estimate spread of solution θ (given policy π_θ).

Fitness and BD are calculated as average and mode over E trajectories, respectively. Insert soln only if its fitness is higher and its spread is lower than corresponding soln in BD niche.

to be remodeled: Altogether, current works in XRL mainly focus either on summary, collecting trajectory highlights and replaying them; causality, modeling how (a sequence of) state-action pairs or environment changes affect trajectory segments; or reward analysis, breaking down the reward or distributing goals to decompose the influences on a trajectory Dazeley et al. (2023). Largely distinct from these approaches, we fix our ultimate goal and environment and instead seek a set of sufficiently-optimal, distinct trajectory families which are attainable from a given point of interest. We use this method to answer the relatively under-explored question "what are my options from here?"

3. Background

We aim to generate explanations for agents which (i) operate on an environment expressed as a *Markov decision process* and (ii) use *Q functions* to define their control policies. Throughout this work, we will use superscripts to denote "named" constants and subscripts to 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.*

In our setting, we require that $R(\cdot, \cdot) \geq 0$. Furthermore, as we aim to explain contiguous trajectories in physical space, we conduct this study with the spatially intuitive MDP on a grid, a common setting in gridworld-based reinforcement learning.

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 $s' \in \mathcal{N}(s)$.

Before we move into describing our proposed algorithm let us present some standard definitions surrounding MDPs.

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,

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

Given Q^*, the optimal action to take at every state can be captured through the optimal policy:

Definition 4 (Optimal Policy) *A policy $\pi : S \to A$ is optimal subject to Q if, for all states $s \in S$,*

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

The optimal policy has an associated value function:

Definition 5 (Optimal Value Function) *The optimal value function $V^* : S \to \mathbb{R}$ is defined for state $s \in S$ as*

$$V^*(s) = \max_{a \in A} Q^*(s, a). \tag{4}$$

Lastly, for any policy π, we can define the value function for π as

Definition 6 (Value Function for Policy π) *The value function $V^\pi : S \to \mathbb{R}$ is defined for state $s_t \in S$ as*

$$V^\pi(s_t) = R(s_t, \pi(s_t)) + \gamma \sum_{s' \in S} T(s_t, a', s') V^\pi(s'). \tag{5}$$

We now move into the discussion of our proposed algorithm.

4. Alternative Policies via Corridor Search

In this work, we propose an algorithm that characterizes the alternative policy "options" from a chosen state by checking for policies which are both (i) comparable in cost to the global optimal policy, and (ii) produce "geometrically distinct" families of trajectories, both of which we will define in the following subsection. We begin with an intuitive motivation, move to definitions, and finally present the "corridor search" algorithm.

4.1. Motivation: Explanation for Humans through "Corridors"

We suppose that an agent has reached a particular state and a human observer is curious about geometrically distinct, reasonable cost policy options to continue the trajectory ("what can we do from here?"). Here, we introduce a simplified framework to quantify qualitative distinctions between possible policies, introducing a concept of "corridors."

As a first-order assessment, we consider the case of distinct individual trajectories. Ideally, policies could be found which each produce a specific trajectory passing through distinct waypoints, resulting in different sequences of states – "distinct" paths. However, in the stochastic MDP environment, there is no one-to-one control over trajectories through the state space; instead, any policy π produces a family of trajectories whose shapes depend on probabilistic transition function T. Thus, training policies to achieve particular paths over S is typically limited to methods of reward shaping, where R is altered to favor selected waypoints s and draw trajectories there (when possible). In these cases, the questions remain which waypoint(s) to select, and how these will alter cost.

Rather than select individual waypoints to reward from s_i, we suggest that the reward-shaping process may be reduced for the context of explanation:

1. **Waypoints $\rightarrow$ Way-regions:** rather than selecting s individually, we select larger *way-regions* $W \subset S$ to be rewarded.
2. **Sequences of Waypoints $\rightarrow$ Corridors:** to reward differing geometric shapes for trajectories, we replace a sequence of multiple waypoints with a *corridor*, a selected region of S outside of which the reward is altered.

