ISSN 2347–3657
Volume 10, Issue 3, Sep 2022

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657
Volume 10, Issue 3, Sep 2022

Estimated Data Set for Partially Observed System Approximate Planning and Reinforcement Learning

Dr K VENKATA SUBBAIAH[1], Mr G SRINIVAS[2], Mr.K.PRAVEEN KUMAR[3]
Mrs.B.SARITHA[4]

Abstract

Reinforcement learning methods may take use of asymmetry, which occurs during offline training in partly viewable virtual environments, to their advantage. If handled correctly, such private data may significantly improve the optimum convergence qualities. Nevertheless, the majority of the present research in asym-metric reinforcement learning relies on empirical assessment and is mostly heuristic, without theoretical guarantees or linkages to underlying theory. This paper first establishes the theory of Asymmetric Policy Iteration, a model-based dynamic programming solution technique; then, it applies relaxations that lead to Asymmetric DQN, a deep reinforcement learning process that does not rely on models. Experimental results corroborate and supplement our theoretical results, which were tested in settings with high levels of partial observability and demanding of information collection techniques and memorizing.

1 INTRODUCTION

One new paradigm in reinforcement learning (RL) is known as offline training and online execution (OTOE). This approach requires learning agents to undergo training in a simulated environment before they can be used in the "real" world. Our approach focuses on the many benefits of OTOE, which include safety guarantees, rapid training, flexibility, and access to sensitive information. In fact, OTOE has become the preferred paradigm in some research cliques due to all these reasons; for example, in multi-agent RL, it is often referred to as CTDE. Offline training provides access to privileged information that is inaccessible during regular online training and/or execution. Depending on the kind of control issue, this may manifest in many ways; for example, in multi-agent RL, it can take the shape of other agents' observations and actions; in partly observable RL, it can take the shape of the system's state; and so on.

Professor & HoD[1], Asst.Professor[2,3,4]
CSE(AIML)
DRK INSTITUTE OF SCIENCE & TECHNOLOGY

ISSN 2347–3657
Volume 10, Issue 3, Sep 2022

This data is accessible in OTOE only during the offline phase, when the simulation's internal state is accessible, and is therefore a transitory privilege. Although this data is unavailable during online execution, it may enhance the agent's online performance by improving its convergence speed and overall training performance if employed correctly. As a mechanism known as asymmetry, PORL often associates actor-critical procedures with OTOE and privileged knowledge. We use the word "asymmetry" more generally to refer to the broad notion of utilizing privileged knowledge during of-fline training—two ideas that overlap significantly in our work—in addition to its extremely precise etymological meaning, which is detailed below. Two distinct models are trained in actor-critic methods: one represents the agent's behavior (the policy model) and the other represents the agent's appraisal of its circumstance (the critic model). Since both models in standard actor-critic approaches get the same information—the agent's history in PORL—we may say that these methods are inherently symmetric. This symmetry is disrupted in an asymmetric actor-critic relationship when the actor gives the critic access to information that should be kept private [Pinto et al., 2018, Foerster et al., 2018, Lowe et al., 2017, Yang et al., 2018, Li et al., 2019, Wang et al., 2020, Warrington et al., 2021, Xiao et al., 2021, Baisero and Amato, 2022, Lyu et al., 2022]. Thanks to the critic's status as a training construct—completely unnecessary and unutilized during execution—this is indeed feasible. While a secondary model similar to the critic is often only used during training, asymmetry has been employed in other DQN-like RL approaches [Rashid et al., 2018, Mahajan et al., 2019, Rashid et al., 2020, Xiao et al., 2020, de Witt et al., 2020]. As a training construct similar to the critic in actor-critic, a second value-based model is provided in such circumstances only for the purpose of asymmetry.

But much of the previous work on asymmetric RL has only suggested heuristic asymmetry models, which have only been tested empirically and don't have a theoretical foundation to ensure proper use of state information. We contend that, if not supported by sound theoretical research, such approaches could employ state data in ways that impede the training of a partly observable agent. A well-known result in partially observable control states, for instance, that an optimal fully observable agent would never take an action that an optimal partially observable agent might take. This includes information-gathering actions that help the partially observable agent learn about the environment state but do nothing for the fully observable agent. Another example is that an optimal partially observable agent might take an action that an optimal fully observable agent would never take.

