Towards Applicable State Abstractions: a Preview in Strategy Games

Linjie Xu, Diego Perez-Liebana
School of Electronic Engineering and Computer Science
Queen Mary University of London, London, UK
{linjie.xu, diego.perez}@qmul.ac.uk

Alexander Dockhorn
Faculty of Electrical Engineering and Computer Science
Leibniz University Hannover
dockhorn@tnt.uni-hannover.de

Abstract

State Abstraction is a methodology that aims to simplify planning problems and enable planners to deal with more complex environments. It is a useful tool that helps Artificial Intelligence (AI) to solve different problems, e.g., controlling, planning, and game playing. Although state abstraction has been applied with much success in large-scale problems, most of these state abstractions are specific to their use case and cannot be generalized to others. More general applications, e.g., planning and reinforcement learning (RL) in general game-playing, have mainly been evaluated in small-scale environments. This paper gives an overview of related studies and highlights three open problems of state abstraction: 1) How to scale state abstraction to large-scale problems? 2) How to deal with the trade-off between abstraction accuracy and training time? 3) How to derive theoretical performance bounds of local state abstraction? Finally, we propose strategy games to become a prior platform to address open problems and study the application of domain-independent state abstraction.

1 Introduction

Sequential decision-making in complex environments, where state and action spaces are large, has been shown to be difficult for artificial intelligence (AI) agents. Specifically, it is challenging for AI agents to explore large state and action spaces and efficiently exploit collected samples. Abstraction is a technique that reduces the size of these spaces, for example eliminating unimportant states or aggregating states into groups to create a smaller state space. As such, it is shown to be a useful tool for AI in many decision-making domains, such as control, planning and game playing. In particular, abstraction helps AI gain superhuman performance in heads-up poker [1]. Consequently, abstraction is discussed by Konidaris [2] as a valuable technique to reach more general AI.

Formally, abstraction is a methodology where a Markov Decision Process (MDP) is simplified by constructing another MDP, whose optimal strategy is close to the optimal strategy of the original MDP. Abstractions can be categorized into two basic types: state and action abstraction. They either eliminate unimportant elements or aggregate elements together from the corresponding space. Figure 1 shows an example of state abstraction that aggregates states and an example of action abstraction that eliminates actions. The presented state abstraction clusters the cells of a game state in MicroRTS [3] into regions. Thereby, similar cells are merged and the granularity of the state representation is reduced. The action abstraction example reduces the granularity of camera movement in the game Minecraft. In contrast to moving the camera freely left or right in the range of 0' to 180', it is found more efficient to discretize the action space to move left 90' pr move right 90' [4].

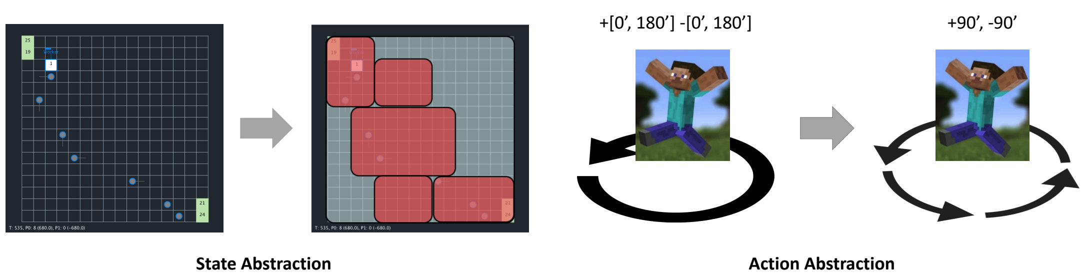

Figure 1: Examples of State Abstraction, in which multiple positions of the map are merged into hyperstates, and Action Abstraction, which shows a discretization of a real-valued rotation.

Next, we aim to give a brief overview of current research on state and action abstraction. For state abstraction, aggregating similar states is currently the most popular approach. Two approaches can reach this goal, recognizing similar states [5, 6] and reducing irrelevant features [7] in the state representation. With the help of Neural Networks (NNs), the state representation could be encoded in a lower-dimensional latent space [8]. Works found that such an NN-based state abstraction [9, 10, 11, 12, 13] yields additional benefits such as better generalization. Besides explicitly aggregating states, there are methods modeling latent structure between states [14] that enable sharing knowledge between states that have a common latent representation. Apart from filtering the action space to construct an action abstraction (as it is commonly done in portfolio-based methods [15, 16]), temporal abstractions are the most prominent type of action abstraction to be found in the literature. Temporal abstractions replace the primitive actions (elements of the original action space) with action sequences. In addition, hybrid techniques which combine state and action abstraction have shown to provide many benefits in Reinforcement Learning (RL) [17, 18] and planning [19].

