

ORIGINAL RESEARCH

Open access

Offline Reinforcement Learning with Implicit Policy Constraints and Energy-Based Modeling

Daniel Kim^{1*}, Jihoon Park¹, Min-seo Kang², Hyun-woo Jung³

Abstract

Offline reinforcement learning (RL) focuses on training policies exclusively from passively gathered datasets, establishing it as a data-driven decision-making framework. A major obstacle in offline RL is the phenomenon of distribution shift, which leads the algorithm to encounter out-of-distribution (OOD) samples. While constraining the divergence between the target policy and the behavior policy can alleviate distribution shift, this strategy can unduly restrict the target policy and degrade algorithmic performance, because it fails to explicitly differentiate between in-distribution and OOD samples. Furthermore, accurately learning and representing a multi-modal behavior policy is highly challenging when datasets are compiled from multiple distinct behavior policies. To resolve these limitations, the authors tackle the distribution shift problem through implicit policy constraints utilizing energy-based models (EBMs), bypassing the explicit modeling of the behavior policy. EBMs offer a powerful framework for characterizing complex multi-modal distributions while effectively distinguishing between in-distribution and OOD samples. Experimental outcomes demonstrate that this approach substantially surpasses explicit policy constraint methods and other baseline techniques. Furthermore, the trained energy model serves as an effective indicator of OOD encounters, providing alerts for potential failures.

Keywords Offline Reinforcement Learning (RL), Distribution shift, Out-Of-Distribution (OOD) samples, Energy-Based Models (EBMs)

*Correspondence:

Daniel Kim

daniel.kim@gmail.com

¹ Department of Offline Reinforcement Learning and Policy Constraints, College of Engineering, Seoul National University, Seoul, South Korea

² Department of Energy-Based Modeling and RL, College of Engineering, Chung-Ang University, Seoul, South Korea

³ Department of Implicit Policy Learning, College of Engineering, Kangwon National University, Chuncheon, South Korea

Introduction

Reinforcement learning (RL) operates via a trial-and-error process that necessitates extensive interaction with the environment. Consequently, simulators are frequently utilized to enable rapid and cost-effective interactions, as seen in gaming environments [1-3] and robotic simulations [4-6]. Nonetheless, developing high-fidelity simulators remains exceptionally difficult for numerous real-world domains, such as healthcare or autonomous driving. By contrast, gathering passive datasets within real-world environments is highly accessible. A primary driver behind the success of computer vision and natural language processing, compared to traditional RL, is their entirely data-driven learning paradigm, which leverages massive datasets to full advantage. Offline RL seeks to optimize policies directly from static datasets, offering a highly encouraging framework for deploying data-driven decision methods across a broader spectrum of real-world challenges.

Behavior cloning [7] represents a supervised learning-style, data-driven decision method that has seen widespread deployment in autonomous land vehicles [8, 9] and robotics [10, 11]. However, behavior cloning depends on costly expert demonstrations, whereas practical datasets are frequently non-expert or suboptimal. Crucially, the expense of recording these suboptimal datasets is substantially lower than that of producing meticulous expert demonstrations. Offline RL can extract effective policies from such suboptimal data and ultimately surpass the performance of the original behavior policy. A core challenge in offline RL is the distribution shift [12] problem, wherein the state-action distribution generated by the target policy diverges from the distribution produced by the behavior policy. In this context, the target policy is optimized by the offline RL algorithms, while the behavior policy is responsible for data collection. This distribution shift causes the agent to encounter out-of-distribution (OOD) state-actions. Because value estimations for OOD samples are inaccurate and prone to inflation, these overestimated Q-values are exploited during greedy policy improvement phases, eventually leading to Q-value divergence via bootstrapping updates [13, 14].