A state-of-the-art asymmetric value-based deep RL algorithm for partly observable control, backed by a solid theoretical analysis, is the ultimate objective of this study. We do this by starting from the ground up and building the theory of asym-metric policy improvement, which describes methods for improving policies utilizing privileged state knowledge while keeping optimum convergence guarantees. We start by creating model-based dynamic programming solution methods called Asymmetric Policy Iteration (API) and Asymmetric Action-Value Iteration. Asymmetric Q-Learning (AQL) is the direct RL successor to API and AAVI. We do this by including components of stochastic training from sample experience. Lastly, we provide value- function approximation, which leads to Asymmetric DQN (ADQN). ADQN is a technique that can be used in a principled manner and is similar to other top-tier deep

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 10, Issue 3, Sep 2022

RL methods. We are proud to have developed theoretically-driven asymmetric value-based RL, which we believe to be the first of its kind.

2 RELATED WORK

Privileged information available offline has been used to improve training performances in a wide range of prior single-agent and multi-agent methods which include both policy-based and value-based methods.

In sigle-agent control, Pinto et al. [2018] employ DDPG with an asymmetric state-based critic to handle robot manipulation tasks; belief-grounded networks [Nguyen et al., 2021] uses a belief-based form of asymmetry and a belief-reconstruction task to train the history representation; Warrington et al. [2021], Chen et al. [2020] use imitation learning to train a partially observable agent via a fully observable agent trained offline. Baisero and Amato [2022] show theoretical issues with state-only forms of asymmetry for policy- gradients, and develop a history-state variant.

In multi-agent control, COMA [Foerster et al., 2018] uses a single centralized asymmetric critic which employs the joint observations and/or the environment state. MAD- DPG [Lowe et al., 2017] and M3DDPG [Li et al., 2019] use multiple centralized asymmetric critics, one for each agent, which employ the joint observations and/or the environment state. R-MADDPG [Wang et al., 2020] uses a recurrent model and a centralized critic which uses the entire histories of all agents; CM3 [Yang et al., 2018] uses a state-onlycritic for reactive control; MacDec-DDRQN [Xiao et al., 2020] uses a centralized value model to learn individual centralized value models. ROLA [Xiao et al., 2021] uses both centralized and individual asymmetric critics which employ local history and/or state information to estimate individual advantage values. QMIX [Rashid et al., 2018], MAVEN [Mahajan et al., 2019], and WQMIX [Rashid et al., 2020] use a centralized but factored value model to train individual agent value models. Lyu et al. [2022] extend thetheory by Baisero and Amato [2022] to the multi-agent case.

3 BACKGROUND

In the next subsection, we present some of the background required to understand our work. Section 3.1 formally de- scribes partially observable control problems. Section 3.2 contains a review of non- asymmetric value-based control, in the form of DQN.

Section 3.3 covers the definition of *history-state* value functions. In Section 3.4, we present operator notation and useful operators.

3.1 POMDPS

A partially observable Markov decision process (POMDP) is a discrete-time control problem represented by tuple $\langle S, A, O, b_0, T, O, R, \gamma \rangle$, where (a) S, A, and O are state, action, and observation spaces, (b) $b_0 \in \Delta S$ is an initial state distribution, (c) $T : S \times A \to \Delta S$ is a stochastic state transition function, (d) $O : S \times A \times S \to \Delta O$ is a stochas- tic observation emission function, (e) $R : S \times A \to R$ is a reward function, and (f) $\gamma \in [0, 1)$ is a discount factor.

Partially observable control is based on observable *histo- ries*, i.e., the sequences of past actions and observations. The *history space* $(A \times O)^*$ represents such se- quences. To simplify notation, we overload symbol R to also denote the expected reward function on histories $R(h, a) = E_{s|h}[R(s, a)]$. General partially observable poli-cies take the form of mappings from *histories* to action distributions $\pi : \Delta$; however, in this work, we will focus exclusively on *deterministic* policies $\pi : \Sigma$. The goal of the control problem is to find a policy which maxi- mizes the episodic expected *return* $E[\sum_t \gamma^t R(s_t, a_t)]$.

