Safe Evaluation For Offline Learning: Are We Ready To Deploy?

Hager Radi
University of Alberta
Alberta, Canada
radi@ualberta.ca

Josiah P. Hanna
University of Wisconsin – Madison
Wisconsin, USA
jphanna@cs.wisc.edu

Peter Stone
The University of Texas at Austin &
Sony AI
Texas, USA
pstone@cs.utexas.edu

Matthew E. Taylor
University of Alberta &
Alberta Machine Intelligence Institute
Alberta, Canada
matthew.e.taylor@ualberta.ca

Abstract

The world currently offers an abundance of data in multiple domains, from which we can learn reinforcement learning (RL) policies without further interaction with the environment. RL agents learning offline from such data is possible but deploying them while learning might be dangerous in domains where safety is critical. Therefore, it is essential to find a way to estimate how a newly-learned agent will perform if deployed in the target environment before actually deploying it and without the risk of overestimating its true performance. To achieve this, we introduce a framework for safe evaluation of offline learning using approximate high-confidence off-policy evaluation (HCOPE) to estimate the performance of offline policies during learning. In our setting, we assume a source of data, which we split into a train-set, to learn an offline policy, and a test-set, to estimate a lower-bound on the offline policy using off-policy evaluation with bootstrapping. A lower-bound estimate tells us how good a newly-learned target policy would perform before it is deployed in the real environment, and therefore allows us to decide when to deploy our learned policy.

1 Introduction

Suppose someone else is controlling a sequential decision making task for you. This could be a person trading stocks for you, a hand-coded controller for a chemical plant, or even a PID for temperature regulation. Off-policy reinforcement learning allows us to learn policies from fixed data. But when would you want to switch from the existing controller to your learned policy? This decision may depend on the cost for continued data collection from the existing controller (e.g., you pay someone to manage your stock portfolio), your risk appetite, and your confidence in the performance of the policy you have learned. This paper takes a critical first step towards the last question:

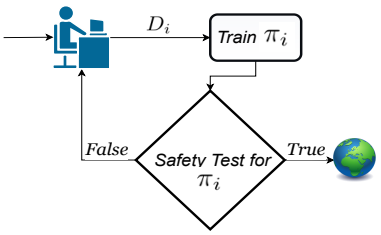

Figure 1: A framework for continual safety-evaluation of offline learning

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

how to learn off-policy and simultaneously evaluate that policy with confidence when we have no access to the environment nor the policy generating the data?

Offline reinforcement learning is a way of training off-policy algorithms offline using already existing data, generated by humans or other controllers. Learning with offline data is challenging because of the distribution mismatch between data collected by the behavior agent and the offline agent (Levine et al. 2020). What is even more challenging is evaluating offline agents in the offline RL setting if we assume no access to the environment; in some domains, we cannot execute our learned policy until it is good enough. This limitation raises the question about the possibility of using off-policy policy evaluation (OPE) methods, where we can estimate the value of a policy using trajectories from another policy, to predict what the performance of the offline agent is, at any point of time during learning. Further, we investigate if we can use high-confidence OPE (HCOPE) to control the risk of overestimating the policy's performance.

We present a framework for safe evaluation of RL agents learning offline. *Safe* refers to defining a lower bound to our policy estimates, a point where we think the new policy is worse than it is in practice, which is important for safety-critical applications. The paper tackles the setup where we combine offline policy improvement to learn a policy from existing data, and off-policy policy evaluation with bootstrapping confidence intervals to provide a lower confidence bound estimate of such a policy, simultaneously. We believe the setup we are tackling is under-studied in the literature. We are aiming to learn a target policy purely out of a data buffer, and evaluate its performance while learning so that we can tell when it is ready for deployment. Our framework can be summarized in Figure 1: we have a source of data, from which we utilize samples. At each step, we split the data into training and testing. After each training step, we test the policy using approximate [1] high-confidence off-policy evaluation (HCOPE). We continue the process of training/testing for a few iterations until the testing shows the policy can outperform the data with a confidence level δ. We dynamically receive samples, continuously performs RL updates, and continuously monitors a confidence interval on the changing policy until it reaches a sufficient level. Our **contributions** are:

