Learning World Value Functions without Exploration

Ebenezer Gelo [*1] **Steven James** [1] **Benjamin Rosman** [1]

Abstract

We explore the application of offline Reinforcement Learning (RL), specifically focusing on learning a goal-oriented knowledge representation framework called World Value Function (WVF). We benchmark the performance of selected offline RL algorithms, including offline Deep Q-Network (DQN) and Batch-Constrained deep Q-learning (BCQ), under varying data buffer sizes. Notably, these selected algorithms were modified to learn goal-oriented value functions. Using a 2D video game and a robotic environment, our experiments span discrete and continuous action domains. The success rates of learned WVF using these algorithms over varying replay data show valuable insights into the efficiency of these algorithms under different conditions and domains, highlighting the significance of possessing a large and diverse dataset for learning WVF in a batch setting.

1. Introduction

Reinforcement Learning(RL) is a machine learning approach that has drawn considerable interest in the artificial intelligence (AI) domain (Sutton & Barto, 2018). RL allows an agent to learn and make choices by performing actions in an environment to maximize an accumulated reward signal. RL has been widely used in different areas such as gaming, finance, robotics, and recommendation systems, among others (Mnih et al., 2015; Silver et al., 2016; Wang et al., 2016).

Although the online RL approach, where the agent interacts with the environment to learn the best actions, is commonly used in RL, it can be impractical or expensive in some situations. To address these challenges, offline RL, also called batch RL, has emerged as a promising alternative (Fujimoto et al., 2019a; Levine et al., 2020; Fujimoto et al., 2019b). This technique involves learning from a batch of datasets without further interacting with the environment.

There has been a growing interest in learning a particular type of goal-oriented value function in the context of compositional RL (Tasse et al., 2022a; 2020; van Niekerk et al., 2018). World Value Function(WVF) captures the value of achieving specific goals in the environment and can be useful for solving complex tasks that require reasoning about long-term goals (Tasse et al., 2022b). However, most of the existing research on learning WVF has focused on online RL settings, where the agent learns from real-time interactions with the environment.

In this work, we seek to show how WVF can be learned without exploration. Our primary focus is to benchmark the performance of selected offline RL algorithms under varying data buffer sizes. We employed offline Deep Q-network (DQN) and discrete Batch-Constrained deep Q-learning (BCQ) in a 2D video game domain with discrete action space called Boxman (van Niekerk et al., 2018; Tasse et al., 2020). We also extend our exploration to the continuous domain using continuous BCQ for WVF in a robotic environment provided by Panda-Gym (Gallouédec et al., 2021).

This approach allowed us to assess the impact of the amount of data in the buffer on the performance of each algorithm when learning WVF. The results from these comparisons provide valuable insights into the efficiency of these algorithms under different conditions and domains.

2. Background

We consider the RL problem that involves an agent making decisions within the environment, getting feedback through rewards, and adjusting its actions based on the outcomes it observes. The standard RL problem is often called a Markov Decision Process (MDP). An MDP is defined by a tuple (S, A, P, R, γ), where: S is the set of possible states in the environment, A represents the set of possible actions an agent can take, $P : S \times A \times S \to [0, 1]$ is the function that describes the probability of transitioning from one state to another after taking an action, $R : S \times A \to \mathbb{R}$ is the reward function that maps the agent's actions and state transitions to a numerical reward signal and $\gamma \in [0, 1]$ is the discount factor determining how much an agent values future rewards compared to immediate rewards.

The goal of RL is to find a policy $\pi : S \to A$ that maximizes the expected cumulative discounted reward:

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \tag{1}$$

Where s_t is the state, a_t is the action at time t, and the expectation is over the distribution of trajectories induced by the policy π and the MDP. This is often achieved by learn-

ing a value function. Once the value function converges, the optimal policy can be obtained by choosing the action that maximizes the expected value at each state. This is expressed as: $\pi^*(s) = \arg\max_a Q^*(s,a)$

2.1. Offline Reinforcement Learning

In contrast to the traditional RL problem, which involves learning from interactions with the environment, Offline RL aims to learn an optimal control policy from a fixed dataset without further interactions with the system. This can be formally defined as:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] \tag{2}$$

The expectation is taken over the trajectories sampled from dataset $\mathcal{D}$. The policy π is restricted to the actions and states in the dataset.

