UNSUPERVISED SUBGOAL DISCOVERY METHOD FOR LEARNING HIERARCHICAL REPRESENTATIONS

Jacob Rafati & David Noelle
Electrical Engineering and Computer Science
University of California, Merced. 5200 North Lake Road, Merced, CA 95343, USA.
{jrafatiheravi,dnoelle}@ucmerced.edu

ABSTRACT

Hierarchical Reinforcement Learning (HRL) methods attempt to address the scalability issues of RL by learning policies at multiple levels of temporal abstraction. Abstraction can be had by subgoal discovery, in concert with the learning of corresponding skill policies to achieve those subgoals. Many approaches to subgoal discovery in HRL depend on the analysis of a model of the environment, but the need to learn such a model introduces its own problems of scale. In this paper, we present a novel method for subgoal discovery based on unsupervised learning methods over the past experiences of an agent. Additionally, we offer biologically inspired methods for learning representations in model-free HRL.

1 INTRODUCTION

The reinforcement learning (RL) problem suffers from serious scaling issues. Methods such as transfer learning (Ammar et al., 2012; Singh, 1992; Taylor & Stone, 2009), and Hierarchical Reinforcement Learning (HRL) attempt to address these issues (Barto & Mahadevan, 2003; Hengst, 2010; Dayan & Hinton, 1992; Dietterich, 2000). HRL is an important computational approach intended to tackle problems of scale by learning to operate over different levels of *temporal abstraction* (Sutton et al., 1999; Parr & Russell, 1997; Krishnamurthy et al., 2016). The acquisition of hierarchies of reusable skills is one of the distinguishing characteristics of biological intelligence (Botvinick et al., 2009; Diuk et al., 2013; Badre et al., 2010), and the learning of such hierarchies is an important open problem in computational reinforcement learning.

A number of approaches to HRL have been suggested. One approach focuses on abstraction over the space of actions, or "options", that appear repeatedly during the learning of a set of tasks (Sutton et al., 1999; Levy & Shimkin, 2011; Fox et al., 2017; Bacon et al., 2017). Such frequently reused subpolicies can be abstracted into skills that can be treated as individual actions at a higher level of abstraction (Pickett & Barto, 2002; Thrun & Schwartz, 1995). A somewhat different approach to temporal abstraction involves identifying a set of states that make for useful *subgoals*. This introduces a major open problem in HRL: that of *subgoal discovery*. Some approaches to subgoal discovery maintain the value function in a large look-up table (Sutton et al., 1999; Goel & Huber, 2003; Şimşek et al., 2005; McGovern & Barto, 2001), and most of these methods require building the state transition graph, providing a model of the environment and the agent's possible interactions with it (Machado et al., 2017; Şimşek et al., 2005; Goel & Huber, 2003).

Once useful subgoals are discovered, an HRL agent should be able to learn the skills to attain those subgoals through the use of *intrinsic motivation* — artificially rewarding the agent for attaining selected subgoals (Singh et al., 2010; Vigorito & Barto, 2010). In such systems, knowledge of the current subgoal is needed to estimate future intrinsic reward, resulting in value functions that consider subgoals along with states (Vezhnevets et al., 2017). Such a parameterized universal value function, $q(s, g, a; w)$, integrates the value functions for multiple skills into a single function, taking the current subgoal, g, as an argument.

It is important to note that *model-free* HRL, which does not require a model of the environment, still often requires the learning of useful internal representations of states (Rafati & Noelle, 2019b). When learning the value function using a nonlinear function approximator, such as a deep neural network, relevant features of states must be extracted in order to support generalization at scale . A

number of methods have been explored for learning such internal representations during model-free RL (Tesauro, 1995; Rafati & Noelle, 2017; Mnih et al., 2015), and deep model-based HRL (Kulkarni et al., 2016; Li et al., 2017). However, selecting the right representation is still an open problem (Maillard et al., 2011).