- A framework combining offline RL and approximate high-confidence OPE.
- Investigating the feasibility of HCOPE methods given constantly-improving target policy (as opposed to a fixed policy previously studied in the literature)

2 Motivation

Offline RL offers an opportunity for learning data-driven policies without environment interaction. In safety-critical applications found in healthcare or autonomous driving, there are plenty of data that we can use to learn RL policies and hence use for decision making (Gottesman et al. 2019). Also, there are domains where data efficiency is essential as the data collection process is either expensive or dangerous. If we want to learn policies in such domains, we need to find a way to tell how good the performance of a policy is before actually deploying it to the real-world environment. However, the execution of a new policy can be costly or dangerous if it performs worse than the policy that is currently being used. Hence, we focus on safety when evaluating offline agents such that the probability that the performance of our agent below a baseline is at most δ, where δ specifies how much risk is reasonable for a certain domain. This will allow us to know when we can trust a policy to take control in the real world. If we are given access to trajectories generated by an unknown behavior policy, we assume an iterative setting where at each iteration we can either request another batch of trajectories from the source or deploy our own policy. Our objective is to only deploy our own policy if a $1 - \delta$ lower bound on its expected return is greater than the expected return of the unknown behavior policy.

3 Related work

The current work intersects with the literature of two camps, which are *offline RL* and *batch RL*. *Offline RL* is about improving a policy from historical data for control, not just evaluation. A survey

[1] We refer to HCOPE as approximate because bootstrapping lower bounds may have error rates larger than δ but they provide a practical alternative to guaranteed bounds that are too loose to use.

paper (Levine et al. 2020) discusses how to categorize model-free offline RL methods into policy constraints that constrain the learned policy to be close to the behavior policy such as the work by Fujimoto, Meger, and Precup (2018), and uncertainty-based methods that attempt to estimate the epistemic uncertainty of Q-values to reduce distributional shift. Non-constrained methods include the traditional Q-learning (Watkins and Dayan 1992) or Soft-Actor Critic (Haarnoja et al. 2018) but they are not always reliable in the fully offline setup. Imitation learning, more specifically behavioral cloning (Bain and Sammut 1999), is another way for learning offline policies from historical data.

Batch RL refers to learning off-policy estimates from historical data, without environment interaction. It has an important property: given data generated by a behavior policy, it will estimate a new evaluation/target policy, guaranteed with high confidence that its performance is not worse than the behavior policy π_b. This camp includes three sub-directions. Off-policy policy evaluation *(OPE)* (Voloshin et al. 2019), high confidence off-policy policy evaluation *(HCOPE)* (Thomas, Theocharous, and Ghavamzadeh 2015a), and safe policy improvement *(SPI)* (Thomas, Theocharous, and Ghavamzadeh 2015b). An empirical study of OPE methods (Voloshin et al. 2019) discussed the applicability of each method and presented method selection guidelines depending on the environment parameters and the mismatch between π_θ and π_b. This study categorized OPE methods into importance sampling methods, directed methods, and hybrid methods that combine aspects from the two worlds. Importance Sampling (IS) (Precup, Sutton, and Singh 2000) is one of the widely used methods for off-policy evaluation where rewards are re-weighted by the ratio between π_θ and π_b. There are later versions such as Weighted Importance Sampling (Mahmood, van Hasselt, and Sutton 2014), Per-Decision Importance Sampling, and Per-Decision Weighted Importance Sampling (Precup, Sutton, and Singh 2000) that are biased, but offer lower-variance estimates. Then, we have the direct methods which rely on regression techniques to directly estimate the value function of the policy. This category includes model-based methods where the transition dynamics and reward are estimated from historical data via a model. Then, the off-policy estimate is computed with Monte-Carlo policy evaluation (Hanna, Stone, and Niekum 2017). Another direct method is Fitted Q-Evaluation (Le, Voloshin, and Yue 2019), which is a model-free approach and acts as the policy evaluation counter-part to batch Q-learning or FQI (Riedmiller 2005). The third category is the hybrid methods that combine different features from IS and direct methods. This category mainly involves doubly-robust methods (Jiang and Li 2015) that uses a direct model to lower the variance of IS.