Every policy π is associated with an action-value function $Q^\pi(h, a)$ which represents the expected return associated with the agent having observed history h, taking action a, and then continuing to behave according to the policy π. Q^π is the unique solution to the Bellman equation,

$$Q^\pi(h, a) = R(h, a) + \gamma E_{o|h,a}[Q^\pi(hao, \pi(hao))] . \quad (1)$$

The action-value function associated with the optimal pol- icy π^* is denoted as Q^*, and is the unique solution to the Bellman *optimality* equation,

$$Q^*(h, a) = R(h, a) + \gamma E_{o|h,a}\left[\max_{a'} Q^*(hao, a')\right] . \quad (2)$$

Notation We use symbols Q, Q^π, Q^*, and $\hat{Q}$ to denote similar but separate concepts. $Q : H \times A \to R$ denotes an arbitrary real-valued function, not necessarily associated with any policy, Q^π denotes the value function associated with a policy, Q^* denotes the value function associated with an optimal policy, while $\hat{Q}$ denotes a (deep) parametric model. We denote the

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 10, Issue 3, Sep 2022

space of all such history-action real functions as $\mathbb{Q} \doteq \{Q \mid Q\colon \mathrm{H} \times \mathrm{A} \to \mathbb{R}\}$. Further, we use $g(Q)$ to denote the policy which acts greedily based on Q,i.e., if $\pi = g(Q)$, then $\pi(h) = \operatorname{argmax}_a Q(h, a)$.

3.2 DQN

Deep Q-Network (DQN) [Mnih et al., 2015] is a highly successful algorithm for training deep neural networks to control high-dimensional fully-observable Markov decision processes (MDPs) based on reward feedback, and the first to achieve human-level performance on a majority of the Atari 2600 games. Rather than relying on a lookup table to track the estimated expected return for each state-action pair (s, a), DQN learns a parametric function $\hat{Q}$:

variables in the relevant equations, and by employing archi- tectures capable of processing history data. *Frame stacking*, i.e., the practice of concatenating a small number of re- cent observations, has been found to be sufficient to tackle problems which feature small amounts of partial observabil- ity [Mnih et al., 2015]. On the other hand, larger amounts of partial observability generally require longer-term memo- rization capabilities. For such problems, the standard choice has become that of combining the DQN training algorithm with a recurrent neural network component used to pro- cess history data, also known as *Deep Recurrent Q-Network* (DRQN) [Hausknecht and Stone, 2015]. Although some practitioners use the DQN label exclusively to indicate the variant which lacks a recurrent component, our view is thatthe essence of the DQN algorithm is in its training regime and its losses, rather than the details of which architecture is used. Therefore, in this document, we use the label DQN more broadly to encompass all architectural variants. In practice, because our work focuses on control problems which feature large amounts of partial observability, we em- ploy appropriate algorithmic and modeling choices, i.e., allmethods and baselines employ a history-based model $\hat{Q}$, and all models which receive history data employ a recurrent network component to process it.

Theorem 3.3 (AQL Optimality)**.** *Assume stepsizes* α_k *sat- isfying the following asymptotic conditions,*

$$\sum_{k=0}^{\infty} \alpha_k = \infty, \qquad \sum_{k=0}^{\infty} \alpha_k^2 < \infty. \tag{15}$$

While it may be possible to approximate this factor in other model-free ways, AQL remains primarily a conceptual al- gorithm also due to the requirement of a tabular model overhistories. Either way, AQL serves as a fundamental basis to derive the next iteration of asymmetric value-based RL.

4.1 ASYMMETRIC DQN

When acting in POMDPs with high-dimensional observa- tions and states, a tabular-lookup method such as AQL be-comes infeasible. In such instances, we must introduce func- tion approximation to generalize over similar experiences. The use of approximation sacrifices the optimal convergence guarantee established by Theorem 4.4, but is necessary to scale algorithms to significantly more challenging partially observable environments. Nevertheless, the value function approximations are an orthogonal matter to how privilegedstate information is used, and we expect the sound theoreti- cal principles upon which AQL is built will help asymmetric deep methods even when relying on function approximation. To generalize over state-action pairs, the algorithm reformu- lates the increment

If Q_0, U_0 *are mutually consistent* ($Q_0 = EU_0$)*, then the sequences* Q_k *and* U_k *generated by AQL converge to* Q^* *and* U^* *with probability 1 (proof in Appendix A.5.)*

[1]The Q-evaluation step of AAVI (Equation (10)) can equiva- lently be expressed as $Q_{k+1} \leftarrow EB_{g(Q_k)}U_k$, which is the form AQL employs to guarantee optimal convergence.

