Published in final edited form as:

Curr Opin Behav Sci. 2019 October ; 29: 1–7. doi:10.1016/j.cobeha.2018.11.005.

On The Necessity of Abstraction

George Konidaris

Computer Science Department, Brown University 115 Waterman Street Providence RI 02906

Abstract

A generally intelligent agent faces a dilemma: it requires a complex sensorimotor space to be capable of solving a wide range of problems, but many tasks are only feasible given the right problem-specific formulation. I argue that a necessary but understudied requirement for general intelligence is the ability to form task-specific abstract representations. I show that the reinforcement learning paradigm structures this question into how to learn action abstractions and how to learn state abstractions, and discuss the field's progress on these topics.

1. Introduction

AI has recently produced a series of encouraging breakthroughs, particularly in reinforcement learning [1], which studies complete agents learning to act in an environment and therefore encapsulates the entirety of the AI problem. These results constitute undeniable progress in constructing generally intelligent AIs. However, one major aspect of general intelligence remains largely unaddressed. Consider chess.

AlphaZero essentially solves chess [2]. It does so purely by self-play, in approximately four hours, with no human intervention and without the benefit of specialist chess knowledge. That is a major achievement. But there *is* some human knowledge embedded here: it is in the *problem formulation* itself. AlphaZero takes as input a neural encoding of an abstract description of a chessboard as an 8 × 8 array of positions, each of which can be empty or contain one of six piece types of one of two colors; the actions available to it are legal moves for that abstract input: the pawn on e2 can be moved to e4. This abstract representation contains just the information relevant to playing chess and nothing else, and it is given to the agent as if it is a natural part of the environment. That is perfectly appropriate for a chess-playing agent, but it finesses a problem that must be solved by a general-purpose AI.

Consider one particular general-purpose agent that reasonably approximates a human—a robot with a native sensorimotor space consisting of video input and motor control output—and the chessboards shown in Figure 1. Those chessboards are actually perceived by the

Declaration of Interests

George Konidaris is the Chief Roboticist of Realtime Robotics, a robotics company that produces a specialized motion planning processor.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

robot as high-resolution color images, and the only actions it can choose to execute are to actuate the motors attached to its joints. A general-purpose robot cannot expect to be given an abstract representation suitable for playing chess, just as it cannot expect to be given one that is appropriate for scheduling a flight, playing Go, juggling, driving cross-country, or composing a sonnet. Nevertheless, it should be able to do all of these things.

The only computationally feasible way for a general AI to learn to play chess is to build a representation like that used by AlphaZero: an abstract representation of the board and the legal moves. The robot must be able to do this irrespective of the particular angle from which it views the board; varying sizes and colors of chess pieces, squares, and boards; different backgrounds, lighting conditions, and other extraneous information; how many joints it can actuate; and what gripper type it has. Those difficulties all have to do with the innate complexity of the *robot*, not the essential complexity of the *task*—the chessboards shown in Figure 1 are all in the same game position, despite their very different appearances, and irrespective of the details of the robot's body. A general-purpose AI can only be effective when it is able to focus solely on the complexity of task. Consequently, a precondition for general AI is the ability to *construct an appropriate—and problem-specific —abstract representation of a new problem.* Humans do this effortlessly, even though such representations cannot be hard-wired into our brain—nothing remotely similar to chess, for example, appears in our evolutionary history.

2. Forms of Abstraction

Fortunately, the reinforcement learning formalization helps structure our reasoning about abstraction. Reinforcement learning problems are typically formalized as Markov Decision Processes (or MDPs), described by a tuple:

$$M = (S, A, R, T, \gamma), \quad (1)$$

where S is a set of states; A is a set of actions; $R(s, a, s')$ returns the reward obtained by executing action a from state s and arriving in state s'; and $T(s'|s, a)$ encodes the task transition dynamics, a distribution over states s' the agent may enter into after executing action a at state s; and $\gamma \in (0, 1]$ is a discount factor expressing a preference for immediate over future reward. Of these, the reward function and discount factor describe the agent's objectives, while the transition function describes the operation of the environment.

It is reasonable to model the operation of a generally intelligent agent as a single MDP—the *base* or *root* MDP—where the state and action space may be high dimensional and continuous, and the reward function (and possibly discount factor) can be varied to reflect the current task.[1] Then since the transition function depends on the state and action set, constructing an abstract task-specific MDP requires two types of abstraction: *state*

[1] Of course, things are more complex for both humans and robots, since (at the very least) the assumption that the state is observable is too strong.

abstraction, where an agent builds an abstract state set $\bar{S}$, and *action abstraction*, where it builds an abstract action set $\bar{A}$.

3. Learning State Abstractions

Learning a state abstraction involves finding a mapping from the original state space S to another more compact space $\bar{S}$ that is sufficient for solving the task at hand. Such approaches have always been data-driven, and most are constructed to accurately but compactly represent some aspect of the agent's learning process, either exactly [3] or with bounded loss [4, 5, 6].

