Novelty and MCTS

Hendrik Baier
Michael Kaisers
hendrik.baier@cwi.nl
michael.kaisers@cwi.nl
Centrum Wiskunde & Informatica
Amsterdam, Netherlands

ABSTRACT

Novelty search has become a popular technique in different fields such as evolutionary computing, classical AI planning, and deep reinforcement learning. Searching for novelty instead of, or in addition to, directly maximizing the search objective, aims at avoiding dead ends and local minima, and overall improving exploration. We propose and test the integration of novelty into Monte Carlo Tree Search (MCTS), a state-of-the-art framework for online RL planning, by linearly combining value estimates with novelty scores during the selection phase of MCTS. Three different novelty measures are adapted from the literature, integrated into MCTS, and tested in four different board games. The initial results are promising and point towards potential for novelty as "online generalization for uncertainty" in more challenging search settings.

CCS CONCEPTS

• **Computing methodologies → Game tree search**.

KEYWORDS

novelty, novelty search, Monte Carlo Tree Search, game tree search

ACM Reference Format:
Hendrik Baier and Michael Kaisers. 2021. Novelty and MCTS. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3449726.3463217

1 INTRODUCTION

Sequential decision-making problems arise in a variety of domains, and significant progress in this area has been achieved by studying search in games. *Monte Carlo Tree Search* (MCTS) in particular handles large search spaces well due to selective sampling of promising actions. It has been shown to converge to the optimal policy in the limit, if exploration and exploitation are traded off properly [11], and it provides approximations at any time. MCTS and its many variants have been successfully applied to countless domains in recent years, for example to General Game Playing [6], to General Video Game Playing [15], and to two-player board games in recent breakthroughs that combined search with deep neural networks [23, 24].

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3463217

The use of *novelty* in search and optimization has been a significant line of research in several different subfields of AI in the last decade, such as in evolutionary computing, in classical AI planning, and in deep reinforcement learning. Searching for novel states or novel behaviors as an *intrinsic* motivation of an AI agent has shown surprising success compared to exclusively trying to optimize the *extrinsically* given objective function, especially in domains where the gradient of improvement with regard to that objective function is sparse or misleading [13].

In this paper, we propose the integration of novelty search techniques into MCTS, and test them in online RL planning. Specifically, we bias search by linearly combining one of three novelty scores with the MCTS node value estimates before adding the UCB exploration bonus. Unlike in many previous applications of novelty to search, we assume that a somewhat effective heuristic state evaluation function is already available to guide MCTS; and unlike typical applications of novelty to deep RL, MCTS is able to use count-based uncertainty estimates in its tree. Nevertheless, our preliminary results indicate that novelty, as an auxiliary objective that generalizes uncertainty across the state space, can lead to better gameplay through further improvements in MCTS exploration.

This paper is structured as follows: Section 2 relates this paper to the literature. Section 3 briefly sketches the three different novelty measures used, and how we integrated them into MCTS. Section 4 presents experimental results in four board game domains, and Section 5 discusses conclusions and future work.

2 RELATED WORK

In evolutionary computation, it has been found that fitness functions, meant to measure progress towards the actual objective of the search, can often be deceptive, and thus lead into dead ends. A sometimes surprisingly effective approach to circumventing this problem is to ignore the objective entirely, and to search only for (behavioral) novelty instead [12, 13]. In this work, we do not aim to completely ignore our objective, i.e. heuristic estimates guiding successful play, but integrate an additional novelty score in order to improve exploration.

In classical AI planning, a simple blind-search procedure called *Iterated Width* (IW) was developed [16]. IW is an iterative breadth-first search that prunes states of insufficient novelty, relaxing the requirement for novelty in every iteration. It was found to be effective in many benchmark planning problems due to their simple enough goal structure, and lead to state-of-the-art performance when integrated with other known planning techniques [16]. As IW does not require knowledge of transitions and goals, it has also been successfully applied to simulation-based planning, both

in Atari games [17] and in General Game Playing (GGP) [8]. Its pruning criterion, i.e. its novelty measure, was extended to take the reward-so-far into account [22], which improved results in Atari games; and it was modified to utilize heuristic value estimates of states instead when those are available [10]. One of the novelty measures we examine in this work is similar to the ones previously proposed for classical planning [10, 22], where it has been called h_{BN}. However, we use it to modify value estimates in the MCTS tree, instead of for improving the node ordering in a traditional best first search queue.

