Probabilistic Offline Policy Ranking with Approximate Bayesian Computation

Longchao Da[1], Porter Jenkins[2], Trevor Schwantes[2], Jeffrey Dotson[2], Hua Wei[1]*

[1]Arizona State University,
[2]Brigham Young University
{longchao,hua.wei}@asu.edu, {pjenkins,jeff_dotson}@cs.byu.edu, Schwantes2@gmail.com

Abstract

In practice, it is essential to compare and rank candidate policies offline before real-world deployment for safety and reliability. Prior work seeks to solve this offline policy ranking (OPR) problem through value-based methods, such as Off-policy evaluation (OPE). However, they fail to analyze special case performance (e.g., worst or best cases), due to the lack of holistic characterization of policies' performance. It is even more difficult to estimate precise policy values when the reward is not fully accessible under sparse settings. In this paper, we present Probabilistic Offline Policy Ranking (POPR), a framework to address OPR problems by leveraging expert data to characterize the probability of a candidate policy behaving like experts, and approximating its entire performance posterior distribution to help with ranking. POPR does not rely on value estimation, and the derived performance posterior can be used to distinguish candidates in worst-, best-, and average-cases. To estimate the posterior, we propose POPR-EABC, an Energy-based Approximate Bayesian Computation (ABC) method conducting likelihood-free inference. POPR-EABC reduces the heuristic nature of ABC by a smooth energy function, and improves the sampling efficiency by a pseudo-likelihood. We empirically demonstrate that POPR-EABC is adequate for evaluating policies in both discrete and continuous action spaces across various experiment environments, and facilitates probabilistic comparisons of candidate policies before deployment.

Introduction

Policies trained in simulation often encounter performance drops when deployed in a different simulated environment (Jayawardana et al. 2022; Wei et al. 2022) or the real world (Hanna and Stone 2017; Da et al. 2023). With a set of candidate policies, evaluating and ranking prior to real deployment is critical for real-world applications. Off-policy evaluation (OPE) allows one to estimate the goodness of a policy (often referred to as target/candidate policy) using data collected from another, possibly unrelated policy (referred to as behavior policy). Such evaluation is important because testing and implementing a policy in the real world can be costly in areas like trading (Liu et al. 2020) and physical retail (Jenkins et al. 2022, 2020), even vital in situations like healthcare (Liao et al. 2020) and transportation (Du et al. 2023; Vlachogiannis et al. 2023; Li et al. 2023).

With growing interest in OPE, the research community has produced a number of estimators, including importance sampling (IS) (Thomas and Brunskill 2016; Farajtabar, Chow, and Ghavamzadeh 2018; Jiang and Li 2016), direct methods (DM) (Harutyunyan et al. 2016; Li et al. 2010; Le, Voloshin, and Yue 2019; Kostrikov and Nachum 2020) and distribution correction estimation (DICE) methods (Dai et al. 2020; Zhang, Liu, and Whiteson 2020; Yang et al. 2022). Importance-sampling-based methods weight the data collected from the behavior policy according to the probability of transitioning to each state under the target policy, yet they assume access to a probability distribution over actions from the behavior policy. DM and DICE do not require knowing the output probabilities of the behavior policy, where DM directly learns an environment or value model from offline data, and DICE methods learn to estimate the discounted stationary distribution ratios. Most of these methods compute the point estimates of the policy's value (Dudík, Langford, and Li 2011; Jiang and Li 2016; Zhang, Liu, and Whiteson 2020; Yang et al. 2020), some of which additionally estimate the value with confidence intervals (Thomas, Theocharous, and Ghavamzadeh 2015; Kuzborskij et al. 2020; Feng et al. 2020; Dai et al. 2020; Kostrikov and Nachum 2020).

While various estimators have been proposed for off-policy evaluation, in many cases, precise policy value estimation is not necessary. Instead, practitioners often place greater emphasis on the correctness of comparison and ranking of candidate policies. Existing work in Supervised Off-Policy Ranking (SOPR) (Jin et al. 2021) takes a supervised learning approach to policy ranking, and requires an adequate training set or policies with explicitly labeled performance. In practice, this approach is challenging because (1) actual policy data is typically limited, and (2) access to labeled performance data is a strict assumption. The behavior policy is usually inaccessible and behavioral data is usually restricted, such as in healthcare or confidential financial trading domains. Additionally, when the policies are hard to differentiate from mean performance, we might care about the performance under special situations like the worst or best cases (Agarwal et al. 2021), which none of the above literature could address.

In this paper, we propose the Probabilistic Offline Policy Ranking (POPR) framework to address the above challenges.