While all kinds of abstraction are studied in the community, this paper will focus on state abstraction as its application in complex scenarios is understudied. In recent years, state abstraction has been studied in different domains, including control systems [20], planning [13], reinforcement learning [21, 22, 23, 24, 12, 25], multi-agent systems [26, 27] and game playing [28, 29, 30, 31]. One of the most prominent works is by Libratus [1], which won against four human professionals in a heads-up no-limit Texas Hold'em Poker competition. State abstraction is applied to the third and fourth betting rounds for aggregating hands (states) into groups. In the third round, 55 million hands are aggregated into 2.5 million groups and 2.4 million states are aggregated into 1.25 million in the following round. This state abstraction is used to approximate the expected reward of each state and has played a key role in boosting the performance of Libratus. The success of state abstraction leads to the study of its broader application. However, the state abstractions in Libratus are designed only for hold'em poker and can not be used in other environments. Similarly, state abstraction methods in strategy game playing are designed for specific game types.

Alternatively, more general state abstractions are extensively studied in the domains of planning and RL. Some latest works are starting to study state abstraction on environments that share characteristics with real-world environments: partial observability, covering long horizons, requiring the coordination of multiple agents, etc. While studying these settings are a first step towards real-world applications of state abstraction, their evaluation remains bounded to small-scale environments. Research in large-scale, complex environments with the settings mentioned above is essential for scaling up the applications of state abstraction to real-world problems. Strategy games are a clear example of a challenge for decision making that is able to provide such settings in large-scale environments. Examples of these games are the popular Civilization (Firaxis) and Total War (Creative Assembly) series.

In the rest of this paper we review the recent studies of state abstraction in planning and RL in Section 2 where we also identify some current open problems. Section 3 reviews state abstraction works in strategy games and Section 4 concludes this paper.

2 Literature Review

In state abstraction methods, two approaches are used to aggregate states into groups: hard state aggregation and soft state aggregation. In hard state aggregation, one state belongs to a single group. In contrast, soft aggregation [32, 33] defines a probability of one state being aggregated into a group. Thus, one state could belong to multiple groups to a varying degree. In state aggregation, the similarity metrics used to recognize similar states result in abstractions of difference loss bounds [34, 22, 17]. For more discussion on similarity metrics, we refer readers to a survey by Visús et al. [35]. Both the hard state aggregation and the soft state aggregation are measuring the similarity between nodes and aggregate them explicitly into groups. Other methods do not explicitly aggregate states, but connect related states implicitly [14]. These connections do not reduce the state space size but enable knowledge sharing between connected states for more efficient planning.

2.1 State Abstraction in Planning

Planning methods such as Monte Carlo Tree Search (MCTS) generate a tree to approximate the state-action value function for decision-making. Nodes in the tree represent states. Therefore, Jiang et al. [5, 19, 36] proposed to merge tree nodes with small approximate MDP homomorphism errors. In a different study, Hostetler et al. [37] proposed to start with a large node group and progressively split groups into smaller groups. Sokota et al. [38] use state similarity to prevent progressive widening from sampling states that are similar to existing states in the tree. Another approach [14] connects tree nodes by modeling their latent structure to enable information sharing, which improves the planning efficiency.

Supervised learning has also been used recently to construct state abstractions. Shah and Srivastava [13] proposed a supervised learning approach that trains a model to predict critical regions for robotic planning. An abstraction is generated by replacing the original state space with predicted critical regions. They found that predicted abstraction improves multi-source planning but does not help in a single-source planning process. Silver et al. [39] utilizes graph neural networks to predict the importance of objects, which significantly speeds up the black-box symbolic planner in different planning problems.

State abstraction is also found beneficial in hierarchical planning. Konidaris [40] proposed a method that abstracts symbolic representation of objects for hierarchical planning. James et al. [41] extend this method by learning reusable object-centric abstractions that can transfer to new tasks that share similar objects.