These datasets can be collected from various sources, such as demonstrations, human expert data, or logged data from previous interactions with the environment. Offline RL offers several advantages, including accelerating learning and improving sample efficiency, as the agent can learn from diverse and rich data collected over time (Levine et al., 2020; Fujimoto et al., 2019b).

2.2. World Value Functions

World Value Function(WVF), a framework presented by (Tasse et al., 2022b), addresses the challenge of creating a general AI capable of learning and representing knowledge for solving multiple tasks in a given world. WVF is a type of general value function that represents how to solve any goal-reaching task.

(Tasse et al., 2022b) build on existing literature in the field of AI and knowledge representation. Previous research has focused on developing methods for creating general value functions (GVFs) that can be used to learn and represent knowledge in a way that allows for solving multiple tasks (Sutton et al., 2011). GVFs provide a way to generalize across tasks and transfer knowledge from one task to another, but they have limitations in terms of deducing the transition function and producing novel behaviors.

In contrast, WVF offers several advantages. One key benefit of WVF is its ability to reuse existing knowledge to solve new tasks through logical composition. By defining its reward function, the agent can modify task rewards based on its internal goal space, which allows it to leverage its previously learned knowledge to adapt to new tasks. This allows the agent to solve any task in the world and possess *mastery* (Veeriah et al., 2018).

3. Offline RL for WVF

For both DQN and BCQ, we modified the Q-learning update rule to incorporate the WVF. This involves using the WVF to compute each state's value function to update the Q-values based on the expected cumulative reward of achieving the agent's internal goals. We extend the Q-learning algorithm given by (Tasse et al., 2022b) into an offline DQN that strictly learns from the experience replay buffer (Agarwal et al., 2019).

Algorithm 1 Offline DQN for WVF

Initialize: WVF $\bar{Q}$ with random weights θ, target network $\bar{Q}'$ with weights $\theta' = \theta$, goal buffer $\mathcal{G}$ replay buffer $\mathcal{D}$, learning rate α, batch size N
for each timestep **do**
  Sample minibatch of transitions (s, g, a, r, s') of size N from $\mathcal{D}$
  for each (s, g, a, r, s') in minibatch **do**
    if s' is absorbing **then**
      $\mathcal{G} \leftarrow \mathcal{G} \cup \{s'\}$
    end if
    for each $g' \in \mathcal{G}$ **do**
      $\bar{r} \leftarrow R_{\text{MIN}}$ if $g' \neq s'$ and $s' \in \mathcal{G}$ else r
      $\delta \leftarrow \left[\bar{r} + \max_{a'} \bar{Q}(s', g', a'; \theta')\right] - \bar{Q}(s, g', a; \theta)$
      Perform a gradient descent step on $(\delta)^2$ with respect to the network parameters θ
    end for
    Every C steps update $\bar{Q}' = \bar{Q}$
  end for
end for

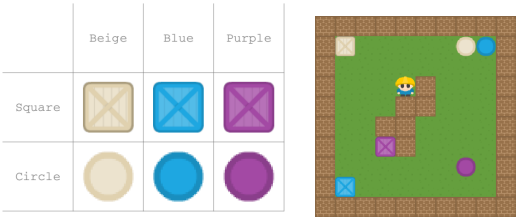
We used a similar approach for BCQ, but the policy is constrained to improve upon a behavior policy that is close to the data collection policy, as in (Fujimoto et al., 2019b). BCQ does this by incorporating a generative model, specifically a Variational Auto-Encoder (VAE), to generate actions likely from the batch of the dataset. It also includes a perturbation model, which further modifies the action. However, the discrete version of BCQ selects the action with the highest Q-value, subject to the constraint that the ratio of its probability under the generative model to the maximum probability is above a certain threshold (Fujimoto et al., 2019a). This is to ensure the chosen action must not only have a high Q-value but also be likely under the generative model.