Our primary algorithmic contribution here is *Asymmetric DQN* (ADQN), an asymmetric variant of DQN derived by introducing value function approximation to AQL. We first replace the tabular-lookup models U and Q of AQL with parametric differentiable models $\hat{U}$ and $\hat{Q}$. In practice, these are implemented as deep neural networks whose architec- tures are chosen according to the structure of the states andobservations emitted by the POMDP.

To facilitate the substitution, we must reformulate the stochastic update rules of AQL as squared-error loss mini- mization. For the rest of the section, due to spacing concerns, we will use $\hat{\pi} = g(\hat{Q})$ as a shorthand to represent actions se- lected greedily on $\hat{Q}$, i.e., $\hat{\pi}(h) = \operatorname{argmax}_a \hat{Q}(h, a)$. Given a single environment interaction (h, s, a, r, s', o), the corre- sponding losses can be defined as

$$\ell_{\hat{U}} = \left(r + \gamma\, \mathrm{SG}\left[\hat{U}(hao, s', \hat{\pi}(hao))\right] - \hat{U}(h, s, a)\right)^2, \tag{16}$$

$$\ell_{\hat{Q}} = \left(r + \gamma\, \mathrm{SG}\left[\hat{U}(hao, s', \hat{\pi}(hao))\right] - \hat{Q}(h, a)\right)^2, \tag{17}$$

where SG is the *stop-gradient* function which indicates thatgradient calculation should not consider the enclosed terms.It is worth noting that $\ell_{\hat{U}}$ and $\ell_{\hat{Q}}$ use the same target to train $\hat{U}$ and $\hat{Q}$. The crucial difference is that $\hat{U}$ is in able to model the target as a function of s, while $\hat{Q}$

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 10, Issue 3, Sep 2022

is unable to do so, and can at only model the expectation of the target over values of s. In a way, these losses approximately enforce a "loose" form of mutual consistency $\hat{Q} \approx E\hat{U}$. In practice, the term is generated by "target networks" that rely on stale parameters to stabilize learning when bootstrapping [Mnih et al., 2015]; the stale parameters are periodically updated by copying the main parameters. The total loss $L_{\hat{U}} + L_{\hat{Q}}$ can be jointly minimized with respect to the parameters by a single backpropagation step, efficiently approximating the interleaved updates of AQL.

When the function approximators are nonlinear (as is oftenthe case for neural networks), training will fail if the gradient updates are conducted on sequentially collected transitions that are not i.i.d. [Mnih et al., 2015]. The second critical modification to AQL is therefore the adoption of experience replay [Lin, 1992] in order to decorrelate training experi- ences. Rather than training on a sample immediately when it is collected, each POMDP transition (h, s, a, r, s', o) is deferred to a first-in first-out replay memory. Periodically, when it is time to train the networks, a minibatch of several experiences is sampled from the replay memory; gradientsfor these samples are computedLand aLveraged together to estimate the true gradient of the joint loss $L_{\hat{U}} + L_{\hat{Q}}$, which in turn is used to improve the parameters.

Why ADQN? Having finally addressed the practical dis- advantages of API, it is worthwhile to reconsider the "why" question again, this time focusing on why one would preferto use ADQN compared to DQN. Ultimately, the purpose of both algorithms is to train an approximate $\hat{Q} \approx Q^*$ through which optimal control can be executed, and both algorithms should *in theory* converge to very similar approximations. What is then the advantage of ADQN over DQN? Similarly to the asymmetric actor- critic case [Baisero and Amato, 2022], the advantage is a practical one associated with the difficulties of learning an appropriate representation of his- tory $\varphi(h)$, which is one of the major bottlenecks in PORL.

History representations are sequence models which notori- ously requires lots of data and processing power for propertraining. To further compound on this issue, the quality of the data used to train the history representation in PORL isdirectly related to the quality of $\hat{Q}(\varphi(h), \cdot, \cdot)$, which in turn depends on the quality of the history representation itself; unsurprisingly, it can be quite hard to bootstrap the training of an improved history representation when starting from a poor history representation. Note, however, that learning an appropriate state representation $\varphi(s)$ is much simpler than learning $\varphi(h)$ due to the non-sequential nature of indi- vidual states, i.e., the $\varphi(s)$ representation model has fixed input and output sizes, and can generally be modeled using a simpler feed-forward architecture. In ADQN, the issues associated with learning a proper history representation areallevіated by the fact that its training is bootstrapped not only on the history representation itself, but also on the state representation.