In (deep) reinforcement learning, novelty has been studied as a form of intrinsic motivation for the RL agent [1], meant to aid in representation learning as well as in improving exploration. Unlike IW-like algorithms, for which a binary classification of states into novel or non-novel is often sufficient, these RL approaches aim for a finer-grained measure of the agent's uncertainty about its environment, in order to guide the agent towards less familiar regions of the state space and thus encourage learning. In analogy to the tabular case, where we can simply count how often each state has been visited and intrinsically reward the agent for visiting states with lower visit counts, RL novelty techniques are often based on *pseudocounts* – a generalization of state counts which allows to generalize uncertainty across large state spaces. Pseudocounts can be defined based on a *density model* that assigns probabilities to observing a given state [4]. Such pseudocounts have been proposed based on state hashing [25], on neural density models [5, 20], or on successor representations [18]. Two of the novelty measures we study in this work are based on the state-visitation density model [4], as well as on a similar model constructed on a feature representation of the state instead of on the raw state [19]. However, we use it for sample-based planning instead of learning.

The closest related work to ours is an unpublished student project with the goal of combining IW-like novelty search and MCTS, applied to GGP [14]. In contrast to our work in which we bias search, novelty was used for hard pruning of the tree, which invalidates the convergence guarantees of MCTS; it was not applied to MCTS using heuristic evaluation functions for state evaluation; and its experimental evaluation was insufficient to draw statistically significant conclusions.

3 NOVELTY AND MCTS

In this section, we describe the three novelty measures we use in this exploratory study, and how we integrate them into the MCTS framework. All novelty measures assume a state space S with internal structure, with *factored* states s that consist of a vector of distinct components or variables and their assigned values. In board games for example, "square d4" could be such a variable. A variable plus assigned value, e.g. "white pawn on d4", is also called a *fact*.

3.1 Defining novelty

The techniques briefly outlined in this subsection aim at measuring the novelty of a newly discovered state, and are slightly modified techniques from the literature.

3.1.1 Evaluation novelty. This technique is adapted from *reward-based novelty* [10, 22]. Given a heuristic state evaluation function $V : S \to \mathbb{R}$ defined on the set of states S, and S_t as the set of states observed until time step t, the *novelty score* of a fact (variable & value) f at time step t is defined as

$$N_t(f) = \begin{cases} \max\limits_{s \in S_t, f \in s} V(s) & \text{if } f \in s \text{ for some } s \in S_t \\ -\infty & \text{otherwise,} \end{cases} \tag{1}$$

i.e. as the highest evaluation of any state with that particular fact seen so far. Given a state s, its *evaluation novelty* $N_E(s)$ is then defined as

$$N_E(s) = \begin{cases} \alpha & \text{if } V(s) > N_t(f) \text{ for some } f \in s \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where α is a tunable parameter. This means that this novelty measure is binary – it distinguishes only between novel and non-novel – and that a state is considered novel iff for at least one of the facts it consists of, the evaluation of the state is higher than that of any state observed before with that particular fact[1].

3.1.2 Raw-state pseudocount novelty. This technique is adapted from the *ϕ-Exploration-Bonus algorithm* [19]. It does not make use of a heuristic evaluation function, but of a probability distribution over states. Given a feature mapping $\phi : S \to T$ from the state space into an M-dimensional feature space T, we define a density model $\rho_t(\phi)$ at time t – a probability distribution over the feature space – as the product of independent factor distributions $\rho_t^i(\phi_i)$ over the M individual features:

$$\rho_t(\phi) = \prod_{i=1}^{M} \rho_t^i(\phi_i). \tag{3}$$

For the factor models, we use the empirical estimator

$$\rho_t^i(\phi_i) = \frac{C_t(\phi_i)}{t}, \tag{4}$$

where $C_t(\phi_i)$ is the number of times feature ϕ_i has been observed until time step t. This allows us to define the *ϕ-pseudocount* for a given state s at time t as

$$\hat{C}_t^{\phi}(s) = \frac{\rho_t(\phi(s))(1 - \rho_{t+1}(\phi(s)))}{\rho_{t+1}(\phi(s)) - \rho_t(\phi(s))}, \tag{5}$$

where ρ_t is the density model before $\phi(s)$ has been observed, and ρ_{t+1} is the model after the observation [4]. The *pseudocount novelty* $N_C(s)$ is then defined as

$$N_C(s) = \frac{\alpha}{\sqrt{\hat{C}_t^{\phi}(s)}}, \tag{6}$$

where α is again a tunable parameter [19].

