Imperial College
London

Exploring state representations for offline RL

Thesis Presentation

Aimilios Hatzistamou

Presentation outline

Motivation & Problem statement

Thesis

With both abstractions and the offline RL setting showing strong promise, the focus of this Master's thesis will be to **evaluate whether offline RL can benefit from explicit representation models**—a question that is still largely unexplored to this day.

Background

State Abstraction

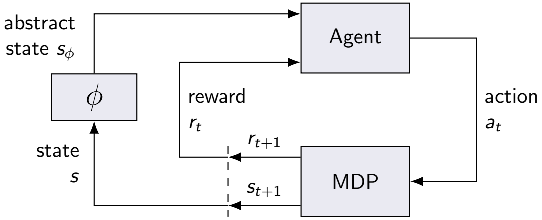


Figure 1: Agent-MDP interaction with state abstraction. Every timestep, the MDP produces a state s and the agent learns from $s_\phi = \phi(s)$.

A good abstraction satisfies (Abel, 2019):

- Efficient creation
- Efficient decision-making
- Near optimality

State Abstraction

Bisimulation Relations and Metrics

Bisimulation groups together states that are indistinguishable in terms of rewards over all possible action sequences tested—the **bisimulation relations** (Givan et al., 2003)

$$\phi(s_1) = \phi(s_2) \implies \begin{cases} r(s_1, a) = \quad r(s_2, a) & \forall a \in \mathcal{A} \\ P(G|s_1, a) = P(G|s_2, a) & \forall G \in \mathcal{S}_B,\ a \in \mathcal{A} \end{cases}$$

where $P(G|s, a) = \sum_{s' \in G} P(s'|s, a)$ and S_B is the set of all groups G of equivalent states under abstraction ϕ.

Bisimulation metrics (Ferns and Precup, 2014) offer a way to quantify the "behavioral similarity" between states:

$$d(s_i, s_j) = \max_{a \in \mathcal{A}} (1 - c)|r(s_i, a) - r(s_j, a)| + c \cdot W_1(P^a_{s_i}, P^a_{s_j}, d) \quad (1)$$

where $c \in [0, 1)$, and W_1 is the Wasserstein-1 distance.

State Abstraction

State representation learning

Common approaches to learning a state representation:

- Auto-Encoders.
- Forward models.
- Inverse models.
- Exploiting rewards.
- Prior knowledge.

In practice, used as auxiliary tasks (additional training objectives).

Intuition: learning to estimate quantities that are relevant to solving the main RL problem over a shared representation will speed up the progress on the main RL task.

Offline RL

Overview

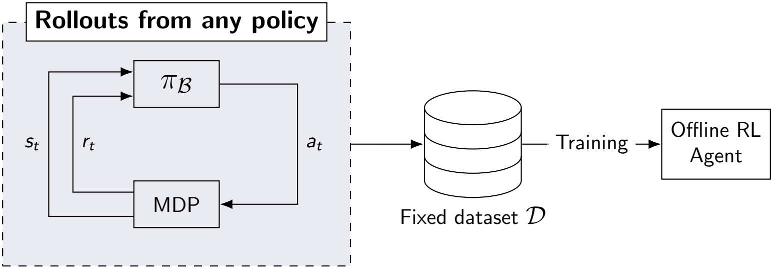


Figure 1: In the offline setting, an agent learns from a fixed dataset containing historical decisions and outcomes. The dataset may contain examples of both desirable and undesirable behaviour, and the policy(ies) that generated the data is typically unknown.

Algorithms & Representations

Algorithms
Behavioral Cloning

Agent learns a policy $\pi_\theta(a_t|s_t)$ mapping states to actions.

Objective function:

Negative log-likelihood of the selected action being from the policy $\pi_\mathcal{B}$ being cloned, i.e: minimizing $-\log p(a_t = \pi_\mathcal{B}(s_t)|s_t)$ across the whole dataset.

DQN uses Q-learning algorithm to estimate optimal action-values

$$Q^*(s, a) = \mathbb{E}_{\pi^*}[R_t | s_t = s, a_t = a]$$

Optimal policy is then constructed as: $\pi^*(s) = \arg\max_a Q^*(s, a)$.

$$\text{DQN loss} = L_\delta(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t))$$

Our implementation uses several common improvements:

- Experience replay (replay buffer = the offline dataset)
- Target Q-network
- N-step bootstrapping (offline data w/ 5-step transitions)
- Double Q-learning (Hasselt, 2010)

Representations

Reward vs Reconstruction

Reconstruction:

- Rich training signal.
- Not aligned with RL problem.

