AFRL-AFOSR-VA-TR-2020-0025

Constructing Abstraction Hierarchies for Robust, Real-Time Control

George Konidaris
BROWN UNIVERSITY

04/22/2020
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTA2
Arlington, Virginia 22203
Air Force Materiel Command

DISTRIBUTION A: Distribution approved for public release.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
10-06-2020	Final Performance	15 Jan 2017 to 14 Jan 2020

4. TITLE AND SUBTITLE
Constructing Abstraction Hierarchies for Robust, Real-Time Control

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-17-1-0124

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
George Konidaris

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
BROWN UNIVERSITY
1 PROSPECT STREET
PROVIDENCE, RI 02912-9100 US

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AF Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR RTA2

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-VA-TR-2020-0025

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This project primarily focused on the theoretical principles underlying which high-level actions an agent should build, and data-efficient algorithms for learning those high-level actions from interaction with an agent's environment. The projected funded a single PhD student for three years, and resulted in 5 publications at top-tier, highly-refereed international conferences, and 3 additional publications either in preparation or currently under review. The report describes these research results and draws appropriate conclusions.

15. SUBJECT TERMS
Automated Planning, Machine Intelligence

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			LAWTON, JAMES
Unclassified	Unclassified	Unclassified	UU		19b. TELEPHONE NUMBER *(Include area code)* 703-696-5999

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Constructing Abstract Hierarchies for Robust, Real-Time Control

AFOSR Young Investigator Award Final Report

George Konidaris
gdk@cs.brown.edu

April 15th 2020

1 Introduction

This project focused on enabling goal-directed decision-making in complex, unstructured tasks. The core obstacle to effective decision-making in such tasks is understanding how to perform high-level reasoning in a low-level world. Robots must necessarily sense the world via a constant stream of noisy, high-dimensional sensations, and can only ultimately act by emitting low-level motor control signals, but decision-making at that level of detail presents immense computational challenges. Two broad approaches have been employed to ameliorate this problem. In one, the agent either acquires or is given a collection of high-level motor controllers, and chooses which of them to execute without considering the details of how execution is actually carried out. This allows the agent to abstract its *actions*. In the second, an agent compresses its low-level state space to discard irrelevant detail and retain only those aspects relevant to decision-making. This allows the agent to abstract its *state*. It has become increasingly clear that *both* state- and action-abstraction are critical to rapid high-level planning and robust low-level execution.

The PIs recent work has established a critical link between high-level actions and abstract representations—showing that a set of high-level actions *directly specifies* the abstract representations that an agent should use to reason about plans composed of those actions. This theory formalizes the intimate link between a robot's actions and the abstract representations it should use for planning, thereby eliminating the representation design problem. The resulting representation is correct by construction and can be learned completely autonomously by an agent, avoiding the need for manually programming the resulting representation, but critically relies on the availability of suitable high-level actions.

Building off that existing work, this project primarily focused on the theoretical principles underlying which high-level actions an agent should build, and data-efficient algorithms for learning those high-level actions from interaction with an agent's environment. The projected funded a single PhD student for three years, and resulted in 5 publications at top-tier, highly-refereed international conferences, and 3 additional publications either in preparation or currently under review.

2 Major Published Results

2.1 A Principled Theoretical Foundation for Transfer

1. D. Abel, Y. Jinnai, Y. Guo, G.D. Konidaris, and M.L. Littman. Policy and Value Transfer for Lifelong Reinforcement Learning. In *Proceedings of the Thirty-fourth International Conference on Machine Learning*, pages 20–29, July 2018.

Abstraction hierarchies are most useful in the *lifelong learning* or *transfer* setting, where an agent is given data from n reinforcement learning tasks, from which it must synthesize knowledge that improves its performance on the $n+1$th task. Prior work has investigated transferring a wide range of forms of knowledge, but the core question of what the in-principle right form to transfer is, and under which circumstances, remained unresolved. This lack of fundamental theoretical understanding has resulted in a large literature on transfer, full of incompatible or incomplete claims.

Our published work on this topic addressed the question of how best to initialize an agent's policy or value function for task $n+1$, given the optimal policies and value functions obtained by solving tasks 1 through n. We restricted our attention to two kinds of knowledge: policies and values. Beginning with policies, we progressed from the simplest setting of constructing the deterministic policy that performs best in expectation for task $n+1$, to the stochastic and belief-space policy cases, the latter of which models learning. In the first two cases, we either derived a new method or formally proved that an existing method is the optimal way to initialize the policy for two classes of task distributions—when just the reward function changes, and when the transition function can also change (Figure 1).

