SKILL DISCOVERY WITH WELL-DEFINED OBJECTIVES

Yuu Jinnai, David Abel, Jee Won Park
Department of Computer Science
Brown University
Providence, USA

D Ellis Hershkowitz
Computer Science Department
Carnegie Mellon University
Pittsburgh, USA

Michael L. Littman, George Konidaris
Department of Computer Science
Brown University
Providence, USA

ABSTRACT

While many skill discovery methods have been proposed to accelerate learning and planning, most are based on heuristic methods without clear connections to how the skills impact the agent's objective. As such, the conditions under which the algorithms are effective is often unclear. We claim that we should pursue skill discovery algorithms with explicit relationships to the objective of the agent to understand in what scenarios skill discovery methods are useful. We analyze two scenarios, planning and reinforcement learning and show that we are able to give bounds to the performance of the option discovery algorithms. For planning, we show that the problem of finding a set of options which minimizes the planning time is NP-hard, and give a polynomial-time algorithm that is approximately optimal under certain conditions. For reinforcement learning, we target goal-based tasks with sparse reward where the agent has to navigate through the state-space to reach the goal state without any reward signals other than the goal state. We show that the difficulty of discovering a distant rewarding state in an MDP is bounded by the *expected cover time* of a random walk over the graph induced by the MDP's transition dynamics. We therefore propose an algorithm which finds an option which provably diminishes the expected cover time.

1 INTRODUCTION

An appropriate set of *skills*, or temporally extended actions, can significantly improve the performance of an agent in many scenarios (Sutton et al., 1999). Thus, many heuristic algorithms have proposed to discover skills based on intuitive descriptions of useful skills (Iba, 1989; McGovern & Barto, 2001; Menache et al., 2002; Stolle & Precup, 2002; Şimşek & Barto, 2004; Şimşek et al., 2005; Şimşek & Barto, 2009; Konidaris & Barto, 2009; Machado et al., 2017; Eysenbach et al., 2019). While empirical results show that these algorithms are useful in some scenarios, the conditions which the methods are effective is often unclear because the relationship between the objective of the skill discovery algorithm and that of the agent is often not established. In fact, Jong et al. (2008) sought to investigate the utility of skills empirically and pointed out that introducing skills might worsen the learning performance.

In order to discover options that are guaranteed to be useful, we claim that we should develop skill discovery algorithms which have an explicit connection to the objective of the agent. In this way, we can analytically evaluate the performance of the skill discovery algorithms instead of relying solely on empirical evaluations on benchmark tasks. For example, one can develop an approximate algorithm with a lower bound on performance improvement over an agent without options.

We show the analysis on two scenarios, planning and reinforcement learning by Jinnai et al. (2018; 2019). We show that by explicitly targeting the objective function of the agent, it is possible to give a guarantee on how much the algorithms improve the agent's objective. For planning, we show that the task of finding an option set which minimizes the planning time is NP-hard. Then, we provide

an approximate algorithm with performance guarantees under certain conditions. For reinforcement learning, we show that minimizing the *expected cover time*—the number of steps required for a random walk to visit every state Broder & Karlin (1989)—reduces the expected number of steps required to reach an unknown rewarding state. We introduce an option discovery method that explicitly aims to minimize the expected cover time and show that the algorithm provably diminishes it.

2 Objective Functions for Skill Discovery

We describe our approach to two objectives: planning and exploration in reinforcement learning. We show that by explicitly targeting the appropriate objective, our analyses result in theoretical guarantees on the utility of the options.