Reward:

- Sparser signal.
- RL-informed: can discard irrelevant information.

Representations

SRL Methods

	Training Signal		Model Architecture		Latent State	
	Recons.	Reward	Forward	Multi-Step	Stoch.	Discrete
PCA	✓					
AE	✓					
VAE	✓				✓	
VQ-VAE,	✓				✓	✓
DBC		✓				
DeepMDP		✓	✓			
VPN		✓	✓	✓		
World Models	✓		✓	✓	✓	

Table 1: Classification of the SRL models we considered for our experiments. Our selected methods are indicated in bold.

Representations

DeepMDP, Gelada et al. (2019)

Minimize two tractable losses: reward predictions and prediction of the distribution over next latent states.

$$J(\phi) = L_r + \alpha L_t = (r_{t+1} - \hat{r}_{t+1})^2 + \alpha \| s_{\phi,t+1} - \hat{s}_{\phi,t+1} \|_2$$

where α is a weighting factor, $\hat{r}_{t+1} = f(s_t, a_t)$ and $\hat{s}_{\phi,t+1} = g(s_t, a_t)$. f and g are neural network function approximators and we use a deterministic transition model for simplicity.

Representations

Deep Bisimulation for Control (DBC), Zhang et al. (2020)

$$J(\phi) = \left(\|s_{\phi,i} - s_{\phi,j}\|_1 - |r_i - r_j| - \gamma W_2(\hat{\mathcal{P}}(\cdot|\bar{s}_{\phi,i}, a_i), \hat{\mathcal{P}}(\cdot|\bar{s}_{\phi,j}, a_j)) \right)^2$$

where r are rewards, and $\bar{s}_\phi$ denotes $\phi(s)$ without gradient propagation.

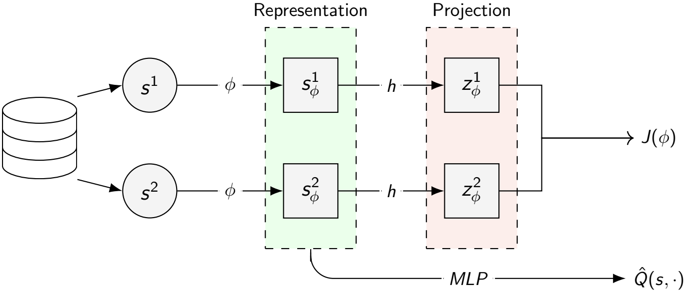


Figure 2: Architecture for learning DBC with an offline DQN.

Representations

Proposed: Contrastive DBC

Problem: DBC is *unstable* offline. We proposed learning the bisimulation metric using contrastive metric embeddings (CME, Agarwal et al. 2021).

Algorithm 1: Learning bisimulation metric with CMEs

Given: State embedding $\phi(\cdot)$, Metric $d(\cdot, \cdot)$, Dataset $\mathcal{D}$ and hyperparameters: temperature $1/\lambda$, Scale β, Total training steps K.;

for *step in k=1...K* **do**

  Sample a pair of batches $B_i \sim \mathcal{D}$, $B_j \sim \mathcal{D}$;

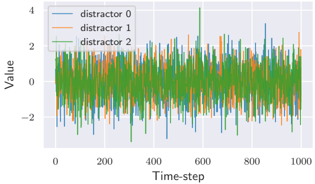
  Update the weights of ϕ to minimize $\mathcal{L}_{CME}$ where $\mathcal{L}_{CME} = \mathbb{E}_{B_i, B_j \sim \mathcal{D}}[L_\phi(B_i, B_j)]$