2.2 State Abstraction in Reinforcement Learning

The study of state abstraction in RL has previously been well-discussed by Konidaris [2], and by Shanahan and Mitchell [42]. Here, we discuss recent works to give a brief overview of how abstraction benefits RL in different manners.

The study of state aggregation in RL [43, 32, 44, 45, 33, 46, 47, 48] brought both theoretical progress and progress in several applications. State aggregation has been utilized to gain faster convergence in different RL methods [43, 48]. Russo [47] showed that the regret for policy gradient with state aggregation is bounded by the maximum same-group state difference. On the application side, Duan et al. [33] proposed an unsupervised soft state aggregation method that beats handcrafted state partition in finding optimal taxi driving policies within simulated NYC traffic.

With neural networks, state aggregation could be done by constraining the state representation in the latent space. The representation learned in this way is found to generalize well among different tasks. Zhang et al. [11] combined the bisimulation similarity metric with an invariant causal prediction to learn a state abstraction that generalizes between environments that share the same latent state space and dynamics. In a multitask setting where all the tasks share the same environment parameters, Zhang et al. [12] proposed to learn a task embedding that captures the behavioral similarity across tasks in the sense of a universal dynamic model. The learned dynamic model has been shown to generalize well in multi-task and meta RL settings. Besides generalization, state abstraction contributes to the exploration of RL both theoretically and practically. Taïga et al. [9] have used state abstraction to improve the theoretical understanding of pseudo-count-based exploration bonuses. Misra et al. [10] proposed a supervised learning method to learn kinematic state abstraction according to the dynamic similarity between states. Evaluated in challenging exploration tasks, their algorithm is empirically exponentially more sample efficient than Proximal Policy Improvement (PPO).

State abstraction also enhances hierarchical RL in different ways. Abel et al. [18] introduced a variant of the option framework that depends on a given state abstraction, resulting in a state-abstraction where the near-optimal behavior is preserved. In a work by Shah et al. [25], a representation learning approach is used to learn a skill-centric state abstraction for hierarchical RL for long-horizon planning problems. The learned representation has been shown to generalize well to novel tasks.

Besides the problem of how to generate the state abstraction, abstraction selection is another significant problem. Jiang et al. [44] analyzed the abstraction comparison under finite sample limit and proposed an abstraction selection algorithm with a hypothesis test. Tamassia et al. [45] proposed an algorithm that chooses a suitable abstraction from the given abstraction set at each time step. The chosen abstraction is the most granular one that has the highest confidence in the action with the highest approximated Q-value.

2.3 Open Problems

While many of the reviewed works have shown the benefits of applying abstractions in the domains of planning and reinforcement learning problems, we could not find promising examples of abstraction in complex real-world applications. In the following, we have identified open problems of state abstraction methods that might restrain their applicability. Studying and solving these problems would make abstraction an even more powerful tool for the simplification of many decision-making problems.

Abstraction of large-scale environments: The works reviewed above show that abstraction enables more efficient planning and improves RL exploration and generalization. However, most of these works are evaluating their methods in small-scale environments (Table 1). Scaling-up abstraction methods is non-trivial since current methods require a large number of observations to create an adequate abstraction. Nevertheless, such methods would be required to ensure the feasibility of abstractions in real-world applications.

Trade-off between abstraction accuracy and training time: A study by Arumugam and Van Roy [49] raised the question of when it is beneficial to learn a state abstraction alongside learning optimal behavior. They proposed a method where the agent maintains an explicit belief over its state abstraction and have shown that this method can benefit the overall performance. Similarly, Xu et al. [50] have studied the application of abstraction in planning problems with limited decision-time. Determining a more accurate abstraction takes off decision-time, but may help in solving the planning problem faster. Studying the relation between an abstractions accuracy and the planner's performance will play an important role in improving the performance of abstraction-based agents.

Theoretical performance bounds of local state abstraction: Since not all MDPs have a global abstraction under some similarity metrics [5], some attention has been given to local abstractions that only cover a part of the state space. Recent works have shown that constructing a local abstraction has been computationally more feasible than the use of global abstractions [5, 1]. Furthermore, the construction of global abstractions could be biased by the underlying samples of the state space [51]. Nevertheless, theoretical performance bounds have only been derived for global abstraction methods so far, similar bounds are missing for local abstractions but would be required for their robust and efficient application.