The earliest state abstraction methods focused on constructing small discrete state spaces. Bisimulation approaches attempt to preserve the complete transition model of the task, either exactly [7] or approximately [8, 9, 10, 11]; unfortunately the resulting model minimization problem is NP-Hard [9]. Later state aggregation approaches [12, 13, 14, 15, 16, 17, 6] collapse sets of states from the original state space into undifferentiated single states in the abstract space, based on measures such as the topology of the original state space.

Several approaches find abstract representations by selectively ignoring state variables, for example by selecting an appropriate abstraction from a library [18, 19, 20, 21, 22], discarding irrelevant state variables [23, 24], or starting with no state relevant variables and adding some back in when necessary [25]. Such approaches became more principled with the introduction of feature selection methods drawn from linear regression [26, 27, 28, 29, 30, 31] to selectively include state features with the aim of learning accurate but compact representations.

Rather than discarding a subset of an existing state space, *representation discovery* approaches construct an entirely new compact state space that preserves some properties of the task. Example methods have centered on preserving the topology of the domain [32, 33], the ability to predict the next state [34, 35], and conformance with our prior knowledge of physical systems [36, 37]. Figure 2 shows an example learned state abstraction.

A substantial shift occurred recently with the application of deep neural networks to reinforcement learning, which has in some cases successfully learned policies directly from raw sensorimotor data [38, 39]. The most impressive of these from a general AI standpoint is the use of a single network architecture and learning algorithm to master a large number of Atari games directly from raw pixel input [39]. At first blush, deep networks are just powerful function approximators, unrelated to state space abstraction. However, that view misses what makes them so powerful. Their deep structure forces state inputs to go through layers of processing, with policies depending only on the final layer. This structure strongly encourages the network to learn a highly processed transformation of the input state into a new representation [40] suitable for supporting a policy. The now-widespread use of autoencoders and pre-training [41, 42, 43, 38, 44, 45, 46]—where a deep network learns to compute a compact feature vector sufficient for reproducing its own input and the policy is learned as a function of that feature vector only—closely corresponds to representation discovery approaches. Therefore, while it may at first seem that deep networks successfully avoid state abstraction, it is likely that their success stems at least partly from their ability to do just that.

4. Learning Action Abstractions

Action abstraction involves constructing a set $\bar{A}$ of higher-level actions (sometimes called *skills*) out of an agent's available low-level (also often called *primitive*) actions. Most research in this area has adopted the *options framework* [47], which provides methods for learning and planning using high-level *options* defined by a tuple $o = \left(I_o, \pi_o, \beta_o\right)$. The agent can execute option *o* in any state in the option's *initiation set* $I_o \subseteq S$, whereafter the agent executes actions according to the *option policy* $\pi_o : S \to A$, until option execution stops in some state *s* according the *termination condition* probability $\beta_o : S \to [0, 1]$. The agent selects options in just the same way that it selects low-level actions.

The core question here has always been *skill discovery*—how to identify, from data or an explicit task description—a useful collection of options. In practice this amounts to identifying the termination condition β_o (often called a *subgoal*), around which the other components can be constructed: a synthetic reward function R_o that rewards entering β_o is used to learn the option policy (now just another reinforcement learning problem), and the initiation set includes only states from which that policy succeeds in reaching β_o.

The majority of work in skill discovery has centered on the somewhat heuristic identification of the desirable properties of subgoals, and then the development of algorithms for constructing options with those properties. Examples include a high likelihood of visiting high-reward or high-novelty states [48, 49, 50, 51], repeated subpolicies [52, 53], reaching various topological features of the state space like bottlenecks, graph clustering boundaries, and high between-ness states [54, 55, 12, 56, 57, 58, 59, 60, 15, 61], reaching specific discretized state-variable values [62, 13], generating diverse behavior [63, 64] or constructing skills that can be chained to solve the task [65, 66]. Figure 3 shows an example subgoal identification using between-ness centrality.

Unfortunately, however, there is evidence that poorly chosen options can slow learning [68]. Possibly stimulated by this result, a new wave of recent work—initiated by Solway et al. [69]—has defined an explicit performance cri-terion that adding options should improve, and sought to optimize it. This has been approached both by constructing algorithms with performance guarantees [69, 70, 71] and by adding parameters describing options to the agent's learning task and directly optimizing them [72, 73, 16, 74, 75, 76]. Unfortunately recent complexity results have shown that even a very simple instantiation of the resulting problem is NP-Hard [71].

A critical distinction here is between approaches that aim to speed the learning of a *single task,* and those which may accept a temporary reduction in learning speed for one task with the aim of improving performance over future tasks, known as *skill transfer* [77, 78, 53]. The challenge in the single-task case is overcoming the additional cost of discovering the options; this results in a narrow opportunity for performance improvements, but a well-defined objective. In the skill transfer case, the key challenge is predicting the usefulness of a particular option to future tasks, given limited data.

5. Combined State and Action Abstraction

The vast majority of current research focuses on attacking state or action abstraction in isolation. This is a perfectly reasonable research strategy given the difficulty of each problem, but it seems unlikely to obtain a coherent model when both are required. A major question is therefore how one type of abstraction can drive the other.

