Option Discovery in the Absence of Rewards with Manifold Analysis

Amitay Bar [1] **Ronen Talmon** [1] **Ron Meir** [1]

Abstract

Options have been shown to be an effective tool in reinforcement learning, facilitating improved exploration and learning. In this paper, we present an approach based on spectral graph theory and derive an algorithm that systematically discovers options without access to a specific reward or task assignment. As opposed to the common practice used in previous methods, our algorithm makes full use of the spectrum of the graph Laplacian. Incorporating modes associated with higher graph frequencies unravels domain subtleties, which are shown to be useful for option discovery. Using geometric and manifold-based analysis, we present a theoretical justification for the algorithm. In addition, we showcase its performance in several domains, demonstrating clear improvements compared to competing methods.

1. Introduction

Reinforcement learning (RL) has attracted much attention in recent years thanks to its success in solving a broad range of challenging tasks. Options (a.k.a. skills) play an important role in RL (Sutton et al., 1999) and have opened the door to a series of studies demonstrating improvement in both learning and exploration (Vezhnevets et al., 2017; Nachum et al., 2018; Eysenbach et al., 2019; Tang et al., 2017; Mannor et al., 2004; Menache et al., 2002). One important class of options consists of options that are not associated with any specific task and are acquired without receiving any reward. Such generic options often lead to efficient learning in various tasks that are not known a-priori, (e.g., (Eysenbach et al., 2019)).

An effective approach to build such options is based on spectral graph theory, assuming a finite state domain in which each state is regarded as a node of a graph, and the graph edges represent the states connectivity. Such an approach led to the introduction of proto-value functions (PVFs)(Mahadevan & Maggioni, 2007), which are the eigenvectors of the graph Laplacian (Chung & Graham, 1997). It was shown that the PVFs establish an efficient representation of the domain. Recently, these PVFs were used for options representation (Machado et al., 2017; 2018). There, eigenoptions were introduced by considering only the dominant eigenvectors (PVFs), where each eigenoption is formed based on a single eigenvector. In a related work, Jinnai et al. (2019) presented cover options using only the Fiedler vector multiple times. On the one hand, option discovery with a graph-based representation is a powerful combination, since it facilitates options that are not task or reward-specific, yet it naturally incorporates the geometry of the domain. On the other hand, existing methods are based only on a single eigenvector or consider only the dominant eigenvectors while omitting the rest, leaving room for improvement and further investigation.

In this paper, we present a new scheme for defining options, relying on all the eigenvectors of the graph Laplacian. More concretely, we form a score function built from the eigenvectors, from which options can be systematically derived. Since the agent acts without receiving reward, it is only natural to discover and analyze the options considering the geometry of the domain. For analysis purposes, we model the domain as a manifold and consequently the graph as a discrete approximation of the manifold, allowing us to incorporate concepts and results from manifold learning, such as the diffusion distance (Coifman & Lafon, 2006). We show that our options lead to improved performance both in learning and exploration compared to the eigenoptions as well as other option discovery schemes.

Our main contributions are as follows. First, we present a new approach to principled option discovery with a theoretical foundation based on geometric and manifold analysis. Second, this analysis includes novel results in manifold learning involving two key components: the stationary distribution of a random walk on a graph and the diffusion distance. To obtain these results, we employ a new concept in manifold learning, in which the entire spectrum of the underlying graph is considered rather than only its leading components. Third, we propose an algorithm for option discovery, applicable in high-dimensional determin-

[1]Viterbi Faculty of Electrical Engineering, Technion, Israel Institute of Technology . Correspondence to: Amitay Bar <amitayb@campus.technion.ac.il>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

istic domains. We empirically demonstrate that the learning performance obtained by our options outperforms competing options on three small-scale domains. In addition, we show extensions to stochastic domains and to large scale domains.

2. Background

2.1. RL and Options

We use the Markov decision process (MDP) framework to formulate the RL problem (Puterman, 2014). An MDP is a 5-tuple $\langle \mathbb{S}, \mathbb{A}, p, r, \gamma \rangle$, where $\mathbb{S}$ is the set of states, $\mathbb{A}$ is the set of actions, p is the transition probability such that $p\left(s'|s,a\right)$ is the probability of moving from state s to state s' by taking an action a, $r(s,a,s')$ is the reward function and $\gamma \in [0,1)$ is a discount factor. Consider an agent operating sequentially so that at time step n it moves from state s_n to state s_{n+1}, receiving a reward $R_{n+1} = r(s_n, a, s_{n+1})$. Its goal is to learn a policy $\pi : \mathbb{S} \times \mathbb{A} \to [0,1]$ which maximizes the expected discounted return $G_n \triangleq \mathbb{E}_{\pi,p}[\sum_{k=0}^{\infty} \gamma^k R_{n+k+1} | s_n]$.

An option is a generalization of an action (also known as a skill or a sub-goal) (Sutton et al., 1999). Formally, an option o is the 3-tuple $\langle \mathbb{I}, \pi_o, \beta \rangle$ where $\mathbb{I}$ is an initiation set $\mathbb{I} \subseteq \mathbb{S}$ (the states at which the option can be invoked), $\pi_o : \mathbb{S} \times \mathbb{A} \to [0,1]$ is the policy of the option to be followed by the agent, and $\beta : \mathbb{S} \to [0,1]$ is the termination condition. By following an option o the agent chooses actions according to the policy of the option π_o until the option is terminated according to the termination condition β.

2.2. Diffusion Distance

The diffusion distance is a notion of distance between two points in a high-dimensional data set (Coifman & Lafon, 2006), where the points are assumed to lie on a manifold. It is widely used in many data science applications, e.g., in Mahmoudi & Sapiro (2009); Bronstein et al. (2011); Lafon et al. (2006); Liu et al. (2009); Lederman & Talmon (2018); Van Dijk et al. (2018), since it captures well the geometric structure of the data. While the formulation of diffusion distance is typically general, here we describe it directly in the MDP setting.