*Corresponding Author

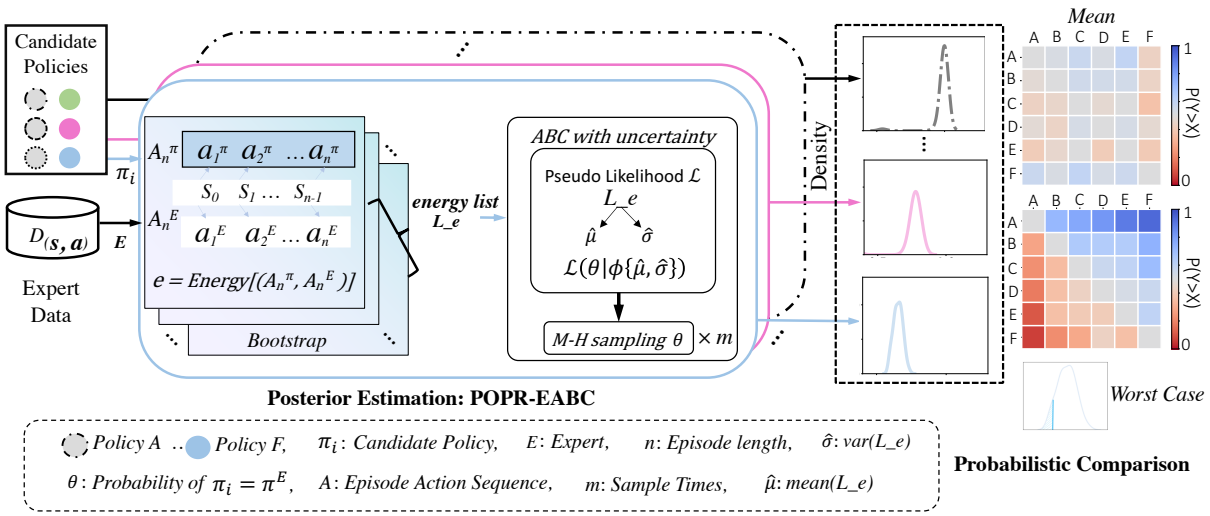


Figure 1: POPR consists of two parts, posterior estimation and probabilistic comparison. (Left) Given expert datasets and candidate policies, (Middle) POPR learns the performance posterior of each candidate relative to an expert, in which we propose POPR-EABC. By bootstrapping, the energy function first calculates a list of energy values between expert action A_n^E and policy π's action A_n^π given the same S sampled from $D_{(s,a)}$, and then a pseudo-likelihood (aware of mean and variance from energy values) is used in M-H algorithm to sample the acceptable probability θ from a proposal distribution. (Right) The sampled proposals approximate the performance posterior, enabling special case comparison: e.g., worst-case analysis. The heat map shows pair-wise comparisons. It is hard to compare on mean but easier on the worst case, showing the benefit of POPR-EABC.

POPR does not require access to behavioral policies, nor the performance value or reward. Instead, it exploits limited expert data to the maximum extent. Our intuition is that we can measure the expected behavior of a candidate policy relative to a static policy generated by an expert, such as a medical doctor or a driving instructor(Kim et al. 2013). The more the candidate policy behaves like experts, the better.

Based on this intuition, the performance posterior distribution is estimated, providing a holistic characterization of the policy's quality, making it possible to compare the performance in the best or worst case. Specifically, we propose a novel method POPR-EABC, an **E**nergy based **A**pproximate **B**ayesian **C**omputation to estimate the posterior distribution. By using a smooth energy function to measure the similarity between expert and policy-generated data, we obviate the need to specify tolerance parameters on summary statistics and improve the efficiency of ABC. We introduce a pseudo-likelihood that parameterizes the energy variance and facilitates Bayesian inference. On both self-trained policies and open-sourced policies, we perform extensive evaluations comparing six baselines under different RL tasks covering both discrete and continuous action spaces. The results prove the effectiveness of POPR-EABC in offline policy evaluation. We demonstrate our method could exploit efficiently at small size expert data and has a high tolerance for data quality.

Preliminaries

In this section, we formalize the Offline Policy Ranking (OPR) problem and the general process of OPR methods.

Formalization of OPR Problem

We consider a Markov Decision Process (MDP), defined by a tuple (S, A, T, R, γ), where S and A represent the state and action spaces, respectively. $T(s'|s,a)$ represents a, possibly unknown, transition function, where s' is the next possible state from s taking action a, $s \in S$, $a \in A$ and $R(s,a)$ represents a reward function. The expected return of a policy π is defined as $V(\pi) = \mathbb{E}[\sum_{t=0}^{\mathcal{H}} \gamma^t r_t]$, where $\mathcal{H}$ is the horizon of the MDP, and t is the index of steps.

Definition 1 (Offline Policy Ranking)**.** *Given an offline dataset $\mathcal{D}$, that consists of N observed behavior trajectories $\mathcal{T}$ from behavior policy μ. $\mathcal{D} = \{\mathcal{T}_i \langle s_t, a_t, r_t, s_{t+1} \rangle\}_{i=1}^N$ with N trajectories, each having a variable length L_i, $t \in L_i$. And given a set of candidate policies $\hat{\Pi} = \{\hat{\pi}^{(1)}, ..., \hat{\pi}^{(k)}\}$, where k represents the policies' index. The goal of offline policy ranking is to acquire a ranked order $O\{\cdot\}$ that represents the true performance of the policies without interacting with the environment online. In the later section, OPR stands for Offline Policy Ranking.*

Solutions to OPR

OPE Methods It is possible to solve OPR tasks using Off-policy evaluation by calculating the expected values $\mathbb{E}[V(\hat{\pi}^{(k)}|\mathcal{D})]$ for each candidate policy $\hat{\pi}^{(k)}$ and ranking each $\hat{\pi}^{(k)} \in \hat{\Pi}$ accordingly to obtain $O\{\hat{\Pi}\}$. These methods (Voloshin et al. 2019; Harutyunyan et al. 2016; Precup 2000) aim to precisely estimate the expected value with an offline dataset $\mathcal{D}$, but they either tend to re-weight (Xie, Ma, and Wang 2019) or provide correction to the original reward

values r (Nachum et al. 2019), such reliance on high-quality and dense value return is a severe challenge for most of OPE methods in practical use. For scenarios with intrinsic sparse reward settings (only receive reward signal when the task is done) or low-quality reward representation (partially observable (s, a) leads to untrackable reward r), the OPE methods are likely to rank the policies incorrectly.

Supervised Off-Policy Ranking (SOPR) SOPR (Jin et al. 2022) is also able to solve OPR problems by training a ranking model using a policy dataset with labeled performance, and then minimizing a ranking loss. It considers the overall performance in the training process but also fails to present a probabilistic result of the comparison, and is not able to conduct worst/best case analysis either.

We summarize the current solutions in Table 1, while some of the OPE methods relax the requirement of the action probability $P(a)$ from behavior policy, none of them could alleviate the reliance on rewards.

Features	OPE	SOPR	POPR (ours)
Not Access Value	✗	✗	✓
Not Access $P(a)$	✗/✓	✓	✓
Probabilistic Ranking	✗	✗	✓
Case Analysis	✗	✗	✓

Table 1: Comparison of different solutions for OPR tasks

When two policies perform similarly with mean performance, we may wonder what is the probability of one outperforming the other and also the comparison in the worst/best cases behavior, which leads to the probabilistic comparison and case analysis, while none of the existing work could solve. Therefore, we provide our probabilistic framework in Section . to estimate the posterior distribution of the performance for policy $\hat{\pi}^{(k)}$ on the offline data $\mathcal{D}$. The posterior should contain all the necessary probabilistic information about $\hat{\pi}^{(k)}$, helping us to analyze the best or worst-case performance. In the implementation, we introduce an energy-based inference method with pseudo-likelihood to estimate the posterior, which helps the evaluation to better consider the intrinsic uncertainty.