Different feature mappings ϕ are imaginable; for the *raw-state* pseudocount novelty tested in this work, we use the atomic facts our states consist of (resulting in N_C^{raw}). An atomic fact in chess could for example be "black knight on e5" or "white pawn on a3".

[1] Note that a higher value $N_t(f)$ of a fact f does not mean that f is considered more novel. The values $N_t(f)$ are simply called "novelty scores" because they allow us to compute the novelty $N_E(s)$ of a given state, where higher values indeed mean more novel.

3.1.3 Feature-based pseudocount novelty. The *feature-based* pseudocount novelty variant is computed analogously to the raw-state variant. The difference is that instead of using atomic facts, we compute the density model with the features used by our heuristic evaluation functions (resulting in N_C^{eval}), as originally intended for this novelty measure [19]. An evaluation function in chess for example could use more abstract features such as "Black still has both knights", or "White controls the f-file with a rook", which are potentially more meaningful in terms of future rewards.

Evaluation novelty could in principle also be defined on atomic state variables as well as on arbitrary features extracted from states – here, we only include experiments with the raw state facts (N_E^{raw}).

3.2 Using novelty within MCTS

When integrating novelty measures into MCTS, our goal is to improve exploration under short time controls, without losing convergence guarantees in the limit. We achieve this through a relatively small change to vanilla MCTS: by combining regular value estimates with novelty scores in the selection phase of MCTS, using a technique originally proposed for *Rapid Action Value Estimates* (RAVE) [9].

Each MCTS simulation produces an evaluation $V(s)$, returned by the heuristic evaluation function, and additionally a novelty score $N(s)$, produced by the chosen novelty measure, of the state s that was just added to the tree. Just like the value estimate in vanilla MCTS, the additional novelty score is then also backpropagated to all states that were visited in the current simulation; and in each tree node representing one of these states, an average $\bar{N}_a$ is maintained of all novelty scores seen in the subtree below the traversed state-action pair, analogously to how value estimates $\bar{V}_a$ in MCTS tree nodes are formed by averaging over all *evaluations* seen in the subtrees below. For each visited tree node from which an action a was chosen, with n_a the number of times that action has been chosen so far, the backpropagation updates are:

$$n_a = n_a + 1, \quad \bar{V}_a = \bar{V}_a + \frac{V(s) - \bar{V}_a}{n_a}, \quad \bar{N}_a = \bar{N}_a + \frac{N(s) - \bar{N}_a}{n_a}$$

During the selection phase, the selection policy of MCTS can now be changed from the classic UCB1 policy that is based on value alone:

$$\bar{V}_a + k\sqrt{\frac{\ln n}{n_a}}, \tag{7}$$

where n is the number of times the given node has been traversed chosing any action, and k is a factor trading off exploration and exploitation, to a new policy that linearly combines value and novelty averages:

$$b\bar{N}_a + (1-b)\bar{V}_a + k\sqrt{\frac{\ln n}{n_a}}, \tag{8}$$

where b is a weighting coefficient as given by

$$\sqrt{\frac{\beta}{3n_a + \beta}}, \tag{9}$$

with β as a tunable parameter regulating how quickly b decays over time, i.e. how quickly novelty is phased out as the node becomes increasingly certain of its value estimates over time. After the search has been completed, MCTS choses the root action with the highest value estimate (ignoring novelty) for execution; chosing the root action with the highest sample count is another popular option, but did not lead to significantly different results in exploratory experiments.

Note that in the limit, the MCTS search will visit every possible state an infinite number of times, and all novelty averages will approach zero. In short search times however, the parameter β helps us to control how quickly MCTS should forget about novelty and focus on value.