Consider a graph $G = (\mathbb{S}, \mathbb{E})$, where the finite set of states $\mathbb{S}$ is the node set and the edge set $\mathbb{E} \subset \mathbb{S} \times \mathbb{S}$ consists of all possible transitions between states. Define a random walk on the graph with transition probability matrix $\boldsymbol{W}$, defined by $\boldsymbol{W}_{ij} = p(s_{t+1} = i | s_t = j)$. Let $\boldsymbol{p}_t^{(l)}$ denote the vector of transition probabilities from state l to all states in t random walk steps defined by the lth column of $\boldsymbol{W}^t$. Throughout the paper the convention is a column-vector representation. With the above preparation, the diffusion distance is defined by

$$D_t\left(s,s'\right) \triangleq \|\boldsymbol{p}_t^{(s)} - \boldsymbol{p}_t^{(s')}\|,$$

where $\|\cdot\|$ is the L_2 norm. In contrast to the standard Euclidean distance, the diffusion distance does not depend solely on two individual points, namely, s and s', but takes into account the structure of the entire data sets. See a prototypical demonstration in the supplementary material (SM). Broadly, in short distances it is closely related to the geodesic distance (shortest path) (Portegies, 2016) and in long distances it demonstrates high robustness to noise and outliers (Coifman & Lafon, 2006). For more details on the advantages of the diffusion distance and its efficient computation using the eigenvectors of the graph Laplacian see the SM.

3. Diffusion Options

In standard, mostly goal-oriented RL, one learns to map states to actions in order to achieve a desired task. In situations with uncertainty (e.g., model uncertainty, reward uncertainty, etc.) exploration is essential in order to reduce uncertainty, thereby improving future actions. Exploration often consists of aspects that are specific to a given task, and aspects that are generic to the domain. For example, in an environment with multiple rooms, one may wish to learn how to reach the door of each room, thereby facilitating learning in later situations where a specific task is given, say, reaching a specific room (or set of rooms). This may also be useful if additional rooms are later added. In both cases (task-based or task-free), options can greatly facilitate the speed of exploration by forming shortcuts (Eysenbach et al., 2019). In this work we present a manifold-based approach to developing generic options that can be later used across multiple task domains.

To encourage exploration, a useful set of options will lead the agent to distant regions, visiting states that the uninformed random walk will seldom lead to. To this end, we exploit the diffusion distance and show that the strength of diffusion distance in the realm of high dimensional data analysis enables us to devise structure-aware options that improve both learning and exploration.

3.1. Algorithm

The derivation of the algorithm for option discovery is carried out in a setting consisting of discrete and deterministic domains with a finite number of states, where the transitions between states are known. This allows us to focus on the development of the representation of the domain with multiple spectral components and on the analysis based on the interface of spectral graph theory and the diffusion distance. Nevertheless, the primary interest is large scale domains, which are only partially known a-priori. In Section 3.2, by

relying on previous work, we show how to accommodate such domains.

The proposed algorithm for systematic option discovery consists of two stages. The first stage involves graph construction. Let G be a graph whose node set is the finite set of states $\mathbb{S}$. Let $\boldsymbol{M} \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{S}|}$ be the symmetric adjacency matrix of the graph, prescribing the possible transitions between states, namely $\boldsymbol{M}_{s,s'} = 1$ if a transition from state s to state s' is possible, and $\boldsymbol{M}_{s,s'} = 0$ otherwise. Based on $\boldsymbol{M}$, define a non-symmetric lazy random walk matrix $\boldsymbol{W} \triangleq \frac{1}{2}(\boldsymbol{I} + \boldsymbol{M}\boldsymbol{D}^{-1})$, where the degree matrix $\boldsymbol{D}$ is a diagonal matrix whose diagonal elements equal the sum of rows of $\boldsymbol{M}$. Applying eigenvalue decomposition to $\boldsymbol{W}$ yields two sets of left and right eigenvectors, denoted by $\{\phi_i\}$ and $\{\tilde{\phi}_i\}$, respectively, and a set of real eigenvalues $\{\omega_i\}$. The sth component of ϕ_i is denoted by $\phi_i(s)$.

The second stage of the algorithm relies on the following score function, $f_t : \mathbb{S} \to \mathbb{R}$, defined on the set of states $\mathbb{S}$, and assigns a score to each state $s \in \mathbb{S}$

$$f_t(s) \triangleq \| \sum_{i\geq 2} \omega_i^t \phi_i(s)\, \tilde{\phi}_i \|^2, \tag{1}$$

where $t > 0$ is a scale parameter representing the diffusion time. By construction, $f_t(s)$ consists of the full spectrum of $\boldsymbol{W}$, including both low and high frequencies, in contrast to common practice. As we show in Proposition 1, $f_t(s)$ is directly related to the average diffusion distance between state s and all other states, making it a promising candidate for an option discovery criterion, as discussed below.

After computing $f_t(s)$, the states at which it attains a local maximum are extracted. We term these states option goal states, and denote them by $\{s_o^{(i)}\}$, where the index i ranges between 1 and the number of local maxima. Each such state is associated with an option, which leads the agent from its current state to the option goal state. The options can start at any state ($\mathbb{I} = \mathbb{S}$), and terminate deterministically once reaching its option goal state, i.e. for option i, $\beta_i\left(s_o^{(i)}\right) = 1$, and $\beta_i(s) = 0 \,\forall s \neq s_o^{(i)}$. In other words, once the agent chooses to act according to an option, it moves to s_o via the shortest path from its current position. We note that the scale parameter t indirectly controls the number of options; since $0 < \omega_i \leq 1$ (Chung & Graham, 1997), the multiplication by ω_i^t in (1) makes $f_t(s)$ smoother as t increases, analogously to a low pass filter effect. In addition, many eigenvalues are often negligible, and therefore, accurate reconstruction of $f_t(s)$ in (1) typically does not require all the spectral components.