State abstraction can drive action abstraction by first constructing a state abstraction and then building a corresponding action set, typically in the form of actions that move between abstract states. All skill discovery algorithms that rely on a clustering or partitioning of the state space already implicitly do this, while some (e.g., [12, 13, 15, 16, 17]) explicitly construct the resulting abstract MDP.

A second, much less explored alternative is to have action abstraction drive state abstraction: first discover abstract actions, and then build a state abstraction that supports planning with them [79, 80, 81, 82, 83]. My own recent work in this area [84] constructs an abstract representation that is provably necessary and sufficient for computing the probability that a sequence of given options can be executed (and the reward obtained if successful). The resulting framework is capable of learning abstract representations directly from sensorimotor experience, to solve a manipulation task on a complex mobile robot platform. An example abstract model of a skill, along with visualizations of the abstract symbolic propositions appearing in it, is shown in Figure 4.

The key challenge in learning state-driven hierarchies is that it is easy to construct state abstractions for which no set of feasible options can be constructed. This occurs because options are constrained to be realizable in the environment—executing the option from the initiation set must result in the agent reaching the termination condition—but no such restrictions hold for state abstractions. This does not occur when learning action-driven hierarchies, where the agent is free to select an abstraction that supports the skills it has built, but then everything hinges on the skill discovery algorithm, which must solve a very hard problem.

6. Conclusions

The tension between narrow and general AI—between producing agents that solve specific problems extremely well, versus agents that can solve a wide variety of problems adequately—has always been a core dilemma in AI research. The modern AI paradigm [85] addresses this challenge by precisely formulating *general classes of problems* (e.g., MDPs) and designing algorithms targeting the entire class, rather than any specific instance of it. That approach led to the widespread development of powerful general-purpose algorithms, but it omits the key step of *having the agent itself formulate a specific problem in terms of a general problem class* as a precursor to applying a general algorithm—a step that is a necessary precondition for solving the AI problem.

Acknowledgements

The author would like to thank Dave Abel and Yuu Jinnai for their useful comments on this manuscript. This research was supported in part by the National Institutes of Health under award number R01MH109177. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The content is solely the responsibility of the author and does not necessarily represent the official views of the National Institutes of Health.

References

[1]. Sutton R, Barto A, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.

[2]. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T, et al., Mastering chess and shogi by self-play with a general reinforcement learning algorithm, arXiv:1712.01815.

[3]. Li L, Walsh T, Littman M, Towards a unified theory of state abstraction for MDPs, in: Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics, 2006, pp. 531–539.

[4]. Roy BV, Performance loss bounds for approximate value iteration with state aggregation, Mathematics of Operations Research 31 (2) (2006) 234–244.

[5]. Abel D, Hershkowitz D, Littman M, Near optimal behavior via approximate state abstraction, in: Proceedings of The 33rd International Conference on Machine Learning, 2016, pp. 2915–2923,* The authors formalize the notion of an approximate (rather than exact) state abstraction, and provide performance bounds for four types of approximate state abstractions.

[6]. Abel D, Arumugam D, Lehnert L, Littman M, State abstractions for lifelong reinforcement learning, in: Proceedings of the 35th International Conference on Machine Learning, 2018, pp. 10–19.

[7]. Dean T, Givan R, Model minimization in Markov decision processes, in: In Proceedings of the Fourteenth National Conference on Artificial Intelligence, 1997, pp. 106–111.

[8]. Dean T, Givan R, Leach S, Model reduction techniques for computing approximately optimal solutions for Markov decision processes, in: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, 1997, pp. 124–131.

[9]. Even-Dar E, Mansour Y, Approximate equivalence of Markov decision processes, in: Learning Theory and Kernel Machines, Springer, 2003, pp. 581–594.

[10]. Ferns N, Panangaden P, Precup D, Metrics for finite Markov decision processes, in: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, 2004, pp. 162–169.

[11]. Ferns N, Castro P, Precup D, Panangaden P, Methods for computing state similarity in Markov decision processes, in: Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence, 2006, pp. 174–181.

[12]. Bakker B, Schmidhuber J, Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization, in: Proceedings of the 8th Conference on Intelligent Autonomous Systems, 2004, pp. 438–445.

[13]. Mugan J, Kuipers B, Autonomous learning of high-level states and actions in continuous environments, IEEE Transactions on Autonomous Mental Development 4 (1) (2012) 70–86.

[14]. Mandel T, Liu Y-E, Brunskill E, Popovic Z, Efficient Bayesian clustering for reinforcement learning, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 1830–1838.

[15]. Krishnamurthy R, Lakshminarayanan A, Kumar P, Ravindran B, Hierarchical reinforcement learning using spatio-temporal abstractions and deep neural networks, arXiv:1605.05359.

[16]. Vezhnevets A, Osindero S, Schaul T, Heess N, Jaderberg M, Silver D, Kavukcuoglu K, FeUdal networks for hierarchical reinforcement learning, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 3540–3549.

[17]. Kompella VR, Stollenga M, Luciw M, Schmidhuber J, Continual curiosity-driven skill acquisition from high-dimensional video inputs for humanoid robots, Artificial Intelligence 247 (2017) 313–335.