Here, we assume that individual states are not qualitatively important to a human user, but rather state-space regions whose sizes may be adjusted; informally, we suppose a human is interested in whether a trajectory travels from some "here to there," rather than "s^1 to s^2." In this case, it is excessive to set individual states as intermediate goals; instead, we suggest that an intermediate goal region and designated set of states through which to arrive there is sufficient to geometrically characterize a "type" of policy.

Thus motivated, we introduce our simplified approach to the search for alternative policies via *corridor search*: given a starting state s_i, we systematically check corridors of states which start at s_i and terminate at some W, optimizing a local policy via corridor-based reward shaping. The resulting policies represent options from s_i which are distinct and have a reasonable cost, providing the human an overview of their choices.

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 families of trajectories which are sufficiently geometrically distinct from each other. We begin with the discretization of S into *cells:*

Definition 7 (Cell) *A cell centered at s' is defined as*

$$c(s') = \{s \in S \mid s'[k] - d \leq s[k] \leq s'[k] + d \quad \forall k\} \tag{6}$$

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

If a human user selects d such that states in each c are considered qualitatively similar, then distinctions between $s \in c$ are explanatorily arbitrary, making c a qualitatively-meaningful region. When the entire state space is grouped into cells, it is called *complete discretization:*

Definition 8 (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 9 (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 $c_i \neq c_j$. At the end of a corridor, we select a *terminal edge* as our way-region:

Definition 10 (Terminal Edge) *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',*

$$S_\Omega = \{s \in c_B \mid s[k] - s'[k] = \alpha d\} \tag{7}$$

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

A last issue to be addressed is 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 11 (ϵ-Optimal Policy) *A policy π with corresponding value function V^π is ϵ-optimal if for a given $\epsilon \geq 0$ it satisfies*

$$V^\pi(s_i) \geq V^*(s_i) - \epsilon. \tag{8}$$

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 of interest. Informally, given a corridor $C = (c_0, ..., c_B)$ and starting state $s_0 \in c_0$, suppose an agent only receives a reward when remaining within the corridor or exiting through terminal cell c_B. We seek a policy which will still achieve a sufficient reward, in particular if any trajectories exiting the corridor at a non-terminal edge immediately terminate. In this subsection, we will establish this local Q problem and then examine the resulting guarantees.

Definition 12 (Local Q Problem) *For corridor $C = (c_0, ..., c_B)$ and terminal edge $S_\Omega \in \mathcal{E}^C$ the local Q problem is a Q-learning problem solving for optimal local policy π_L on MDP $\mathcal{M}_L$ with $S_L = S$, $A_L = A$,*

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

where δ_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 the notion of a local optimal policy π_L defined, we can define a related policy on the global MDP:

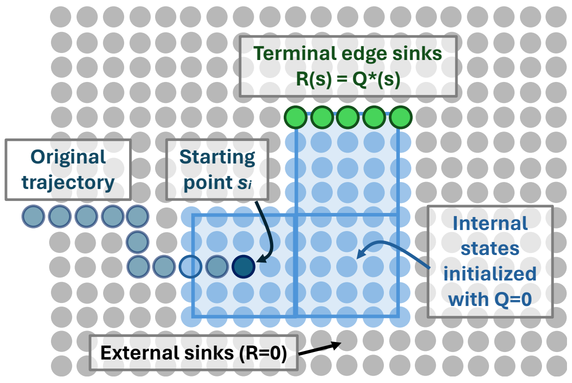

Figure 1: **(Local Q-Learning Illustrated)** Simulated corridor $|C| = 3$ with local MDP reward shaping applied.

Definition 13 (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*

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

Then an alternative policy for corridor C, defined on augmented state $\tilde{s} = \langle s[0], ..., s[k], \Delta \rangle$ 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 (8).