While the distribution shift problem can be resolved in online RL by continuously updating the replay buffer, the replay buffer in offline RL remains completely static due to the absence of further environmental interaction. Policy constraint-based offline RL techniques manage distribution shift by restricting the divergence between the target policy and the behavior policy [13–16]. Nevertheless, directly enforcing a behavior policy-based constraint introduces several shortcomings. First, this strategy can over-constrain the target policy and suppress its performance potential, as it lacks a mechanism to directly separate in-distribution samples from OOD samples. Second, available datasets are frequently gathered from a variety of distinct behavior policies, making an explicit representation of the multi-modal behavior policy exceptionally difficult. To overcome these challenges, we introduce an Implicit Policy Constraint (IPC) framework that leverages energy-based models (EBMs) [17]. EBMs have been effectively deployed for OOD detection [18, 19] by allocating low energy metrics to in-distribution data and high energy metrics to OOD data. By integrating the EBM into offline RL, we aim to relax the excessive constraints typically imposed on the target policy. Rather than explicitly learning the behavior policy, we utilize the energy value as an unnormalized probability to implicitly characterize it, yielding a robust framework for representing multi-modal behavior policies.

The primary contributions of this work are summarized as follows. First, we incorporate the EBM into offline RL to explicitly characterize OOD samples and model multi-modal behavior policies. To our knowledge, this work represents the initial effort to implement EBMs within an offline RL framework. Second, we introduce the energy-based IPC algorithm, termed IPC, and analytically establish the policy optimization objective of IPC using KL divergence. Third, we assess the performance of the proposed algorithm on the D4RL Gym benchmarks; the empirical results demonstrate that our energy-based IPC method significantly outperforms direct behavior policy-based approaches as well as other established baselines. Finally, we confirm the capacity of the trained EBM to detect OOD visitations and issue warnings regarding potential imminent failures within a practical deployment setting.

Related Work

Offline RL

Our study is closely connected to policy constraint-based offline RL [12, 20] strategies. Within this category, BCQ [13] incorporates a conditional variational auto-encoder [21, 22] generative framework to produce candidate actions, thereby preventing the exploitation of OOD errors. While we similarly adopt a generative framework, the EBM, our approach utilizes it to calculate energy values rather than to output actions. BEAR [14] restricts the target policy by leveraging the support set of the behavior policy instead of action probabilities, which softens the intensity of the policy constraints. BRAC [16] evaluates multiple policy constraint variants—including kernel MMD, KL divergence, and Wasserstein distance—and demonstrates that these distinct divergence metrics yield comparable performance. TD3BC [15] introduces a streamlined policy constraint framework by incorporating a behavior cloning mean square error loss component directly into the actor optimization objective. WPC [23] moderates the severity of policy constraints by applying weights to the constraint, thereby enhancing the baseline performance of TD3BC. In Fisher-BRC [24], the behavior policy is explicitly trained to provide regularization for both the actor and the critic.

Critic regularization represents another heavily investigated domain in offline RL. CQL [25] penalizes the values of OOD actions to establish lower bounds for the true values, sampling OOD actions simultaneously from the current policy and a uniform distribution. Our methodology also involves sampling OOD actions to train the EBM; however, unlike CQL, our trained EBM operates independently of the Q-function and serves to constrain the policy rather than regularize the critic network. MCQ [26] reduces over-pessimism in Q-values by proactively training OOD actions with appropriate pseudo-target values. EDAC [27] and PBRL [28] utilize ensemble-based uncertainty estimation to resolve the OOD value estimation issue. RORL [29] presents an innovative conservative smoothing method for perturbed states, enhancing resilience against adversarial observation perturbations. Rather than explicitly forcing conservation onto OOD Q-values, ConsNet [30] introduces a novel critic network architecture designed to automatically generate conservative estimations for OOD samples.