Given the previous work, none of them discussed the feasibility of evaluating offline RL agents during learning, and how much we can trust high-confidence OPE as an approach for testing. The setup we are studying is quite different from the current literature because previous work assumed a behavior policy π_b and a target policy π_θ, where both policies are fixed and may be related. As an example, in a study for HCOPE methods by Thomas, Theocharous, and Ghavamzadeh (2015a), the target policy is initialised as a subset of the behavior policy such that they are close to each other. In another study for safe improvement (Thomas, Theocharous, and Ghavamzadeh 2015b), the Daedulus algorithm learns a safe target policy as a continuous improvement over the behavior policy. This is quite relevant to what we have here but Daedulus uses data from an older version of the policy it currently improves to perform the improvement step and safety tests. With the Doubly-robust estimator, authors present the results with different versions of π_θ such that π_θ is always a mixture of π_b and π_θ with different degrees (Jiang and Li 2015). To the best of our knowledge, none of the previous work showed how OPE or HCOPE methods would perform if the target policy is improved independently from the behavior policy, which is the case for offline learning.

4 Methodology

In this work, we present a framework to close the gap between offline RL and approximate high-confidence off-policy evaluation as a feasible evaluation method for offline agents in safety-critical domains. In our framework, we sample data dynamically at each iteration, perform policy updates, and continuously monitor its performance with a confidence interval till it reaches a good performance and is ready to be deployed. The policy we are learning offline does not interact with the environment unless it passes the safety test. Specifically, we have k iterations where in each iteration i, the current data size n is denoted as D_i. D_i is split between training our target policy $\pi_{\theta i}$ and performing the safety test using approximate high-confidence off-policy evaluation methods. We return when the learned policy is ready to be deployed with appropriate confidence. To summarize an interaction between an agent and a data source: 1) an agent requests a set of data D_i without knowledge of the

Algorithm 1 Offline Safe Evaluation Framework

Input: initial π_θ, dataset D of n trajectories, confidence level $\delta \in [0, 1]$, number of bootstrap estimates B, policy improvement method Φ, HCOPE method Ψ
Output: π_θ, $\hat{v_\delta}(\pi_\theta)$: $1 - \delta$ lower-bound on $v(\pi_\theta)$

Let $i = 0$
while $\hat{v_\delta}(\pi_\theta) \leq \hat{v}(\pi_b)$ **do**
  Request data-set of trajectories D_i with size n
  Split D_i into D_{train} and D_{test}
  Improve policy $\pi_{\theta_i} = \Phi(D_{train})$
  Evaluate policy $\hat{v_\delta}(\pi_\theta) = \Psi(D_{test})$
end while
return π_θ, $\hat{v_\delta}(\pi_\theta)$

behavior policy, 2) we improve a policy $\pi_{\theta i}$ offline using D_{train} with any offline RL algorithm, 3) we perform a safety check using D_{test} with high-confidence evaluation methods, and 4) our new policy is either ready to be deployed or we go back to step 1).
Instantiating our framework requires selecting an offline policy improvement method and a method for computing off-policy lower-bound estimates. For these two components, we study a variety of methods. We refer to each offline policy improvement method with Φ, and high-confidence off-policy evaluation method with Ψ. Our safe evaluation framework is further explained in Algorithm 1.

4.1 Policy improvement

We investigate different offline learning techniques such as naive off-policy methods (Category A), imitation learning (Category B), and policy constraint approaches specific for offline learning (Category C). We use Q-Learning (QL) in Category A, Behavioral Cloning (BC) in Category B, and Batched Constrained Q-Learning (BCQ) (Fujimoto et al. 2019) in Category C.