end

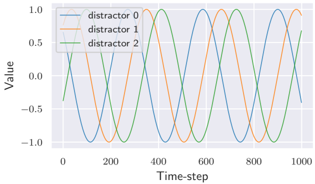
State representation in offline RL

Classic control (Cartpole) with distractions

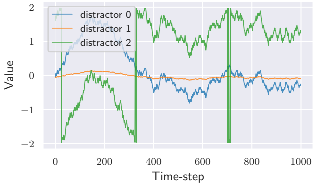
Distractor types



(a) Gaussian



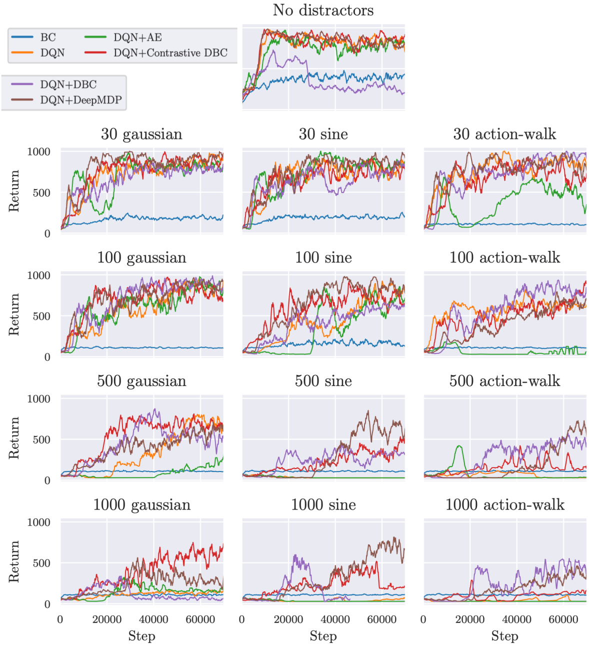
(b) Sine



(c) Action-walk

Figure 2: Example distractors plotted over 1k time-steps.

Results



BC
DQN
DQN+AE
DQN+Contrastive DBC
DQN+DBC
DQN+DeepMDP
No distractors
30 gaussian
30 sine
30 action-walk
100 gaussian
100 sine
100 action-walk
500 gaussian
500 sine
500 action-walk
1000 gaussian
1000 sine
1000 action-walk
Return
1000
500
0
0
20000
40000
60000
Step

Results

Summary

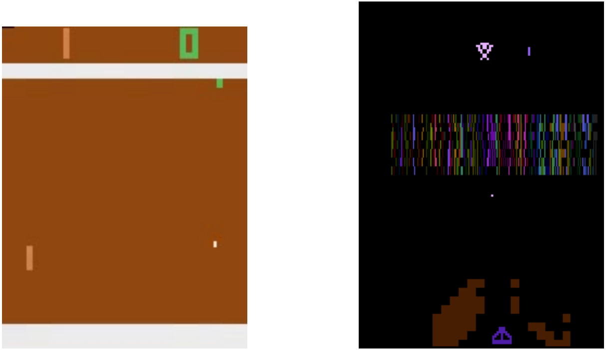
Main takeaways:

- Explicit representations can help offline learners discover significantly better policies.
- Reward-informed representations outperformed reconstruction and baselines.
- Contrastive DBC, more stable and outperforms DBC in several settings.
- High distractions, no reward $\longrightarrow$ convergence failures.

Large-scale tasks

Large-scale tasks

Atari Unplugged (Gulcehre et al., 2021)



(a) Pong (b) Yars Revenge

Figure 3: Sample frames from the two Atari games we pick. We use datasets of transitions with sticky actions, 4 stacked frames. 500M transitions per game, 375GB and 1.5TB for Pong and Yars respectively.

Large-scale tasks

Atari results

Without tuning $\longrightarrow$ no clear benefit of using representations on these tasks.

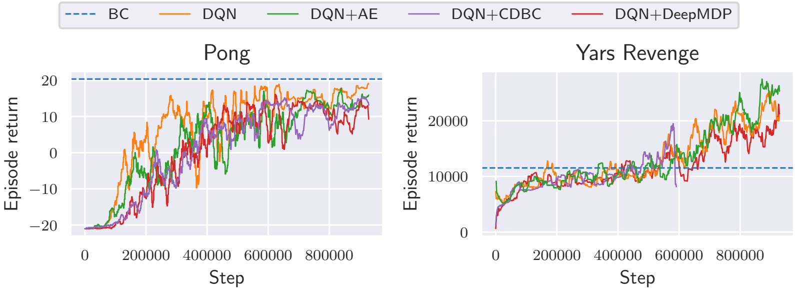


Figure 3: Training offline agents with different representation objectives on offline Atari data. No hyperparameters were tuned due to time/compute constraints.