Because value overestimation stems from encountering OOD actions, IQL [31] employs SARSA-like updates alongside expectile regression to optimize the Q-function, utilizing exclusively the actions present in the datasets. SAW [32] estimates the state advantage via expectile regression within a QSS-style [33] learning framework. VEM [34] introduces expectile value learning to ensure the training process stays within the support of the offline datasets. Decision Transformer (DT) [35] and TT [36] apply the Transformer network architecture to offline RL, treating the offline task as a sequential modeling problem. More recently, diffusion probabilistic models have been integrated into offline RL. The diffuser framework [37] models an entire trajectory analogously to an image and executes the initial action of the trajectory sequence. Diffusion-QL [38] leverages the diffusion model to train a highly expressive target policy, utilizing Q-values to guide the generative process. Furthermore, data augmentation can enhance existing offline RL algorithms. ROMI [39] creates conservative rollouts by training a reverse dynamics model, offering informed data augmentation for model-free offline RL. CABI [40] synthesizes reliable OOD transitions by cross-checking the forward and inverse dynamics models, boosting the generalization performance of model-free offline RL. Lastly, TATU [41] truncates synthetic trajectories based on accumulated uncertainty to yield reliable imagined trajectories.

Energy-based Model

EBMs [17, 42] represent unnormalized probabilistic frameworks that capture dependencies by assigning a scalar energy value to each configuration of variables. SQL [43] adopts an energy-based distribution to learn intricate multi-modal policies, thereby improving exploration capabilities and compositionality. JEM [44] reinterprets a standard discriminative classifier of $p(y|x)$ as an EBM representing the joint distribution $p(x, y)$, using the resulting probabilistic scores for downstream OOD detection. Liu *et al.* [18] directly utilize non-probabilistic energy scores as a scoring function to identify OOD samples. Elflein Sven [19] analyzes the OOD detection performance of EBMs relative to alternative generative approaches, showing that architectural adjustments and supervision can enhance OOD detection capability. Florence *et al.* [45] introduce an implicit behavior cloning approach by training an EBM and deriving actions from it, demonstrating that formulating supervised policy learning with an implicit model generally outperforms standard explicit alternatives.

Preliminaries

Offline RL

In RL, the interaction between an agent and its environment is typically formulated as a Markov decision process (MDP). An MDP is represented by the tuple $(S, A, P, r, \rho, \gamma)$, where S and A denote the state space and action space, respectively, r signifies the immediate reward, P represents the transition dynamics, ρ indicates the initial state distribution, and γ is the discount factor. The policy $\pi(a|s)$ establishes the probability distribution over an action a given a state s . The primary objective of the RL agent is to optimize a policy that maximizes the expected cumulative discounted reward within the designated MDP:

$$J(\pi) = \mathbb{E}_{s_0 \sim p, a_t \sim \pi(s_t), s_{t+1} \sim P(\cdot | s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t) \right]. \quad (1)$$

Contemporary RL frameworks optimize policies utilizing the actor–critic paradigm. For any specified policy π , a value function is trained to approximate the discounted reward via the Bellman operator T^π :

$$T^\pi Q(s, a) = \mathbb{E}_{s' \sim P(\cdot | s, a), a' \sim \pi(\cdot | s')} [r(s, a) + \gamma Q(s', a')]. \quad (2)$$

The objective for policy optimization is defined as follows:

$$\pi_\theta := \operatorname{argmax}_{\pi_\theta} \mathbb{E}_{s \sim \mathcal{B}, a \sim \pi_\theta(s)} [Q(s, a)], \quad (3)$$

where $\mathcal{B}$ represents the replay buffer.

In the context of offline RL, value estimation errors are exploited by the greedy actor as described in [Eq. 3](#). When integrated with value bootstrapping, the Q-function can diverge toward infinitely large values [\[13, 14\]](#). Off-policy RL addresses this vulnerability by periodically refreshing the replay buffer to rectify inflated Q-values. Conversely, the replay buffer in offline RL is never updated. Prior offline RL frameworks have introduced various strategies to prevent the agent from visiting these inaccurately estimated state-action pairs, namely the OOD samples. Policy constraint approaches tackle this issue by bounding the divergence between the target policy and the behavior policy:

$$\pi_\theta := \operatorname{argmax}_{\pi_\theta} \mathbb{E}_{s \sim \mathcal{B}, a \sim \pi_\theta(s)} [Q(s, a) - \beta D(\pi_\theta(s), \pi_b(s))], \quad (4)$$