4.2 Approximate high-confidence off-policy evaluation with bootstrapping

We use off-policy estimators to evaluate a policy that is continuously learning. An off-policy estimator is a method for computing an estimate $\hat{v}(\pi_\theta)$ on the true value of the target policy $v(\pi_\theta)$ using trajectories D from another policy π_b, which is what Off-policy policy evaluation (OPE) methods do. For high-confidence OPE, this extends to lower-bounding the performance of the target policy π_θ where we combine OPE with bootstrapping confidence intervals. Bootstrapping is considered semi-safe since it requires the assumption that the bootstrap distribution can represent the statistic of interest which is not true for a finite sample, which in result makes our HCOPE approximate. However, it is safe enough as well as data efficient for high-risk domains like medical predictions (Morrison et al. 2007). When evaluating, we use new samples each time, to avoid the multiple comparisons problem[2]. This also ensures that we do not overfit our OPE estimates or tune training parameters to reduce the estimate error. The evaluation approximates a confidence lower bound of $\hat{v_\delta}(\pi_\theta)$ on $v(\pi_\theta)$ such that $\hat{v_\delta}(\pi_\theta) \leq v(\pi_\theta)$ with probability at least $1 - \delta$. There are 3 main categories of estimators as mentioned in Section 3; we use weighted importance sampling with bootstrapping (WIS), direct model-based bootstrapping (MB), and weighted doubly-robust with bootstrapping (WDR) to represent each of the 3 categories of off-policy estimators. Further details can be found in Appendix A.2.

5 Experiments

In our experiments, we focus on discrete control in the classic MountainCar (Sutton and Barto 2018) with a shortened horizon[3]. To mimic the setup of Figure 1, we simulate a data source with an online partially-trained actor-critic that collects data of a medium quality. To include exploratory

[2]The problem occurs when conducting multiple statistical tests simultaneously with reusing data so the confidence bound does not strictly hold anymore

[3]Further details in Appendix A.3

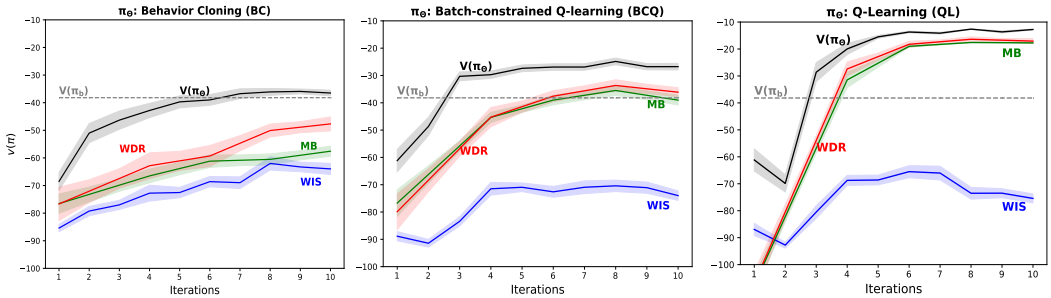

Figure 2: Results of safe evaluation using different offline improvement algorithms on MountainCar. QL and BCQ outperforms the data distribution while BC reaches π_b at most. WIS, MB, and WDR refer to the evaluation methods: weighted importance sampling, model-based, and weighted doubly-robust respectively.

behavior in data, an agent takes a random action 30% of the time instead of following the online policy when collecting trajectories. We improve a policy offline using 3 different improvement methods (BC, BCQ, and QL), and evaluate each policy simultaneously using 3 different off-policy evaluation methods (WIS, MB, and WDR). We follow Algorithm 1, but we limit the loop to maximum iterations of 10 because not all evaluation methods will be able to achieve the stopping condition. In each iteration, we sample 300 trajectories from the data source and split between training and testing such that training uses 20 trajectories only since evaluation requires more data for tighter bounds. For each improvement method, we pass training data and do m policy updates where m is tuned per algorithm such that the offline policy converges within the total number of iterations. For bootstrapping, we use ($\delta = 0.05$) to get a 95% confidence lower-bound using $B = 2000$ bootstrap estimates as recommended by practitioners (Efron 1979). For WDR only, we use a value of $B = 224$ to avoid the heavy computation; this can still get us a good approximation as suggested by MAGIC (Thomas and Brunskill 2016).