4 EXPERIMENTAL RESULTS

We tested novelty-enhanced MCTS against vanilla MCTS in four different board game domains: *Connect 4*, *Othello*, *Breakthrough*, and *Knightthrough*. These are fully observable, deterministic, alternating-turn, two-player games, although our approach does not require these constraints. All MCTS players use traditional linear heuristic evaluation functions instead of random rollouts for state evaluation; these evaluation functions also provide us with feature representations for N_C^{eval} (see Section 3.1.3). All experiments allowed for either 1000 or 5000 MCTS simulations per move, in order to test whether novelty can improve the sample efficiency of the search. Vanilla MCTS has one parameter: the exploration factor k of UCB1. All novelty-based approaches have two additional parameters: a parameter α that controls the magnitude of the novelty term, and a parameter β that controls how quickly the novelty term's influence on the search decreases. The parameters of all agents were first tuned (requiring about 24,000 games per agent), followed by a test of the best found parameter settings with at least 2000 additional games. The results of these tests are presented here. In all tables, boldface indicates statistical significance at the 95% confidence level of an improvement over the vanilla MCTS baseline (winrate of novelty-enhanced MCTS higher than 50%).

Our experiments are divided into three groups, depending on the novelty measure used. In the first group, we enhanced MCTS with pseudocount novelty based on the raw state itself (N_C^{raw}). The results are shown in Table 1. In the second set of experiments, we kept the pseudocount novelty measure, but now computed it based on the same features that are used by the heuristic evaluation function in each domain (N_C^{eval}). Table 2 shows the results. In the third group of experiments finally, we used the evaluation-based novelty measure of Subsection 3.1.1, again computed on raw states (N_E^{raw}). The results are given in Table 3.

We can summarize the results with three observations. First, novelty-enhanced exploration seems to be promising in principle: It led to statistically significant improvements over vanilla UCB1 selection in 14 out of 24 conditions. Second, it seems to be more promising when higher search budgets are available: Only 4 of 12 conditions at 1000 simulations per move are improved, but 10 of 12 conditions at 5000 simulations per move. And interestingly, differences between domains and search budgets were relatively robust to the choice of novelty measure. Whether novelty helps or not seems to depend more on the domain and/or the evaluation function used – possibly on how frequently it is misleading – than on the precise technique chosen for computing novelty.

As expected, the optimal values for α and β determined by our experiments tended to be higher for conditions where novelty has

Table 1: Winrate of MCTS using raw-state pseudocount novelty ($N_C^{\mathbf{raw}}$) vs. baseline MCTS

Game	simulations/move	
	1000	5000
Connect 4	**59.5%**	**64.1%**
Othello	48.8%	51.7%
Breakthrough	52.7%	**59.9%**
Knightthrough	49.3%	**55.9%**

Table 2: Winrate of MCTS using feature-based pseudocount novelty ($N_C^{\mathbf{eval}}$) vs. baseline MCTS

Game	simulations/move	
	1000	5000
Connect 4	51.1%	**63.1%**
Othello	50.5%	**52.8%**
Breakthrough	49.6%	**57.7%**
Knightthrough	**53.2%**	**53.4%**

Table 3: Winrate of MCTS using raw-state evaluation novelty ($N_E^{\mathbf{raw}}$) vs. baseline MCTS

Game	simulations/move	
	1000	5000
Connect 4	**58.0%**	**65.3%**
Othello	51.2%	49.2%
Breakthrough	50.6%	**57.6%**
Knightthrough	**52.9%**	**54.9%**

a stronger positive effect, and zero or close to zero for conditions where novelty does not help, minimizing its effect. For example, N_C^{raw} in Connect 4 at 1000 simulations per move worked best with $\alpha = 2$ and $\beta = 0.01$, while for Othello $\alpha = 0.003$ and $\beta = 0.0006$ were returned by our optimization. The algorithms were not very sensitive to these parameters, and sometimes using a higher α or using a higher β even seemed interchangeable to a degree, as they both increase the influence of novelty. While in Connect 4 at 1000 simulations, as just mentioned, α was much higher than β for example, the opposite was true for Breakthrough at 5000 simulations with $\alpha = 0.02$ and $\beta = 1$ – with similar performance. Note however that comparisons of the absolute values of these parameters in different domains are difficult, as they depend on the variance of the heuristic evaluation function used: When all heuristic values fall between 0.499 and 0.501, a very small weight to a novelty score can already make a large difference when choosing moves.