2.1 Finding Options that Minimize Planning Time

First, we consider a planning problem with the value iteration algorithm. We formalize what it means to find the set of options that is optimal for planning. More precisely, we consider the problem of finding a subset of options from a candidate set of options so that planning converges within a given iteration limit:

Definition 1 MOMI (MinOptionMaxIteration):
Given *an MDP $M = (\mathcal{S}, \mathcal{A}, R, T, \gamma)$, a non-negative real-value ϵ, a candidate option set $\mathcal{O}'$, and an integer ℓ,* **return** *$\mathcal{O}$ minimizing $|\mathcal{O}|$ subject to $L(\mathcal{O}) \leq \ell$ and $\mathcal{O} \subseteq \mathcal{O}'$, where $L(\mathcal{O})$ is the number of value iteration passes to solve the MDP using the option set $\mathcal{O}$.*

MOMI has the following complexity results:

Theorem 1.

1. *MOMI is $\Omega(\log n)$ hard to approximate even for deterministic MDPs unless $P = NP$.*
2. *MOMI is $2^{\log^{1-\epsilon} n}$-hard to approximate for any $\epsilon > 0$ even for deterministic MDP unless $NP \subseteq DTIME(n^{poly \log n})$.*

Here we describe the outline of the proof (see the Appendix for the full description). The proof is by reduction from the label cover and the set cover problem respectively to a special case of the problem where the set of options are constrained to be *point options*. A point option is a type of option which has exactly one state in initiation set and one state with termination probability set to one. Even for this limited setting, finding a set of point options is $2^{\log^{1-\epsilon} n}$-hard to approximate, and $\Omega(\log n)$-hard to approximate even for deterministic tasks. As we showed the inapproximability results for the special case, MOMI is also NP-hard to approximate. *Thus, Finding an optimal set of options for planning is NP-hard in general*.

This inapproximability results suggest that efficient option discovery algorithms only exist in a more restricted setting that the above cases. We now present a polynomial-time algorithm A-MOMI for approximately computing the optimal set of point options for tasks with bounded return and goal states. The overview of the procedure is as follows.

1. Compute $d : \mathcal{S} \times \mathcal{S} \to \mathbb{Z}_{\geq 0}$ for every state pair where d is the number of iterations for s_i to reach ϵ-optimal if we add a point option from s_j to g, minus one.
2. For every state s_i, compute a set of states X_{s_i} within $\ell - 1$ distance of reaching s_i. The set X_{s_i} represents the states that converge within ℓ steps if we add a point option from s_i to g.
3. Let $\mathcal{X}$ be a set of X_{s_i} for every $s_i \in \mathcal{S} \setminus X_g^+$, where X_g^+ is a set of states that converges within ℓ without any options (thus can be ignored).
4. Solve the set-cover optimization problem to find a set of subsets that covers the entire state set using the approximate algorithm by Chvatal (1979). This process corresponds to finding a minimum set of subsets $\{X_{s_i}\}$ that makes every state in $\mathcal{S}$ converge within ℓ steps.
5. Generate a set of point options with initiation states set to one of the center states in the solution of the asymmetric k-center, and termination states set to the goal.

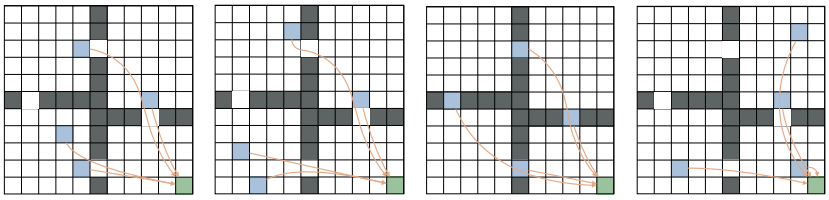
The algorithm has the following properties:

Theorem 2.