The proposed algorithm for option discovery appears in Algorithm 1. We term the discovered options *diffusion options* because they are built from the eigenvalue decomposition of a discrete diffusion process, i.e., the lazy random walk on the graph. In addition, in Section 3.3, we show a tight relation to the diffusion distance. Algorithm 1 exhibits several advantages. First, the algorithm prescribes a systematic way to derive options which are not associated with any particular task or reward. Second, we empirically demonstrate the acceleration of the learning process and more efficient exploration in prototypical domains compared to competing methods for option discovery. Third, the computationally heavy part is performed only once and in advance. Fourth, the scale parameter t enables to control the number of options and facilitates multiscale option discovery.

We remark that the eigenvalue decomposition of $\boldsymbol{W}$ used for the construction of $f_t(s)$ is related to the eigenvalue decomposition of the normalized graph Laplacian $\boldsymbol{N}$, which traditionally forms the spectral decomposition of a graph. See the SM for details.

Algorithm 1 Diffusion Options

Input: Adjacency matrix $\boldsymbol{M}$ and scale parameter $t > 0$
Output: K options with policies $\{\pi_o^{(i)}\}_{i=1}^K$

1: Compute the degree matrix $\boldsymbol{D}$ from $\boldsymbol{M}$
2: Compute the random walk matrix $\boldsymbol{W} = \frac{1}{2}(\boldsymbol{I} - \boldsymbol{M}\boldsymbol{D}^{-1})$
3: Apply EVD to $\boldsymbol{W}$ and obtain $\{\phi_i\}$, $\{\tilde{\phi}_i\}$ and $\{\omega_i\}$
4: Construct $f_t(s) = \| \sum_{i\geq 2} \omega_i^t \phi_i(s)\, \phi_i \|^2$
5: Find the states $\{s_o^{(i)}\}_{i=1}^K$ of the local maxima of $f_t(\cdot)$
6: **for** $i \in \{1, \ldots, K\}$ **do**
7: Build an option with policy $\pi_o^{(i)}$ s.t. it leads to $s_o^{(i)}$
8: **end for**

3.2. Extension to Large Scale Domains

The exposition thus far focused on domains, whose full transition matrix is at hand when learning the representation of the domain. Suppose now that the considered set of states $\mathbb{S}$ is only a subset of the entire set of states. The extension of diffusion options discovered by Algorithm 1 to unseen states $s \notin \mathbb{S}$ requires the extension of $f_t(s)$ and the extension of the option policies. Since the option policies can be trained off-policy as shown by Jinnai et al. (2020), here we focus on extending $f_t(s)$.

In the SM, we show that the extension of $f_t(s)$ to unseen states $s \notin \mathbb{S}$ involves the extension of the eigenvectors ϕ and $\tilde{\phi}$ taking part in the construction of $f_t(s)$ in (1). Since the eigenvectors admit a particular algebraic structure, their extension is naturally regularized, and therefore, often more accurate than a generic function extension. This fact was recently exploited by Wu et al. (2019), who developed a method that was later demonstrated by Jinnai et al. (2020), to compute the eigenvectors of the graph Laplacian in large scale domains. Another approach for extending the eigenvectors was proposed by Machado et al. (2018) using deep

successor representation. We note that due to the low pass filter effect in (1) not all the eigenvectors need to be extended. Additionally, only the locations of the local maxima of $f_t(s)$ are used in Algorithm 1, rather than all its values, thus we can extend a sufficient number of eigenvectors, so that the same local maxima are attained as in the construction with all the spectral components.

After the score function $f_t(s)$ is approximated, extracting its local maxima requires not only going over all its entries, but also considering their connectivity. This additional complexity is negligible when the underlying graph is sparsely connected. Importantly, the more connected the graph is, the less significant the options are (e.g., as demonstrated by Jinnai et al. (2019)), and therefore, in the context of this paper, only sparsely connected graphs are of interest.

3.3. Analysis

We start the analysis with our main result relating $f_t(s)$ to the diffusion distance. The proof is provided in the SM.

Proposition 1. *The function $f_t : \mathbb{S} \to \mathbb{R}$ defined as $f_t(s) \triangleq \| \sum_{i \geq 2} \omega_i^t \phi_i(s) \tilde{\phi}_i \|^2$ is equal to the mean squared diffusion distance between state s and all other states, up to a constant independent of s, namely*

$$f_t(s) = \left\langle D_t^2(s, s') \right\rangle_{s' \in \mathbb{S}} + const, \tag{2}$$

where $\langle g(x) \rangle_{x \in \mathbb{X}}$ represents the average on $\mathbb{X}$:

$$\langle g(x) \rangle_{x \in \mathbb{X}} \triangleq \frac{1}{|\mathbb{X}|} \sum_{s \in \mathbb{X}} g(x).$$

An immediate consequence of Proposition 1 is that

$$\max_s f_t(s) = \max_s \left\langle D_t^2(s, s') \right\rangle_{s' \in \mathbb{S}},$$

implying that the option goal states, $\{s_o^{(i)}\}$, are the farthest states from all other states in terms of average squared diffusion distance. Broadly, moving to such far states encourages exploration as the agent systematically travels through the largest number of states without, for example, the repetitions involved in the uninformed random walk. Additionally, by reaching different option goal states, the agent reaches different and distant regions of the domain, which also benefits exploration. The particular notion of diffusion distance efficiently captures the geometry of the domain and demonstrates important advantages over the Euclidean and even the geodesic distances. See the SM for an illustrative example. The averaging operation $\langle \cdot \rangle$ incorporates the fact that the options are not related to a specific task, and therefore, the start state, the goal state, and the states at which the options are invoked, are all unknown a-priori.

Empirically we will demonstrate that the diffusion distance is related to the domain difficulty (see Section 4.4). The larger the average pairwise diffusion distance is, the more difficult the domain is. As a result, when the agent follows options leading to distant states in terms of the diffusion distance, in effect, it reduces the domain difficulty. In addition, we demonstrate that such goal states are typically "special" states such as corners of rooms or bottleneck states such as doors (see Fig. 1(h) and Section 4).