[18]. Diuk C, Li L, Leffler B, The adaptive k-meteorologists problems and its application to structure learning and feature selection in reinforcement learning, in: Proceedings of the 26th International Conference on Machine Learning, 2009, pp. 249–256.

[19]. Konidaris G, Barto A, Efficient skill learning using abstraction selection, in: Proceedings of the Twenty First International Joint Conference on Artificial Intelligence, 2009, pp. 1107–1112.

[20]. van Seijen H, Whiteson S, Kester L, Efficient abstraction selection in reinforcement learning, Computational Intelligence 30 (4) (2013) 657–699.

[21]. Cobo L, Subramanian K, Isbell C, Lanterman A, Thomaz A, Abstraction from demonstration for efficient reinforcement learning in high-dimensional domains, Artificial Intelligence 216 (2014) 103–128.

[22]. Jiang N, Kulesza A, Singh S, Abstraction selection in model-based reinforcement learning, in: Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 179–188.

[23]. Jong N, Stone P, State abstraction discovery from irrelevant state variables, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence, 2005, pp. 752–757.

[24]. Mehta N, Ray S, Tadepalli P, Dietterich T, Automatic discovery and transfer of MAXQ hierarchies, in: Proceedings of the Twenty Fifth International Conference on Machine Learning, 2008, pp. 648–655.

[25]. McCallum A, Learning to use selective attention and short-term memory in sequential tasks, in: From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, 1996, pp. 315–324.

[26]. Kroon M, Whiteson S, Automatic feature selection for model-based reinforcement learning in factored MDPs, in: Proceedings of the International Conference on Machine Learning and Applications, 2009, pp. 324–330.

[27]. Kolter J, Ng A, Regularization and feature selection in least-squares temporal difference learning, in: Proceedings of the 26th International Conference on Machine Learning, 2009, pp. 521–528.

[28]. Johns J, Painter-Wakefield C, Parr R, Linear complementarity for regularized policy evaluation and improvement, in: Advances in Neural Information Processing Systems 23, 2010, pp. 1009–1017.

[29]. Nguyen T, Li Z, Silander T, Leong T-Y, Online feature selection for model-based reinforcement learning, in: Proceedings of the 30th International Conference on Machine Learning, 2013, pp. 498–506.

[30]. Rosman B, Feature selection for domain knowledge representation through multitask learning, in: Proceedings of the 2014 Joint IEEE Conference on Development and Learning and Epigenetic Robotics, 2014, pp. 216–221.

[31]. Wookey D, Konidaris G, Regularized feature selection in reinforcement learning, Machine Learning 100 (2–3) (2015) 655–676.

[32]. Mahadevan S, Maggioni M, Ferguson K, Osentoski S, Learning representation and control in continuous Markov decision processes, in: Proceedings of the Twenty First National Conference on Artificial Intelligence, 2006, pp. 1194–1199.

[33]. Luciw M, Schmidhuber J, Low complexity proto-value function learning from sensory observations with incremental slow feature analysis, in: Proceedings of the International Conference on Artificial Neural Networks, 2012, pp. 279–287.

[34]. Sprague N, Predictive projections, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009, pp. 1223–1229.

[35]. Boots B, Siddiqi S, Gordon G, Closing the learning-planning loop with predictive state representations, The International Journal of Robotics Research 30 (7) (2011) 954–966.

[36]. Scholz J, Levihn M, Isbell C, Wingate D, A physics-based model prior for object-oriented MDPs, in: Proceedings of the 31st International Conference on Machine Learning, 2014, pp. 1089–1097.

[37]. Jonschkowski R, Brock O, Learning state representations with robotic priors, Autonomous Robots 39 (3) (2015) 407–428.

[38]. Levine S, Finn C, Darrell T, Abbeel P, End-to-end training of deep visuomotor policies, Journal of Machine Learning Research 17 (1) (2016) 1334–1373.

[39]. Mnih V, Kavukcuoglu K, Silver D, Rusu A, Veness J, Bellemare M, Graves A, Riedmiller M, Fidjeland A, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D,

Legg S, Hassabis D, Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533. [PubMed: 25719670]

[40]. Bengio Y, Courville A, Vincent P, Representation learning: A review and new perspectives, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8) (2013) 1798–1828. [PubMed: 23787338]

[41]. Lange S, Riedmiller M, Deep auto-encoder neural networks in reinforcement learning, in: Proceedings of the 2010 International Joint Conference on Neural Networks, 2010.

[42]. Lange S, Riedmiller M, Voigtlander A, Autonomous reinforcement learning on raw visual input data in a real world application, in: Proceedings of the 2012 International Joint Conference on Neural Networks, 2012.

[43]. Oh J, Guo X, Lee H, Lewis R, Singh S, Action-conditional video prediction using deep networks in Atari games, in: Advances in Neural Information Processing Systems 28, 2015, pp. 2863–2871.

[44]. Finn C, Tan X, Duan Y, Darrell T, Levine S, Abbeel P, Deep spatial autoencoders for visuomotor learning, in: Proceedings of the 2016 IEEE International Conference on Robotics and Automation, 2016, pp. 512–519.