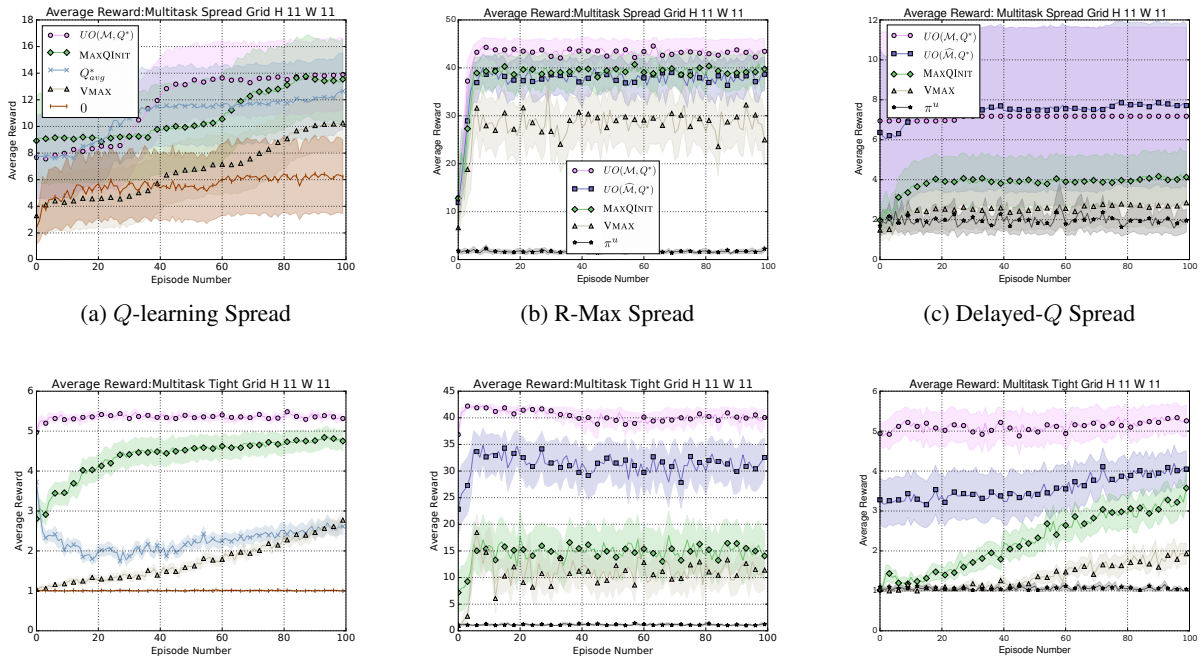
Π	$\mathcal{R} \sim D$	$G \sim D$
$\Pi_d : \mathcal{S} \mapsto \mathcal{A}$	Avg. MDP [Ramachandran and Amir, 2007]	[Singh et al., 1994]
$\Pi_s : \mathcal{S} \mapsto \Pr(\mathcal{A})$	Avg. MDP	[Singh et al., 1994]
$\Pi_b : \mathcal{S} \times \Pr(\mathcal{M}) \mapsto \mathcal{A}$	Belief MDP [Åström, 1965]	Belief MDP

Figure 1: Methods for computing the in-expectation optimally performing policy for a new task, for various learning settings and policy classes. The first column describes the policy *class* that can be reused in a new task: deterministic, stochastic, and belief-space policies (the last of which models learning). The second column considers the setting where the reward function varies across tasks, while the third column considers the setting where a *goal* (additional completion reward plus terminating state) varies across tasks. This table shows that the right way to, for example, construct the stochastic policy with highest expected reward for the $n+1$th task when reward varies is to construct and solve the MDP with a reward function averaged over those previously observed.

We then turned to value-function initialization, focusing on methods that preserve PAC-MDP guarantees but minimize required learning via optimistic value-function initialization. Our investigations resulted in a practical new method, MaxQInit, that lowers both the empirical and theoretical sample complexity of lifelong learning via value-function-based transfer (Figure 2).

2.2 Skill Discovery for Planning

2. Y. Jinnai, D. Abel, D. Hershkowitz, M.L. Littman, and G.D. Konidaris. Finding Options that Minimize Planning Time. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3120–3129, June 2019.



(a) Q-learning Spread (b) R-Max Spread (c) Delayed-Q Spread

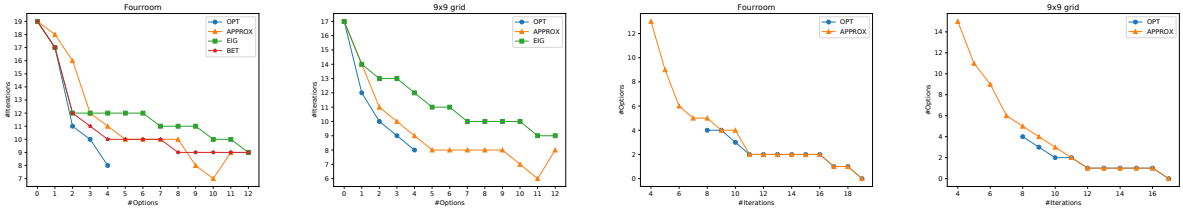
(d) Q-learning Tight (e) R-Max Tight (f) Delayed-Q Tight

Figure 2: Comparison of Q-learning, Delayed-Q, and R-Max with Q-function initialized by VMAX and MaxQInit (and average MDP for Q-learning). Plots show reward averaged over 100 MDPs. Note that the UO algorithms are (impractical) idealizations of MaxQInit, serving to upper-bound its possible performance.