In this paper, we investigate the integration of multiple mechanisms, including the learning of internal state representations, temporal abstraction, automatic subgoal discovery, and intrinsic motivation learning, all within the model-free HRL framework. We propose efficient and general methods for subgoal discovery using unsupervised learning and anomaly (outlier) detection. These methods do not require information beyond that which is typically collected by the agent during model-free reinforcement learning, such as a small memory of recent experiences (agent trajectories).

2 Problem Statement

In an RL problem, the agent should implement a policy, π, from states, $\mathcal{S}$, to possible actions, $\mathcal{A}$, to maximize its expected return from the environment (Sutton & Barto, 2017). At each cycle of interaction, the agent receives a state, s, from the environment, takes an action, a, and one time step later, the environment sends a reward, $r \in \mathbb{R}$, and an updated state, s'. Each cycle of interaction, $e = (s, a, r, s')$ is called a transition *experience*. The goal is to find an optimal policy that maximizes the expected value of the return, i.e. the cumulative sum of future rewards, $G_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'+1}$, where $\gamma \in [0, 1]$ is a discount factor, and T as a final step. It is often useful to define a parametrized value function $Q(s, a; w)$ to estimate the expected value of the return. Q-learning is a Temporal Difference (model-free RL) algorithm that attempts to find the optimal value function by minimizing the loss function, $L(w)$, which is defined over a recent *experience memory*, $\mathcal{D}$:

$$L(w) \triangleq \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\Big[\big(r + \gamma \max_{a'} Q(s', a'; w) - Q(s, a; w)\big)^2\Big]. \tag{1}$$

Learning representations of the value function is challenging for tasks with sparse and delayed rewards, since $r = 0$ for most experiences. Even if the agent accidentally visits a rare rewarding state, where $r > 0$, the experience replay mechanism often fails to learn the value of those states (Mnih et al., 2015). Another major problem in RL is the exploration-exploitation trade-off. Common approaches, such as the ϵ-greedy method, are not sufficiently efficient in exploring the state space to succeed on large-scale complex problems (Bellemare et al., 2016; Vigorito & Barto, 2010). As a simple example, consider the task of navigation in the *4-room environment with a key and a lock* in Figure 2(a). The agent is rewarded for entering the grid square containing the key, and it is more substantially rewarded for entering the grid square with the lock after obtaining the key. The other states are not rewarded. Learning even this simple task is challenging for a reinforcement learning agent.

Our intuition, shared with other researchers, is that hierarchies of abstraction will be critical for successfully solving problems of this kind. To be successful, the agent should represent knowledge at multiple levels of spatial and temporal abstraction. Appropriate abstraction can be had by identifying a relatively small set of states that are likely to be useful as *subgoals* and jointly learning the corresponding skills of achieving those subgoals, using intrinsic motivation.

3 Model-Free Hierarchical Reinforcement Learning

We introduce a unified method for model-free HRL. The major components of our framework, and the information flow between them, are sketched in Figure 1 (a). Inspired by Kulkarni et al. (2016), we start by using two levels of hierarchy (Figure 1(b)). The more abstract level of this hierarchy is managed by a *meta-controller* which guides the action selection processes of the lower level *controller*. Separate value functions are learned for the meta-controller and the controller.

At time step t, the meta-controller receives a state observation, $s = s_t$, from the environment. It has a policy for selecting a *subgoal*, $g = g_t$, from a set of subgoals, $\mathcal{G}$. In our implementation, this policy arises from estimating the value of each subgoal, $Q(s, g; \mathcal{W})$, and selecting the subgoal of highest estimated value. With the current subgoal selected, the controller uses its policy to select an action, $a \in \mathcal{A}$, based on the current state, s, and the current subgoal, g. In our implementation, this policy involves selecting the action that results in the highest estimate of the controller's

