Learning Complementary Representations of the Past using Auxiliary Tasks in Partially Observable Reinforcement Learning

Extended Abstract

Andrea Baisero
Northeastern University
Boston, Massachusetts, USA
baisero.a@northeastern.edu

Christopher Amato
Northeastern University
Boston, Massachusetts, USA
c.amato@northeastern.edu

KEYWORDS

reinforcement learning; partial observability; state representations

ACM Reference Format:
Andrea Baisero and Christopher Amato. 2020. Learning Complementary Representations of the Past using Auxiliary Tasks in Partially Observable Reinforcement Learning. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

Partially observable Markov decision processes (POMDPs) define discrete-time sequential control problems [3, 11, 20]. In partially observable reinforcement learning (RL), an agent lacks access to the system state or domain model, and has to rely on the observable past (aka *history-state*) for decision making [20]. History-states are intrinsically complex, and extracting more appropriate representations is very challenging albeit necessary for general POMDPs. We refer to this as the *history representation learning* problem.

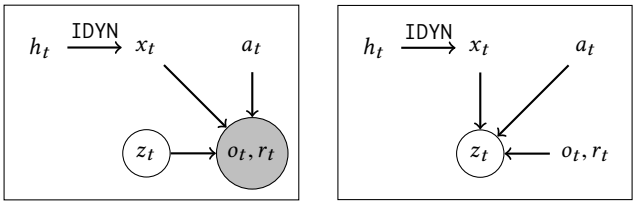
Part of the difficulty of learning history representations is intrinsic to RL and its characteristically weak learning signal. Inspired by recent efforts in complementing RL with auxiliary tasks [10, 14, 18], we argue that auxiliary tasks can be used to learn richer and more structured history representations, which in turn benefit the agent.

2 BACKGROUND

We use calligraphy $\mathcal{X}$ to indicate sets, lowercase $x \in \mathcal{X}$ to indicate set elements, uppercase X to indicate random variables which take values from $\mathcal{X}$, and $\Delta\mathcal{X}$ to denote the set of distributions on $\mathcal{X}$.

POMDPs. A POMDP is composed of *state-*, *action-*, and *observation-*spaces $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{O}$; *dynamics* $\mathrm{D}\colon \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S} \times \mathcal{O})$; *reward function* $\mathrm{R}\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$; and *discount factor* $\gamma \in [0, 1]$. We define the *reward-space* $\mathcal{R} \doteq \{\mathrm{R}(s, a) \mid s \in \mathcal{S}, a \in \mathcal{A}\}$. A *return* $g_t \doteq \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ is defined as the discounted sum of future rewards. A *history* $h \in \mathcal{H}$ is a sequence of past interactions, and a *belief* $b(h) \doteq \Pr(S \mid h)$ is the distribution over states given a history. In model-free partially observable RL, a *history-policy* $\pi_{\mathcal{H}}\colon \mathcal{H} \to \Delta\mathcal{A}$ is trained to maximize the expected episodic return $\mathbb{E}[G_0]$.

Internal State Representations. We view the history-policy as the composition $\pi_{\mathcal{H}} \equiv \pi_{\mathcal{X}} \circ \phi$ of a feature extraction model $\phi\colon \mathcal{H} \to \mathcal{X}$ and a policy model $\pi_{\mathcal{X}}\colon \mathcal{X} \to \Delta\mathcal{A}$. We refer to $\mathcal{X}$ as the *internal-state* (istate) set. Consequently, we refer to ϕ as the *istate representation* (irepresentation), and to $\pi_{\mathcal{X}}$ as the *istate policy* (istate-policy).



(a) Generative model (decoder) $p(z, o, r; h, a)$ **(b) Inference model (encoder)** $q(z; h, a, o, r)$

Figure 1: VAE models for the one-step prediction task, which train a shared irepresentation model `IDYN`.

Reactive irepresentations [15, 16] concatenate the m most recent actions and observations. They are easy to use, require a minimal number of trainable parameters, and have inherent temporal structure. However, they generalize poorly, and performance is critically dependent on the memory parameter m.

Recurrent representations are modeled by recurrent neural networks (RNNs) [6, 12, 22]. They theoretically can exhibit infinite-term memory and generalize well. However, training them to exhibit either property remains challenging. In practice, they usually exhibit some mid-term memory and some form of generalization.