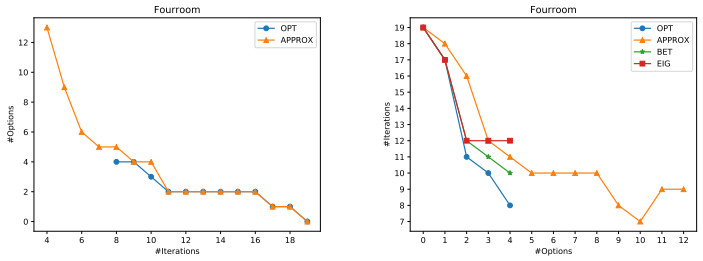
1. A-MOMI *runs in polynomial time.*
2. *The MDP will be solved within ℓ iterations using the option set acquired by* A-MOMI.
3. *If the MDP is deterministic, the option set is at most $O(\log k)$ times larger than the smallest option set possible to solve the MDP within ℓ iterations.*

See the Appendix for the proof. To our knowledge, our method is the first option discovery algorithm with a performance guarantee.

We also consider MIMO, the complementary problem of finding a set of k options that minimize the number of iterations until convergence. The problem is also NP-hard and exists a polynomial-time approximate algorithm, A-MIMO. See the Appendix for the proof.



(a) optimal $k = 4$ (b) approx. $k = 4$ (c) Betweenness (d) Eigenoptions



(e) Four Room (MOMI) (f) Four Room (MIMO)

Figure 1: (a)–(d) Comparison of the optimal point options with options generated by the approximation algorithm. The green square represents the termination state and the blue squares the initiation states. Observe that the approximation algorithm is similar to that of optimal options. Note that the optimal option set is not unique: there can be multiple optimal option sets, and we are visualizing just one returned by the solver. (e)–(f) Figures show the number of options generated by A-MOMI and A-MIMO. OPT: an optimal set of options. APPROX: a bounded suboptimal set of options generated by A-MIMO an A-MOMI. BET: betweenness options. EIG: eigenoptions.

We empirically evaluated the performance of the approximate algorithms against the optimal option set and two heuristic approaches for option discovery, betweenness options (Şimşek & Barto, 2009) and eigenoptions (Machado et al., 2017) in simple grid-world tasks. The results indicate that the approximation algorithm is on par with other heuristic algorithms. While heuristic algorithms have no theoretical performance guarantees (e.g. betweenness options are not necessarily helpful when the task has no bottleneck states), our algorithm offers a performance guarantee in any domain.

2.2 Finding Options that Minimize Learning Time for Hard Exploration Tasks

We now consider reinforcement learning tasks, where the environment model is not available. In particular, we consider how options can improve exploration in goal-based tasks with sparse reward. We model the initial exploratory behavior of a reinforcement learning agent in a sparse reward task by a random walk induced by a fixed stationary distribution. This is because (1) it is a reasonable model for an agent with no prior knowledge of the task and (2) it serves as a worst-case analysis: it is reasonable to assume that efficient exploration algorithms explore faster than the random policy.

We aim to minimize the time required by an agent to explore the task. More precisely, we aim to minimize the *expected cover time*: the expected number of steps required for a random walk to visit all the vertices in a graph (Broder & Karlin, 1989). The expected cover time quantifies how quickly a random walk reaches to a rewarding state.

Theorem 3. *Assume a stochastic shortest path problem to reach a goal state $g \in \mathcal{S}$ where a non-positive reward $r_c \leq 0$ is given for non-goal states and $\gamma = 1$. Let P be a random walk transition matrix: $P(s, s') = \sum_{a \in A} \pi(s) T(s, a, s')$:*

$$\forall g : V_g^\pi(s) \geq r_c \mathbb{E}[C(G)],$$

where $C(G)$ is the expected cover time of the graph G.

See the Appendix for the proof. The theorem suggests that *the smaller the expected cover time, the easier exploration tends to be*. Now the question is how to reduce the expected cover time of the random walk without prior information about the task.