Our successor work asked a similar question for the skill discovery setting. We considered the planning case, where the right set of options allows an agent to probe more deeply into the search space with a single computation. Here, there is an abundance of work using the options framework [Sutton et al., 1999]. Indeed, previous work has offered substantial support that abstract actions can accelerate planning [Mann and Mannor, 2014, Silver and Ciosek, 2012]. But little is known about how to find the right set of options for planning; prior work often seeks to codify an intuitive notion of effectiveness, which often captures important aspects of the role of options in planning, but results in heuristic in that algorithms not based on optimizing any precise performance-related metric. The question of how to discover the optimal set of options—even in-principle—in various settings remains therefore remained unresolved.

Our work formalized the question of finding the set of options that is optimal for planning, and used the resulting formalization to develop an algorithm with performance guarantees and a principled theoretical foundation. Specifically, we considered the problem of finding the smallest set of options so that planning converges in fewer than ℓ value iterations (VI). Our main result was that this problem is *NP*-hard. More precisely, the problem:

1. is $2^{\log^{1-\epsilon} n}$-hard to approximate for any $\epsilon > 0$ unless $NP \subseteq DTIME(n^{\text{poly}\log n})$, where n is the input size;
2. is $\Omega(\log n)$-hard to approximate even for deterministic MDPs unless $P = NP$;
3. has an $O(n)$-approximation algorithm;



(a) Four Room (MIMO) (b) 9×9 grid (MIMO) (c) Four Room (MOMI) (d) 9×9 grid (MOMI)

Figure 3: MIMO and MOMI evaluations. Parts (a)–(b) show the number of iterations for VI using options generated by A-MIMO. Parts (c)–(d) show the number of options generated by A-MOMI to ensure the MDP is solved within a given number of iterations. OPT: optimal set of options. APPROX: a bounded suboptimal set of options generated by A-MIMO an A-MOMI. BET: betweenness options. EIG: eigenoptions.

4. has an $O(\log n)$-approximation algorithm for deterministic MDPs.

We introduced A-MOMI, a polynomial-time approximation algorithm with $O(n)$ suboptimality in general and $O(\log n)$ suboptimality for deterministic MDPs. The expression $2^{\log^{1-\epsilon} n}$ is only slightly smaller than n: if $\epsilon = 0$ then $\Omega(2^{\log n}) = \Omega(n)$. Thus, A-MOMI is close to the best possible approximation factor.

In addition, we consider the complementary problem of finding a set of k options that minimize the number of VI iterations until convergence. We show that this problem is also *NP*-hard, even for a deterministic MDP and introduce A-MIMO, a polynomial time approximation algorithm. Finally, we empirically evaluated the performance of these algorithms against two standard heuristic approaches for option discovery [Şimşek and Barto, 2009, Machado et al., 2017] (Figure 3).

To the best of our knowledge, twenty years after the introduction of the options framework, these are *the first complexity results for option discovery, and the first option discovery algorithms with formal performance guarantees*.

2.3 Skill Discovery for Exploration

5. Y. Jinnai, J. Park, D. Abel, and G.D. Konidaris. Discovering Options for Exploration by Minimizing Cover Time. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3130–3139, June 2019.