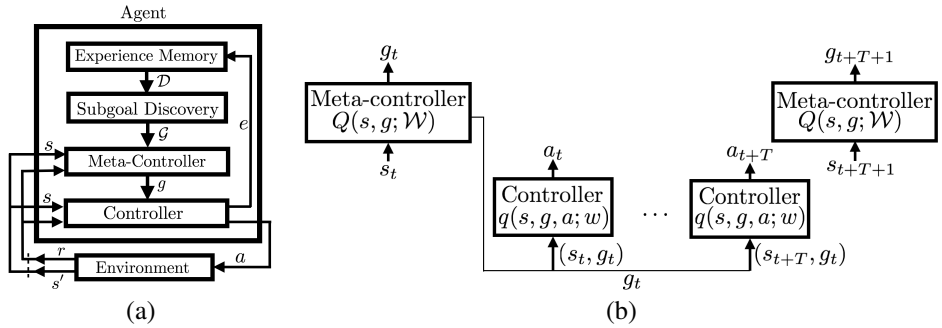

Figure 1: (a) The information flow in our unified model-free hierarchical reinforcement learning framework. (b) Temporal abstraction in the meta-controller/controller architecture.

value function, $q(s, g, a; w)$. Actions continue to be selected by the controller while an internal critic monitors the current state, comparing it to the current subgoal, and delivering an appropriate *intrinsic reward*, $\tilde{r}$, to the controller on each time step. Each transition experience, $(s, g, a, \tilde{r}, s')$, is recorded in the controller's experience memory set, $\mathcal{D}_1$, to support learning. When the subgoal is attained, or a maximum amount of time has passed, the meta-controller observes the resulting state, $s_{t'} = s_{t+T+1}$, and selects another subgoal, $g' = g_{t+T+1}$. The transition experience for the meta-controller, $(s, g, G, s_{t'})$ is recorded in the meta-controller's experience memory set, $\mathcal{D}_2$. For training the meta-controller value function, we minimize a loss function based on the reward received from the environment, $\mathcal{L}(\mathcal{W}) \triangleq \mathbb{E}_{(s,g,G,s_{t'})\sim\mathcal{D}_2}\big[\big(G + \gamma \max_{g'} Q(s', g'; \mathcal{W}) - Q(s, g; \mathcal{W})\big)^2\big]$, where $G = \sum_{t'=t}^{t+T} \gamma^{t'-t} r_{t'}$ is the return between the selection of consecutive subgoals. The controller improves its subpolicy, $\pi(a|s, g)$, by learning its value function, $q(s, g, a; w)$, over the set of recorded transition experiences. The controller updates its value function approximator parameters, w, so as to minimize its loss function, $L(w) \triangleq \mathbb{E}_{(s,g,a,\tilde{r},s')\sim\mathcal{D}_1}\big[\big(\tilde{r} + \gamma \max'_a q(s', g, a'; w) - q(s, g, a; w)\big)^2\big]$. Intrinsic motivation learning is the core idea behind the learning of the value function for the controller. The intrinsic critic in this HRL framework can send much more regular feedback to the controller, since it is based on attaining subgoals, rather than ultimate goals. As an example, our implementation typically assigns an intrinsic reward of $+1$ when the agent attains the current subgoal, g, and -1 for any other state transition. Identifying a good set of candidate subgoals is an additional prerequisite for success, and it is discussed next.

4 Unsupervised Subgoal Discovery

The performance of the meta-controller/controller framework depends critically on selecting good candidate subgoals for the meta-controller to consider. In our framework, a subgoal is a state, or a set of related states, that satisfies at least one of these conditions: (1) It is close (in terms of actions) to a rewarding state. (2) It represents a set of states, at least some of which tend to be along a state transition path to a rewarding state. Our strategy involves applying unsupervised learning methods to a recent experience memory, $\mathcal{D}$, to identify sets of states that may be good subgoal candidates. We focus specifically on two kinds of analysis that can be performed on the set of transition experiences. We hypothesize that good subgoals might be found by (1) attending to the states associated with *anomalous* transition experiences and (2) clustering experiences based on a similarity measure and collecting the set of associated states into a potential subgoal. Thus, our proposed method merges *anomaly (outlier) detection* with the K-means clustering of experiences. The unsupervised subgoal discovery method is summarized in Algorithm 1. See Rafati & Noelle (2019a) for more details.