Advantage Actor Critic. Advantage Actor Critic (A2C) [5, 21] is a policy gradient variant which trains *actor* $\pi(h; \theta_\pi) \in \Delta\mathcal{A}$ and *critic* $v(h; \theta_v) \in \mathbb{R}$ models. The overall objective $\mathcal{L}_{\mathrm{A2C}}(\theta_\pi, \theta_v) = \mathcal{L}_\pi(\theta_\pi) + \mathcal{L}_v(\theta_v) + \alpha\mathcal{L}_S(\theta_\pi)$, is composed of *policy*, *critic*, and weighed *entropy* losses,

$$\nabla\mathcal{L}_\pi(\theta_\pi) \propto -\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \delta_t \nabla \log \pi(A_t; H_t, \theta_\pi)\right] \tag{1}$$

$$\nabla\mathcal{L}_v(\theta_v) \approx \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \nabla \delta_t^2\right] \tag{2}$$

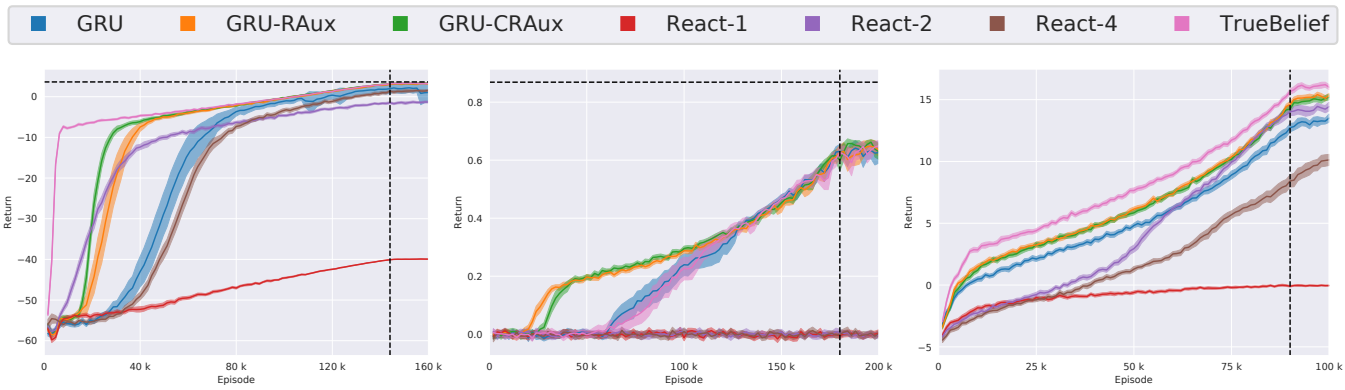
$$\nabla\mathcal{L}_S(\theta_\pi) = -\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \nabla \mathbb{H}[\pi(\,\cdot\,; H_t, \theta_\pi)]\right] \tag{3}$$

where $\delta_t \doteq R_t + \gamma v(H_{t+1}; \theta_v) - v(H_t; \theta_v)$ is the TD error.

3 ONE-STEP PREDICTION TASK

Interactions with the environment contain a certain "learning potential" not fully captured by the RL problem alone; Auxiliary learning tasks can exploit this potential to train richer irepresentations [7, 10, 17, 18], ultimately helping the RL agent. In the context

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand. © 2020 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.



(a) Shopping-5. **(b) Heavenhell-3.** **(c) Rocksample-5-6.**

Figure 2: Training performance averaged over 40 independent runs, with shaded areas showing 2 standard errors of the mean.

of the representation learning problem, we argue that the ideal auxiliary task satisfies the following properties:

a) To kick-start and accelerate learning, it should constitute an *easy* learning problem; we primarily consider self-supervision.
b) To promote sample efficiency, a single trajectory should constitute multiple input-output data-points; the task should be *well-defined* for every time-step.
c) To further promote sample efficiency, the task should be *stationary* and invariant w.r.t. a non-stationary agent; a replay memory could then be used to exploit off-policy experiences.
d) To promote generalization, belief-states should be a *sufficient statistic* for histories; we can exclude task like the reconstruction of histories via autoencoders, since they require considerably more information than given by the belief-state.

The theory of predictive state representations (PSRs) proves [13, 19] that an irepresentation which encodes sufficient information for the prediction of observable futures is a sufficient statistic of the observable past and, equivalently, of the underlying belief-state. Taking inspiration from PSRs, we define the *one-step prediction task* (PRED) as the problem of training a predictive model $p\colon \mathcal{H} \times \mathcal{A} \to \Delta(\mathcal{O}\times\mathcal{R})$ to estimate $\Pr(o, r \mid h, a)$; To simplify the reward prediction problem, we treat it as a classification of discrete rewards.

A Variational Autoencoder for One-Step Predictions. We address the PRED task with variational autoencoders [2, 4, 8] (VAEs). VAEs are used to train stochastic generative models, and have been used for model-based RL [1, 9]. In a VAE, generative and inference models p and q are trained to maximize the *evidence lower bound* (ELBO):

$$\mathcal{L}_{\text{ELBO}}(h, a, o, r) = \mathop{\mathbb{E}}_{z\sim q(z;h,a,o,r)}\left[\log \frac{p(o, r; z, h, a)}{q(z; h, a, o, r)}\right] \qquad (4)$$

Figure 1 shows our VAE model. Our use of VAEs is unconventional in that our primary interest is that of training the irepresentation model, and we do not fully use the generative model.