Probabilistic Offline Policy Ranking

In order to correctly rank the candidate policies, we can compare the performance of the candidate policies with experts as an indicator of the goodness of their performance. Below, we define a statistic value representation θ, which is the result of an estimation, that can be used to rank over $\hat{\Pi}$, and we provide a formal definition as below.

Definition 2 (Probabilistic Offline Policy Comparison)**.** *Given an expert dataset $\mathcal{D}_e = \{\langle s_t^{(i)}, a_t^{(i)}, r_t^{(i)}, s_{t+1}^{(i)} \rangle_i\}_{i=1}^N$, if we define a statistic $\theta^{(k)} \in [0, 1]$, that measures how consistent the candidate policy $\hat{\pi}^{(k)}$ is with the expert policy, the posterior distribution of $\theta^{(k)}$ is defined as:*

$$p(\theta^{(k)}|\mathcal{D}_e, \hat{\pi}^{(k)}) := p[\pi_e(s_t) = \hat{\pi}^{(k)}(s_t)|\ \mathcal{D}_e] \quad (1)$$

Note that posterior $p(\theta^{(k)}|\mathcal{D}_e, \hat{\pi}^{(k)})$ can be recognized as the formation of a bag of $\theta^{(k)}$ samples.

Under this definition, $\theta^{(k)}$ is the probability that the candidate policy $\hat{\pi}^{(k)}$ produces the same behavior as the expert π_e, given the state s_t. To reduce notational clutter, we use $p(\theta^{(k)}|\mathcal{D}_e, \hat{\pi}^{(k)})$ and $p(\theta^{(k)}|\mathcal{D}_e)$ interchangeably in this paper, additionally, we interchange $\pi_e(s_t)$ and a_e since we do not assume to know the form of the expert policy. We will refer target policy and behavior policy in OPE scope, whereas candidate policy and expert policy are in the POPR scope.

Posterior Estimation Due to the limited number of expert trajectories, real-world environments' stochasticity, and the target policy's decision variance, there is intrinsic uncertainty in $\theta^{(k)}$ when describing the performance of a policy, which introduces bias to the statistic measurement, further causing unreliable evaluation. Holistic depictions considering the variance help to better policies' performance, so we seek to estimate the holistic posterior distribution of $\theta^{(k)}$ for:

$$p(\theta^{(k)}|\mathcal{D}_e) \propto p(\mathcal{D}_e|\theta^{(k)})p(\theta^{(k)}) \quad (2)$$

Based on the meaning of $\theta^{(k)}$, posterior $p(\theta^{(k)}|\mathcal{D}_e)$ provides holistic information on policy $\hat{\pi}^{(k)}$'s behavior, and supports the evaluation of the candidate policies. Posterior estimation approaches are not limited, such as Bayesian Inference, Markov Chain Monte Carlo, etc. If we notate the posterior process as $f(\cdot)$, we could represent the framework process:

$$O\{\hat{\Pi}\} = G(f(\hat{\pi}^{(k)}|\mathcal{T}_i \sim \mathcal{D}_e)) \quad (3)$$

where $\mathcal{T}_i \in \mathcal{D}_e$ containing n trajectories, $f(\cdot)$ is the posterior estimation process based on sampling trajectories $\mathcal{T}$ from dataset $\mathcal{D}_e$, please note that practitioner could sample multiple times not limited to the total amount n of trajectories. And function $G(\cdot)$ could be any post-process and analysis procedures on posterior samples $\mathcal{S}_\theta^{(k)}$, such as statistical calculation or comparison introduced in Section . $\mathcal{S}_\theta^{(k)}$ represents a bag of sampled θ for candidate policy $\hat{\pi}^{(k)}$.

Scoring Functions for Ranking and Comparison From $f(\cdot)$ process, the derived posterior samples $\mathcal{S}_\theta^{(k)}$ summarize all of the performance information learned from the behavior of candidate policies $\hat{\pi}^{(k)}$. We can conduct various tasks such as policy ranking or probabilistic pair-wise comparison, and furthermore, the special cases analysis from the posterior samples. The different tasks will lead to different instantiations of a concrete function $G(\cdot)$.

• *Ranking on Average:* To conduct a ranking task considering the overall performance, we could use the mean of the whole samples, we do: $\mathbb{E}[\theta^{(k)}] = \frac{1}{|\mathcal{S}|}\sum_{s \in \mathcal{S}_\theta^{(k)}} s$, where $|\mathcal{S}|$ stands for the total amount of $\theta^{(k)}$ samples, and then sort $\hat{\Pi} = \{\hat{\pi}^{(1)}, ..., \hat{\pi}^{(k)}\}$ by $\mathbb{E}[\theta^{(k)}]$, we get the resulting $O\{\hat{\Pi}\}$.

• *Worst/Best-case Analysis:* To conduct special cases analysis, either ranking or pair-wise comparison, we only need to conduct one step selection on the pre-ordered $\mathcal{S}_\theta^{(k)}$. E.g., in this paper, when it comes to worst-case comparison, only the bottom $5\% \times |\mathcal{S}|$ of samples will be selected to keep with the above two analysis procedures. Note that the proportion of observation could vary according to the necessity.

• *Pair-wise Comparison:* To conduct a pair-wise comparison between a group of policies, we compare by the expected Monte Carlo samples (θs), e.g., $\hat{\pi}^{(k)}$, and $\hat{\pi}^{(l)}$ by computing $p(\theta^{(k)} > \theta^{(l)}|\mathcal{D}_e)$. In other words, $p(\theta^{(k)} > \theta^{(l)}|\mathcal{D}_e) = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}_\theta^{(k)}} \mathbb{1}(\mathcal{S}_\theta^{(k)}[i] > \mathcal{S}_\theta^{(l)}[i])$, i is the index for each $\theta \in \mathcal{S}_\theta^{(k)}$.

POPR-EABC

Following the POPR framework, the primary task is to estimate the posterior of a candidate policy $\hat{\pi}^{(k)}$'s performance, i.e., to derive $\mathcal{S}_\theta^{(k)}$. Bayesian inference typically requires the specification of a likelihood function, where the data are assumed to be independent and identically distributed (iid) to make the likelihood computation tractable. However, such an assumption is not suitable for policy evaluations since the observed states and actions in MDP's are determined by the environment dynamics, and are *not* independent from each other. Thus, we seek to leverage Approximate Bayesian Computation (ABC), which relies on simulation rather than likelihood, to measure the relevance of parameters to the data. An overview of our method is shown in Figure 1.