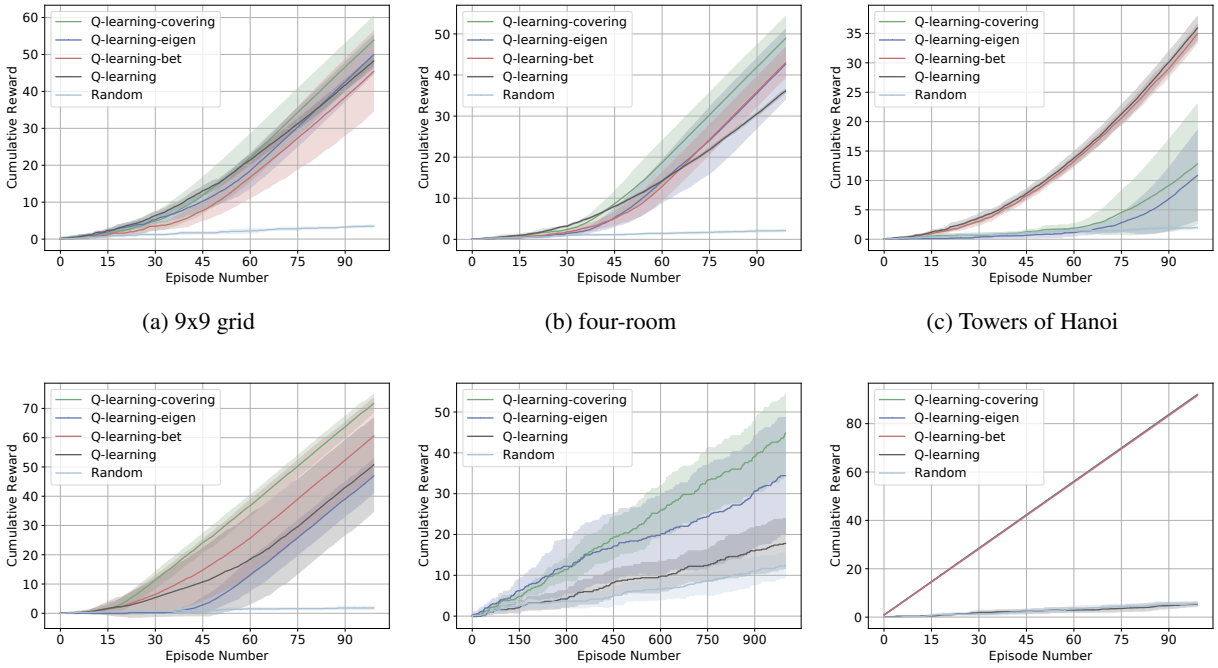
6. Y. Jinnai, J. Park, M.C. Machado, and G.D. Konidaris. Exploration in Reinforcement Learning with Deep Covering Options. In *Proceedings of the Eighth International Conference on Learning Representations*, April 2020.

We next turned to formally considering the properties that make discovered options optimal for *exploration* in reinforcement learning—where an agent is placed in a sparse-reward setting and must obtain data by interacting with the environment. In such settings the agent must discover skills in the absence of reward, which is often so hard to obtain as to be effectively absent.

This is a more complex problem than planning because the agent only ever has partial data about the world, and also because repeatedly executing the wrong options can harm performance since agent is stuck with the

results (unlike the planning setting). While it is a common claim that options can improve exploration, existing approaches are fundamentally heuristic (just as in the planning setting) and lack a principled theoretical grounding, so their effectiveness can only be evaluated empirically.

We introduced an option discovery method that explicitly aims to improve exploration in sparse reward domains by minimizing the expected number of steps required to reach an unknown rewarding state. We achieved this by modeling the behavior of an agent early in learning (i.e., before observing the reward signal) as a uniform random walk over the task state graph. We showed that minimizing the graph *cover time*—the number of steps required for a random walk to visit every state [Broder and Karlin, 1989]—reduces the expected number of steps required to reach an unknown rewarding state. We then introduced a polynomial time algorithm to find a set of options guaranteed to reduce the expected cover time using the transition function either given to or learned by the agent, by using an approximate method [Ghosh and Boyd, 2006] to minimize the upper bound of the expected cover time as a function of the algebraic connectivity of the graph Laplacian [Chung, 1996]. This is, to the best of our knowledge, *the first option discovery algorithm for exploration with any kind of performance guarantee*. Our empirical results from our first paper on this topic demonstrated that the approach generally performs on par with, or better than, previous state-of-the-art methods in discrete domains (Figure 4).