Proposition 2 offers an alternative perspective on $f_t(s)$, relating it to the stationary distribution of the graph, denoted by $\boldsymbol{\pi}_0$. The proof is in the SM.

Proposition 2. *$f_t(s)$ can be recast as*

$$f_t(s) = \| \boldsymbol{p}_t^{(s)} - \boldsymbol{\pi}_0 \|^2,$$

where $\boldsymbol{\pi}_0$ is the stationary distribution of the lazy random walk $\boldsymbol{W}$ on the graph G. In addition, $f_t(s)$ is bounded from above by

$$f_t(s) \leq \omega_2^{2t} \left(\frac{1}{\boldsymbol{\pi}_0(s)} - 1 \right).$$

The first part of Proposition 2 relates $f_t(s)$ to the difference between the transition probability from state s and the stationary distribution. As t grows to infinity, the transition probability approaches the stationary distribution. For a fixed t, the states at which $f_t(s)$ gets a maximum value are the states that their transition probability differ the most from the stationary distribution.

States s for which $\pi_0(s)$ is small are states that are least visited by an agent following a standard random walk. Arguably, these are exactly the states the agent should visit, for example by following options, to improve exploration. Indeed, we observe that the upper bound in Proposition 2 implies that these states allow for large $f_t(s)$ values. We further discuss the relation between $f_t(s)$ and π_0 in a multi-dimensional grid domain in the SM.

Establishing the relation of $f_t(s)$ to the stationary distribution is important by itself because the stationary distribution is a central component in many applications and algorithms. Perhaps the most notable are PageRank (Page et al., 1999) and its variants (Kleinberg, 1999), where the purpose is to discover important web pages that are highly connected and therefore can be considered as network hubs. In the exploration-exploitation terminology, one could claim that PageRank favors exploitation by identifying central pages. Conversely, the diffusion options lead the agent toward states that are least connected (with small stationary distribution values), and therefore, they encourage exploration.

We end this section with two remarks. First, the upper bound in Proposition 2 generalizes a known bound on the convergence of the transition probability, starting from node a in a graph, to the stationary distribution at node b (Spielman,

2018),

$$|p_t(b) - \pi_0(b)| \leq \sqrt{\frac{d(b)}{d(a)}}\omega_2^t,$$

where $d(a)$ and $d(b)$ are the degrees of nodes a and b, respectively.

Second, combining Proposition 1 and Proposition 2 relates the diffusion distance to the distance from the stationary distribution of a random walk. This relation may have consequences in a broader context, when either the diffusion distance or the stationary distribution are used.

3.4. Extension to Stochastic Domains

In the deterministic setting we considered thus far, we assumed that an action definitively leads the agent to a particular state, i.e., given an action a and a state s the probability $p(s'|s,a)$ is concentrated at a single state.

Alternatively, one could consider a setting, where the domain is stochastic, and its stochasticity introduces uncertainty and decouples the action from the transition, namely, $p(s'|s,a)$ can be supported on more than one state. As a result, the agent following a random walk experiences a different number of transitions between states. The corresponding transition probability matrix leads to a non-symmetric normalized graph Laplacian $\boldsymbol{N}$. This poses a challenge since the eigenvalue decomposition of $\boldsymbol{N}$ is not guaranteed to be real, and therefore, the construction of $f_t(s)$ in (1) needs a modification. Note that other settings could lead to a non-symmetric Laplacian as well.

Here, we propose a remedy to support such cases. Our solution follows the work presented by Mhaskar (2018), which is based on the polar decomposition. Concretely, consider the polar decomposition of $\boldsymbol{N} = \boldsymbol{R}\boldsymbol{U}$, where $\boldsymbol{R}$ is a positive semi-definite matrix and $\boldsymbol{U}$ is a unitary matrix. Since $\boldsymbol{R}$ is uniquely determined, the spectral analysis applied to $\boldsymbol{N}$ in the deterministic case can be applied to $\boldsymbol{R}$ in a similar manner. As observed by Mhaskar (2018), there exist efficient algorithms for computing $\boldsymbol{R}$ (Nakatsukasa et al., 2010). Accordingly, the required modification applied to the option discovery in Algorithm 1 is minimal. After the computation of $\boldsymbol{N}$, its polar decomposition is computed. Then, the eigenvalue decomposition of the positive part $\boldsymbol{R}$ is used for the construction of $f_t(s)$. See the SM for the modified algorithm. In Section 4.3, we demonstrate its performance.

4. Experimental Results

We demonstrate empirically that the diffusion options are generic and useful, allowing for improvement in both learning unknown tasks and in exploring domains efficiently. Particularly, using Q learning (Watkins & Dayan, 1992), we show that equipped with the diffusion options, which are computed in a reward-free domain, the agent is able to learn tasks that are unknown a-priori faster and to explore a domain more effectively. In addition, we demonstrate the relation between the diffusion options and the stationary distribution.

We focus on three domains: a Ring domain, which is the 2D manifold of the placement of a 2-joint robotic arm (Verma, 2008), a Maze domain (Wu et al., 2019), and a 4Rooms domain (Sutton et al., 1999). The set of actions are: left, right, up and down. In every domain, we pre-define a single start state and a set of goal states.

The agent performs several trials, where each trial is associated with a different goal state from the set of goal states. In each trial, the agent starts at the same start state and is assigned with the task of reaching the trial goal state. We implement Q learning (Watkins & Dayan, 1992) with $\alpha = 0.1$ and $\gamma = 0.9$ for 400 episodes, containing 100 steps each. The agent follows the Q function at states for which it exists, and otherwise chooses a primitive action or an option with equal probability. In case the agent does not reach the goal state after 100 steps, a default value of 101 is set for the number of steps.

Since options typically consist of multiple steps, for a fair comparison, we take them into account in the total steps count at each episode. Note that this might lead to terminating an option without reaching its option goal state in case the episode reaches 100 steps.

We compare the diffusion options with the eigenoptions presented by Machado et al. (2017) and with the cover options from Jinnai et al. (2019). As a baseline, we also show results for a random walk consisting of only primitive actions without options. In the SM, we include comparison to random options as well.