Energy-based Similarity

The standard ABC paradigm faces the difficulty of designing a good summary statistic for efficient sampling. In practice, these methods suffer from very low acceptance rates and long sampling times (Turner and Van Zandt 2012). To overcome this challenge, we define a continuous, scalar-valued energy function, $e = E(\mathcal{T}_e, \hat{\mathcal{T}}), e \in [0, 1]$ to avoid specifying a heuristic discrepancy statistic, where $\mathcal{T}_e$ is the expert trajectory and $\hat{\mathcal{T}}$ is the simulated trajectory taken by candidate policy $\pi^{(k)}$, given the same observation sequence S_e. We draw bootstrapped samples of trajectories from our dataset, $d_e \in \mathcal{D}$, and generate simulated data from our candidate policy, $\hat{d}$. We then calculate the normalized energy between these two data subsets, $E(d_e, \hat{d}) = 1 - \frac{\rho(d_e,\hat{d})}{Z}$, where $Z = |d_e|$ is a normalizing constant, and ρ is could be any universal distance metric. Intuitively, when the similarity between the two bootstrapped data sets is high, E approaches unity; when similarity is low, E approaches zero. In our experiments, we find out that Jensen–Shannon (JS) divergence (Endres and Schindelin 2003) is more efficient compared to other similarity measures, and use it as the default setting.

Calculating the Pseudo-likelihood

The above energy-based statistic helps mitigate the ABC algorithm's heuristic nature by providing a smooth measure of similarity between datasets. To estimate the posterior distribution we design a pseudo-likelihood, which uses the bootstrapped energy values to provide an approximation of the joint probability of the data in a computationally simpler way (Besag 1975). Rather than specify a formal likelihood, we fit a density function to the empirical energy values. This density contains distributional information about the behavior of the candidate policy relative to the expert. The pseudo-likelihood, along with the prior, facilitates the estimation of the posterior.

More formally, we approximate the likelihood as function of M bootstrapped energy values: $\mathcal{L}(\mathcal{D}_e|\theta^{(k)}) \approx p(\theta^{(k)}|\{e_1, ...e_M\})$. The energy values, $e_1, ...e_M$ are calculated by drawing M bootstrapped datasets $\{\bar{\mathcal{D}}_1, ...\bar{\mathcal{D}}_M\}$ from the expert data, $\mathcal{D}_e$. This bootstrapping routine induces diversity in both the expert data, and the candidate policy, and facilitates an estimate of variability in θ. We assume that the pseudo-likelihood follows a beta distribution, $p(\theta|\{e_1, ...e_M\}) \sim \text{Beta}(\alpha, \beta)$, since the support for the beta distribution lies in $[0, 1]$ and is conducive for estimating probabilities. Because the bootstrapped energy values are all sampled independently, we can use the relatively simple method of moments estimator (Fielitz and Myers 1975) to fit the pseudo-likelihood to our data.

$$\hat{\alpha} = \hat{\mu}\left[\frac{\hat{\mu}(1-\hat{\mu})}{\hat{\sigma}^2} - 1\right], \hat{\beta} = (1-\hat{\mu})\left[\frac{\hat{\mu}(1-\hat{\mu})}{\hat{\sigma}^2} - 1\right] \quad (4)$$

The parameters, $\hat{\alpha}$ and $\hat{\beta}$, as shown in Eq. 4, are calculated based on the mean $\mu = \frac{1}{M}\hat{\sum}_{i=1}^{M} e_i$ and variance $\hat{\sigma}^2 = \frac{1}{M-1}\sum_{i=1}^{M}(e_i - \hat{\mu})^2$ of the bootstrapped energy values, which determine the shape and scale of the Beta density and specify a plausible range of values of $\theta^{(k)}$, $\mathcal{L}(\theta^{(k)}|\hat{\alpha}, \hat{\beta}) = \text{Beta}(\hat{\alpha}, \hat{\beta})$. This, along with the prior, $p(\theta^{(k)})$, form the acceptance criteria. Detailed explanations to better illustrate the equation are shown in the Appendix. Intuitively, $\mathcal{L}(\theta^{(k)}|\hat{\alpha}, \hat{\beta})$ outputs a likelihood over the domain of $\theta^{(k)}$, given the bootstrapped energy values. Figure 2 provides visualization of the behavior of this function at different $\theta^{(k)}$ and energy values.

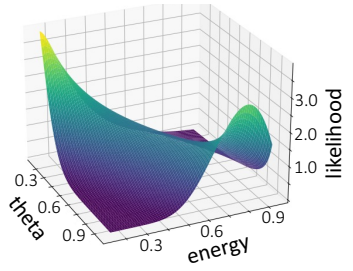


Figure 2: A visual depiction of an example pseudo-likelihood surface. For a given θ, energy tuple, the likelihood of the combination is shown on the vertical axis. For high energy values (indicating high agreement between expert and candidate), large θ will have a high likelihood (indicating a high acceptance probability). We incorporate this into the ABC-MH (Metropolis-Hastings) sampling algorithm to learn the posterior $p(\theta^{(k)}|\mathcal{D}_e)$ of policy performance.

Sampling the Posterior

Consequently, we can apply an adapted Metropolis-Hastings (M-H) algorithm (Turner and Van Zandt 2012) in POPR-EABC to sample from the posterior by replacing the likelihood term with the pseudo-likelihood. After the execution of algorithm, the output of POPR-EABC is a set

Algorithm 1: POPR-EABC Algorithm

Input: Dataset $\mathcal{D}_e$, candidate policy $\hat{\pi}^{(k)}$, prior distribution $p(\cdot)$, proposal distribution $q(\cdot)$, energy function $E(\cdot)$, pseudo-likelihood $\mathcal{L}(\cdot)$