5 CONCLUSIONS AND FURTHER WORK

In this preliminary work, we tested three different state novelty measures with the goal of improving the exploration behavior of MCTS. Results in several board games were promising.

Our method of linearly combining novelty scores with MCTS value estimates exposes a nuance that may be more subtle than using novelty as an added reward bonus in the literature on intrinsic motivation, or as the sole objective in evolutionary search: novelty is here exploited as a heuristic value estimate. We compute this novelty-as-a-value-estimate as a sort of prior, and combine it with the regular action value estimates of the MCTS tree using a weight factor. This weight factor decays to zero as online observations replace the prior that biases the exploration of search. With novelty providing a form of *online generalization for uncertainty*, it appears complementary to the exploration/uncertainty term of UCB1 when sufficiently many samples are available.

Future work includes the testing of additional novelty measures, and scaling up to more, and more varied, test domains. Time-controlled experiments should be conducted in order to take the overhead of the different novelty computations into account, not just their sample efficiency. Multiple types of novelty could be combined during search in order to exploit different ways of generalizing uncertainty online, for example as in *Multiple Estimator MCTS* [3]. Novelty could also be compared to, and combined with, a variety of MCTS enhancements that generalize *value* online, such as for example RAVE and its variants [9], MAST/PAST/FAST [7], or OMA [2].

Furthermore, it would be interesting to examine the relationship between the quality of the heuristic evaluation function (state value estimator), and the usefulness of novelty-enhanced search. Perfect value estimates would of course make search and exploration unnecessary; and novelty is meant to help in particular with deceptive and misleading heuristics, whose gradients do not necessarily lead the agent (directly enough) to its goal [13]. In first experiments with MCTS guided by random rollouts instead of evaluation functions, we did not find novelty to work for any of the tested domains – even though the random-rollout-guided MCTS players are far weaker than the ones guided by heuristic value estimates. This deserves further study.

Future work could also shed light on whether novelty-based approaches can improve the exploration behavior of MCTS guided by neural networks, such as in the AlphaZero or MuZero frameworks [21, 23]. Here, novelty could potentially help improve both learning *and* planning performance.

ACKNOWLEDGMENTS

This work is part of the project *Flexible Assets Bid Across Markets* (FABAM, project number TEUE117015), funded within the Dutch Topsector Energie / TKI Urban Energy by Rijksdienst voor Ondernemend Nederland (RvO).

REFERENCES

[1] Arthur Aubret, Laëtitia Matignon, and Salima Hassas. 2019. A survey on intrinsic motivation in reinforcement learning. *CoRR* abs/1908.06976 (2019). arXiv:1908.06976

[2] Hendrik Baier and Michael Kaisers. 2020. Guiding Multiplayer MCTS by Focusing on Yourself. In *2020 IEEE Conference on Games (CoG 2020)*. 550–557.

[3] Hendrik Baier and Michael Kaisers. 2021. ME-MCTS: Online Generalization by Combining Multiple Value Estimators. In *30th International Joint Conference on Artificial Intelligence (IJCAI 2021)*.

[4] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. 2016. Unifying Count-Based Exploration and Intrinsic Motivation. In *Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS 2016)*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). 1471–1479.

[5] Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. 2018. Exploration by Random Network Distillation. *CoRR* abs/1810.12894 (2018). arXiv:1810.12894

[6] Hilmar Finnsson. 2012. Generalized Monte-Carlo Tree Search Extensions for General Game Playing. In *Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2012)*, Jörg Hoffmann and Bart Selman (Eds.). AAAI Press.

[7] Hilmar Finnsson and Yngvi Björnsson. 2010. Learning Simulation Control in General Game-Playing Agents. In *Twenty-Fourth AAAI Conference on Artificial Intelligence, (AAAI 2010)*, Maria Fox and David Poole (Eds.). AAAI Press.