We evaluate the performance using three objective measures. The first measure is the standard learning convergence. We compute the average number of steps to a goal over all learning trials (goal states), where each trial consists of 30 Monte Carlo iterations. The average number of steps is presented as a function of the learning episode. Second, we present the average number of visitations at each state during learning (over all episodes and goal states). Third, to evaluate the exploration efficiency, we compute the number of steps between every two states, following Machado et al. (2017).

The main hyperparameter of the algorithm is t. In our implementation, we set $t = 4$. Our empirical study shows that different values of t lead to similar results. For results using other t values and for a further discussion on the choice of t, see the SM.

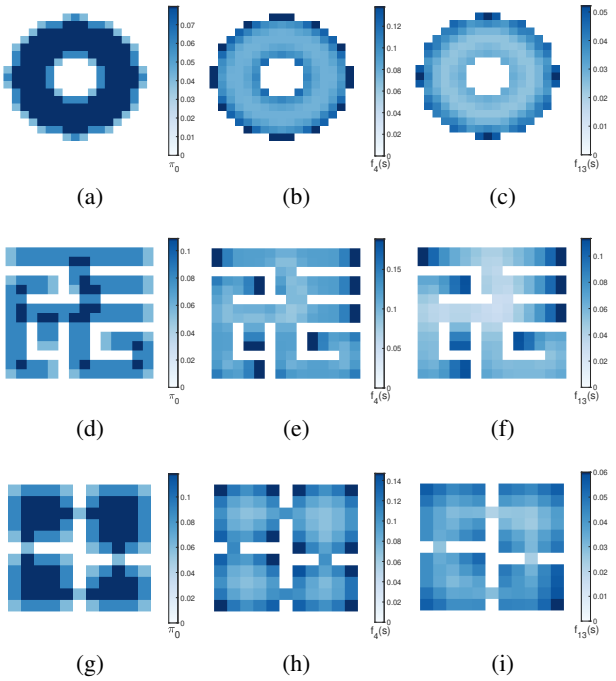

Figure 1. The domains colored according to (a,d,g) the stationary distribution π_0, (b,e,h) the options generating function $f_4(s)$, and (c,f,i) the options generating function $f_{13}(s)$.

4.1. Diffusion Options Generation

In Fig. 1, we plot the options generating function $f_t(s)$ for two values of t as well as the stationary distribution. First, we observe the low pass filter effect obtained by increasing the scale parameter t. Particularly, we see that $f_{13}(s)$ is smoother, containing fewer peaks, than $f_4(s)$. Based on our empirical tests, using only the dominant 10-20 eigenvectors leads to f_t with the same local maxima, resulting in the same options. As discussed in Section 3.2, this facilitates the extension to large scale domains. We emphasize that using fewer eigenvectors is insufficient and does not capture well the geometry of the domain. Second, we observe that the minima of the stationary distribution coincide with the local maxima of $f_t(s)$ for some cases, in accordance with Proposition 2. For example, note the corners of the rooms and the doors in the 4Rooms domain (Figs. 1(g) and 1(h)). Nevertheless, we observe that the local minima of the stationary distribution might also capture irrelevant states in evolved domains. For example, in the Maze domain, in contrast to the stationary distribution, $f_t(s)$ captures the end of the corridors *only* (see Figs. 1(d) and 1(f)), which are important for efficient exploration and learning in this domain.

4.2. Exploration and Learning

Figure 2 presents the results obtained by setting $t = 4$ for all domains. We observe in the visitation count plots that the diffusion options lead the agent to the goal states through the shortest path, e.g., in the Ring domain, following the inner ring. Importantly, these results are obtained by the diffusion options that were built in advance without access to the location of the start and goal states. Conversely, we observe that the eigenoptions lead the agent less efficiently, for example, in the Ring domain, through both the inner and the outer rings. While both the diffusion options and the eigenoptions result in informed trajectories to the goal, we observe that the naïve random walk tends to concentrate near the start state.

Figure 2 also shows that the diffusion options demonstrate the fastest learning convergence, followed by the eigenoptions and then the random walk. In addition, the diffusion options lead to convergence to shorter paths to a goal compared to the eigenoptions. These convergence results coincide with the visitation count. For example in the Ring domain, by employing the eigenoptions, the agent travels via states at the outer ring which are not on the shortest path to the goal. The significant gap in performance between the diffusion options and the eigenoptions in the Maze domain may be explained by the fact that the option goal states of the diffusion options are located at the end of the corridors (see Fig. 1), leading to efficient exploration, and in turn, to this fast learning convergence. We note that the zero variance in the learning curves at the beginning of the learning implies that the agent did not reach its goal during the episode, so the same default value was set.

For a fair comparison, we use the same number of options in both algorithms with the same Q learning configuration described above. In the SM, we present results, where the number of eigenoptions is tuned to attain maximal performance. Even after tuning, the diffusion options outperform the eigenoptions.

Table 1 shows the number of steps between states. We note that in contrast to eigenoptions and diffusion options, cover options are point options; see further discussion in Section 5. We observe that the diffusion options lead to more efficient transitions between states compared to the eigenoptions, cover options, and a random walk. This suggests that diffusion options demonstrate better exploration capabilities.