Output: Set of posterior samples $\mathcal{S}_\theta^{(k)}$

Initialize posterior $\theta^{(k)}$ and sample set $\mathcal{S}_\theta^{(k)}$
for $i = 1 : N$ do
  Get M bootstrapped trajectories $\bar{\mathcal{D}}_e \sim \mathcal{D}_e$
  Initialize array of energy values, $\mathcal{E} = \{\}$
  for $j = 1 : M$ do
    Get episode, $ep_j = \bar{\mathcal{D}}_e[j]$ with length l
    Initialize synthetic dataset, $\hat{\mathcal{D}} = \{\}$
    for $t = 1 : l$ do
      $s_t = ep_j[t]$, $\hat{a_t} = \hat{\pi}^{(k)}(s_t)$
      $\hat{\mathcal{D}}.append([s_t, \hat{a_t}])$
    end
    Evaluate energy $e = E(\bar{\mathcal{D}}_e, \hat{\mathcal{D}})$
    $\mathcal{E}.append(e)$
  end
  Calculate $\hat{\mu}$, $\hat{\sigma}$, $\hat{\alpha}$, and $\hat{\beta}$ from $\mathcal{E}$ with Eq.4
  Propose new theta $\theta^* \sim q(\theta^*|\theta_i^{(k)})$
  Compute acceptance probability:
  $\tau = min\left[1, \frac{\mathcal{L}(\theta^*|\hat{\alpha},\hat{\beta})p(\theta^*)q(\theta_i^{(k)}|\theta^*)}{\mathcal{L}(\theta_i^{(k)}|\hat{\alpha},\hat{\beta})p(\theta_i^{(k)})q(\theta^*|\theta_i^{(k)})}\right]$
  if $\tau < \phi \sim uniform(0,1)$ then
    Accept proposal: $\mathcal{S}_\theta.append(\theta^*)$
    $\theta_i^{(k)} \leftarrow \theta^*$
  end
  else
    Reject proposal: $\mathcal{S}_\theta.append(\theta_i)^{(k)}$
  end
  Return $\mathcal{S}_\theta$
end

of Monte Carlo samples, $\mathcal{S}_\theta^{(k)}$, which approximate posterior distribution, $p(\theta^{(k)}|\mathcal{D}_e, \hat{\pi}^{(k)})$. The full description of the POPR-EABC algorithm can be found in the Algorithm 1.

We execute the POPR-EABC algorithm with a burn-in period of $B = 10$ iterations, and $N = 500$ sampling iterations. Additionally, we set $M = 5$ for the number of bootstrapped samples at each iteration. We use a $Beta(0.5, 0.5)$ prior, and a $Beta$ proposal distribution with parameters, $\alpha = 4.0$, and $\beta = 1e-3$.

Experiments

Experimental Settings

Environments We first designed our `ToyEnv` to verify the proposal estimation of the posterior, then, we use POPR-EABC and baseline OPE algorithms to solve the OPR problem on widely-used complex environments with discrete or continuous action spaces in the Gym environment. Detailed descriptions of the experiment and code can be found in the repository[1].

Baselines and Variants We compare POPR with the first four representative baselines OPE algorithms with their popular implementation [2], DICE follows (Voloshin et al. 2019). Among the methods, Fitted Q-Evaluation (FQE) (Le, Voloshin, and Yue 2019; Kostrikov and Nachum 2020) is a Q-estimation-based OPE method that learns a neural network to approximate the Q-function of the target policy by leveraging Bellman expectation backup operator (Sutton and Barto 2018). $Q^\pi(\lambda)$ (Harutyunyan et al. 2016) and Tree-Backup(λ) (Precup, Sutton, and Singh 2000; Munos et al. 2016) can be viewed as two types of generalization from FQE. Model-based method (MBased) (Paduraru 2012; Kostrikov and Nachum 2020; Fu et al. 2021) estimates the environment model to derive the expected return of the target policy, and Bayesian Distribution Correction Estimation known as (BayesDICE) (Yang et al. 2022) is the state-of-the-art offline policy ranking method that estimates the posteriors of distribution correction ratios in DICE methods (Dai et al. 2020; Zhang, Liu, and Whiteson 2020; Yang et al. 2020). It assumes the individuality of policies during policy ranking.

We also developed a variant of POPR without the probabilistic capacity as AgreeRank, simply measuring the agreement $Agree(\cdot)$ of π to the expert π_e directly: $p_i = Agree(A_\pi, A_{\pi_e}|D_e)$, where s is sampled from the state list S of D, and A_π is a action list generated by $\pi(s)$ in order with the A_{π_e} taken by $\pi_e(s)$. The p_i represents the performance of policy π_i, which can be used to rank accordingly in the candidate policy set $\hat{\Pi}$ to get the $O\{\cdot\}$. The experiment uses negative Euclidean Distance for continuous action space and $1 - \frac{A_{same}}{A_{total}}$ for discrete, where $A_{same} = \sum_{a_\pi^i, a_{\pi_e}^i} \mathbb{1}(a_\pi^i = a_{\pi_e}^i)$ and $A_{total} = |A|$. Note that the dataset we used fits the common setting in that it does not contain the probability of behavior policy; therefore, the existing IS methods in OPE cannot be utilized.

Evaluation Metrics We evaluate POPR and baseline OPE algorithms with two metrics to reflect their accuracy of ranking candidate policies: widely used ranking metric Normalized Discounted Cumulative Gain (NDCG) (Wang et al. 2013), and Spearman's Rank Correlation Coefficient (SRCC), adapted by (Paine et al. 2020; Jin et al. 2022). Detailed implementation of metrics is introduced in the Appendix. The ranges of NDCG and SRCC are [0, 1] and [-1, 1] respectively. The higher, the better.

Experimental Results

Ranking on Average for Policies with Differentiable Mean Performance We first evaluate POPR and baseline OPE on multi-level differentiable policies (by mean performance), the policies adopt the same network architecture but are trained with different epochs, we provide a detailed description of the used policies, training steps, validated ground-truth rank in the Appendix [3], and we have released pre-trained models and

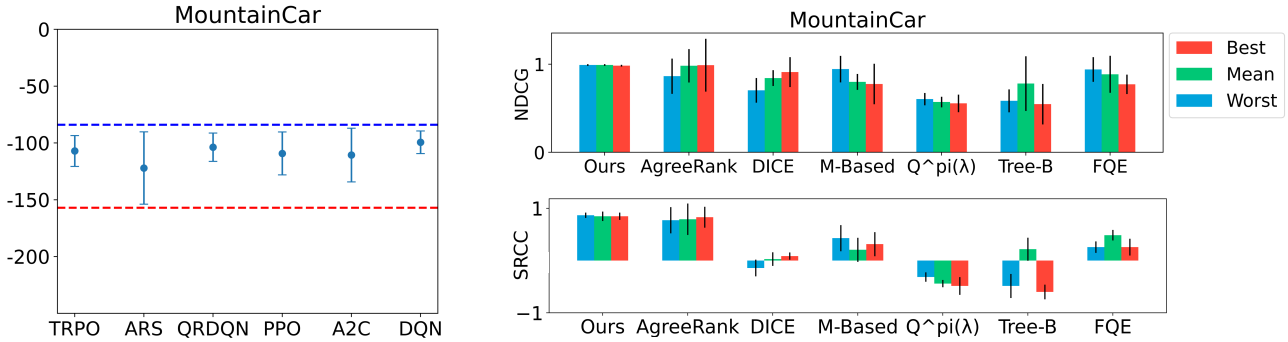
[1] https://github.com/LongchaoDa/POPR-EABC.git

[2] https://github.com/clvoloshin/COBS

[3] Please find Appendix in arXiv version.