The rationale for investigating these algorithms for learning WVF lies in the potential benefits of solely leveraging historical data to accelerate learning and improve sample efficiency. Offline RL can effectively utilize large-scale datasets often available in practical applications to learn value functions. This can be particularly advantageous in scenarios where online data collection is limited.

4. Experiments

4.1. Discrete Domain

To show that we can learn WVF in a batch setting, we used a video game environment called Boxman presented by (van Niekerk et al., 2018). In this domain, the agent gathers different shaped and colored objects, starting from any given location. The state of the environment is represented as an 84×84 RGB image. The agent can traverse in all four cardinal directions. Additionally, it has a pick-up function that enables it to collect an object when it is directly above it.



(a) (b)

Figure 1. The base tasks (a) and a layout (b) of the environment we used for this experiment (van Niekerk et al., 2018).

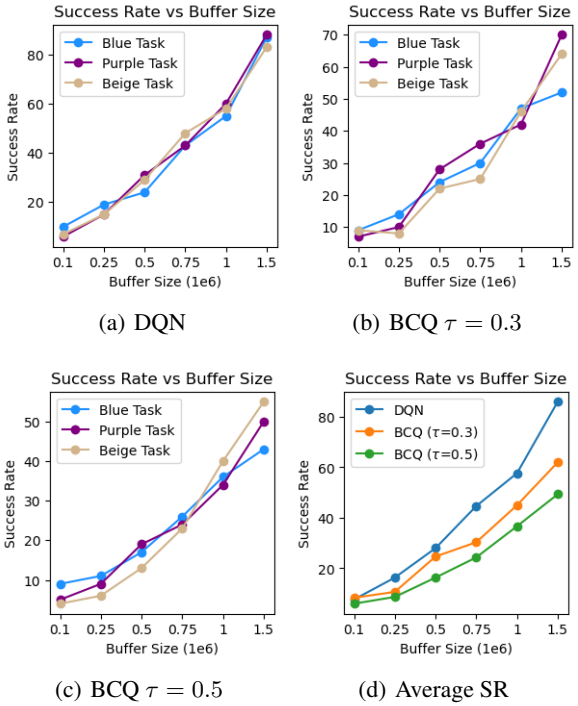
For each color, we trained a separate DQN. The reward function for each DQN was designed to incentivize the model to collect its corresponding object successfully.

After training each DQN, we consolidated the replay buffers generated from all agents to form a comprehensive dataset. This dataset encapsulated the trajectories for reaching each goal. This is important since WVF is a goal-oriented knowledge representation framework. The generated dataset was the foundation for training our offline DQN algorithm and the discrete BCQ.

We conducted training on the offline DQN and discrete BCQ using various dataset sizes to assess their performance under different data sizes. We used the same number of steps for each of the segmented datasets. This approach allowed us to observe how the performance varies with changes in dataset size.

In the context of discrete BCQ, setting the threshold parameter, τ, to 0 results in Q-learning, while setting τ to 1 yields an imitation of the actions present in the dataset (Fujimoto et al., 2019a). Given this behavior, we also compared how it performs under different thresholds - $\tau = 0.3$ and $\tau = 0.5$.

We defined a success rate to evaluate the trained algorithms, which is determined by counting the number of times the specified goal condition is met within a span of 100 episodes following the learned WVF. In other words, a counter is increased each time the desired outcome is achieved. The final count after 100 episodes gives us the success rate. This allowed us to quantitatively compare how the two algorithms perform given varying numbers of data.