Theorem 14 (ϵ-Optimality Guarantee) *Let V_L^* be the value function corresponding to the local Q-learning problem in Definition 12. Then we have*

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

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

Since our algorithm guarantees (Line 19) that $V_L^*(s) \geq V^*(s) - \epsilon$ for all $s \in S_{in}$, Theorem 14 allows us to claim that $V^{\hat{\pi}}(\langle s, 0 \rangle) \geq V^*(s) - \epsilon$ 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 $\hat{\pi}$.

4.4. Algorithm: Application and Complexity.

The corridor search algorithm seeks suitable alternative policies for a human user by creating corridors and calculating an expected reward for their use. Clearly, the number of policies presented as "options" will depend on the number of corridors checked. Here, the family of possible corridors is dependent on their maximum length B, the cell dimension d, and the cell discretization scheme of S.

As a first-order approach, we apply a discretization scheme which uniformly places cells centered at distance $2d$ from each other along each dimension k; a 2D result is shown in Figure 1. This clearly ensures that the resulting cell set $\mathcal{C}$ satisfies completeness by Definition 8.

Regarding corridors, we recall that c_b, c_{b+1} in C must be adjacent. Our chosen scheme results in c which are n-dimensional (hyper)cubes with exactly one adjacent c' per (hyper)face; thus, for a corridor segment ending in cell c_b, there are exactly $2k$ possible c_{b+1} to extend the corridor. For each corridor, there are $2k$ terminal edges, corresponding to $2k$ separate local Q problems. Therefore, given a starting state $s_i \in c_0$, and state space dimension K, there are

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

potential local Q problems to be solved. To further mitigate complexity, however, we note that if the policy corresponding to a corridor $(c_0, ..., c_\beta)$ is not $\epsilon-$optimal for any terminal edge of c_β (Line 19), there is no $\epsilon-$optimal policy for any extension $(c_0, ..., c_{\beta+1})$; thus, the search for all C beginning with segment $(c_0, ..., c_\beta)$ may be truncated, eliminating $\sum_{b=0}^{B-\beta}(2k)^{b+1}$ local problems. We also eliminate any 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 thus the same Q learning problem.

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 running Algorithm 1 on 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. The task of the agent, an 'elf' in this environment, is to begin from the *initial state* at the top-left corner and reach the *goal state* at the bottom-right corner of the domain. Meanwhile, the elf must carefully navigate across the 'frozen lake,' avoiding *holes* modeled as absorbing states, while slipping is possible via a stochastic transition function. Here, four actions a_N, a_S, a_E, a_W corresponding to cardinal directions are available; each transitions one step 'forward' (i.e., in the intended direction) with probability 0.9, and left or right of the intended direction with probability 0.05.

Figure 2 shows the corridor and the corresponding local policy (as arrows) generated by Algorithm 1 when allowing $\epsilon = 0.99$ level of sub-optimality in Q_L. This means that taking the local policy, where the agent is penalized for exiting the corridor, still expects a reward that is 99% of that for the global optimal policy; by our guarantees, this implies the true expectation from the initial state is at least as good. Intuitively, if the corridor is exited, the elf will take the global optimal policy from there; this cannot be worse than the immediate sink simulated in the local problem.

Returning to our ultimate goal, however, we still wonder: "what are our options?" For this selected value of ϵ, the algorithm has only recovered only one corridor that is ϵ-optimal. If we are willing to relax our criterion to $\epsilon = 0.9$, we receive another option which is clearly geometrically distinct. Figure 3 shows the additional corridor and associated policy that are returned. Qualitatively, it can be seen that this corridor describes longer trajectories than those in Figure 2; thus, it