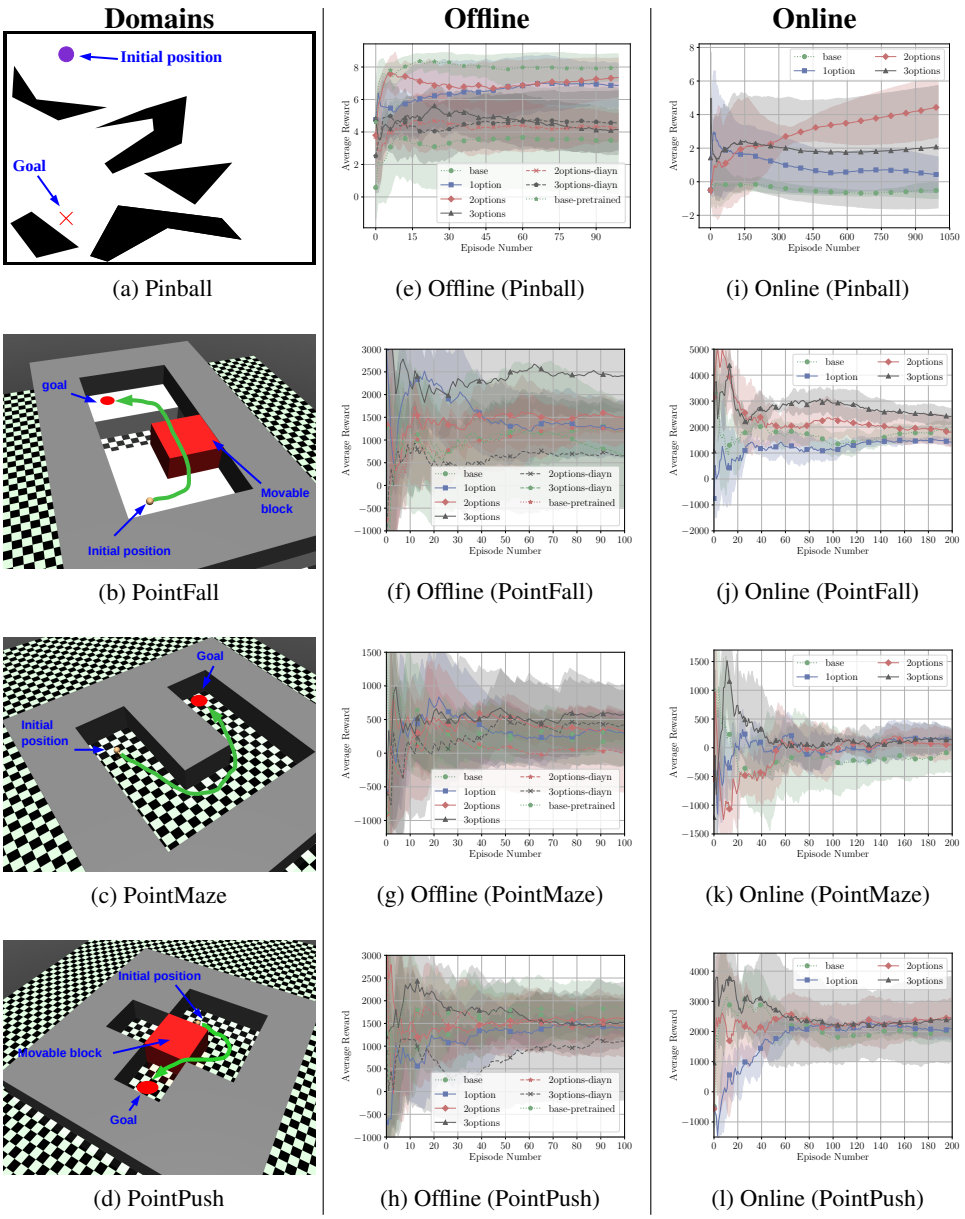


(a) 9x9 grid (b) four-room (c) Towers of Hanoi

(d) Taxi (e) Parr's Maze (f) Race Track

Figure 4: Performance of different option generation methods. Options are generated offline from the adjacency matrix for 9x9grid, four-room, Towers of Hanoi, and Taxi. Options are generated offline from an incidence matrix for Parr's maze and Race Track. Reward is not used for generating options.

Next, we scaled this approach up to high-dimensional continuous domains, by exploiting recent developments in eigenfunction estimation of the Laplacian [Wu et al., 2019]. The resulting skill discovery algorithm is computationally tractable and applicable to environments with large (or continuous) state-spaces; it marries



(a) Pinball (e) Offline (Pinball) (i) Online (Pinball)

(b) PointFall (f) Offline (PointFall) (j) Online (PointFall)

(c) PointMaze (g) Offline (PointMaze) (k) Online (PointMaze)

(d) PointPush (h) Offline (PointPush) (l) Online (PointPush)

Figure 5: Skill discovery performance, averaged over 5 runs. In PointFall (Figure 5b), the agent must push the movable block into a chasm to make a bridge that allows it to reach the goal. In PointMaze (Figure 5c), the agent must first move away from the goal (in terms of L2 distance) to successfully reach it, since the corridor is U-shaped. The green arrow shows successful trajectories. In PointPush (Figure 5d) a greedy agent would move forward and push the movable block into the path to reach the goal. To reach the goal, it must push a movable block to the right to clear the path towards the goal.

the flexibility, scalability, and performance of nonlinear function approximation with the principled theoretical basis of covering options. As a result, it could be applied to high-dimensional control problems, where it substantially improved baseline performance (Figure 5).

2.4 Skill Discovery for Reinforcement Learning

7. A. Bagaria and G.D. Konidaris. Option Discovery using Deep Skill Chaining. In *Proceedings of the Eighth International Conference on Learning Representations*, April 2020.