4.3. Stochastic Domains

We revisit the 4Rooms domain with the addition of a stochastic wind blowing downwards. The presence of wind is translated to the probability of $1/3$ that the agent moves down, regardless of its chosen action. As a result, the agent is

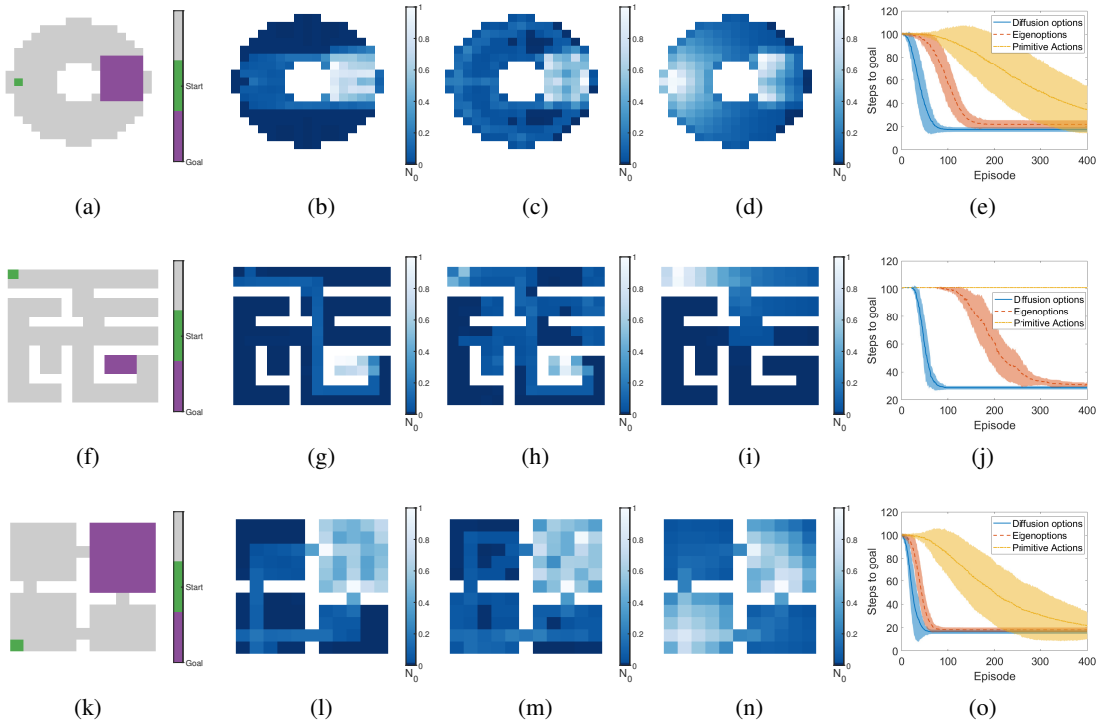

Figure 2. Learning results on the Ring domain (top row), the Maze domain (middle row), and 4Rooms domain (bottom row). (a,f,k) The start state (green) and goal states (purple). (b-d,g-i,l-m) Normalized visitation count N_0 obtained based on (b,g,l) the diffusion options, (c,h,m) the eigenoptions, and (d,i,n) a random walk (d). For visualization purposes, the visitation number is normalized to the range of $[0, 1]$ by dividing by the maximum number of visitations. (e,j,o) The learning convergence depicting the average number of steps to goal for each learning episode. The solid line represents the mean value and the light colors represent the standard deviation.

more likely to visit states at the bottom of the domain, so in principle, the desired options should favor states at the upper parts of the domain.

In Fig. 3(a), we observe that $f_4(s)$ now exhibits high values at the upper part of the rooms, rather than high values at the corners and boundaries as in Fig. 1(h) without the wind. To compare the learning convergence, we adapt the eigenoptions to the stochastic domain by considering the eigenvectors of the positive part of the polar decomposition of the Laplacian as eigenoptions. Figure 3(b), presenting the learning convergence, shows a clear advantage to the use of the diffusion options compared to the eigenoptions in this stochastic setting.

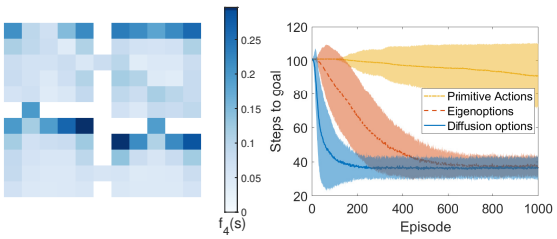

Figure 3. 4Rooms domain with stochastic wind blowing downwards. (a) The domain is colored by $f_4(s)$, where we observe that the local maxima are at the top rooms, compensating for the wind. See Fig. 1 for comparison to the result without wind. (b) The obtained learning convergence.

4.4. Diffusion Distance and Domain Difficulty

We empirically show that the diffusion distance is related to the "domain difficulty". We propose to approximate the difficulty by the average diffusion distance between every pair of states, and compare it with two other measures of difficulty: the average time duration required for learning a task using primitive actions, i.e. the learning rate, and the average number of steps between pairs of states. Note that the computation of diffusion distances is intrinsic, i.e., it takes into account only the geometry of the domain. Consequently, it can be computed per domain a-priori without any task assignment or access to rewards. Conversely, the learning rate and the average number of steps are computed in the context of learning particular tasks and rewards, and as a result, convey their difficulties as well.

Table 1. Number of steps between any pair of states using options induced by $t = 4$ and by $t = 13$. We report the median value and the interquartile range (IQR) over all pairs. See the SM, for mean and standard deviation.

Domain (#states)	t	#options	Diffusion Options		Eigenoptions		Cover Options		Random Walk	
			Median	IQR	Median	IQR	Median	IQR	Median	IQR
Ring (192)	4	32	**217**	101	301	210	361	536	565	160
	13	28	**219**	110	279	232	363	481	565	160
Maze (148)	4	19	**282**	194	446	573	525	812	1280	960
	13	14	**249**	160	641	781	498	842	1280	960
4Rooms (104)	4	20	**147**	137	160	114	179	512	487	104
	13	15	**140**	96	162	151	175	442	487	104

For each domain, the average diffusion distance between all states is computed. To account for the domain size, we multiply the average diffusion distance by the number of accessible states. In addition, we compute the average of diffusion distance over 100 different scales of t from a regular grid between 1 and 1000.

The results are: 13.6, 20.5, and 8.6 for the Ring, the Maze, and the 4Rooms domains, respectively. We observe that the obtained value in the Maze is higher than the obtained value in the Ring, despite having fewer states. Indeed, the learning convergence in the Maze is slower (see Figs. 2(j) and 2(e)) and the average number of steps between states is higher as well (see Table 1).