Results: Figure 2 shows how different offline policy improvement methods perform given medium-quality data along with HCOPE estimators, indicating which method can tell when $v(\pi_\theta) > v(\pi_b)$. Behavior cloning as an offline policy improvement method can only perform as well as the data by π_b while Q-learning and BCQ were able to outperform π_b. The true value of a target policy is calculated as the average return when running the policy in the actual environment (not possible in practice) for 1000 episodes. All reported results are average of 40 runs while the shaded area shows the standard error. The value of the behavioral policy $\hat{v}(\pi_b)$ is the sum of undiscounted rewards of the data set. Since π_b is not known, we first estimate $\hat{\pi_b}$ given the test data (Hanna, Niekum, and Stone 2021).

For the safe evaluation of each offline agent, weighted importance sampling (WIS) with bootstrapping failed to detect that the offline policy outperforms π_b for all improvement methods. The model-based estimator (MB) and weighted doubly-robust with bootstrapping (WDR) have much less error with the true value of a policy and were able to inform when an offline agent outperforms π_b and hence ready to be deployed. For instance, with 95% confidence, in case of offline improvement with QL, we were able to tell that our new target policy is better than the behavior policy at iteration 4 using two estimators (MB and WDR). Our ability to detect an improved policy is a function of how much better that policy is and how good our OPE methods are. It is better to rely on estimators that do not take π_b into account (e.g. direct methods or hybrid methods) so that estimates are less affected by the offline improvement method and its divergence from π_b (as the case for WIS). Further analysis can be found in Appendix A.4. Accordingly, high-confidence off-policy estimators (e.g. direct methods or hybrid methods) are a safe evaluation method for offline learning with enough confidence that controls the risk of overestimating the true performance of an offline policy.

6 Conclusion

In this paper, we propose a framework for safe evaluation of offline RL methods. While dynamically receiving data, we train offline RL agents and run safety tests to estimate a lower-bound on the value of the target policy and control the risk of overestimating its true value. This is essential for safety-critical applications to be able to tell when it is safe to deploy a new policy. We believe safe

evaluation is an important step for offline RL; offline RL has great potential for control in the actual environment, if they are good enough, where our proposed framework is applicable. In future work, we will test this framework on more complex environments, both discrete and continuous. We will include learning offline from multiple sources of data with different qualities and explore how the quality of the data affects safe evaluation. Moreover, we can use safety testing to develop algorithms which can estimate in advance how much data is needed to learn a good-enough policy that can take over the real environment with maximum data efficiency.

Acknowledgments and Disclosure of Funding

This work has taken place in the Intelligent Robot Learning (IRL) Lab at the University of Alberta, which is supported in part by research grants from the Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; NSERC; and Compute Canada. A portion of this work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (CPS-1739964, IIS-1724157, FAIN-2019844), ONR (N00014-18-2243), ARO(W911NF-19-2-0333), DARPA, Lockheed Martin, GM, Bosch, and UT Austin's Good Systems grand challenge. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed andapproved by the University of Texas at Austin in accordance with its policy on objectivity in research.

References

Bain, M.; and Sammut, C. 1999. A Framework for Behavioural Cloning. In *Machine Intelligence 15, Intelligent Agents [St. Catherine's College, Oxford, July 1995]*, 103–129. GBR: Oxford University. ISBN 0198538677.

Efron, B. 1979. Bootstrap Methods: Another Look at the Jackknife. *Ann. Statist."*, 7.