Even when the history representation is poor, we can expect the state representation to contain sufficient contextual information to allow $\hat{U}(\varphi(h), \varphi(s), \cdot)$ to model meaningful values, which in turns h·elps further bootstrap the learning of the history representation $\varphi(h)$, the history model $\hat{Q}(\varphi(h), \cdot)$, and the respective implicit policy $g(\hat{Q})$.

Next, we consider some variants of interest of ADQN.

4.1.1 Variance-Reduced ADQN

In this variant, we approximate the target of $L_{\hat{Q}}$ from Equa- tion (17) as $r + \gamma \hat{U}(hao, s' \sim \hat{n}(hao)) \approx \hat{U}(h, s, a)$, which holds particularly well once $\hat{U}$ has been trained sufficiently. Therefore, this variant uses the following losses,

$$L_{\hat{U}} = (r + \gamma \, \mathrm{SG}[\hat{U}(hao, s', \hat{n}(hao))] - \hat{U}(h, s, a))^2, \quad (18)$$

$$L_{\hat{Q}} = (\mathrm{SG}[\hat{U}(h, s, a)] - \hat{Q}(h, a))^2. \quad (19)$$

This approximate target results in lower variance throughout the entire training process at the cost of introducing bias primarily in the early stages of training; a trade-off which may result advantageous in some control problems.

4.1.2 State-Only ADQN

Some prior work in asymmetric RL has adopted heuristic forms of asymmetry which uses state-only (i.e., history-less) value functions $U(s, a)$. Such form of asymmetry is how- ever associated with fundamental theoretical issues which may severely compromise the learning performance, rang- ing from potentially being ill-defined, to introducing bias into the learning process [Baisero and Amato, 2022]. Such issues are inherently related to partial observability, and their effects scale with the amount of partial observability that the agent is subject to, as well as the agent's "reactiveness", i.e. the amount of history that is willfully ignored by the agent itself to select actions. Nonetheless, this state-only form of value function may be more useful than others in control problems which have small amounts of partial observability, such as vision-based tasks with an occlusion-free view of the environment [Pinto et al., 2018, Baisero and Amato, 2022]. Although our main focus is control problems with significant amounts of partial observability, we are still in- terested in formulating a state-only variant of ADQN as an additional baseline, and as reference for future work

whichmay focus on the kinds of control problems where it thrives.

In this state-only variant, we redefine the parametric model $\hat{U}(s, a)$ to ignore history, and adopt the following losses,

$$L_{\hat{U}} = (r + \gamma\, \mathrm{SG}[\hat{U}(s', \hat{\pi}(hao))] - \hat{U}(s, a))^2, \quad (20)$$

$$L_{\hat{Q}} = (r + \gamma\, \mathrm{SG}[\hat{U}(s', \hat{\pi}(hao))] - \hat{Q}(h, a))^2. \quad (21)$$

4.1.3 Reduced-Variance State-Only ADQN

This variant applies a state-only variant of the variance reduction approximation from Section 4.4.1 to state-only ADQN. In this case, we approximate the target of $L_{\hat{Q}}$ from Equation (21) as $r + \gamma \hat{U}(s', \hat{\pi}(hao)) \approx \hat{U}(s, a)$,

$$L_{\hat{U}} = (r + \gamma\, \mathrm{SG}[\hat{U}(s', \pi(hao))] - \hat{U}(s, a))^2, \quad (22)$$

$$L_{\hat{Q}} = (\mathrm{SG}[\hat{U}(s, a)] - \hat{Q}(h, a))^2. \quad (23)$$

4 EVALUATION

We perform an empirical evaluation of our proposed ADQN method and its variants in a variety of environments which feature significant amounts of partial observability.

Methods We compare the performances of 5 value-based PORL algorithms, denoted as follows:

- **DQN** is the standard non-asymmetric DQN algorithm;
- **ADQN** and **ADQN-VR** are the history-state ADQN algorithms from Sections 4.4 and 4.4.1; and
- **ADQN-State** and **ADQN-State-VR** are the state-only ADQN algorithms from Sections 4.4.2 and 4.4.3.