The anomaly (outlier) detection process identifies states associated with experiences that differ significantly from the others. In the context of subgoal discovery, a relevant anomalous experience would be one that includes a substantial positive reward in an environment in which reward is sparse. We propose that the states associated with these experiences make for good candidate subgoals. For example, in the rooms task, transitions that arrive at the key or the lock are quite dissimilar to most transitions, due to the large positive reward that is received at that point (see Figure 2 (b-d)). In difficult computer games like Montezuma's Revenge, whenever the agent enters a new room, the new

Algorithm 1 Unsupervised Subgoal Discovery Algorithm

for each $e = (s, a, r, s')$ stored in $\mathcal{D}$ **do**
  if experience e is an outlier (anomaly) **then**
    Store s' to the subgoals set $\mathcal{G}$
    Remove e from $\mathcal{D}$
  end if
end for
Fit a K-means Clustering Algorithm on $\mathcal{D}$ using previous centroids as initial points
Store the updated centroids to the subgoals set $\mathcal{G}$

state becomes very different than the states from the previous room, allowing an anomaly detection method to identify the door that leads to a new room as a potentially useful subgoal.

The idea behind using the clustering of experiences involves both "spatial" state space abstraction and dimensionality reduction with regard to the internal representations of states. If a collection of transition experiences are very similar to each other, this might suggest that the associated states are all roughly equally good as subgoals. Thus, rather than considering all of those states, the learning process might be made faster by considering representative states (or smaller sets of states), such as cluster centroids, as candidate subgoals. Furthermore, using a simple clustering technique like K-means clustering to find a small number of centroids in the space of experiences is likely to produce centroid subgoals that are dissimilar from each other. Since rewards are sparse, this dissimilarity will be dominated by state features. For example, in the rooms task, the centroids of K-means clusters, with $K = 4$ (Figure 2 (b)), lie close to the geometric center of each room, with the states within each room coming to belong to the corresponding subgoal's cluster. In this way, the clustering of transition experiences can approximately produce a coarser representation of state space, in this case replacing the fine grained "grid square location" with the coarser "room location".

We have implemented this approach as an artificial neural network, shown in Figure 2 (e). The meta-controller value function receives a one-hot encoding of the current state, computed by converting the current state to the index of the corresponding subgoal. The meta-controller outputs a one-hot encoding of the best subgoal. The controller receives a Gaussian-blurred representation of current state variables (Cartesian coordinates) gated by the current subgoal, and it produces a sparse conjunctive encoding over hidden units using a k-Winners-Take-All mechanism, akin to lateral inhibition in cortex (Rafati & Noelle, 2015; O'Reilly & Munakata, 2001). This is then mapped onto the controller value function output for each possible action. Most of the existing subgoal discovery methods focus on finding the doorways (funnel type subgoals) (Goel & Huber, 2003; Şimşek et al., 2005). With $K = 4$, the doorways can be discovered as boundaries between adjacent clusters. Note that our method is not strongly task dependent, so the choice of K is not crucial to the learning of meaningful representations. The results of clustering for different values of K are shown in Figure 2 (b-d). The average return for the unified HRL method, and regular RL is shown in Figure 2 (f).

We have also applied this method to the difficult ATARI 2600 game called Montezuma's Revenge. Discovered subgoals are shown in Figure 3 (a) and results in (b-c). In this game, the controller was initially trained to navigate the man in red to random regions on the screen (i.e., random subgoals). Using this strategy, the agent learned navigation skills and recorded anomalous states: *key* and *doors*. Also, K-means clustering found useful subgoal regions, such as ladders, stages, and the rope. Using HRL with these discovered subgoals solved the room, while deep Q-learning networks (Mnih et al., 2015) could not. See Rafati & Noelle (2019a;c) for more details on numerical simulations, and mathematical interpretations behind the unsupervised subgoal discovery method.