Our final piece of published work addressed reward-driven skill discovery in reinforcement learning—where skills should be constructed to reliably achieve a goal that the agent is able to reach (in contrast to the exploration case, where the primary difficulty is finding the goal itself). This work extended PI Konidaris's early work on skill chaining [Konidaris and Barto, 2009], where an agent constructs a sequence of options that target the goal. The skills are constructed so that successful execution of each option in the chain allows the agent to execute another option, which brings it closer still to its eventual goal.

While skill chaining was capable of discovering skills in continuous state spaces, it could only be applied to relatively low-dimensional state-spaces with discrete actions. Our new work combined the core insights of skill chaining with recent advances in using non-linear function approximation. The resulting algorithm, *deep skill chaining*, scales to high-dimensional problems with continuous state and action spaces. Through a series of experiments on five challenging domains in the MuJoCo physics simulator [Todorov et al., 2012], we show that deep skill chaining can solve tasks that otherwise cannot be solved by non-hierarchical agents in a reasonable amount of time. Furthermore, the new algorithm outperforms state-of-the-art deep skill discovery algorithms [Bacon et al., 2017, Levy et al., 2019] in these tasks (Figure 6).

3 Results in Submission or in Progress

This section briefly covers three ongoing efforts resulting from this project, which should result in publications appearing this year.

8. A. Bagaria and G.D. Konidaris. Planning and Exploration by Building Skill Graphs. To be submitted, *Neural Information Processing Systems*, 2020.

This work extends our prior work on skill chaining—which focuses on constructing chains of skills towards a well-defined goal—to instead construct general *graphs* of inter-connected skills that can be used by the agent to reach any area of the state space. The resulting skills are constructed in a task-agnostic manner, without the use of a reward function, and are subsequently useful in two scenarios. First, when an agent is given a new goal, a pre-learned skill graph will allow it to plan to a known location near that goal, from where it can use reinforcement learning to learn to reach it. This is the task-agnostic generalization of skill chaining, and we expect it will be useful for building abstract layers for many difficult control tasks. Second, for very long-horizon, sparse reward tasks, skills graphs enable an agent to move between the exploratory skills discovered by covering options (project outputs 5 and 6). The result is a network of skills enabling an agent to effectively explore the frontiers of its knowledge in very large MDPs.