Environments Evaluations are run on 5 partially observ-able navigation tasks which require information gathering strategies and memorization of the past:

- **Heaven-Hell-3** and **Heaven-Hell-4** [Bonet, 1998], cor- ridor environments where the agent must reach the exit to *heaven* and avoid the exit to *hell*, but must first back- track to visit a *priest* to learn which exit is which;
- **Cleaner** [Jiang and Amato, 2021], a maze environment where two agents must reach all tiles to clean them. Inour experiments, the two agents are treated as a single agent, and controlled in a centralized fashion; and
- **GV-MemoryFourRooms-7x7** [Baisero and Katt, 2021], a dynamically generated gridworld with 4 con- nected rooms, where the agent must reach the *good* exit and avoid the *bad* exit, but must first find and memorize a *beacon* to learn which is which.

A more thorough description of these environments can be found in Appendix C of Baisero and Amato [2022].

Each method is trained and evaluated using code available as a public repository[2]. For each environment and algorithm, we perform an independent grid-search over some hyper- parameters of interest (see Appendix D), and select the com- bination of hyper-parameters which results in the best finalperformance and learning stability (prioritizing final perfor- mance if necessary). To improve the statistical significance of the results, each combination of environment, algorithm, and hyper-parameters is run 20 independent times.

4.1 RESULTS AND DISCUSSION

Figure 1 shows the results of these evaluations, which broadly confirm our theoretical analysis on asymmetric value-based PORL, the practical advantage of employing history-state forms of evaluation to aid partially observable control, and confirm the superiority of ADQN compared to other similar symmetric and asymmetric variants. This eval- uation further confirms other recently developed theoretical analysis on asymmetric PORL, i.e., that state-only forms ofasymmetry are inadequate to handle non-trivial amounts of partial observability [Baisero and Amato, 2022].

Across the board, **ADQN** and **ADQN-VR** outperform all baselines in final performance, convergence speed, and/or overall learning stability. The contrast between methods is particularly stark in Figures 1a and 1b, where **ADQN** and **ADQN-VR** are not just the only methods that demon- strate any substantial improvement, but are also able to reach optimal performance. On the other hand, the state- only variants fail to outperform even the **DQN** baseline in most environments (with a single exception discussed later), which further confirms the theoretical issues that have been recently associated with state-

ISSN 2347–3657
Volume 10, Issue 3, Sep 2022

reduced variants **ADQN-VR** and **ADQN-State-VR** only differ in relatively minor ways from their default counterparts. Such differences can be found in Figures 1a and 1d, where **ADQN** is more stable or has better convergence properties than **ADQN-State-VR**, and in Figures 1c and 1d, where **ADQN-State-VR** has better final convergence values than **ADQN-State**. This seems to

- **Car-Flag** [Nguyen, 2021], a 1-dimensional continuouscontrol variant of **Heaven-Hell**;

6 CONCLUSIONS

When agents are taught in OTOE's simulated environment, they have temporary access to privileged information that would otherwise be unavailable, such as the status of the partly viewable environment. This framework is used in RL. A typical way to leverage such privileged knowledge during training is asymmetry, which, when done well, may significantly improve learning performance and efficiency. But much recent asymmetric RL work relies on heuristics that haven't been tested and don't have any theoretical backing. We created the theory of asymmetric value-based RL to address this gap in the literature. By starting with the fundamentals of asymmetric policy improvement and working our way up from there, we were able to accomplish our major objective of creating a theoretically sound asymmetric value-based RL algorithm. The API was the conceptual solution approach that was used, which had specific practical restrictions but guaranteed optimum convergence. Then, to overcome these limits, we relaxed API in a number of ways that led to ADQN, a competitive and viable deep RL algorithm. Using a battery of carefully chosen environments with high levels of partial observability, information-gathering strategies, and memorization of the past, we conducted an empirical evaluation to compare the performances of ADQN and its variants to standard non-asymmetric DQN. Even when faced with control issues that regular DQN couldn't handle, ADQN consistently outperformed in all of these settings. The general effectiveness of our ADQN method in partly observable control problems, as well as the significance of leveraging privileged knowledge in principled and theoretically-guided ways, were all verified by our assessment. Extending ADQN to the multi-agent control case, which presents additional learning challenges, identifying tasks in vision-based robotics that could benefit from state-only ADQN (e.g., tasks with little partial observability), and expanding the evaluation of ADQN in more complex partially observable vision-based tasks are all potential areas for future work.