We now present Covering option, an algorithm which discovers options that minimize the expected cover time. The algorithm is approximate since the problem of finding such a set of options is computationally intractable; even a good solution is hard to find due to the Braess's paradox (Braess, 1968; Braess et al., 2005), which states that the expected cover time does not monotonically decrease as edges are added to the graph. Thus, expected cover time is often minimized indirectly via maximizing algebraic connectivity (Fiedler, 1973; Chung, 1996). The expected cover time is upper bounded by a quantity involving the algebraic connectivity, and by maximizing it the bound can be minimized (Broder & Karlin, 1989). As adding a set of edges to maximize the algebraic connectivity is still NP-hard (Mosk-Aoyama, 2008), we use the approximation method by Ghosh & Boyd (2006). The algorithm is as follows:

1. Compute the second smallest eigenvalue and its corresponding eigenvector (i.e., the Fiedler vector) of the Laplacian $\mathcal{L}$ of the state transition graph G.
2. Let v_i and v_j be the state with largest and smallest value in the eigenvector respectively. Generate two point options; one with $\mathcal{I} = \{v_i\}$ and $\beta = \{v_j\}$ and the other with $\mathcal{I} = \{v_j\}$ and $\beta = \{v_i\}$.
3. Set $G \leftarrow G \cup \{(v_i, v_j)\}$ and repeat the process until the number of options reaches k.

The algorithm is guaranteed to reduce the upper bound of the expected cover time:

Theorem 4. *Assume that a random walk induced by a policy π is a uniform random walk and the multiplicity of the second smallest eigenvalue of $\mathcal{L}$ is one. Adding the two options identified by the algorithm improves the upper bound of the cover time:*

$$\mathbb{E}[C(G')] \leq \frac{n^2 \ln n}{\lambda_2(\mathcal{L}) + F}(1 + o(1)), \tag{1}$$

where $\mathbb{E}[C(G')]$ is the expected cover time of the resulting random walk, $F = \frac{(v_i - v_j)^2}{6/(\lambda_3 - \lambda_2) + 3/2}$, v_i, v_j are the maximum and minimum values of the Fiedler vector, and λ_2 is the second smallest eigenvalue of $\mathcal{L}$, and n is the number of states. If the multiplicity of the second smallest eigenvalue is greater than one, then adding any single option cannot improve the bound.

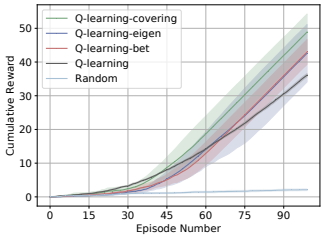
See the Appendix for the proof. Note that the procedure is similar to eigenoptions, proposed by Machado et al. (2017). Both algorithms use the eigenvectors of the Laplacian matrix to generate options. While eigenoptions have no performance guarantees, by explicitly targeting an objective we are able to derive a lower bound on improving the expected cover time and also achieve better empirical performance (Fig 2). Table 2a shows our preliminary results on comparing the expected cover time on simple tabular domains. Our algorithm successfully generates a set of options which reduce the cover time more than the eigenoptions (Machado et al., 2017). In addition, Covering option is fast to compute as it only needs to compute the Fiedler vector. Although computing the whole graph spectrum is a computationally complex matrix operation, the Fiedler vector can be computed efficiently even for very large graphs (Koren et al., 2002).

We now evaluate the utility of each type of discovered options when learning. We used Q-learning (Watkins & Dayan, 1992) ($\alpha = 0.1, \gamma = 0.95$) for 100 episodes of 100 timesteps each and generated 8 options with each algorithms using the adjacency matrix representing the state-transition of the MDP. Figure 2 shows the comparison of accumulated rewards averaged over 5 runs. In all experiments, Covering option outperformed or was on par with eigenoptions.

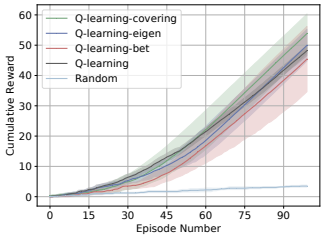
fourroom	λ_2	Cover Time
Covering option	**0.065**	**672.0**
Eigenoptions	0.054	695.9
No options	0.023	1094.8

9x9 grid	λ_2	Cover Time
Covering option	**0.24**	**258.6**
Eigenoptions	0.19	261.5
No options	0.12	460.5

(a) λ_2 and Expected Cover Time



(b) four-room



(c) 9x9 grid

Figure 2: (a) Comparison of the algebraic connectivity and the expected cover time. For Covering option and eigenoptions we add 8 options. (b)–(c) Comparison of performance with different option generation methods. Options are generated offline from the adjacency matrix for four-room and 9x9grid. Reward information is not used for generating options.