	ToyEnv		MountainCar		AcroBot		Pendulumn	
	NDCG	SRCC	NDCG	SRCC	NDCG	SRCC	NDCG	SRCC
FQE	0.8608	0.8228	0.6135	-0.3771	0.5970	-0.1000	0.6620	-0.087
Tree-Backup (λ)	**1.0000**	**1.0000**	0.7321	-0.0629	0.6722	0.1619	0.7562	-0.325
$Q^\pi(\lambda)$	0.6650	-0.274	0.7039	0.1943	0.6108	-0.1070	0.8133	-0.0390
MBased	0.7004	0.0857	0.7093	-0.0001	0.5785	-0.1640	0.5906	-0.2460
BayesDICE	0.5913	-0.466	0.9005	0.2571	0.9033	0.8571	0.7251	0.2976
AgreeRank	**1.0000**	**1.0000**	0.9829	0.9357	**1.0000**	0.9950	0.9421	0.8190
POPR-EABC	**1.0000**	**1.0000**	**0.9992**	**0.9663**	**1.0000**	**1.0000**	**0.9908**	**0.9047**

Table 2: Ranking evaluation on multi-level differentiable policies w.r.t. NDCG and SRCC. The higher, the better. Mean values across 5 times of experiments are shown. Best (bold) and second (underline) indicate POPR-EABC performs ideally.



(a) Online performance.

(b) The evaluation results given by different methods.

Figure 3: The evaluation on policies with similar mean. (a) The mean and standard deviation of the online performance for different candidate policies. The policies have similar mean, making it hard to rank on the mean. (b) The ranking performance of POPR-EABC and baseline methods to rank under best/worst/mean case scenario.

training scripts in code. The implementation of the policies is based on a public codebase (Raffin et al. 2021) [4].

Since the policies are from different epochs, they are clearly differentiable candidates for each task. Therefore, we use the order ranked by mean of the accumulated reward of each policy, when deployed for $n = 1000$ times running as ground truth $O\{\hat{\Pi}_{mean}\}$. Then we conduct ranking following section of Probabilistic Offline Policy Ranking, where the ranking methods' performances are measured through the evaluation metrics by comparing $O\{\hat{\Pi}\}$ and $O\{\hat{\Pi}_{mean}\}$. Table 2 contains experimental results in different environments.

The results in Table 2 show POPR achieves a higher rank correlation coefficient and cumulative gain than baseline algorithms, which means POPR can provide ranking results for different policies with higher accuracy. In addition, POPR performs the most stably, not having negative rank correlation results in all the tasks, whereas each baseline OPE algorithm has one or more negative rank correlation results.

Best/worse-case Analysis for Policies with Similar Mean Performance In this section, we evaluate POPR and baseline OPE algorithms on some open-source policies that show similar mean performance and would like to differentiate the policies through best- or worst-case performance. All the policies are publicly available and well-trained by various RL algorithms, including DQN, QRDQN, TRPO, PPO, A2C, and ARS. (Raffin 2020) [5]. Figure 3(a) shows the mean and standard deviation of their online performance with 10000 rounds of rollouts in `MountainCar`. It can be found that the performance of different policies is quite similar, while some show larger standard deviations. Under such cases where the candidate policies show similar mean performance, their best/worst case performance would be helpful in practice.

To get the ground truth ranking of the policies under best/worse case performance, we run each open-sourced policy on the same environment setting for $n = 10000$ rounds and log the reward for each round, upon which we take their lowest and highest 5% values' mean as their worst case and best case performance ground-truth: $O\{\hat{\Pi}_{worst}\}$ and $O\{\hat{\Pi}_{best}\}$. Figure 3 shows the results on environments of `MountainCar`, while we also validate in `Pendulumn` in the Appendix. From Figure 3, we observe as following:

• POPR-EABC outperforms other baseline evaluation methods with higher NDCG and SRCC results in all cases, i.e., best, worst, and mean. In the best/worst cases evaluation, POPR-EABC is able to outperform other baseline methods because it benefits from the performance posterior derived, which we could pay attention to the cases we are caring, while other OPE methods could only produce an expected policy

[4] https://github.com/DLR-RM/stable-baselines3

[5] https://github.com/DLR-RM/rl-baselines3-zoo

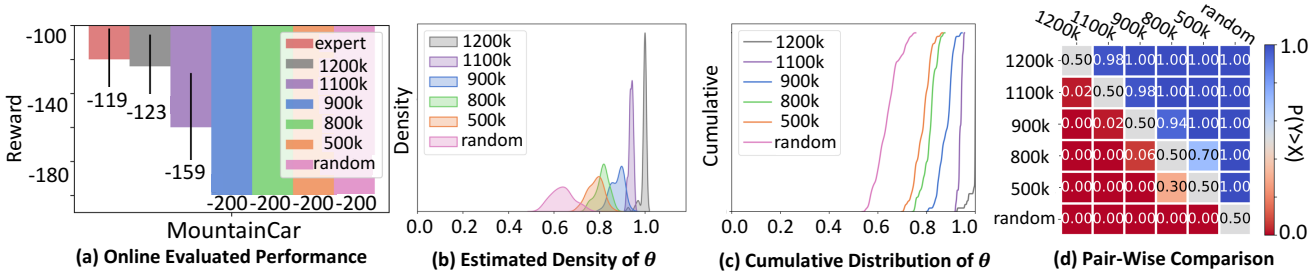


Figure 4: Probabilistic evaluation on policies. (a) Mean performance with standard deviation by rolling the policies in the online environment. Point estimates are hard to tell the differences between some policies as they have the same mean. (b) Kernel Density Estimates from the posteriors given by POPR-EABC. (c) The Cumulative Probability Estimates from POPR-EABC on different policies. The more the line closer to the lower right, the policy performs better. POPR-EABC can differentiate different levels of policy. (d) Pair-wise comparison of different policies. The darker blue the color is, the policy at Y is better than X.

value, and AgreeRank only able to produce an action similarity value. These two groups of approaches fail to effectively and correctly tell the differences under special cases.