(a) DQN (b) BCQ $\tau = 0.3$

(c) BCQ $\tau = 0.5$ (d) Average SR

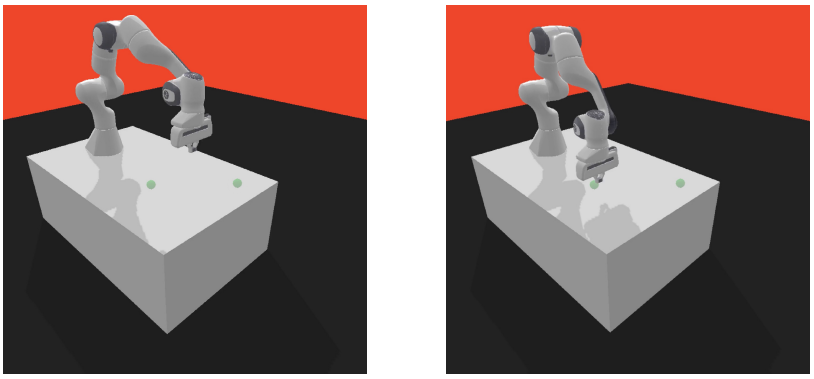
Figure 2. (a), (b), and (c) show the success rate for each goal condition averaged over three runs. (d) compares the average success rate for the three approaches we used in the experiment.

From (d), we can see that offline DQN performs better than BCQ, which can be attributed to BCQ's Q-learning update rule that handles the constraints of batch learning (Fujimoto et al., 2019b). We believe that because the offline DQN does not have this constraint, it can learn better than BCQ given the combined replay buffer. We, therefore, conclude that offline DQN is more suitable for learning WVF in a discrete domain without exploration.

4.2. Continous Domain

We extended our experiment to a continuous domain using an environment provided by Panda-Gym. Panda-Gym is a robotic environment that is based on the Franka Emika Panda robot. We created a custom task by extending the `PandaReach_v3` task (Gallouédec et al., 2021), adding another reachable goal region, thus transforming the domain into a multi-goal one. The targets were randomly generated within a volume of 30^3cm. The reward is sparse for this experiment, and the control mode is the end-effector displacement.

For each target task, we trained a separate Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015). We then used the learned behavioral policy to generate buffers



(a) Start State (b) Terminal State

Figure 3. The figure presented illustrates the application of learned WVF. The learned WVF encapsulates all tasks related to reaching the agent's internal goals (Tasse et al., 2022b). Therefore, the robot depicted is capable of conclusively accomplishing both target tasks shown in green.

of varying sizes. Following this, we then train a continuous BCQ as in (Fujimoto et al., 2019b) on the combined buffer to learn the WVF. In this experiment, we only compare the performance of the continuous BCQ against varying numbers of data. We use the same evaluation as in the discrete domain, where BCQs are trained for the same number of steps but using different data sizes, and success rates are computed using learned WVF over 100 episodes averaged over three runs.

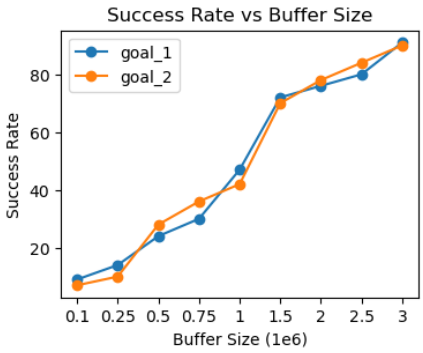


Figure 4. Success rate for both target tasks over varying replay data sizes.

From the results shown in (*Figure* 4), we can learn that while continuous BCQ has mechanisms to handle issues like extrapolation error and overestimation bias, the effectiveness of these mechanisms is greatly enhanced with more data. Therefore, BCQ requires large data when learning WVF for optimal performance.

5. Conclusion

In this study, we have demonstrated the capability to learn WVF without the necessity for exploration in both discrete and continuous domains. This was achieved by employing offline DQN and discrete BCQ for the discrete domain and continuous BCQ for the continuous domain. Our findings underscore the significance of having ample data when working with offline RL algorithms. The results suggest that the performance of these algorithms can be substantially improved with the availability of more data.

Looking ahead, there are several promising directions for future research. One potential area of focus could be the development of more efficient data collection strategies to improve the performance of many other offline RL algorithms further. Additionally, research could explore methods for better handling distributional shifts and overestimation bias, particularly in scenarios with multiple tasks.

6. Acknowledgments

Thank you to my supervisors, Dr. Steven James and Prof. Benjamin Rosman, for their invaluable guidance, advice, and support throughout the research.

Thank you to Geraud Nangue Tasse for supplying the foundational implementations and patiently addressing all my questions about the framework.