3 State Abstraction for Strategy Game Playing

To date, complex strategy games remain a challenge due to their large state and action spaces, partial observability, opponent modeling, real-time unit coordination, dynamic environments and long planning horizons. State abstraction has been utilized as a powerful technique to solve some of the challenges mentioned above. Recognizing similar states using the structural analogy has been shown to improve the performance of solving a sub-problem in the game Civilization [52]. State abstraction that separates maps into regions are a popular method in strategy games such as *Capture the Flag* [53] and *StarCraft* [28, 29, 30]. Dockhorn et al. [31] proposed a method to select entity features for state abstraction in strategy games, and [8] proposed a supervised learning method to abstract the image state into grids where each position predicts the object type, enhancing their RL agent in the game *Battle City*.

While these works designed the formation of state abstractions to fit specific game types, their methods cannot generalize to different games. On one hand, the current studies of state abstraction in planning and RL evaluate their method in small-scale problems. On the other, state abstraction that provide good results on large-scale strategy games are designed for specific games. Only recently, Linjie et al. [50] show preliminary results of domain-independent state abstraction in multi-unit games. This comparison positions strategy games as a promising large-scale challenge to study domain-independent state abstraction. Additionally, most of the real-world decision-making problems take place in partial observable environments, cover long horizons and require the coordination of multiple entities. These challenges are common in strategy games, which makes it a reasonable step to study state abstraction in strategy games towards the real-world application.

3.1 Applying State Abstractions

Most of the work referenced in this paper tackles the problem of large state spaces with the aim of reducing the dimensionality of the problem, in order to facilitate decision-making. In general, these approaches attempt to abstract the state space following a determined technique (be this by hand, using domain knowledge, or by using different machine learning algorithms), employing the reduced state representation to tackle the problem as a whole.

While this could be a valid approach for certain domains, especially those related to a single task or contextualized in non-highly dynamic environments, we believe that there is much to explore and gain by looking at different ways to use multiple state abstractions. In the domain of strategy games (and, by extension, many real-world problems) we can observe decision-making applied at different levels. For instance, decisions can be made to attend to short-term urgencies (lack of particular resources, enemy units at the player's base, etc.) or long-term planning (the composition of the army to be built, where to settle the next base, which branch in the research tree should the player prioritize, etc.). Similarly, decision-making for a civilian unit (e.g. a worker, or a settler) is fundamentally different from that of a combat unit (an archer, or a horseman). It is naive to think that the *same* state abstraction, understood as a simplification and or clustering of the state space into a reduced representation, can serve equally well all these decision-making problems. The same can be said about decisions that are made at the opening phase of a game, which must consider different aspects of the world (terrain, resources, map layout), in contrast to those required in later stages, where players strive to achieve victory by one of the game's win conditions.

In light of this, we consider that different types of abstractions can be made available for different purposes. In strategy games, one possible abstraction could consist of removing elements from the state that are of no interest to particular decision-making aspects, such as ally or enemy units, workers, buildings, or even entire players. This can be done for all states of the game, or it could depend on the particular state the game is, potentially learning a correspondence between states and the *kind* of state abstraction they require. For example, strategy games go through phases where production or gathering of resources must be prioritized, or when a defensive play is preferred. An attractive line of research could study how to select the appropriate state abstraction algorithm given a particular state or set of game states. Similarly, different state abstractions can be devised considering for which units or players decisions need to be made, or which game elements can be removed from the game state representation based on other factors (such as, for instance, distance in the board to the acting player). In a way, this procedure can be seen as the player *injecting* partial observability (PO) to the state space, but a kind of PO that, rather than being imposed by the game (fog of war), decides to omit the contents of the game states that are not relevant to the decision-making problem.

Finally, it's worth considering that these abstractions are not mutually exclusive. Given the original un-abstracted state space, several abstraction mechanisms can simultaneously produce different abstracted spaces. These spaces can be used by distinct units owned by the player, or even at different time horizons in a decision plan: for example, using finer abstractions for immediate next steps (where decisions may require precision) and coarser ones for farther look-ahead (when decisions can lead to broader consequences). Some initial work has already been done in this direction (by [45]), although in a simpler environment: the game Pac-Man. Overall, we believe that there is much to learn not only on how abstractions are produced but also how are they used, especially in complex and dynamic environments where different decisions are influenced by many distinct factors.