• POPR-EABC shows smaller std in terms of NDCG and SRCC because it parameterizes the energy value variance in Eq.4 and considers such information in pseudo-likelihood to promote a stable and fast convergence to potential posterior.

Probabilistic Pairwise Comparison Different from existing work, POPR-EABC evaluates policies with detailed probabilistic values. Hence, we provide our comparison results from POPR-EABC on policies trained with different epochs in `MountainCar` as a case study. Figure 4(a) shows the *mean* rewards of each policy 100 rollouts, we could notice point estimates can not tell the differences between some policies as they have the same value of -200, indicating values from online rollouts are sometimes inadequate to differentiate between policies, requiring further evaluations on behaviors.

Figure 4(b) and 4(c) present the probabilistic evaluations by POPR-EABC with the estimated density of certain PDF, and its cumulative distribution respectively. Since θ represents the probability of the current candidate policy being as good as the potential expert policy, the faster one reaches 1 (right-top line) in Figure 4(c), the better performance of the evaluated policy, indicating there are more samples of θ closer to 1 during the evaluation of the policy. Reflected by the densities, The approximation for the probability distribution of θ is in Figure 4(b). Benefit from the (b) characterization of a holistic policy feature, in Figure 4(c), POPR-EABC differentiates between policies by estimated performance posterior.

Figure 4(d) presents the pairwise comparison of these policies given by POPR-EABC. Each cell value represents the probability of one policy from the Y-axis being better than the other from the X-axis. The value of 0 means Y has a probability of zero to be better than X and vice versa when the value is 1. The results in Figure 4(d) suggest that POPR provides effective pair-wise probabilistic analysis.

More Analysis More analyses are shown in arXiv Appendix: the effect of size and quality of expert data, different similarity measurements, and prior selection choices.

Related Work

Offline Policy Ranking is relevant to Q-function selection by choosing the best Q-function from a set of candidate functions. Different from OPE, these methods focus on Q-function, whereas in the real world, the target policy may not be in the form of a Q-function. Offline policy ranking has also been studied (Doroudi, Thomas, and Brunskill 2017; Paine et al. 2020; Jin et al. 2022), which considers point estimates rather than estimating a distribution. Another work (Yang et al. 2022) in OPR estimates the distribution by transforming it into an optimization problem with constraints, whereas this paper uses statistical simulation methods to estimate the posterior distribution. **Off-policy Evaluation** (OPE) has been focused on estimating the expected value of the target policy. Plenty of OPE methods provide point estimates for the expected value (Jiang and Li 2016; Zhang, Liu, and Whiteson 2020; Yang et al. 2020). There are some OPE methods additionally estimate the value with confidence intervals (Thomas, Theocharous, and Ghavamzadeh 2015; Kuzborskij et al. 2020; Feng et al. 2020; Dai et al. 2020; Kostrikov and Nachum 2020). Recently, another direction is estimating and bounding the CDF of returns (Chandak et al. 2021; Huang et al. 2021), although these methods are leveraging the distribution estimation, they either require knowledge of action probabilities under the behavior policy or rely on dense returns which restrict the scope of applicability.

Conclusion

This paper introduces POPR, a framework of a probability-based, statistically rigorous solution for offline policy ranking. Specifically, POPR-EABC is proposed to derive the holistic posterior of candidate policies performance as an implementation of POPR, based on an energy pseudo-likelihood, it profiles the policy behavior through a probabilistic manner, perceives action variance in Approximate Bayesian Computation process, brings awareness to the intrinsic uncertainty of the system. It helps estimate the policy's performance and facilitates probabilistic pair-wise candidate policies' comparison before deployment.

Acknowledgments

The work was partially supported by NSF award #2153311. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Agarwal, R.; Schwarzer, M.; Castro, P. S.; Courville, A. C.; and Bellemare, M. 2021. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34: 29304–29320.

Besag, J. 1975. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 24(3): 179–195.

Chandak, Y.; Niekum, S.; da Silva, B.; Learned-Miller, E.; Brunskill, E.; and Thomas, P. S. 2021. Universal off-policy evaluation. *Advances in Neural Information Processing Systems*, 34: 27475–27490.

Da, L.; Gao, M.; Mei, H.; and Wei, H. 2023. Prompt to transfer: Sim-to-real Transfer for Traffic Signal Control with Prompt Learning. *arXiv preprint arXiv:2308.14284*.

Dai, B.; Zhang, R.; Li, L.; and Schuurmans, D. 2020. GenDICE: Offline Generalized Stationary Distribution Correction Estimation.

Doroudi, S.; Thomas, P. S.; and Brunskill, E. 2017. Importance Sampling for Fair Policy Selection. In *Conference on Uncertainty in Artificial Intelligence*.

Du, W.; Ye, J.; Gu, J.; Li, J.; Wei, H.; and Wang, G. 2023. Safelight: A reinforcement learning method toward collision-free traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 14801–14810.

Dudík, M.; Langford, J.; and Li, L. 2011. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 1097–1104.

Endres, D. M.; and Schindelin, J. E. 2003. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49: 1858–1860.

Farajtabar, M.; Chow, Y.; and Ghavamzadeh, M. 2018. More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, 1447–1456. PMLR.

Feng, Y.; Ren, T.; Tang, Z.; and Liu, Q. 2020. Accountable off-policy evaluation with kernel bellman statistics. In *International Conference on Machine Learning*, 3102–3111. PMLR.

Fielitz, B. D.; and Myers, B. L. 1975. Concepts, Theory, and Techniques: Estimation of Parameters in the Beta Distribution. *Decision Sciences*, 6(1): 1–13.