3 Related Work

While many option discovery algorithms are relying on a heuristic, several works have proposed methods with well-defined objectives.

Several works have proposed learning the policy and the termination condition of the option by gradient descent using the observed rewards (Mankowitz et al., 2016; Bacon et al., 2017; Harb et al., 2018). Bacon et al. (2017) proposed the option-critic framework and generated options which directly minimize the expected accumulative reward (i.e. the objective of the agent). Harb et al. (2018) proposed to generate options which minimize the sum of expected accumulative reward and the deliberation cost (Simon, 1957) using the option-critic framework (Bacon et al., 2017). The method successfully sped up the learning time by taking into account of the deliberation cost to prefer options with long duration. As they require the reward information, options discovered are task-dependent. Levy et al. (2019) proposed an architecture to learn goal-conditioned policies (i.e. options) to reach certain goal states. They showed that the method can speed up the learning even in long-horizon problems by discovering short horizon subtasks automatically. Brunskill & Li (2014) targeted the lifelong reinforcement learning setting and proposed an option generation method for lifelong reinforcement learning. They analyzed the sample complexity of RMAX using options and proposed an option discovery targeting to minimize the sample complexity. Solway et al. (2014) formalized an optimal behavioral hierarchy as a model which fits the behavior of the agent in tasks the best.

Several works have shown empirically that adding a particular set of options or macro-operators can speed up planning algorithms (Sutton & Barto, 1998; Hauskrecht et al., 1998; Silver & Ciosek, 2012; Konidaris, 2016). Mann et al. (2015) analyzed the convergence rate of approximate value iteration with and without options and showed that options lead to faster convergence if their durations are longer and the value function is initialized pessimistically. As in reinforcement learning, how to find efficient temporal abstractions for planning automatically remains an open question.

4 Conclusions

In this paper, we analyzed two scenarios, planning and reinforcement learning. For planning, we considered the problem of minimizing the size of the option set given a maximum number of iterations (MOMI) and showed that the problem is computationally intractable. We described a polynomial-time approximate algorithm for solving MOMI under certain conditions. For reinforcement learning, we proposed Covering option and showed that it has a guarantee on how much it improves the expected cover time of a random walk. These theoretical guarantees are available because the skill discovery algorithms are directly tailored to the objective of the agent.

There are multiple directions to pursue. First, developing a theory and an algorithm for multitask planning is future work. While the NP-hardness for generating optimal options in Sec. 2.1 implies that it is not useful to generate options for single-task planning since it takes more computational

effort to find the options than to solve the task, it may be useful for multitask planning where the agent can use the acquired skills for future tasks. Second, we aim to scale up the framework of automatically discovering skills for reinforcement learning to continuous state-space tasks. Recent work by Wu et al. (2019) showed that the spectral analysis can be applied to continuous state-space MDPs. Combining their method with option discovery algorithms may enable to discover skills in continuous state-space MDPs. Third, our analysis for reinforcement learning assumes that the state-transition graph is an unweighted undirected graph. This is a severe simplification for MDPs and we plan to extend our method to weighted directed graphs.

REFERENCES

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1726–1734, 2017.

Dietrich Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12(1): 258–268, 1968.

Dietrich Braess, Anna Nagurney, and Tina Wakolbinger. On a paradox of traffic planning. *Transportation science*, 39(4):446–450, 2005.

Andrei Z Broder and Anna R Karlin. Bounds on the cover time. *Journal of Theoretical Probability*, 2(1):101–120, 1989.