4 Conclusion

In this paper, we reviewed the current literature on state abstraction in different domains, as well as identified open problems and possible lines of future research on this area. Concretely, we raise 3 open questions that are essential for the application of state abstraction: abstractions in large-scale environments, accuracy vs. training time trade-off and performance bounds for local state abstractions. Nowadays, most works on state abstractions are applied to small-scale problems, thus we position strategy game playing as an ideal large-scale platform to study how to create and use state abstractions. Strategy games share many challenges with real-world problems, such as partial observability, short vs long-term planning, coordination of multiple units/elements, and more. The complexities of this type of games allow for the investigation of how different state abstractions can be used, either in parallel or alternatively, to solve difficult decision-making problems that span across multiple dimensions. Our expectation is that addressing these concerns in strategy games will pave the way for the application of state abstraction and decision making in real-world scenarios.

References

[1] Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

[2] George Konidaris. On the necessity of abstraction. *Current opinion in behavioral sciences*, 29:1–7, 2019.

[3] Santiago Ontanón. The combinatorial multi-armed bandit problem and its application to real-time strategy games. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

[4] Linjie Xu and Yihong Chen. A hierarchical approach for marlö challenge. In *2019 IEEE Conference on Games (CoG)*, pages 1–4. IEEE, 2019.

[5] Nan Jiang, Satinder Singh, and Richard Lewis. Improving uct planning via approximate homomorphisms. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1289–1296, 2014.

[6] Jesse Hostetler, Alan Fern, and Tom Dietterich. State aggregation in monte carlo tree search. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[7] Nicholas K Jong and Peter Stone. State abstraction discovery from irrelevant state variables. In *IJCAI*, volume 8, pages 752–757. Citeseer, 2005.

[8] Yu Chen, Huizhuo Yuan, and Yujun Li. Object-oriented state abstraction in reinforcement learning for video games. In *2019 IEEE Conference on Games (CoG)*, pages 1–4, 2019.

[9] Adrien Ali Taïga, Aaron Courville, and Marc G Bellemare. Approximate exploration through state abstraction. *arXiv preprint arXiv:1808.09819*, 2018.

[10] Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International conference on machine learning*, pages 6961–6971. PMLR, 2020.

[11] Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *International Conference on Machine Learning*, pages 11214–11224. PMLR, 2020.

[12] Amy Zhang, Shagun Sodhani, Khimya Khetarpal, and Joelle Pineau. Learning robust state abstractions for hidden-parameter block mdps. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[13] Naman Shah and Siddharth Srivastava. Using deep learning to bootstrap abstractions for hierarchical robot planning. *In Proc. of the 21st International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2022)*, 2022.

[14] Julia Grosse, Cheng Zhang, and Philipp Hennig. Probabilistic dag search. In *Uncertainty in Artificial Intelligence*, pages 1424–1433. PMLR, 2021.

[15] Diego Perez-Liebana, Cristina Guerrero-Romero, Alexander Dockhorn, Linjie Xu, Jorge Hurtado, and Dominik Jeurissen. Generating diverse and competitive play-styles for strategy games. In *2021 IEEE Conference on Games (CoG)*, pages 1–8, 2021.

[16] Alexander Dockhorn, Jorge Hurtado-Grueso, Dominik Jeurissen, Linjie Xu, and Diego Perez-Liebana. Portfolio search and optimization for general strategy game-playing. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 2085–2092, 2021.

[17] Ariel Rosenfeld, Matthew E Taylor, and Sarit Kraus. Speeding up tabular reinforcement learning using state-action similarities. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1722–1724, 2017.

[18] David Abel, Nate Umbanhowar, Khimya Khetarpal, Dilip Arumugam, Doina Precup, and Michael Littman. Value preserving state-action abstractions. In *International Conference on Artificial Intelligence and Statistics*, pages 1639–1650. PMLR, 2020.

[19] Ankit Anand, Aditya Grover, Parag Singla, et al. Asap-uct: Abstraction of state-action pairs in uct. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[20] Rupak Majumdar, Necmiye Ozay, and Anne-Kathrin Schmuck. On abstraction-based controller design with output feedback. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, pages 1–11, 2020.

[21] Aijun Bai, Siddharth Srivastava, and Stuart Russell. Markovian state and action abstractions for mdps via hierarchical MCTS. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3029–3039. IJCAI/AAAI Press, 2016.