The relation between the domain difficulty and the diffusion distance gives another justification to the proposed algorithm. By Proposition 1, acting according to a diffusion option leads the agent to a distant state in terms of the diffusion distance. As a result, it can be seen as a way to effectively reduce the domain difficulty.

5. Relation to Existing Work

Option discovery has attracted much interest in recent years, resulting in numerous methods from various perspectives such as information theoretic (Mohamed & Rezende, 2015; Florensa et al., 2017; Hausman et al., 2018), learning hierarchy (Bacon et al., 2017; Vezhnevets et al., 2017), and curiosity (Pathak et al., 2017), to name but a few. Discovering options without reward has been a recent active research subject. Combining information theory and skill discovery, Eysenbach et al. (2019) proposed to view skills as mixtures of policies, and to derive policies without a reward using an information theoretic objective function. There, a two-stage approach, similar to the present paper, was presented. In the first stage, the domain is scanned with no reward and the options are computed, and in the second stage, the options are utilized for learning in the context of particular rewards.

The notion of "bottleneck" states has assumed a central role in option discovery. For example, Menache et al. (2002); Şimşek et al. (2005); Mannor et al. (2004) propose to define and to identify bottleneck states using graph and spectral clustering methods. Unfortunately, these approaches fail in domains such as the Ring domain, for which clustering is not well defined. An alternative approach presented by Stolle & Precup (2002) defines bottleneck states as frequently visited states. Recently, Goyal et al. (2019) showed that this definition might lead to the discovery of redundant options in domains such as a T-shaped domain.

Perhaps the closest algorithm to ours for option discovery was presented by Machado et al. (2017). There, the agent uses a subset of eigenvectors of the graph Laplacian of the domain. Each eigenvector (up to a sign) prescribes a value function assigned to each option (termed eigenoption). The agent follows the eigenvector until it reaches a local extremum, where the option terminates. A natural question that arises is why the extrema of the eigenvectors are good option goal states. Here we offer a plausible answer from a diffusion distance perspective. Cheng et al. (2019) defined the diffusion distance to a subset $\mathbb{B} \subset \mathbb{S}$, and derived a lower bound. In the present paper notation, the formulation of the bound is as follows. Let $d_{\mathbb{B}_i}(s)$ be the smallest number of steps, such that the random walk starting from state s reaches the subset $\mathbb{B}_i$ with probability greater than $\frac{1}{2}$. Then for $\mathbb{B}_i = \{s \in \mathbb{S} : -\epsilon \leq \psi_i(s) \leq \epsilon\}$ the following holds:

$$d_{\mathbb{B}_i}(s) \log\left(\frac{1}{|1-\nu_i|}\right) \geq \log\left(\frac{|\psi_i(s)|}{\|\psi_i\|_{L^\infty}}\right) - \log\left(\frac{1}{2}+\epsilon\right),$$

where ν_i and ψ_i are a pair of eigenvalue and its associated eigenvector of the normalized graph Laplacian $\boldsymbol{N}$. For small ϵ, the set $\mathbb{B}_i$ is the set of states for which $\psi_i(s)$ is close to zero. By following an eigenoption defined by the eigenvector ψ_i, the agent moves toward states that are distant from the states in $\mathbb{B}_i$. For instance, consider a domain that is comprised of 2 clusters. For such a domain, $\mathbb{B}_2$, derived from ψ_2, is the set of bottleneck states separating the 2 clusters. Thus, the eigenoption leads the agent away from bottleneck states.

In contrast, by Proposition 1, the goal states of diffusion options are states that are distant from *all* states (on average).

Diffusion distance, which is closely related to the proposed options via Proposition 1, takes into account the structure of the domain, including bottlenecks. In addition, Proposition 2 also implies on the tight relation between diffusion options and bottleneck states because bottlenecks often lie at the minima of the stationary distribution.

Other graph-based options are cover options (Jinnai et al., 2019), which are based on different principles than diffusion options. Cover time is the number of steps it takes to reach every state at least once by a random walk, and cover options attempt to minimize the cover time. Diffusion distance is based on the difference between transition probabilities, and diffusion options attempt to reach states that are seldom visited by a random walk. Perhaps the strongest evidence for the difference between the two options is the fact that cover options are derived only from the Fiedler vector (multiple times), whereas diffusion options from the entire spectrum. We claim that using multiple spectral components captures better the structure of the domain. Another difference is that cover options are point options with limited initiation sets. This limitation does not apply to diffusion options.

Our options-generating function $f_t(s)$ in (1) is related to recent work in data analysis as well. Similar functions to $f_t(s)$, constructed from the eigenvectors of the graph Laplacian, were proposed for anomaly detection and clustering ((Cheng et al., 2018; Cheng & Mishne, 2018), respectively). Particularly, Cheng & Mishne (2018) introduced and analyzed a function called spectral norm, and showed that the proliferation of eigenvectors is beneficial for clustering. In this work, we show that the same approach of combining all eigenvectors together, rather than using them separately (as the common practice is, for instance in PCA), is beneficial for option discovery.

6. Conclusions

We presented a method to derive options based on the full spectrum of the graph Laplacian. The main ingredient in the derivation and the subsequent analysis is the diffusion distance, a notion that was introduced in the context of manifold learning primarily for high-dimensional data analysis. We tested our options using Q learning in three domains, demonstrating improved exploration and learning compared to competing options.

We believe that a similar approach with such geometric considerations can be beneficial in other problems. Particularly, in future work we plan to explore its use for state aggregation (Singh et al., 1995; Duan et al., 2019). States that belong to the same partition have the same transition probabilities, and as a consequence, the diffusion distance between them is zero. Therefore, it seems only natural to utilize this notion of distance for this problem. In addition, we will study the possibility to combine model-based state transition learning with the formation of an empirical graph Laplacian.