Fujimoto, S.; Conti, E.; Ghavamzadeh, M.; and Pineau, J. 2019. Benchmarking Batch Deep Reinforcement Learning Algorithms. *CoRR*, abs/1910.01708.

Fujimoto, S.; Meger, D.; and Precup, D. 2018. Off-Policy Deep Reinforcement Learning without Exploration. *CoRR*, abs/1812.02900.

Gottesman, O.; Johansson, F.; Komorowski, M.; Faisal, A.; Sontag, D.; Doshi-Velez, F.; and Celi, L. 2019. Guidelines for reinforcement learning in healthcare. *Nature Medicine*, 25: 16–18.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1861–1870. PMLR.

Hanna, J. P.; Niekum, S.; and Stone, P. 2021. Importance Sampling in Reinforcement Learning with an Estimated Behavior Policy. *Machine Learning (MLJ)*.

Hanna, J. P.; Stone, P.; and Niekum, S. 2017. Bootstrapping with Models: Confidence Intervals for Off-Policy Evaluation. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, 538–546. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Jiang, N.; and Li, L. 2015. Doubly Robust Off-policy Evaluation for Reinforcement Learning. *CoRR*, abs/1511.03722.

Le, H. M.; Voloshin, C.; and Yue, Y. 2019. Batch Policy Learning under Constraints. *CoRR*, abs/1903.08738.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *CoRR*, abs/2005.01643.

Mahmood, A. R.; van Hasselt, H. P.; and Sutton, R. S. 2014. Weighted importance sampling for off-policy learning with linear function approximation. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Morrison, A. C.; Bare, L. A.; Chambless, L. E.; Ellis, S. G.; Malloy, M.; Kane, J. P.; Pankow, J. S.; Devlin, J. J.; Willerson, J. T.; and Boerwinkle, E. 2007. Prediction of Coronary Heart Disease Risk using a Genetic Risk Score: The Atherosclerosis Risk in Communities Study. *American Journal of Epidemiology*, 166(1): 28–35.

Precup, D.; Sutton, R. S.; and Singh, S. P. 2000. Eligibility Traces for Off-Policy Policy Evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, 759–766. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 1558607072.

Riedmiller, M. 2005. Neural Fitted Q Iteration – First Experiences with a Data Efficient Neural Reinforcement Learning Method. volume 3720, 317–328. ISBN 978-3-540-29243-2.

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

Thomas, P. S. 2015. *Safe reinforcement learning*. Ph.D. thesis, University of Massachusetts Libraries.

Thomas, P. S.; and Brunskill, E. 2016. Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning. *CoRR*, abs/1604.00923.

Thomas, P. S.; Theocharous, G.; and Ghavamzadeh, M. 2015a. High-Confidence Off-Policy Evaluation. In *AAAI*.

Thomas, P. S.; Theocharous, G.; and Ghavamzadeh, M. 2015b. High Confidence Policy Improvement. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, 2380–2388. JMLR.org.

Voloshin, C.; Le, H. M.; Jiang, N.; and Yue, Y. 2019. Empirical Study of Off-Policy Policy Evaluation for Reinforcement Learning. *CoRR*, abs/1911.06854.

Watkins, C. J. C. H.; and Dayan, P. 1992. Q-learning. In *Machine Learning*, 279–292.

A Appendix

A.1 Bootstrapping

Consider a sample D of n random variables H_j for $j = 1, 2, ..., n$ where we can sample H_j from some $i.i.d.$ distribution of data. From the sample of data D, we can compute a sample estimate $\hat{\theta}$ of a parameter θ such that $\hat{\theta} = f(D)$ where f is the function to compute θ. Given a dataset D, we create B resamples with replacement, where B is the number of bootstrap resamples, and compute θ, $\hat{\theta}$, on each of these resamples. Bootstrapping (Efron 1979) allows us to estimate the distribution of $\hat{\theta}$ with confidence intervals. The estimates computed with different resamples will be used to determine the $1 - \delta$ confidence interval. In our setup, the parameter of interest θ is the expected return of a policy $v(\pi)$.