This work has been inspired, in part, by theories of reinforcement learning in the brain. These theories often involve interactions between the striatum and neocortex. There is some evidence that temporal abstraction in HRL might map onto regions within the dorsolateral and orbital prefrontal cortex (PFC) (Botvinick et al., 2009), allowing the PFC to provide hierarchical representations to the basal ganglia. More recent discoveries reveal a potential role for medial temporal lobe structures, including the hippocampus, in planning and spatial navigation (Botvinick & Weinstein, 2014), utilizing a hierarchical representation of space (Chalmers et al., 2016). There are also studies of interactions between the hippocampus and the PFC that are directly related to our unsupervised subgoal discovery method. Preston & Eichenbaum (2013) illustrated how novel memories (like *anomalous*

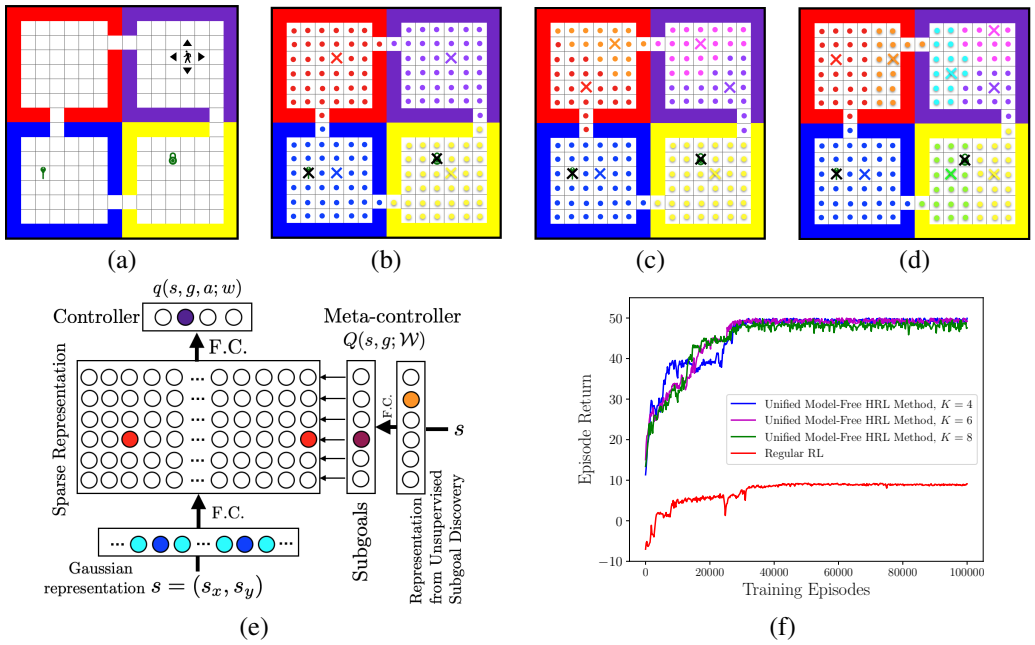

Figure 2: (a) The *4-room* task with a key ($r = +10$) and a lock ($r = +40$). (b-d) The results of the unsupervised subgoal discovery algorithm with *anomalies* marked with black Xs and *centroids* with colored ones. The number of K-means clusters was set to (b) $K = 4$, (c) $K = 6$, (d) $K = 8$. (e) Integrated meta-controller and controller network architecture. (f) The average episode return.

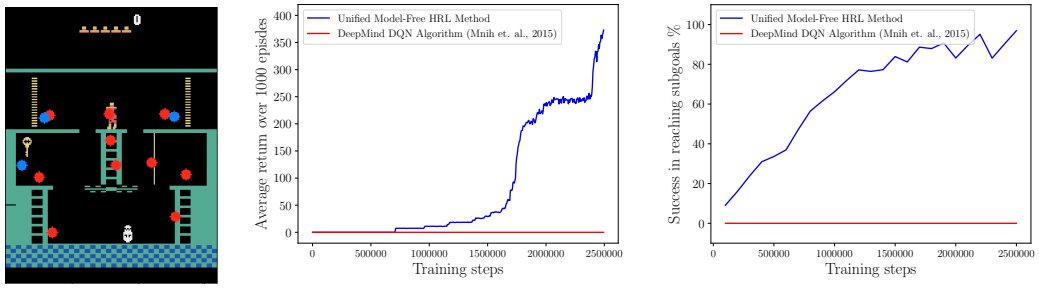

Figure 3: (a) The results of the unsupervised subgoal discovery algorithm in the first room of Montezuma's Revenge. Blue circles are the anomalous subgoals and the red ones are the centroids of clusters. (b) The average game score. (c) The success of the controller in reaching subgoals.