where β serves as a hyperparameter to control the constraint intensity, and D represents a divergence metric such as the KL divergence, maximum mean discrepancy, or Wasserstein distance. TD3BC [\[15\]](#) represents a minimalist formulation of policy constraint-based offline RL, directly utilizing the behavior cloning loss as the divergence metric as follows:

$$\pi_\theta := \operatorname{argmax}_{\pi_\theta} \mathbb{E}_{s, a \sim \mathcal{B}} [Q(s, \pi_\theta(s)) - \beta (\pi_\theta(s) - a)^2]. \quad (5)$$

Energy-based models

EBMs generate a scalar energy metric $E_\omega(x, y): R^{m+n} \rightarrow R$ for a given sample (x, y) ; it captures the dependency between the variables x and y by assigning lower energy values to dataset samples and higher energy values to OOD configurations. This energy value can be converted into a conditional probability density $p(y|x)$ via the Gibbs distribution:

$$p(y|x) = \frac{e^{-E_\omega(x, y)/T}}{Z_\omega(x)}, \quad (6)$$

where $E_\omega(x, y)$ represents the energy model, T denotes the temperature parameter, and $Z_\omega(x)$ signifies the partition function defined as:

$$Z_w(x) = \int e^{-E_w(x,y)/T} dy. \quad (7)$$

Eq. 6 establishes a direct link between energy values and the behavior policy; thus, EBMs can be interpreted as an implicit, unnormalized behavior policy representation or an unnormalized action density estimator for the datasets.

Materials and Methods

Because the EBM depends exclusively on the dataset, we separate the training process into two distinct phases. First, an EBM is optimized on the offline dataset to accurately fit the underlying data distribution. The model is trained to map dataset samples to low energy values and OOD samples to high energy values. Subsequently, the trained EBM is utilized to implicitly represent the behavior policy and enforce constraints on the target policy.

Learning the EBM

Various strategies exist for optimizing EBM frameworks [42], such as maximum likelihood training with Markov chain Monte Carlo, score matching, and noise contrastive estimation. In this study, we follow the approach established in implicit behavior cloning (BC) [45] by employing an InfoNCE-style [46] loss function to train our EBM:

$$\begin{aligned} \mathcal{L}_{EBM} &= -\frac{1}{N} \sum_{i=1}^N \log p(a_i | s_i) = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{-E_w(s_i, a_i)/T}}{Z_w(s_i)} \\ &= -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{-E_w(s_i, a_i)/T}}{e^{-E_w(s_i, a_i)/T} + \sum_{j=1}^{N^{neg}} e^{-E_w(s_i, \tilde{a}_j)/T}} \end{aligned} \quad (8)$$

where N represents the batch size and N^{neg} denotes the number of negative samples. Given that the EBM is designed to assign lower energy values to dataset instances and higher energy values to OOD instances, negative samples function as OOD representations. For each dataset state s_i , we draw N^{neg} actions from a uniform distribution $[a_{min}, a_{max}]$. Alternative distributions for negative sampling, such as a Gaussian distribution, can also be utilized. Minimizing this InfoNCE-style loss function simultaneously suppresses the energy of dataset samples and inflates the energy of negative samples.

The behavior policy represented through energy in **Eq. 8** demonstrates that it concurrently maximizes the probabilities of dataset actions and minimizes the density of negative actions. By contrast, explicit behavior policy modeling techniques—such as optimizing a Gaussian behavior policy—focus solely on maximizing the likelihood probabilities of dataset actions, which can inadvertently assign high probabilities to OOD actions. Consequently, policy constraints applied to these OOD actions become ineffective. The energy-based conditional action probabilities can be interpreted as a dataset density estimation, which varies slightly from the actual behavior policy. In practice, identifying the ground-truth behavior policy is impossible because we only possess finite trajectory samples from it. We therefore estimate the dataset density as a close approximation of the behavior policy.