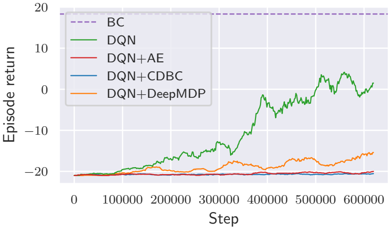
Large-scale tasks

Extra distractions

Figure 3: Example Pong frames with background substitution. Approach inspired by work of Zhang et al. (2020) on DeepMind Control Suite.

Large-scale tasks

Extra distractions



(a) Offline training of different SRL methods on Pong with background substitution.

(b) Render of DQN evaluation

Large-scale tasks

Illustrating tuning sensitivity

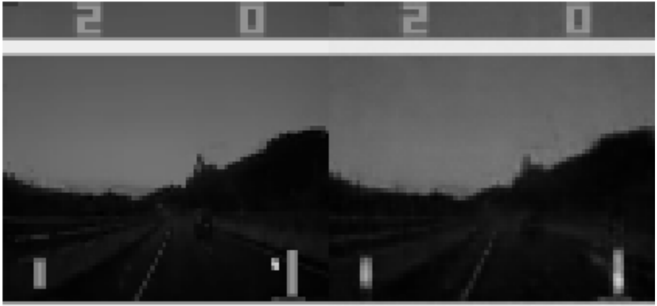


Figure 3: **Left**: a frame captured during online evaluation. **Right**: its reconstruction by the autoencoder. Despite the propagation of gradients from DQN to encoder, the jointly-trained representation is unable to capture the location of the ball.

Conclusion

Conclusion

Contributions

1. **Evaluation of several SRL methods on offline tasks**. Our empirical evidence suggest that introducing a jointly-trained representation loss can improve the performance of offline policy learning algorithms, but is sensitive to tuning.
2. **Distractor benchmark**. We proposed a benchmark environment to evaluate the robustness of different SRL methods against distractions in the observation space
3. **Contrastive DBC**. Contrastive loss to embed the bisimulation metric, effectively grouping states that are behaviorally equivalent. Better experimental properties than DBC.
4. **Insights and discussions**

Conclusion

Future Work

1. **Different SRL objectives**. Explore other classes of methods (e.g. sequence models).
2. **Real world**. Extend our results to real-world domains (e.g. robotic manipulation).
3. **Hyperparameters**. Which representations are most sensitive to the exact values of hyperparameters, and what techniques can render training more robust?
4. **Offline RL algorithms**. Evaluate SRL with other algorithms like CQL or BRAC.

Questions?

References I

Abel, D.
2019. A Theory of State Abstraction for Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9876–9877.

Agarwal, R., M. C. Machado, P. S. Castro, and M. G. Bellemare
2021. Contrastive Behavioral Similarity Embeddings for Generalization in Reinforcement Learning. *arXiv:2101.05265 [cs, stat]*.

Ferns, N. and D. Precup
2014. Bisimulation Metrics are Optimal Value Functions. *Uncertainty in Artificial Intelligence - Proceedings of the 30th Conference, UAI 2014*.

References II

Gelada, C., S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare
2019. DeepMDP: Learning Continuous Latent Space Models for Representation Learning. *arXiv:1906.02736 [cs, stat]*.

Givan, R., T. Dean, and M. Greig
2003. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 147(1).

Gulcehre, C., Z. Wang, A. Novikov, T. L. Paine, S. G. Colmenarejo, K. Zolna, R. Agarwal, J. Merel, D. Mankowitz, C. Paduraru, G. Dulac-Arnold, J. Li, M. Norouzi, M. Hoffman, O. Nachum, G. Tucker, N. Heess, and N. de Freitas
2021. RL Unplugged: A Suite of Benchmarks for Offline Reinforcement Learning. *arXiv:2006.13888 [cs, stat]*. arXiv: 2006.13888.

References III

Hasselt, H.
2010. Double Q-learning. In *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, eds., volume 23. Curran Associates, Inc.

Zhang, A., R. McAllister, R. Calandra, Y. Gal, and S. Levine
2020. Learning Invariant Representations for Reinforcement Learning without Reconstruction. *arXiv:2006.10742 [cs, stat]*.