Acknowledgements

We thank the reviewers for their helpful suggestions and especially for bringing cover options (Jinnai et al., 2019; 2020) to our attention. This research was partially supported by the Technion Hiroshi Fujiwara cyber security research center and and by the Pazy Foundation. The work of RM is partially supported by the Ollendorff Center of the Viterbi Faculty of Electrical Engineering at the Technion, and by the Skillman chair in biomedical sciences.

References

Bacon, P.-L., Harb, J., and Precup, D. The option-critic architecture. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Bronstein, A. M., Bronstein, M. M., Guibas, L. J., and Ovsjanikov, M. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics (TOG)*, 30(1):1–20, 2011.

Cheng, X. and Mishne, G. Spectral embedding norm: Looking deep into the spectrum of the graph Laplacian. *Journal on Imaging Sciences (SIAM)*, 2018.

Cheng, X., Mishne, G., and Steinerberger, S. The geometry of nodal sets and outlier detection. *Journal of Number Theory*, 185:48–64, 2018.

Cheng, X., Rachh, M., and Steinerberger, S. On the diffusion geometry of graph Laplacians and applications. *Applied and Computational Harmonic Analysis*, 46(3): 674–688, 2019.

Chung, F. R. and Graham, F. C. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

Coifman, R. R. and Lafon, S. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

Duan, Y., Ke, T., and Wang, M. State aggregation learning from markov transition data. In *Advances in Neural Information Processing Systems*, pp. 4488–4497, 2019.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. *International Conference on Learning Representations (ICLR)*, 2019.

Florensa, C., Duan, Y., and Abbeel, P. Stochastic neural networks for hierarchical reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2017.

Goyal, A., Islam, R., Strouse, D., Ahmed, Z., Botvinick, M., Larochelle, H., Levine, S., and Bengio, Y. Infobot: Transfer and exploration via the information bottleneck. *International Conference on Learning Representations (ICLR)*, 2019.

Hausman, K., Springenberg, J. T., Wang, Z., Heess, N., and Riedmiller, M. Learning an embedding space for transferable robot skills. *International Conference on Learning Representations (ICLR)*, 2018.

Jinnai, Y., Park, J. W., Abel, D., and Konidaris, G. Discovering options for exploration by minimizing cover time. *International Conference on Machine Learning (ICML*, 2019.

Jinnai, Y., Park, J. W., Machado, M. C., and Konidaris, G. Exploration in reinforcement learning with deep covering options. In *International Conference on Learning Representations*, 2020.

Kleinberg, J. M. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

Lafon, S., Keller, Y., and Coifman, R. R. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on pattern analysis and machine intelligence*, 28 (11):1784–1797, 2006.

Lederman, R. R. and Talmon, R. Learning the geometry of common latent variables using alternating-diffusion. *Applied and Computational Harmonic Analysis*, 44(3): 509–536, 2018.

Liu, J., Yang, Y., and Shah, M. Learning semantic visual vocabularies using diffusion distance. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 461–468. IEEE, 2009.

Machado, M. C., Bellemare, M. G., and Bowling, M. A Laplacian framework for option discovery in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2295–2304. JMLR. org, 2017.

Machado, M. C., Rosenbaum, C., Guo, X., Liu, M., Tesauro, G., and Campbell, M. Eigenoption discovery through the deep successor representation. *International Conference on Learning Representations (ICLR)*, 2018.

Mahadevan, S. and Maggioni, M. Proto-value functions: A Laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8(Oct):2169–2231, 2007.

Mahmoudi, M. and Sapiro, G. Three-dimensional point cloud recognition via distributions of geometric distances. *Graphical Models*, 71(1):22–31, 2009.

Mannor, S., Menache, I., Hoze, A., and Klein, U. Dynamic abstraction in reinforcement learning via clustering. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 71. ACM, 2004.

Menache, I., Mannor, S., and Shimkin, N. Q-cut—dynamic discovery of sub-goals in reinforcement learning. In *European Conference on Machine Learning*, pp. 295–306. Springer, 2002.

Mhaskar, H. N. A unified framework for harmonic analysis of functions on directed graphs and changing data. *Applied and Computational Harmonic Analysis*, 44(3): 611–644, 2018.

Mohamed, S. and Rezende, D. J. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pp. 2125–2133, 2015.

Nachum, O., Gu, S. S., Lee, H., and Levine, S. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 3303–3313, 2018.

Nakatsukasa, Y., Bai, Z., and Gygi, F. Optimizing halley's iteration for computing the matrix polar decomposition. *SIAM Journal on Matrix Analysis and Applications*, 31 (5):2700–2720, 2010.

Page, L., Brin, S., Motwani, R., and Winograd, T. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

Portegies, J. W. Embeddings of riemannian manifolds with heat kernels and eigenfunctions. *Communications on Pure and Applied Mathematics*, 69(3):478–518, 2016.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Şimşek, Ö., Wolfe, A. P., and Barto, A. G. Identifying useful subgoals in reinforcement learning by local graph partitioning. In *Proceedings of the 22nd international conference on Machine learning*, pp. 816–823. ACM, 2005.

Singh, S. P., Jaakkola, T., and Jordan, M. I. Reinforcement learning with soft state aggregation. In *Advances in neural information processing systems*, pp. 361–368, 1995.

Spielman, D. Lecture notes. *http://www.cs.yale.edu/homes/spielman/561/syllabus.html*, 2018.

Stolle, M. and Precup, D. Learning options in reinforcement learning. In *International Symposium on abstraction, reformulation, and approximation*, pp. 212–223. Springer, 2002.

Sutton, R. S., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211, 1999.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, O. X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pp. 2753–2762, 2017.

Van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A. J., Burdziak, C., Moon, K. R., Chaffer, C. L., Pattabiraman, D., et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3): 716–729, 2018.

Verma, N. Mathematical advances in manifold learning. *Technical Report. San Diego: University of California*, 2008.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. Feudal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3540–3549. JMLR. org, 2017.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Wu, Y., Tucker, G., and Nachum, O. The Laplacian in RL: Learning representations with efficient approximations. *International Conference on Learning Representations (ICLR)*, 2019.