[45]. Higgins I, Pal A, Rusu A, Matthey L, Burgess C, Pritzel A, Botvinick M, Blundell C, Lerchner A, DARLA: Improving zero-shot transfer in reinforcement learning, in: Proceedings of the Thirty-Fourth International Conference on Machine Learning, 2017, pp. 1480–1490.

[46]. Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, Lerchner A, β-VAE: Learning basic visual concepts with a constrained variational framework, in: Proceedings of the Fifth International Conference on Learning Representations, 2017.

[47]. Sutton R, Precup D, Singh S, Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning, Artificial Intelligence 112 (1–2) (1999) 181–211.

[48]. McGovern A, Barto A, Automatic discovery of subgoals in reinforcement learning using diverse density, in: Proceedings of the Eighteenth International Conference on Machine Learning, 2001, pp. 361–368.

[49]. Stolle M, Precup D, Learning options in reinforcement learning, in: Proceedings of the 5th International Symposium on Abstraction, Reformulation and Approximation, 2002, pp. 212–223.

[50]. Şimşek Ö, Barto A, Using relative novelty to identify useful temporal abstractions in reinforcement learning, in: Proceedings of the Twenty-First International Conference on Machine Learning, 2004, pp. 751–758.

[51]. Asadi M, Huber M, Effective control knowledge transfer through learning skill and representation hierarchies, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007, pp. 2054–2059.

[52]. Pickett M, Barto A, PolicyBlocks: An algorithm for creating useful macro-actions in reinforcement learning, in: Proceedings of the Nineteenth International Conference of Machine Learning, 2002, pp. 506–513.

[53]. Topin N, Haltmeyer N, Squire S, Winder J, DesJardins M, Mac-Glashan J, Portable option discovery for automated learning transfer in object-oriented Markov decision processes, in: Proceedings of the 24th International Conference on Artificial Intelligence, 2015, pp. 3856–3864.

[54]. Menache I, Mannor S, Shimkin N, Q-cut—dynamic discovery of sub-goals in reinforcement learning, in: Proceedings of the Thirteenth European Conference on Machine Learning, 2002, pp. 295–306.

[55]. Mannor S, Menache I, Hoze A, Klein U, Dynamic abstraction in reinforcement learning via clustering, in: Proceedings of the Twenty-First International Conference on Machine Learning, 2004, pp. 560–567.

[56]. Şimşek Ö, Wolfe A, Barto A, Identifying useful subgoals in reinforcement learning by local graph partitioning, in: Proceedings of the Twenty-Second International Conference on Machine Learning, 2005, pp. 816–823.

[57]. Şimşek Ö, Barto A, Skill characterization based on betweenness, in: Advances in Neural Information Processing Systems 22, 2008, pp. 1497–1504.

[58]. Metzen J, Online skill discovery using graph-based clustering, in: Proceedings of the 10th European Workshop on Reinforcement Learning, 2012, pp. 77–88.

[59]. Moradi P, Shiri M, Rad A, Khadivi A, Hasler M, Automatic skill acquisition in reinforcement learning using graph centrality measures, Intelligent Data Analysis 16 (1) (2012) 113–135.

[60]. Bacon P-L, On the bottleneck concept for options discovery: Theoretical underpinnings and extension in continuous state spaces, Master's thesis, McGill University (2013).

[61]. Machado M, Bellemare M, Bowling M, A Laplacian framework for option discovery in reinforcement learning, in: Proceedings of the Thirty-Fourth International Conference on Machine Learning, 2017, pp. 2295–2304,* A skill discovery approached based on analysis of the topology of the state space, via the graph Laplacian. Because the approach is based on the graph induced by sample transitions, and not the state space directly, it is able to scale up to high-dimensional continuous domains.

[62]. Hengst B, Discovering hierarchy in reinforcement learning with HEXQ, in: Proceedings of the Nineteenth International Conference on Machine Learning, 2002, pp. 243–250.

[63]. Daniel C, Neumann G, Kroemer O, Peters J, Hierarchical relative entropy policy search, The Journal of Machine Learning Research 17 (1) (2016) 3190–3239.

[64]. Eysenbach B, Gupta A, Ibarz J, Levine S, Diversity is all you need: Learning skills without a reward function, arXiv: 1802.06070.

[65]. Konidaris G, Barto A, Skill discovery in continuous reinforcement learning domains using skill chaining, in: Advances in Neural Information Processing Systems 22, 2009, pp. 1015–1023.

[66]. Florensa C, Held D, Geng X, Abbeel P, Automatic goal generation for reinforcement learning agents, in: Proceedings of the 35th International Conference on Machine Learning, 2018, pp. 1515–1528.

[67]. Şimşek Ö, Behavioral building blocks for autonomous agents: description, identification, and learning, Ph.D. thesis, University of Massachusetts Amherst (2008).

[68]. Jong N, Hester T, Stone P, The utility of temporal abstraction in reinforcement learning, in: Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems, 2008, pp. 299–306.