References

Baldo Baisero and Amato Christopher. Uniform Asymptotic Reinforcement Learning with Partial Perceptions. Pages 44–52 of the 2022 Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems.
Alessandro Baisero and Samantha Katt. "gym-gridverse": Grid-world environments for reinforcement learning with and without partial observation, 2021. Please visit https://github.com/abaisero/gym-gridverse for more information.
Bonet, Blai. Solving big POMDPs with real time dynamic programming. Proceedings of the AAAI Fall Symposium on POMDPs, 1998.

Philipp Krähenbühl, Vladlen Koltun, Brady Zhou, and Dian Chen. Gaining knowledge via deceit. On pages 66–75 of the Conference on Robot Learning. 2020 PMLR.

Bei Peng, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson are all involved. Decentralized Continuous Cooperative Control via Deep Multi-Agent Reinforcement Learning. January 2020, as a preprint on arXiv:2003.06709.

Triantafyllos Afouras, Jakob Foerster, Gregory Farquhar, Nantas Nardelli, and Shimon Whiteson made up the group. Policy gradients for hypothetical multi-agent scenarios. Part of the 2018 AAAI Artificial Intelligence Conference Proceedings, volume 32.

Peter Stone and Matthew Hausknecht. Deep recurrent q-learning for partly observable mdps... Published in the 2015 AAAI Fall Symposium Series.

Hiring: Shuo Jiang and Christopher Amato since 2018. Guided exploration and selective memory reuse in multi-agent reinforcement learning. Pages 777–784 of the 2021 ACM Symposium on Applied Computing Proceedings.
Xinyue Cui, Fei Fang, Honghua Dong, Yi Wu, Stuart Russell, and Shihui Li were all involved. Deep deterministic policy gradient with minimax for robust multi-agent reinforcement learning. Volume 33, pages 4213–4220, 2019 AAAI Conference on Artificial Intelligence Proceedings.
Machine learning, 8(3):293-321, 1992. Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning, and teaching.

ISSN 2347–3657
Volume 10, Issue 3, Sep 2022

Igor Mordatch, OpenAI Pieter Abbeel, Aviv Tamar, Jean Harb, Yi I. Wu, and Ryan Lowe. In settings that include cooperative and competitive elements, multi-agent actor-critic. On pages 6379–6390 of the 2017 edition of Advances in Neural Information Processing Systems.
Christophe Amato, Yuchen Xiao, Xueguang Lyu, and Andrea Baisero. The Role of State-Based Critics in MAL with a More Nuanced Perspective. Published in the 2022 AAAI Artificial Intelligence Conference Proceedings.
Contributors: Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven supports multi-agent variational exploration. Within the 2019 edition of Advances in Neural Information Processing Systems, on pages 7613-7624.

Andreas K. Fidjeland, Koray Kavukcuoglu, Volodymyr Mnih, David Silver, An- drei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, and Martin Riedmiller. The use of deep reinforcement learning for control at the human level. Nature, 518(7540):529, 2015.
Hello Lê Văn.URL: https://github.com/hai-h-nguyen/pomdp-domains, POMDP Robot Domains, 2021.
Kevin Daley, Xinchao Song, Hai Nguyen, Christopher Am-ato, and Robert Platt. Belief-Grounded Networks for Partial Observability-Assisted Robot Learning. Proceedings of the Conference on Robot Learning, volume 16, pages 1640–1653. 2021 PMLR.

The authors are Pieter Abbeel, Wojciech Zaremba, Marcin Andrychowicz, Peter Welinder, and Lerrel Pinto. Asymmetric Actor Critic for Image-Based Robot Learning. In 2018's Robotics: Science and Systems Proceedings.

The following individuals are listed: Christian Schroeder, Jakob Foerster, Shimon Whiteson, Christian Rashid, and Mikayel Samvelyan. Deep multi-agent reinforcement learning with Qmix: factoring monotonic value functions. Pp. 4295–4304 in 2018's International Conference on Machine Learning (PMLR).

Bei Peng, Gregory Farquhar, Tabish Rashid, and Shimon Whiteson. Weighted qmix: Deep multi-agent reinforcement learning with expanding monotonic value function factorization. Proceedings of the 33rd annual conference on neural information processing systems, pp. 10199–10211. October 2020.

Andrew G. Barto and Richard S. Sutton. A primer on reinforcement learning. In 2018, the MIT Press appeared.