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 $s \to s'$ 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 $required\_corridor$
```

is not surprising that this corridor was only suggested for a larger degree of sub-optimality. However, a human observer may still prefer to take the policy recommended in Figure 3 over the one from Figure 2 for qualitative reasons, as the longer corridor is more sparsely littered with dangerous holes. Indeed, the alternative policy may achieve the goal with a higher probability, a qualitative criterion for "safety," while still offering sufficient optimality.

Finally, we examine the local policies for each corridor, shown as arrows in Figures 2 and 3, and find that the states highlighted in red in Figure 3 differ in the two corridors. It is important to note that, while our approach is able to generate ϵ-optimal policies, it is unique for its ability to ensure that these policies are sufficiently *distinct*; by providing a corridor associated with each policy, the algorithm is capable of qualifying the difference in the two policies in an explainable geometric way, rather than only via optimality.

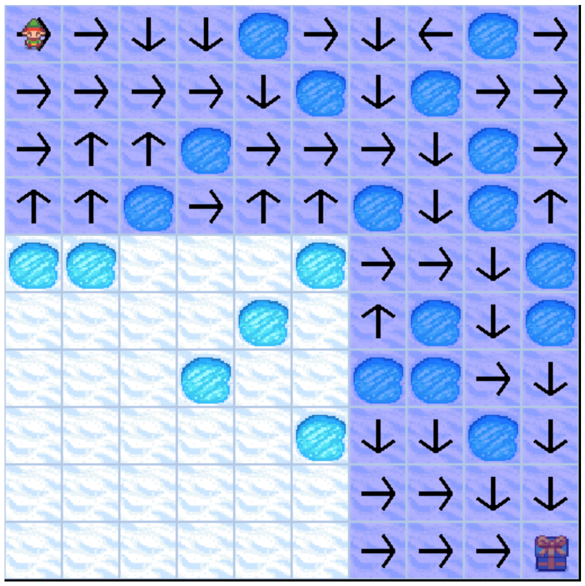

Figure 2: **(Sole corridor at $\epsilon = 0.99$ sub-optimality)** The corridor (shaded blue) and the corresponding policy (arrows) that guarantee 99% optimality.

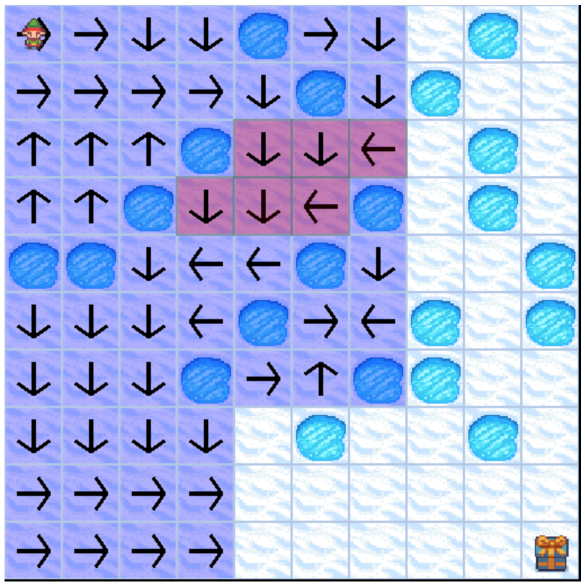

Figure 3: **(Additional corridor at $\epsilon = 0.9$ sub-optimality)** The region highlighted in red emphasizes the change in policy required to follow the new corridor.

6. Conclusion and Future Work

The corridor search algorithm for Q-learning based agents produced qualitatively distinct policies, successfully identifying "alternative options" from a point of interest on an MDP. These policies defined local alternatives corresponding to different regions of the state space, producing trajectory families which were visibly different. Future work would further 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.

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.

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.

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.

Richard Dazeley, Peter Vamplew, and Francisco Cruz. Explainable reinforcement learning for broad-xai: a conceptual framework and survey. *Neural Computing and Applications*, 35(23): 16893–16916, 2023.

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.

Raphael C Engelhardt, Marc Oedingen, Moritz Lange, Laurenz Wiskott, and Wolfgang Konen. Iterative oblique decision trees deliver explainable rl models. *Algorithms*, 16(6):282, 2023.

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.

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.

Zoe Juozapaitis, Anurag Koul, Alan Fern, Martin Erwig, and Finale Doshi-Velez. Explainable reinforcement learning via reward decomposition. In *IJCAI/ECAI Workshop on explainable artificial intelligence*, 2019.