[69]. Solway A, Diuk C, Cordova N, Yee D, Barto A, Niv Y, Botvinick M, Optimal behavioral hierarchy, PLOS Computational Biology 10 (8),** The authors formalize an objective function for hierarchies—average learning performance across a set of possible future tasks—and then provide experimental evidence that the hierarchies discovered by humans are maximizing it.

[70]. Brunskill E, Li L, PAC-inspired option discovery in lifelong reinforcement learning, in: Proceedings of the Thirty-First International Conference on Machine Learning, 2014, pp. 316–324.

[71]. Jinnai Y, Abel D, Hershkowitz D, Littman M, Konidaris G, Finding options that minimize planning time, ArXiv: 1810.07311.

[72]. Bacon P-L, Harb J, Precup D, The option-critic architecture, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 1726–1734.

[73]. Levy A, Platt R, Saenko K, Hierarchical actor-critic, ArXiv:1712.00948.

[74]. Levy A, Platt R, Saenko K, Hierarchical reinforcement learning with hindsight, ArXiv: 1805.08180.

[75]. Harb J, Bacon P-L, Klissarov M, Precup D, When waiting is not an option: Learning options with a deliberation cost, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 3165–3172.

[76]. Nachum O, Gu S, Lee H, Levine S, Data-efficient hierarchical reinforcement learning, in: Advances in Neural Information Processing 31, 2018.

[77]. Konidaris G, Barto A, Building portable options: Skill transfer in reinforcement learning, in: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, 2007, pp. 895–900.

[78]. Konidaris G, Kuindersma S, Grupen R, Barto A, Autonomous skill acquisition on a mobile manipulator, in: Proceedings of the Twenty-Fifth Conference on Artificial Intelligence, 2011, pp. 1468–1473.

[79]. Jetchev N, Lang T, Toussaint M, Learning grounded relational symbols from continuous data for abstract reasoning, in: Proceedings of the 2013 ICRA Workshop on Autonomous Learning, 2013.

[80]. Konidaris G, Kaelbling L, Lozano-Perez T, Constructing symbolic representations for high-level planning, in: Proceedings of the Twenty-Eighth Conference on Artificial Intelligence, 2014, pp. 1932–1940.

[81]. Ugur E, Piater J, Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2015, pp. 2627–2633.

[82]. Ugur E, Piater J, Refining discovered symbols with multi-step interaction experience, in: Proceedings of the 15th IEEE-RAS International Conference on Humanoid Robots, 2015, pp. 1007–1012.

[83]. Konidaris G, Kaelbling L, Lozano-Perez T, Symbol acquisition for probabilistic high-level planning, in: Proceedings of the Twenty Fourth International Joint Conference on Artificial Intelligence, 2015, pp. 3619–3627.

[84]. Konidaris G, Kaelbling L, Lozano-Perez T, From skills to symbols: Learning symbolic representations for abstract high-level planning, Journal of Artificial Intelligence Research 61 (2018) 215–289,** This paper shows how to construct a necessary and sufficient state abstraction for planning using a collection of high-level skills, and applies that theory to learn an abstract representation of a robot manipulation task.

[85]. Russell S, Norvig P, Artificial Intelligence: A Modern Approach, Prentice Hall, Eaglewood Cliffs, New Jersey, 1995.

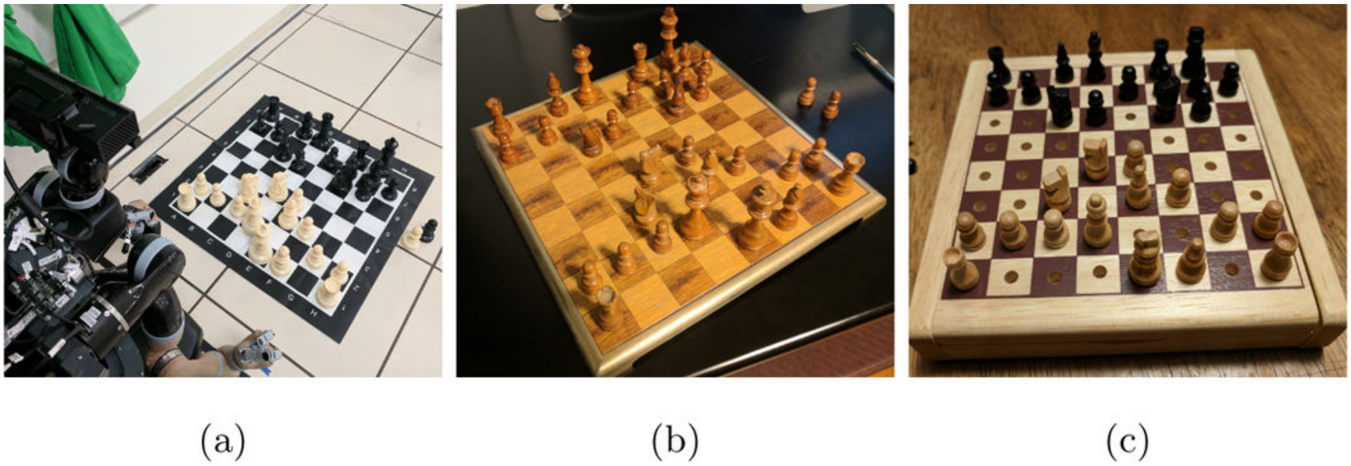


Figure 1:

Three different chessboards with the pieces in the same position.