Emma Brunskill and Lihong Li. PAC-inspired option discovery in lifelong reinforcement learning. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 316–324, 2014.

Fan RK Chung. *Spectral graph theory*. American Mathematical Society, 1996.

Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *Proceedings of the Seventh International Conference on Learning Representations*, 2019.

Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2): 298–305, 1973.

Arpita Ghosh and Stephen Boyd. Growing well-connected graphs. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 6605–6611. IEEE, 2006.

Jean Harb, Pierre-Luc Bacon, Martin Klissarov, and Doina Precup. When waiting is not an option: Learning options with a deliberation cost. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Milos Hauskrecht, Nicolas Meuleau, Leslie Pack Kaelbling, Thomas Dean, and Craig Boutilier. Hierarchical solution of Markov decision processes using macro-actions. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 220–229. Morgan Kaufmann Publishers Inc., 1998.

Glenn A Iba. A heuristic approach to the discovery of macro-operators. *Machine Learning*, 3(4): 285–317, 1989.

Yuu Jinnai, David Abel, D Ellis Hershkowitz, Michael Littman, and George Konidaris. Finding options that minimize planning time. *arXiv preprint arXiv:1810.07311*, 2018.

Yuu Jinnai, Jee Won Park, David Abel, and George Konidaris. Discovering options for exploration by minimizing cover time. *arXiv preprint arXiv:1903.00606*, 2019.

Nicholas K Jong, Todd Hester, and Peter Stone. The utility of temporal abstraction in reinforcement learning. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pp. 299–306. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

George Konidaris. Constructing abstraction hierarchies using a skill-symbol loop. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 1648–1654, 2016.

George Konidaris and Andrew Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Proceedings of the 22nd Conference on Advances in Neural Information Processing Systems*, pp. 1015–1023, 2009.

Yehuda Koren, Liran Carmel, and David Harel. Ace: A fast multiscale eigenvectors computation for drawing huge graphs. In *Proceedings of the IEEE Symposium on Information*, pp. 137–144, 2002.

Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. In *Proceedings of the Seventh International Conference on Learning Representations*, 2019.

Marios C Machado, Marc G Bellemare, and Michael Bowling. A Laplacian framework for option discovery in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2295–2304, 2017.

Daniel J Mankowitz, Timothy A Mann, and Shie Mannor. Adaptive skills adaptive partitions (ASAP). In *Advances in Neural Information Processing Systems*, pp. 1588–1596, 2016.

Timothy A Mann, Shie Mannor, and Doina Precup. Approximate value iteration with temporally extended actions. *Journal of Artificial Intelligence Research*, 53:375–438, 2015.

Amy McGovern and Andrew G Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. In *Proceedings of the 18th International Conference on Machine Learning*, 2001.

Ishai Menache, Shie Mannor, and Nahum Shimkin. Q-cut – dynamic discovery of sub-goals in reinforcement learning. In *European Conference on Machine Learning*, pp. 295–306, 2002.

Damon Mosk-Aoyama. Maximum algebraic connectivity augmentation is NP-hard. *Operations Research Letters*, 36(6):677–679, 2008.

David Silver and Kamil Ciosek. Compositional planning using optimal option models. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Herbert A Simon. *Models of man; social and rational.* Wiley, 1957.

Özgür Şimşek and Andrew G Barto. Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 95. ACM, 2004.

Özgür Şimşek and Andrew G Barto. Skill characterization based on betweenness. In *Proceedings of the Advances in neural information processing systems*, pp. 1497–1504, 2009.

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

Alec Solway, Carlos Diuk, Natalia Córdova, Debbie Yee, Andrew G Barto, Yael Niv, and Matthew M Botvinick. Optimal behavioral hierarchy. *PLoS computational biology*, 10(8):e1003779, 2014.

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

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

Yifan Wu, George Tucker, and Ofir Nachum. The Laplacian in RL: Learning representations with efficient approximations. In *Proceedings of the Seventh International Conference on Learning Representations*, 2019.