[22] David Abel, David Hershkowitz, and Michael Littman. Near optimal behavior via approximate state abstraction. In *International Conference on Machine Learning*, pages 2915–2923. PMLR, 2016.

[23] Ondrej Biza and Robert Platt Jr. Online abstraction with MDP homomorphisms for deep learning. In Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor, editors, *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pages 1125–1133. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[24] Elise van der Pol, Thomas Kipf, Frans A. Oliehoek, and Max Welling. Plannable approximations to MDP homomorphisms: Equivariance under actions. In Amal El Fallah Seghrouchni, Gita Sukthankar, Bo An, and Neil Yorke-Smith, editors, *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020*, pages 1431–1439. International Foundation for Autonomous Agents and Multiagent Systems, 2020.

[25] Dhruv Shah, Alexander T Toshev, Sergey Levine, and brian ichter. Value function spaces: Skill-centric state abstractions for long-horizon reasoning. In *International Conference on Learning Representations*, 2022.

[26] Frans Oliehoek, Stefan Witwicki, and Leslie Kaelbling. A sufficient statistic for influence in structured multiagent environments. *Journal of Artificial Intelligence Research*, 70:789–870, 2021.

[27] Elise van der Pol, Herke van Hoof, Frans A Oliehoek, and Max Welling. Multi-agent MDP homomorphic networks. In *International Conference on Learning Representations*, 2022.

[28] Gabriel Synnaeve and Pierre Bessiere. A bayesian tactician. In *Computer Games Workshop at ECAI 2012*, pages pp–114, 2012.

[29] Alberto Uriarte and Santiago Ontanón. Game-tree search over high-level game states in rts games. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.

[30] Alberto Uriarte and Santiago Ontanón. High-level representations for game-tree search in rts games. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.

[31] Alexander Dockhorn, Jorge Hurtado-Grueso, Dominik Jeurissen, Linjie Xu, and Diego Perez-Liebana. Game state and action abstracting monte carlo tree search for general strategy game-playing. In *2021 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2021.

[32] Satinder Singh, Tommi Jaakkola, and Michael Jordan. Reinforcement learning with soft state aggregation. *Advances in neural information processing systems*, 7, 1994.

[33] Yaqi Duan, Tracy Ke, and Mengdi Wang. State aggregation learning from markov transition data. *Advances in Neural Information Processing Systems*, 32, 2019.

[34] Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. *ISAIM*, 4(5):9, 2006.

[35] Álvaro Visús, Javier García, and Fernando Fernández. A taxonomy of similarity metrics for markov decision processes. *arXiv preprint arXiv:2103.04706*, 2021.

[36] Ankit Anand, Ritesh Noothigattu, Parag Singla, et al. Oga-uct: On-the-go abstractions in uct. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 26, 2016.

[37] Jesse Hostetler, Alan Fern, and Thomas G Dietterich. Progressive abstraction refinement for sparse sampling. In *UAI*, pages 365–374, 2015.

[38] Samuel Sokota, Caleb Y Ho, Zaheen Ahmad, and J Zico Kolter. Monte carlo tree search with iteratively refining state abstractions. *Advances in Neural Information Processing Systems*, 34:18698–18709, 2021.

[39] Tom Silver, Rohan Chitnis, Aidan Curtis, Joshua B. Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Planning with learned object importance in large problem instances using graph neural networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 11962–11971. AAAI Press, 2021.

[40] George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018.

[41] Steven James, Benjamin Rosman, and George Konidaris. Autonomous learning of object-centric abstractions for high-level planning. In *International Conference on Learning Representations*, 2022.

[42] Murray Shanahan and Melanie Mitchell. Abstraction for deep reinforcement learning. *arXiv preprint arXiv:2202.05839*, 2022.

[43] D.P. Bertsekas and D.A. Castanon. Adaptive aggregation methods for infinite horizon dynamic programming. *IEEE Transactions on Automatic Control*, 34(6):589–598, 1989.

[44] Nan Jiang, Alex Kulesza, and Satinder Singh. Abstraction selection in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 179–188. PMLR, 2015.

[45] Marco Tamassia, Fabio Zambetta, William L Raffe, Florian'Floyd' Mueller, and Xiaodong Li. Dynamic choice of state abstraction in q-learning. In *ECAI 2016*, pages 46–54. IOS Press, 2016.