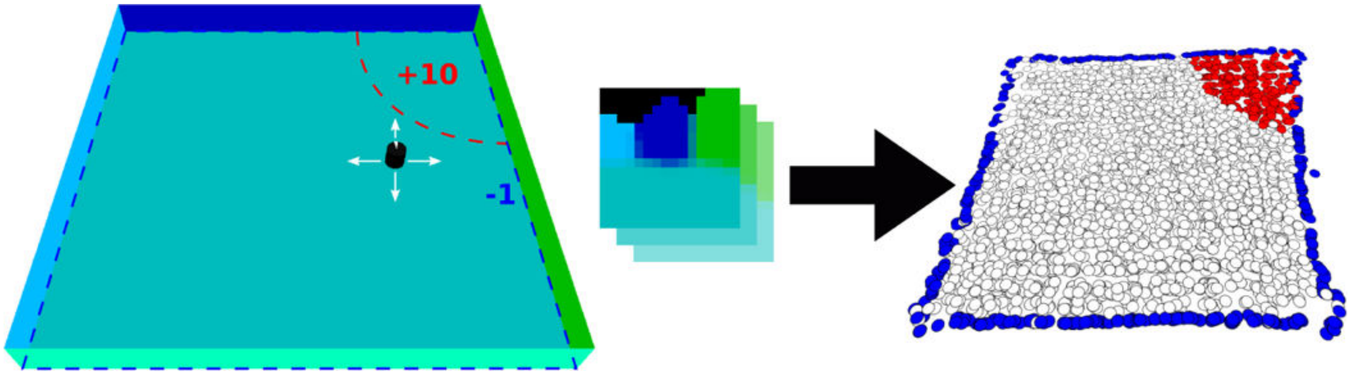


Figure 2:

A robot operates in a square room, where its reward is −1 per step everywhere except the upper right corner, where it receives a positive reward (left). Its native state space is a short history of images observed with a front-facing camera (middle). Using a representation discovery algorithm based on physical priors [37], the robot discovers a low-dimensional representation that accurately reects the topology of the task (right) from its complex sensor space. Reused with permission from Jonschkowski and Brock [37].