[8] Tomas Geffner and Hector Geffner. 2015. Width-Based Planning for General Video-Game Playing. In *Eleventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2015)*, Arnav Jhala and Nathan Sturtevant (Eds.). 23–29.

[9] Sylvain Gelly and David Silver. 2007. Combining Online and Offline Knowledge in UCT. In *Twenty-Fourth International Conference on Machine Learning (ICML 2007) (ACM International Conference Proceeding Series, Vol. 227)*, Zoubin Ghahramani (Ed.). ACM, 273–280.

[10] Michael Katz, Nir Lipovetzky, Dany Moshkovich, and Alexander Tuisov. 2017. Adapting Novelty to Classical Planning as Heuristic Search. In *Twenty-Seventh International Conference on Automated Planning and Scheduling (ICAPS 2017)*, Laura Barbulescu, Jeremy Frank, Mausam, and Stephen F. Smith (Eds.). 172–180.

[11] Levente Kocsis and Csaba Szepesvári. 2006. Bandit Based Monte-Carlo Planning. In *17th European Conference on Machine Learning (ECML 2016)*. 282–293.

[12] Joel Lehman and Kenneth O. Stanley. 2008. Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. In *Artificial Life XI: Eleventh International Conference on the Synthesis and Simulation of Living Systems*, Seth Bullock, Jason Noble, Richard A. Watson, and Mark A. Bedau (Eds.). 329–336.

[13] Joel Lehman and Kenneth O. Stanley. 2011. Abandoning Objectives: Evolution Through the Search for Novelty Alone. *Evol. Comput.* 19, 2 (2011), 189–223.

[14] Wilkins Leong. 2017. *General Game Playing Using Search for Novelty.* Student research project. University of Melbourne, Melbourne, Australia.

[15] Diego Perez Liebana, Spyridon Samothrakis, Julian Togelius, Tom Schaul, Simon M. Lucas, Adrien Couëtoux, Jerry Lee, Chong-U Lim, and Tommy Thompson. 2016. The 2014 General Video Game Playing Competition. *IEEE Trans. Comput. Intell. AI in Games* 8, 3 (2016), 229–243.

[16] Nir Lipovetzky and Hector Geffner. 2012. Width and Serialization of Classical Planning Problems. In *20th European Conference on Artificial Intelligence (ECAI 2012) (Frontiers in Artificial Intelligence and Applications, Vol. 242)*, Luc De Raedt, Christian Bessiere, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas (Eds.). 540–545.

[17] Nir Lipovetzky, Miquel Ramírez, and Hector Geffner. 2015. Classical Planning with Simulators: Results on the Atari Video Games. In *Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, Qiang Yang and Michael J. Wooldridge (Eds.). 1610–1616.

[18] Marlos C. Machado, Marc G. Bellemare, and Michael Bowling. 2020. Count-Based Exploration with the Successor Representation. In *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)*. 5125–5133.

[19] Jarryd Martin, Suraj Narayanan Sasikumar, Tom Everitt, and Marcus Hutter. 2017. Count-Based Exploration in Feature Space for Reinforcement Learning. In *Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*, Carles Sierra (Ed.). 2471–2478.

[20] Georg Ostrovski, Marc G. Bellemare, Aäron van den Oord, and Rémi Munos. 2017. Count-Based Exploration with Neural Density Models. In *34th International Conference on Machine Learning (ICML 2017) (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). 2721–2730.

[21] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy P. Lillicrap, and David Silver. 2019. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *CoRR* abs/1911.08265 (2019). arXiv:1911.08265

[22] Alexander Shleyfman, Alexander Tuisov, and Carmel Domshlak. 2016. Blind Search for Atari-Like Online Planning Revisited. In *Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016)*, Subbarao Kambhampati (Ed.). 3251–3257.

[23] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. 2018. A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-Play. *Science* 362, 6419 (2018), 1140–1144.

[24] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. 2017. Mastering the Game of Go without Human Knowledge. *Nature* 550, 7676 (2017), 354–359.

[25] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. 2017. #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning. In *Thirtieth Annual Conference on Neural Information Processing Systems (NIPS 2017)*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 2753–2762.