subgoals) could be reinforced into permanent storage. Additionally, their studies suggest how PFC may be important for finding new meaningful representations from memory replay of experiences. This phenomena is similar to our clustering of experience memory.

5 Conclusions

We have proposed a novel method for subgoal discovery, using unsupervised learning over a small memory of the most recent experiences of the agent. Unsupervised subgoal discovery is integrated with the intrinsic motivation learning of a controller (low level learner), and the discovered subgoals provide rich representations to be considered by a meta-controller (high level learner). This results in a unified model-free HRL framework that incorporates the learning of useful internal representations of states, automatic subgoal discovery, intrinsic motivation learning of skills, and the learning of subgoal selection by a meta-controller.

REFERENCES

Haitham B. Ammar, Karl Tuyls, Matthew E. Taylor, Kurt Driessens, and Gerhard Weiss. Reinforcement learning transfer via sparse coding. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, 2012. URL http://dl.acm.org/citation.cfm?id=2343576.2343631.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *AAAI*, 2017. URL http://arxiv.org/abs/1609.05140.

David Badre, Andrew Kayser, and D'Esposito. Frontal cortex and the discovery of abstract action rules. *Neuron*, 66:315–26, 2010.

Andrew G. Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(1):41–77, Jan 2003. ISSN 1573-7594. doi: 10.1023/A:1022140919877. URL https://doi.org/10.1023/A:1022140919877.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 1471–1479. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6383-unifying-count-based-exploration-and-intrinsic-motivation.pdf.

Matthew Botvinick and Ari Weinstein. Model-based hierarchical reinforcement learning and human action control. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369 (1655), 2014. doi: 10.1098/rstb.2013.0480.

Matthew M. Botvinick, Yael Niv, and Andrew C. Barto. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, 113(3):262 – 280, 2009. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2008.08.011. URL http://www.sciencedirect.com/science/article/pii/S0010027708002059.

Eric Chalmers, Artur Luczak, and Aaron J. Gruber. Computational properties of the hippocampus increase the efficiency of goal-directed foraging through hierarchical reinforcement learning. *Frontiers in Computational Neuroscience*, 10, 2016. URL https://www.frontiersin.org/article/10.3389/fncom.2016.00128.

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.

Peter Dayan and Geoffrey E. Hinton. Feudal reinforcement learning. In *NeurIPS*, 1992. URL http://papers.nips.cc/paper/714-feudal-reinforcement-learning.

Thomas G Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. 13:227–303, 2000. URL https://arxiv.org/abs/cs/9905014.

Carlos Diuk, Anna Schapiro, Natalia Crdova, Jos Ribas-Fernandes, Yael Niv, and Matthew Botvinick. Divide and conquer: hierarchical reinforcement learning and task decomposition in humans. In *Computational and robotic models of the hierarchical organization of behavior*, pp. 271–291. Springer, 2013. URL https://link.springer.com/chapter/10.1007%2F978-3-642-39875-9_12.

Roy Fox, Sanjay Krishnan, Ion Stoica, and Kenneth Y. Goldberg. Multi-level discovery of deep options. *arXiv preprint arXiv:1703.08294*, 2017. URL https://arxiv.org/abs/1703.08294.

Sandeep Goel and Manfred Huber. Subgoal discovery for hierarchical reinforcement learning using learned policies. In *FLAIRS Conference*, pp. 346–350. AAAI Press, 2003.

Bernhard Hengst. *Hierarchical Reinforcement Learning*, pp. 495–502. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_363. URL https://doi.org/10.1007/978-0-387-30164-8_363.