Valentin Macé, Raphaël Boige, Felix Chalumeau, Thomas Pierrot, Guillaume Richard, and Nicolas Perrin-Gilbert. The quality-diversity transformer: Generating behavior-conditioned trajectories with decision transformers. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, page 1221–1229, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701191. doi: 10.1145/3583131.3590433. URL https://doi.org/10.1145/3583131.3590433.

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.

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.

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

Appendix A. Proof of Theorem 14

Let $\mathcal{M} = \langle S, A, T, R, \gamma \rangle$ be an MDP and let $V^\pi : S \to \mathbb{R}$ be the value function corresponding to some policy π for it and let V^* be the optimal policy. Let $(\underline{\mathrm{S}}, \bar{S})$ be a partition of S and define a new

MDP $\mathcal{M}' = \langle S, A, T', R', \gamma \rangle$ where

$$T'(s,a,s') = \begin{cases} T(s,a,s') & s \in \bar{S} \\ \delta_{s=s'} & s \in \underline{S}, \end{cases} \tag{10}$$

$$R'(s,a) = \begin{cases} R(s,a) & s \in \bar{S} \\ (1-\gamma)V^\pi(s) & s \in \underline{S}. \end{cases} \tag{11}$$

Let V'^π be the value function corresponding to policy π over MDP $\mathcal{M}'$. Then we can claim the following

Lemma 15 $V'^\pi = V^\pi$

Proof: Let $V^0 = V^\pi$ and let us now perform value iteration in an attempt to arrive at the value function corresponding to policy π for $\mathcal{M}'$. For any $s \in S$ we have

$$\begin{aligned} V^1(s) &= R'(s,\pi(s)) + \gamma \sum_{s' \in S} T'(s,\pi(s),s')V^0(s') \\ &\overset{(a)}{=} \begin{cases} (1-\gamma)V^\pi(s) + \gamma V^\pi(s) & s \in \underline{S} \\ R(s,\pi(s)) + \gamma \sum_{s' \in S} T(s,\pi(s),s')V^\pi(s') & s \in \bar{S}. \end{cases} \\ &\overset{(b)}{=} \begin{cases} V^\pi(s) & s \in \underline{S} \\ V^\pi(s) & s \in \bar{S}. \end{cases} = V^0(s) \end{aligned}$$

where equality (a) results from (11) and (10), while (b) results from Definition 6 for $\mathcal{M}$. We see that value iteration converges to V^π in a single step. □

By further partitioning $\underline{S}$ as $\underline{S}^e$ and $\underline{S}^t$, we may construct another MDP $\mathcal{M}_L = \langle S, A, T_L, R_L, \gamma \rangle$ with

$$T_L = T' \tag{12}$$

$$R_L(s,a) = \begin{cases} R(s,a) & s \in \bar{S} \\ (1-\gamma)V^*(s) & s \in \underline{S}^e \\ 0 & s \in \underline{S}^t \end{cases} \tag{13}$$

Let V_L be the optimal value function and let π_L be the corresponding optimal policy. Recalling that all $R(\cdot,\cdot) \geq 0$, we know that $V^* \geq 0$. Then that $\mathcal{M}_L$ is obtained from $\mathcal{M}'$ by reducing the rewards of select states and holding the rest constant. Thus, the optimal value function at every state can only be lower than when policy π_L is implemented over $\mathcal{M}'$. In other words

$$V_L \leq V'^{\pi_L} = V^{\pi_L}$$

where the last equality comes from Lemma 15. Noting that $\bar{S} = S_{in}$, $\underline{S}^e = S_\Omega$ and $\underline{S}^t = S \backslash (S_\in \cup S_\Omega)$ completes the proof.