[46] Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *Advances in Neural Information Processing Systems*, 33:13399–13412, 2020.

[47] Daniel Russo. Approximation benefits of policy gradient methods with aggregated states. *arXiv preprint arXiv:2007.11684*, 2020.

[48] Guanting Chen, Johann Demetrio Gaebler, Matt Peng, Chunlin Sun, and Yinyu Ye. An adaptive state aggregation algorithm for markov decision processes. *arXiv preprint arXiv:2107.11053*, 2021.

[49] Dilip Arumugam and Benjamin Van Roy. Randomized value functions via posterior state-abstraction sampling. *arXiv preprint arXiv:2010.02383*, 2020.

[50] Linjie Xu, Alexander Dockhorn, Jorge Hurtado-Grueso, Dominik Jeurissen, and Diego Perez-Liebana. Elastic monte carlo tree searchwith state abstraction for strategy game playing. In *To appear in 2022 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2022.

[51] Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Improved representations via sampling-based state similarity for markov decision processes. *Advances in Neural Information Processing Systems*, 34, 2021.

[52] Thomas R Hinrichs and Kenneth D Forbus. Analogical learning in a turn-based strategy game. In *IJCAI*, pages 853–858, 2007.

[53] Michael Chung, Michael Buro, and Jonathan Schaeffer. Monte carlo planning in rts games. In *CIG*. Citeseer, 2005.

[54] Aijun Bai, Siddharth Srivastava, and Stuart J Russell. Markovian state and action abstractions for mdps via hierarchical mcts. In *IJCAI*, pages 3029–3039, 2016.

[55] Diego Perez-Liebana, Alexander Dockhorn, Jorge Hurtado-Grueso, and Dominik Jeurissen. The Design Of Stratega: A General Strategy Games Framework. *arXiv preprint arXiv:2009.05643*, 2020.

[56] Alexander Dockhorn, Jorge Hurtado Grueso, Dominik Jeurissen, and Diego Perez-Liebana. Stratega: A General Strategy Games Framework. In *Artificial Intelligence for Strategy Games Decision, AIIDE 2020 Workshop*, 2020.

A Appendix

Method Name	Condition	Data collection	Tree operation	Abstraction Type	Domains
2014 Approximate Homomorphism [5]	$\mathcal{T}$ and Q	Batch	Node Aggregation	State Abstraction	Planning/Othelo
2014 State Aggregation [6]	/	/	Node Aggregation	Given Abstraction	Blackjack
2015 Progressive Abstraction [37]	The same $a*$ or $V*$	Incremental	Progressive Refinement	State Abstraction	Planning/Racetrack, Spanish Blackjack, Academic Advicing
2015 State-Action Abstraction [19]	$\mathcal{T}$ and Q	Batch	Aggregate state-action node	State-Action Pair	Planning/SailingWind, GameOfLife, Navigation (RDDL)
2016 Hierarchical MCTS [54]	/	/	/	Given Abstraction	POMDP Planning/Rooms
2016 OGA-UCT [36]	$\mathcal{T}$ and Q	Incremental	split-combine	State-Action Pair	Planning/Acadvicing, Navigation, SailingWind, RaceTrack, GameOfLife, Sysadmin

Table 1: State abstraction with MCTS in planning.

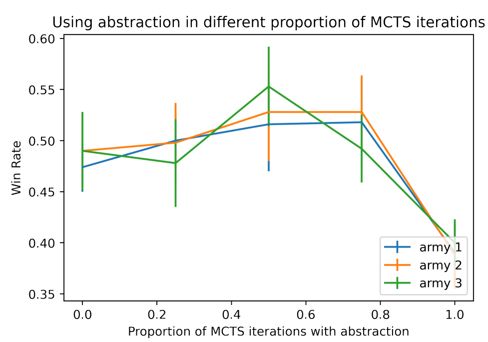

Figure 2: Experiments of MCTS with state abstraction playing a multi-unit combat game of the Stratega framework [55, 56]. In each step, the available MCTS iterations are limited. In this Figure, we compare an agent's performance using different proportions of MCTS iterations running with state abstraction. We tested 3 army compositions including of 4 to 10 units. The results shows that using state abstraction with different numbers of MCTS iterations influence the performance. Using state abstractions all the time shows worse performance than using state abstraction for a lesser number of iterations.