The computations in this work were carried out using the High-Performance Computing Infrastructure (MSL Cluster), which is maintained by the Mathematical Sciences Support Unit at the University of the Witwatersrand.

References

Agarwal, R., Schuurmans, D., and Norouzi, M. Striving for simplicity in off-policy deep reinforcement learning. *CoRR*, abs/1907.04543, 2019. URL http://arxiv.org/abs/1907.04543.

Fujimoto, S., Conti, E., Ghavamzadeh, M., and Pineau, J. Benchmarking batch deep reinforcement learning algorithms, 2019a.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration, 2019b.

Gallouédec, Q., Cazin, N., Dellandréa, E., and Chen, L. panda-gym: Open-Source Goal-Conditioned Environments for Robotic Learning. *4th Robot Learning Workshop: Self-Supervised and Lifelong Learning at NeurIPS*, 2021.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020.

Lillicrap, T., Hunt, J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *CoRR*, 09 2015.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, February 2015. ISSN 00280836.

Prudencio, R. F., Maximo, M. R. O. A., and Colombini, E. L. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–0, 2023. doi: 10.1109/tnnls.2023.3250269. URL https://doi.org/10.1109%2Ftnnls.2023.3250269.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. doi: 10.1038/nature16961.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., and Precup, D. Horde: a scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In Sonenberg, L., Stone, P., Tumer, K., and Yolum, P. (eds.), *AAMAS*, pp. 761–768. IFAAMAS, 2011. ISBN 978-0-9826571-5-7. URL http://dblp.uni-trier.de/db/conf/atal/aamas2011.html#SuttonMDDPWP11.

Tasse, G. N., James, S., and Rosman, B. A boolean task algebra for reinforcement learning, 2020.

Tasse, G. N., James, S., and Rosman, B. Generalisation in lifelong reinforcement learning through logical composition. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=ZOcX-eybqoL.

Tasse, G. N., Rosman, B., and James, S. World value functions: Knowledge representation for learning and planning, 2022b.

van Niekerk, B., James, S., Earle, A., and Rosman, B. Will it blend? composing value functions in reinforcement learning, 2018.

Veeriah, V., Oh, J., and Singh, S. Many-goals reinforcement learning, 2018.

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., and de Freitas, N. Dueling network architectures for deep reinforcement learning, 2016.

Appendix

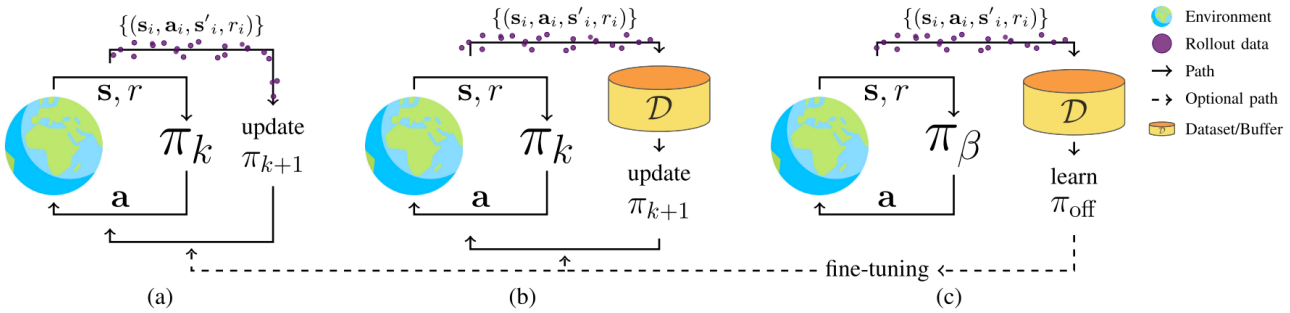


Figure 5. Online RL (a), off-policy RL (b), and offline RL (c) illustrated by (Prudencio et al., 2023). Online RL collects new experiences with the current policy, while off-policy RL reuses previous experiences and gathers new ones. Offline RL solely uses past experiences from a fixed dataset to learn a policy. The learned policy can be further refined with online or off-policy RL techniques.