Hence, with a confidence level $\delta \in [0, 1]$ and B resamples of the dataset of trajectories D, we use bootstrapping methods to approximate a confidence lower bound of $v_\delta(\pi)$ on $v(\pi)$ such that $v_\delta(\pi) \leq v(\pi)$ with probability at least $1 - \delta$.

As the size of data $n \to \infty$, bootstrapping has strong guarantees but it lacks guarantees for finite samples as it is not an exact method. To use bootstrapping, we have to assume that the bootstrap distribution is representative of the distribution of the statistic of interest, which is not the case for finite samples (Hanna, Stone, and Niekum 2017). As a result, bootstrapping is considered semi-safe but it is still safe enough for high risk medical predictions in practice with a known record of producing accurate confidence intervals (Morrison et al. 2007).

A.2 High-confidence off-policy evaluation techniques

Importance sampling with bootstrapping Importance Sampling (IS) (Precup, Sutton, and Singh 2000) is a way for handling mismatch between distributions and hence presented as a consistent and unbiased off-policy estimator. For a trajectory $H \sim \pi_b$ of length L, as $H = S_1, A_1, .., S_L, A_L$, we can define the importance sampling up to time t for policy π_θ as follows: IS$(\pi_\theta, \pi_b, D) = \sum_{i=1}^{m} \rho_L^H R^i, \rho_t^H := \prod_{j=0}^{t} \frac{\pi_\theta(A_j|S_j)}{\pi_b(A_j|S_j)}$ In our setup given no access to π_b, we estimate the behavior policy π_b from data but it results in a biased estimator (Hanna, Niekum, and Stone 2021). To produce a high-confidence IS estimate, we first compute the importance weighted returns then use bootstrapping to get the lower-bound estimate. A popular method is Bias Corrected and accelerated (BCa) bootstrapping (Efron 1979). A pseudo code of BCa is well-described as Algorithm 3 in (Thomas, Theocharous, and Ghavamzadeh 2015b). We use weighted importance sampling (Mahmood, van Hasselt, and Sutton 2014) for discrete control along with bootstrapping.

Direct model-based with bootstrapping Model-based estimation is another off-policy estimator that lies under the direct methods. The model-based off-policy estimator MB computes $v(\pi_\theta)$ by building a model using all the available trajectories D to build a model $\hat{M} = (S, A, \hat{P}, r, \gamma, \hat{d}_0)$ where $\hat{P}$ and $\hat{d}_0$ are estimated with trajectories sampled from π_b. Then, MB will compute $\hat{v}(\pi_\theta)$ as the average return of trajectories simulated in the estimated model $\hat{M}$ while following π_θ. Despite having lower variance than IS methods, MB is an inconsistent estimator such that as $n \to \infty$, the model estimates may converge to a value different from $V(\pi_\theta)$. This is because it is dependant on the modeling assumptions we make, whether we assume a linear or a non-linear model. To get a lower confidence bound on the MB estimate, we use bootstrap confidence intervals; the exact algorithm used is detailed in Hanna, Stone, and Niekum (2017). This bootstrapping method is quite similar to what we mentioned in the previous paragraph but much simpler in implementation and adaptable to different OPE methods. We rely on the model-based estimator with bootstrapping in our study as one of OPE direct-methods.