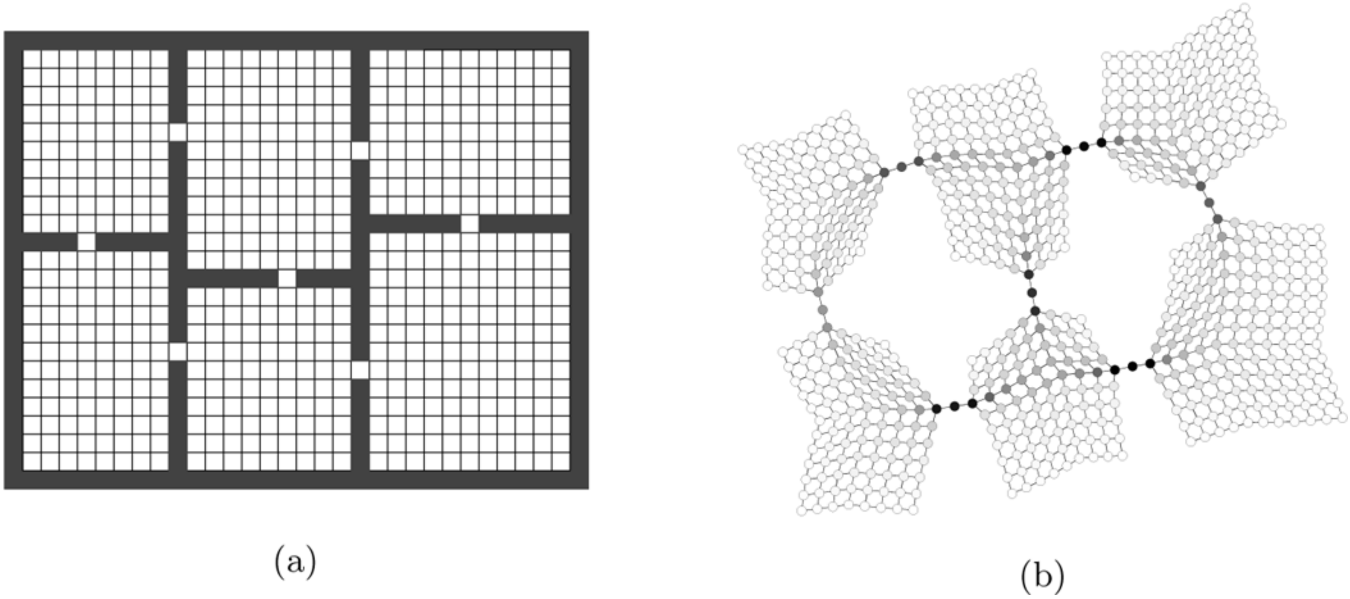
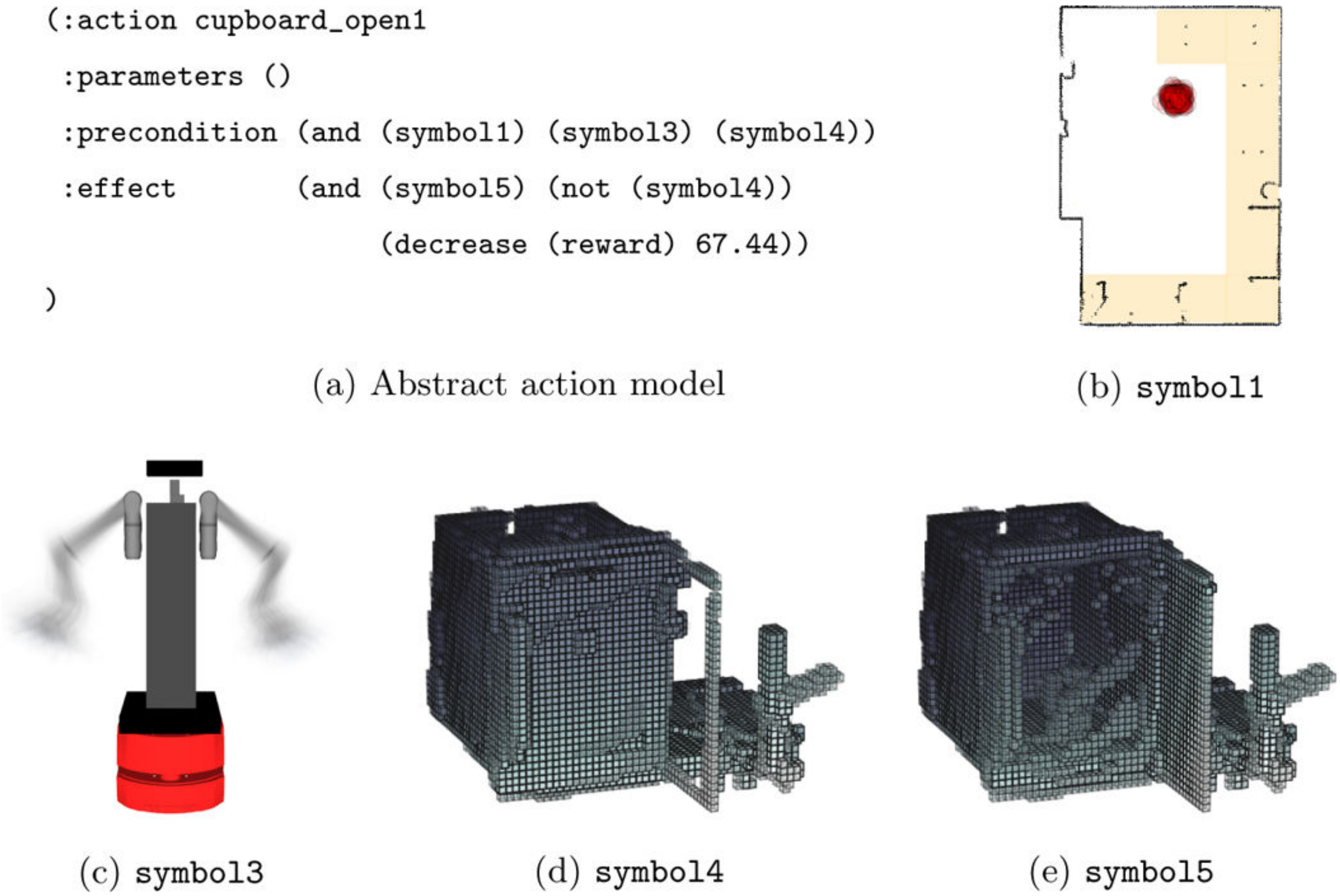


Figure 3:

Skill discovery using between-ness centrality, a measure of the likelihood that a state lies on the shortest path between any two other states. When applied to a gridworld with multiple rooms (a), the doorways between rooms are local maxima of between-ness centrality (b), indicating that they might be useful subgoals. From S_im_sek [67], used with permission.



```
(:action cupboard_open1
 :parameters ()
 :precondition (and (symbol1) (symbol3) (symbol4))
 :effect       (and (symbol5) (not (symbol4))
                    (decrease (reward) 67.44))
)
```

(a) Abstract action model

(b) symbol1

(c) symbol3

(d) symbol4

(e) symbol5

Figure 4:

An abstract learned model (a) of the skill for opening a cupboard, along with the learned groundings for the symbols from which it is constructed. Each learned symbol is visu-alized using samples drawn from the corresponding sensor grounding, which is a probability distribution over the robot's map location, joint positions, or the data reported by its depth sensor. Successfully executing the motor skill requires the robot's location in the map to be in front of the cupboard (symbol1, b) with its arms in the stowed position, which indicates that it is not carrying an object (symbol3, c). Execution switches o_ symbol4, which indicates that the cupboard is closed (d), and switches on symbol5, indicating that it is open (e). The grounded symbolic vocabulary and the abstract model built using it are learned autonomously. Reused with permission from Konidaris et al. [84].