Fu, J.; Norouzi, M.; Nachum, O.; Tucker, G.; Wang, Z.; Novikov, A.; Yang, M.; Zhang, M. R.; Chen, Y.; Kumar, A.; et al. 2021. Benchmarks for deep off-policy evaluation. *arXiv preprint arXiv:2103.16596*.

Hanna, J.; and Stone, P. 2017. Grounded action transformation for robot learning in simulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Harutyunyan, A.; Bellemare, M. G.; Stepleton, T.; and Munos, R. 2016. Q(λ) with Off-Policy Corrections. In *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings*, 305–320. Springer.

Huang, A.; Leqi, L.; Lipton, Z.; and Azizzadenesheli, K. 2021. Off-policy risk assessment in contextual bandits. *Advances in Neural Information Processing Systems*, 34: 23714–23726.

Jayawardana, V.; Tang, C.; Li, S.; Suo, D.; and Wu, C. 2022. The impact of task underspecification in evaluating deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 23881–23893.

Jenkins, P.; Wei, H.; Jenkins, J. S.; and Li, Z. 2020. A probabilistic simulator of spatial demand for product allocation. *arXiv preprint arXiv:2001.03210*.

Jenkins, P.; Wei, H.; Jenkins, J. S.; and Li, Z. 2022. Bayesian Model-Based Offline Reinforcement Learning for Product Allocation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 12531–12537.

Jiang, N.; and Li, L. 2016. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, 652–661. PMLR.

Jin, Y.; Zhang, Y.; Qin, T.; Zhang, X.; Yuan, J.; Li, H.; and Liu, T.-Y. 2021. Supervised off-policy ranking. *arXiv preprint arXiv:2107.01360*.

Jin, Y.; Zhang, Y.; Qin, T.; Zhang, X.; Yuan, J.; Li, H.; and Liu, T.-Y. 2022. Supervised Off-Policy Ranking. In *International Conference on Machine Learning*, 10323–10339. PMLR.

Kim, B.; Farahmand, A.-m.; Pineau, J.; and Precup, D. 2013. Learning from limited demonstrations. *Advances in Neural Information Processing Systems*, 26.

Kostrikov, I.; and Nachum, O. 2020. Statistical bootstrapping for uncertainty estimation in off-policy evaluation. *arXiv preprint arXiv:2007.13609*.

Kuzborskij, I.; Vernade, C.; Gyorgy, A.; and Szepesvari, C. 2020. Confident Off-Policy Evaluation and Selection through Self-Normalized Importance Weighting. In *International Conference on Artificial Intelligence and Statistics*.

Le, H. M.; Voloshin, C.; and Yue, Y. 2019. Batch Policy Learning under Constraints. In *International Conference on Machine Learning*, 3703–3712. PMLR.

Li, L.; Chu, W.; Langford, J.; and Wang, X. 2010. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Web Search and Data Mining*.

Li, S.; Mei, H.; Li, J.; Wei, H.; and Xu, D. 2023. Toward Efficient Traffic Signal Control: Smaller Network Can Do More. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, 8069–8074. IEEE.

Liao, P.; Greenewald, K.; Klasnja, P.; and Murphy, S. 2020. Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1): 1–22.

Liu, Y.; Liu, Q.; Zhao, H.; Pan, Z.; and Liu, C. 2020. Adaptive quantitative trading: An imitative deep reinforcement

learning approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 2128–2135.

Munos, R.; Stepleton, T.; Harutyunyan, A.; and Bellemare, M. 2016. Safe and efficient off-policy reinforcement learning. *Advances in neural information processing systems*, 29.

Nachum, O.; Chow, Y.; Dai, B.; and Li, L. 2019. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in Neural Information Processing Systems*, 32.

Paduraru, C. 2012. Off-policy Evaluation in Markov Decision Processes.

Paine, T. L.; Paduraru, C.; Michi, A.; Gulcehre, C.; Zolna, K.; Novikov, A.; Wang, Z.; and de Freitas, N. 2020. Hyperparameter Selection for Offline Reinforcement Learning. *ArXiv*, abs/2007.09055.

Precup, D. 2000. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, 80.

Precup, D.; Sutton, R. S.; and Singh, S. 2000. Eligibility Traces for Off-Policy Policy Evaluation. In *International Conference on Machine Learning*.

Raffin, A. 2020. RL Baselines3 Zoo. https://github.com/DLR-RM/rl-baselines3-zoo.

Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; and Dormann, N. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *J. Mach. Learn. Res.*, 22: 268:1–268:8.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Thomas, P.; and Brunskill, E. 2016. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, 2139–2148. PMLR.

Thomas, P.; Theocharous, G.; and Ghavamzadeh, M. 2015. High-confidence off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Turner, B. M.; and Van Zandt, T. 2012. A tutorial on approximate Bayesian computation. *Journal of Mathematical Psychology*, 56(2): 69–85.

Vlachogiannis, D. M.; Wei, H.; Moura, S.; and Macfarlane, J. 2023. HumanLight: Incentivizing Ridesharing via Human-centric Deep Reinforcement Learning in Traffic Signal Control. *arXiv preprint arXiv:2304.03697*.

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

Wang, Y.; Wang, L.; Li, Y.; He, D.; Chen, W.; and Liu, T.-Y. 2013. A theoretical analysis of NDCG ranking measures. In *Proceedings of the 26th annual conference on learning theory (COLT 2013)*, volume 8, 6.

Wei, H.; Chen, J.; Ji, X.; Qin, H.; Deng, M.; Li, S.; Wang, L.; Zhang, W.; Yu, Y.; Linc, L.; et al. 2022. Honor of kings arena: an environment for generalization in competitive reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 11881–11892.

Xie, T.; Ma, Y.; and Wang, Y.-X. 2019. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. *Advances in Neural Information Processing Systems*, 32.

Yang, M.; Dai, B.; Nachum, O.; Tucker, G.; and Schuurmans, D. 2022. Offline policy selection under uncertainty. In *International Conference on Artificial Intelligence and Statistics*, 4376–4396. PMLR.

Yang, M.; Nachum, O.; Dai, B.; Li, L.; and Schuurmans, D. 2020. Off-Policy Evaluation via the Regularized Lagrangian. *Advances in Neural Information Processing Systems*, 33: 6551–6561.

Zhang, S.; Liu, B.; and Whiteson, S. 2020. GradientDICE: Rethinking Generalized Offline Estimation of Stationary Values. 11194–11203.