Weighted doubly-robust with bootstrapping Weighted Doubly-Robust (WDR) (Thomas and Brunskill 2016) is a hybrid method for off-policy estimation, presented as an extension to the doubly-robust (DR) method (Jiang and Li 2015). DR is an unbiased estimator of $v(\pi_\theta)$ that uses an approximate model of the MDP to reduce the variance of importance sampling (Jiang and Li 2015). Although biased, WDR is based on per-decision weighted importance sampling (PDWIS) and serves as an improvement over DR method as it significantly balances the bias-variance trade-off. The approximate model value functions act as a control variate for PDWIS (Thomas and Brunskill 2016).

$$PDWIS(\pi_\theta, D, \pi_b) = \sum_{i=1}^{m} \sum_{t=0}^{L-1} \frac{\rho_t^i}{\sum_{j=1}^{m} \rho_t^j} \gamma^t R_t^i \tag{1}$$

$$WDR(\pi_\theta, D, \pi_b) = PDWIS(\pi_\theta, D, \pi_b) - \sum_{i=1}^{n} \sum_{t=0}^{L-1} \gamma^t (w_t^i \hat{q}_{\pi_\theta}(S_t^i, A_t^i) - w_{t-1}^i \hat{v}_{pi_\theta}(S_t^i)) \tag{2}$$

In previous methods, we use bootstrapping with WDR to provide lower confidence bound estimates over $v(\pi_\theta)$. WDR with bootstrapping is guaranteed to converge to the correct estimate as n increases given the statistical

consistency of PDWIS (Hanna, Stone, and Niekum 2017). For the approximate model, a single model is estimated with the available trajectories D, and used to compute the value functions of WDR for each bootstrap data. We choose the weighted doubly robust estimator to represent OPE hybrid methods in our study.

A.3 Experimental Details

Environment: We use the classic MountainCar (Sutton and Barto 2018) with a continuous state (velocity and position), and 3 possible discrete actions. At each time-step, the reward is -1 except for in a terminal state when it is 0. However, we used the modified version of MountainCar as described by Thomas (2015). This means we shorten the horizon of the problem by holding an action a_t constant for 4 updates of the environment state. We also change the start state such that an episode starts with a random position and random velocity (Jiang and Li 2015; Thomas 2015).

A.4 Analysis of evaluation methods

To better understand how safety testing is affected by the policy improvement methods, there are multiple measures to tell the similarity between two probability distributions (as a policy is a distribution over states and actions). One of the practical measures is the total variation (TV) distance. TV distance is a way to measure the difference between action probabilities taken under two policies given the data set in hand. TV would be the sum of differences in probabilities between the behavior policy π_b and the target policy π_θ for each state-action pair in the test set of data. Since π_b is not known, we estimate $\hat{\pi_b}$ given the test data (Hanna, Niekum, and Stone 2021).

We analyze how the improvement method is affecting TV distance between $\hat{\pi_b}$ and π_θ and hence affecting the high-confidence off-policy evaluation estimates. TV distance is correlated to KL-Divergence and hence shows how two policies are different from each other. Figure 3 shows the total variation distance between the offline policy π_θ and the estimated behavior policy $\hat{\pi}_b$ across the different offline learning methods. It show that behavioral cloning (imitation learning) can achieve a much lower distance than other improvement methods whether they are constrained or non-constrained (Q-learning and batch-constrained Q-learning); this is because behavioral cloning forces its target policy to be close to the behavior policy while other methods do not. This result also explains why weighted importance sampling achieves the lowest error between the estimate and the true value in the case of behavioral cloning; the error grows for other methods that do not constrain the policy to be close to the data distribution.

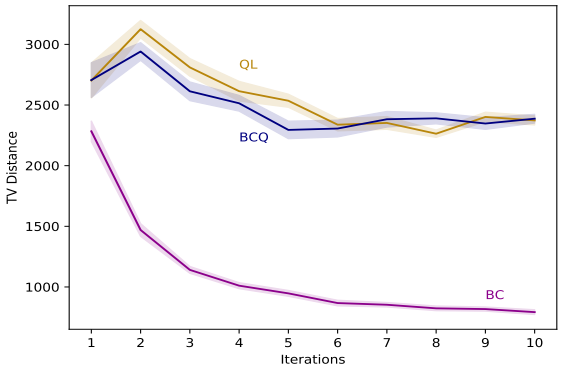

Figure 3: Total Variation Distance between $\hat{\pi}_b$ & π_θ