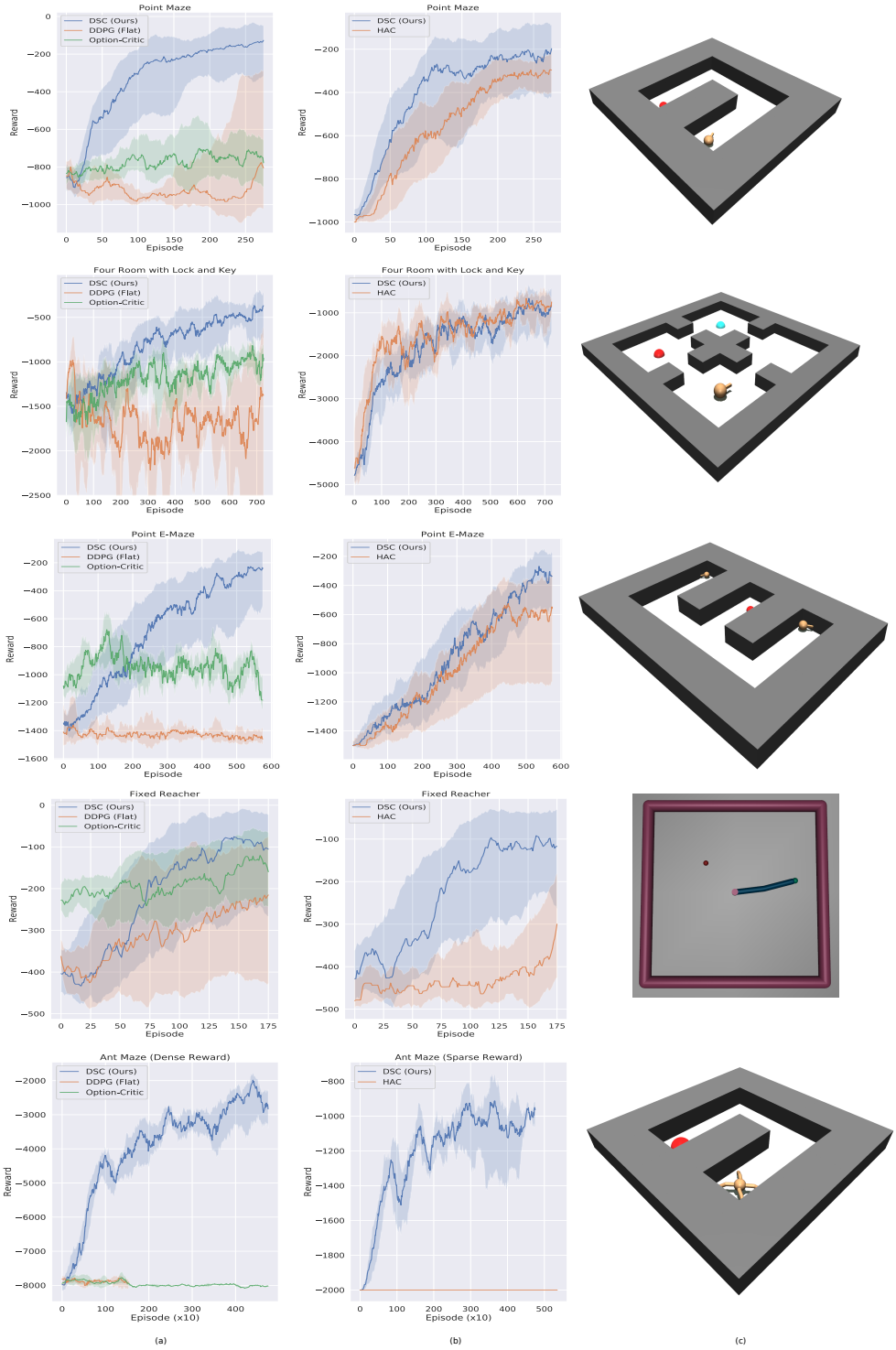


Figure 6: (a) Learning curves comparing deep skill chaining (DSC), a flat agent (DDPG) and Option-Critic. (b) Comparison with Hierarchical Actor Critic (HAC). (c) the continuous control tasks corresponding to the learning curves in (a) and (b). All curves are averaged over 20 runs, except for Ant Maze which was averaged over 5 runs.

9. G.D. Konidaris, S. James, D. Abel, and A. Levy. Constructing Hierarchies of Markov Decision Processes. To be submitted, *Journal of Machine Learning Research*.

This work is a primarily theoretical, and shows how to create a hierarchy of increasingly abstract Markov decision processes by alternating action- and state-abstraction phases. It establishes the semantics of such a hierarchy by showing that abstract states must ground to distributions over, rather than a partition of, lower-level states. The paper introduces a planning algorithm that exploits the resulting hierarchical structure to rapidly find a high-level plan, and can use additional computation time to iteratively refine that plan by increasing the probability of success and the resulting expected reward.

10. S. James, B. Rosman, and G.D. Konidaris. Learning Object-Centric Abstractions for High-Level Planning. Under review, *Proceedings of the 37th International Conference on Machine Learning*, 2020.

This work builds on PI Konidaris's symbol-learning framework, which is limited because it learns highly task-specific task-specific representations that must be relearned for any new task, or even any small change to an existing task. That is impractical for long-lived agents that solve multiple tasks in complex environments. We therefore extended that method by including additional structure—namely, that the world consists of objects, and that similar objects are common amongst tasks. For example, when we play video games, we solve the game quickly by leveraging our existing knowledge of objects and their affordances (such as doors and ladders which occur across multiple levels) [Dubey et al., 2018]. Similarly, robot manipulation tasks often use the same robot and a similar set of physical objects in different configurations. This can substantially improve learning efficiency, because an object-centric model can be reused wherever that same object appears in a problem, and can also be generalized across similar objects—object *types*.

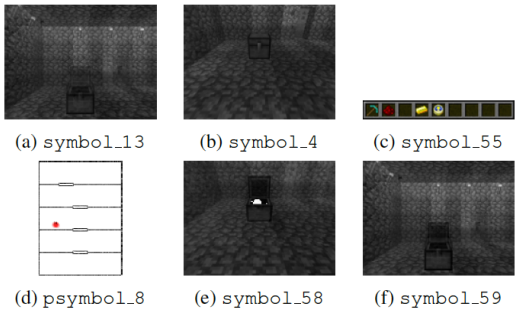
This work introduces a method for building object-centric abstractions that specify both the abstract object attributes that support high-level planning, and an object-relative lifted transition model that can be instantiated in a new problem. This reduces the number of samples required to learn a new task by allowing the agent to avoid relearning the dynamics of previously-seen objects. We apply it to a series of Minecraft tasks [Johnson et al., 2016], resulting in an agent that autonomously learns an abstract representation of a complex, high-dimensional task, from raw pixel input (Figure 7). Our results show that an agent can leverage these portable abstractions to learn a representation of new Minecraft tasks using a diminishing number of samples, allowing the agent to construct plans consisting of hundreds of low-level actions (Figure 8).

4 Conclusions and Future Directions

This project focused on enabling goal-directed decision-making in complex, unstructured tasks, through principled approaches to learning action abstractions and symbolic perceptual abstractions. It has succeeded in advancing the state of the art in both theory and practice in these two areas.