Ramnandan Krishnamurthy, Aravind S. Lakshminarayanan, Peeyush Kumar, and Balaraman Ravindran. Hierarchical reinforcement learning using spatio-temporal abstractions and deep neural networks. *CoRR*, abs/1605.05359, 2016.

Tejas D. Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 3675–3683, 2016.

Kfir Y Levy and Nahum Shimkin. Unified inter and intra options learning using policy gradient methods. In *European Workshop on Reinforcement Learning*. Springer, 2011.

Zhuoru Li, Akshay Narayan, and Tze-Yun Leong. An efficient approach to model-based hierarchical reinforcement learning, 2017. URL https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14771.

Marlos C. Machado, Marc G. Bellemare, and Michael H. Bowling. A laplacian framework for option discovery in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 2295–2304, 2017.

Odalricambrym Maillard, Daniil Ryabko, and Rémi Munos. Selecting the state-representation in reinforcement learning. In *Advances in Neural Information Processing Systems 24*, pp. 2627–2635. Curran Associates, Inc., 2011. URL http://papers.nips.cc/paper/4415-selecting-the-state-representation-in-reinforcement-learning.pdf.

Amy McGovern and Andrew G. Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 361–368, 2001. ISBN 1-55860-778-1. URL http://dl.acm.org/citation.cfm?id=645530.655681.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Randall C. O'Reilly and Yuko Munakata. *Computational Explorations in Cognitive Neuroscience*. MIT Press, Cambridge, Massachusetts, 2001.

Ronald Parr and Stuart J Russell. Reinforcement learning with hierarchies of machines. In *NeurIPS*, 1997. URL https://papers.nips.cc/paper/1384-reinforcement-learning-with-hierarchies-of-machines.

Marc Pickett and Andrew G. Barto. Policyblocks: An algorithm for creating useful macro-actions in reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 506–513, 2002.

Alison R. Preston and Howard Eichenbaum. Interplay of hippocampus and prefrontal cortex in memory. *Current Biology*, 23:764 – 773, 2013.

Jacob Rafati and David C. Noelle. Lateral inhibition overcomes limits of temporal difference learning. In *37th Annual Cognitive Science Society Meeting, Pasadena, CA, USA*, 2015.

Jacob Rafati and David C. Noelle. Sparse coding of learned state representations in reinforcement learning. In *Conference on Cognitive Computational Neuroscience, New York City, NY, USA*, 2017.

Jacob Rafati and David C Noelle. Learning representations in model-free hierarchical reinforcement learning. *arXiv e-print (arXiv:1810.10096)*, 2019a. URL https://arxiv.org/abs/1810.10096.

Jacob Rafati and David C. Noelle. Learning representations in model-free hierarchical reinforcement learning. In *33rd AAAI Conference on Artificial Intelligence (AAAI-19), Honolulu, HI, USA.*, 2019b.

Jacob Rafati and David C. Noelle. Unsupervised methods for subgoal discovery during intrinsic motivation in model-free hierarchical reinforcement learning. In *33rd AAAI Conference on Artificial Intelligence (AAAI-19). Workshop on Knowledge Extraction From Games. Honolulu, HI, USA.*, 2019c.

S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transaction on Autonomous Mental Development*, 2(2):70–82, 2010.

Satinder P. Singh. Transfer of learning by composing solutions of elemental sequential tasks. 8:323–339, 1992. URL https://link.springer.com/article/10.1007/BF00992700.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 2nd edition, 2017.

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.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. 10(Jul):1633–1685, 2009. URL http://www.jmlr.org/papers/v10/taylor09a.html.

Gerald Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38 (3), 1995.

Sebastian Thrun and Anton Schwartz. Finding structure in reinforcement learning. In *Advances in Neural Information Processing Systems 7*, pp. 385–392. MIT Press, 1995.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *ICML*, 2017. URL https://arxiv.org/abs/1703.01161.

Christopher M. Vigorito and Andrew G. Barto. Intrinsically motivated hierarchical skill learning in structured environments. *IEEE Transactions on Autonomous Mental Development*, 2(2):132–143, 2010.