Training with RL and PRED. We consider two schemes for representation learning. In the first, a single representation is trained on both tasks (*RAux*). In the second, two *complementary* representations are trained (*CRAux*) as described below.

Consider a hypothetical: assuming access to a useful *black-box* irepresentation, how this benefit the RL agent? One option is to concatenate the black-box features with the features of a trainable irepresentation. If done like so, we expect the black-box features to not only kick-start the policy learning, but also allow the trainable irepresentation to focus on *complementary* features. While we cannot reasonably assume access to a black-box irepresentation in RL, we propose treating an irepresentation trained on one task as a black-box for to the other task (i.e. blocking gradients), and vice versa. Each irepresentation is then encouraged to learn features not provided by the other task. We believe this will result in the RL task bootstrapping its own learning by encouraging the PRED representation to learn features which are not easily learned by the RL task, which are thus more useful for the RL task.

4 EVALUATION

We evaluate the effectiveness of the PRED task, and both the RAux and the CRAux training schemes, comparing with standard reactive and recurrent baselines. We use 3 domains, each posing a qualitatively different type of representation learning problem: *Shopping-5* poses a more flexible task, and can be solved by a variety of irepresentations. *Heavenhell-3* poses a more rigid task, which can only be solved with mid-term memorization. *Rocksample-5-6* poses a significantly larger and mildly more stochastic task.

We compare the following irepresentations: **TrueBelief** is the belief-state representation, used as a reference for good performance. **React-{1,2,4}** are reactive irepresentations with associated memory parameters, trained on the RL objective. **GRU** is a recurrent irepresentation based on a GRU trained on the RL objective. **GRU-RAux** is a recurrent irepresentation based on a GRU trained with the RAux scheme. **GRU-CRAux** is a recurrent irepresentation based on two GRUs trained with the CRAux scheme. The RL task is solved using A2C, and the PRED task using the VAE model.

Results. Figure 2 shows selected results. To summarize, **GRU-RAux** and **GRU-CRAux** learn irepresentations which consistently outperform not only the baselines, but also the belief-state itself. This implies that the irepresentations learned through the PRED task have richer structure which can be exploited by the policy.

REFERENCES

[1] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. 2017. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252* (2017).

[2] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. 2017. Variational inference: A review for statisticians. *J. Amer. Statist. Assoc.* 112, 518 (2017), 859–877.

[3] Anthony R. Cassandra, Leslie Pack Kaelbling, and Michael L. Littman. 1994. Acting optimally in partially observable stochastic domains. In *AAAI*, Vol. 94. 1023–1028.

[4] Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* (2016).

[5] Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. 2004. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research* 5, Nov (2004), 1471–1530.

[6] Matthew Hausknecht and Peter Stone. 2015. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*.

[7] Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. 2019. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3796–3803.

[8] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *The Journal of Machine Learning Research* 14, 1 (2013), 1303–1347.

[9] Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. 2018. Deep Variational Reinforcement Learning for POMDPs. *arXiv preprint arXiv:1806.02426* (2018).

[10] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. 2016. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397* (2016).

[11] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101, 1 (1998), 99–134. http://www.sciencedirect.com/science/article/pii/S000437029800023X

[12] Guillaume Lample and Devendra Singh Chaplot. 2017. Playing FPS games with deep reinforcement learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[13] Michael L. Littman and Richard S. Sutton. 2002. Predictive representations of state. In *Advances in neural information processing systems*. 1555–1561.

[14] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J. Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, and Koray Kavukcuoglu. 2016. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673* (2016).

[15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, and Georg Ostrovski. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[17] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 16–17.

[18] Evan Shelhamer, Parsa Mahmoudieh, Max Argus, and Trevor Darrell. 2016. Loss is its own reward: Self-supervision for reinforcement learning. *arXiv preprint arXiv:1612.07307* (2016).

[19] Satinder Singh, Michael R. James, and Matthew R. Rudary. 2004. Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 512–519.

[20] Satinder P. Singh, Tommi Jaakkola, and Michael I. Jordan. 1994. Learning without state-estimation in partially observable Markovian decision processes. In *Machine Learning Proceedings 1994*. Elsevier, 284–292.

[21] Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*. 1057–1063.

[22] Daan Wierstra, Alexander Förster, Jan Peters, and Jürgen Schmidhuber. 2010. Recurrent policy gradients. *Logic Journal of the IGPL* 18, 5 (2010), 620–634.