I would like to make two specific conclusions:

1. These abstract representations—especially those built in our in-submission ICML paper—are learned using *tens* or *hundreds* of experiences, and result in agents capable of *much* more complex behavior than agents attempting to solve tasks like Minecraft using end-to-end deep learning, which require *millions or billions* of experiences to learn to perform much simpler tasks. *These approaches are the*



(a) symbol_13 (b) symbol_4 (c) symbol_55

(d) psymbol_8 (e) symbol_58 (f) symbol_59

```
(:action Open-Chest-partition-0
 :parameters (?w - type0 ?x - type6 ?y - type9)
 :precondition (and (notfailed) (symbol_13 ?w)
                    (symbol_4 ?x) (symbol_55 ?y)
                    (psymbol_8))
 :effect (and (symbol_58 ?x) (symbol_59 ?w)
              (not (symbol_4 ?x)) (not (symbol_13 ?w)))
)
```

(g) A learned typed PDDL operator for the `Open-Chest` skill. The predicate in red indicates a problem-specific symbol that must be relearned for each new task, while the rest of the operator can be safely transferred.

Figure 7: Our approach learns that, in order to open a chest, the agent must be standing in front of a chest (`symbol_13`), the chest must be closed (`symbol_4`), the inventory must contain a clock (`symbol_55`) and the agent must be standing at a certain location (`psymbol_8`). The result is that the agent finds itself in front of an open chest (`symbol_58`) and the chest is open (`symbol_59`). `type0` refers to the "agent" class, `type6` the "chest" class and `type9` the "inventory" class.

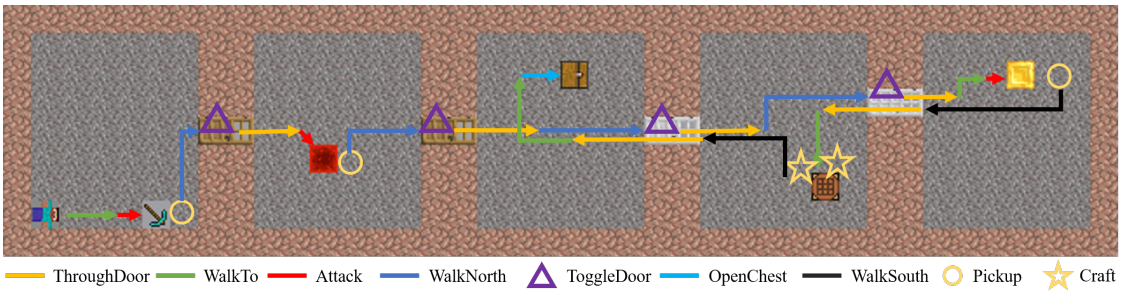


Figure 8: The path traced by the agent solving the a complex Minecraft task. Colored lines and shapes represent different option executions.

most promising approach to drastically improving the state of the art in learning to solve complex, sequential, long-horizon tasks like Minecraft from raw pixel data.

2. These approaches provide a *principled means of bridging classical AI concepts like high-level abstract planning and object-centric knowledge representation*, which are necessary for generating complex, goal-directed behavior, and ultimately for general-purpose AI. The representation shown in Figure 7 is a *learned, grounded, symbolic representation* that enables long-horizon, compositional planning, and offers a bridge to 50 years of research in classical AI. The approaches developed in this project have constructed a link between these two primary models of AI research, offering a principled way to combine them that supports the strength of both.

References

Karl J Åström. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

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

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

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

R. Dubey, P. Agrawal, D. Pathak, T.L. Griffiths, and A.A. Efros. Investigating human priors for playing video games. In *International Conference on Machine Learning*, 2018.

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

M. Johnson, K. Hofmann, T. Hutton, and D. Bignell. The malmo platform for artificial intelligence experimentation. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 4246–4247, 2016.

G.D. Konidaris and A.G. Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems*, pages 1015–1023, 2009.

A. Levy, G.D. Konidaris, R. Platt, and K. Saenko. Learning multi-level hierarchies with hindsight. In *Proceedings of the Eighth International Conference on Learning Representations*, 2019.

Marlos C Machado, Marc G Bellemare, and Michael Bowling. A Laplacian framework for option discovery in reinforcement learning. In *Proceedings of the Thirty-fourth International Conference on Machine Learning*, 2017.

Timothy Mann and Shie Mannor. Scaling up approximate value iteration with options: Better policies with fewer iterations. In *International Conference on Machine Learning*, pages 127–135, 2014.

D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1–4, 2007.

D Silver and K Ciosek. Compositional planning using optimal option models. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1063–1070, 2012.

Özgür Şimşek and Andrew G Barto. Skill characterization based on betweenness. In *Advances in Neural Information Processing Systems*, pages 1497–1504, 2009.

S.P. Singh, T.S. Jaakkola, and M.I. Jordan. Learning without state-estimation in partially observable Markovian decision processes. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 284–292, 1994.

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

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

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.