Energy-based implicit policy constraint

The behavior policy can be characterized by the EBM, as shown in **Eq. 6**. Consider the KL divergence between the target policy and the behavior policy:

$$\begin{aligned}
 D_{KL}(\pi_\theta(s), \pi_b(s)) &= \int \pi_\theta(a|s) \log \frac{\pi_\theta(a|s)}{\pi_b(a|s)} da \\
 &= - \int \pi_\theta(a|s) \log \pi_b(a|s) da - H(\pi_\theta(\cdot|s)) \\
 &= - \int \pi_\theta(a|s) \log \frac{e^{-E_\omega(s,a)/T}}{Z_\omega(s)} da - H(\pi_\theta(\cdot|s)) \\
 &= \mathbb{E}_{a \sim \pi_\theta(s)} [E_\omega(s, a)/T + \log \pi_\theta(s)] + Z_\omega(s),
 \end{aligned} \tag{9}$$

where H represents the policy entropy. Consequently, the optimization objective for the energy-based IPC can be formulated as follows:

$$\begin{aligned}
 \pi_\theta &:= \operatorname{argmax}_{\pi_\theta} \mathbb{E}_{s \sim \mathcal{B}, a \sim \pi_\theta(s)} [Q(s, a) - \beta D_{KL}(\pi_\theta(s), \pi_b(s))] \\
 &= \operatorname{argmax}_{\pi_\theta} \mathbb{E}_{s \sim \mathcal{B}, a \sim \pi_\theta(s)} [Q(s, a) - \beta (E_\omega(s, a)/T + \log \pi_\theta(s))],
 \end{aligned} \tag{10}$$

where the partition factor $Z_\omega(s)$ is omitted since it does not influence the policy optimization gradient.

The energy-based IPC naturally derives an entropy maximization component, which closely mirrors maximum entropy RL frameworks like SAC [47]. It additionally incorporates an energy minimization objective. The trained energy function $E_\omega(s, a)$ operates as a differentiable neural network—similar to a critic network—enabling action gradients over a to backpropagate via the chain rule. Because the energy function is optimized to yield low energy values for dataset samples, minimizing energy forces the policy to remain within low-energy regions, thereby restricting it to the in-distribution space. The hyperparameter β regulates the trade-off between maximizing Q-values and mitigating the risk of selecting OOD actions.

Ensemble-based uncertainty estimation techniques [27, 28] are frequently deployed in existing offline RL methods to conservatively approximate OOD Q-values. Conversely, the EBM utilizes a single neural network rather than an ensemble to isolate OOD samples, which incurs a lower computational overhead and eliminates the need to tune ensemble sizes. Compared to the conservative Q-learning algorithm CQL [25], our approach directly isolates in-distribution and OOD samples using the trained EBM, whereas CQL prevents OOD selection by actively depressing OOD Q-values. Because the EBM functions independently of the Q-values, it can be utilized to monitor whether the current state-action pair stays within or deviates from the dataset distribution, providing a mechanism to flag potential failures.

Algorithm summary

The execution of the IPC algorithm is structured into two independent phases. Initially, an EBM is optimized to capture the dataset distribution; subsequently, this EBM is integrated into the policy optimization objective to confine the target policy within the dataset distribution. We employ the InfoNCE-style loss function defined in Eq. 8 to optimize the EBM model. For each state extracted from the dataset, we generate negative samples by drawing 200 actions from the uniform distribution $[a_{\min}, a_{\max}]$. The IPC offline RL framework is constructed on top of the SAC architecture, with the energy minimization component appended as an auxiliary loss term as illustrated in Eq. 10. The corresponding pseudocode routines are detailed in Algorithm 1 and Algorithm 2.

```

for  $i = 1$  to  $\text{max\_steps}$  do
    Sample batch  $B = (s, a, r, s', d)$  from
    dataset  $\mathcal{D}$ 
     $y = r + \gamma(1 - d)\min_{\bar{\phi}_{1,2}} Q_{\bar{\phi}_{1,2}}(s', \tilde{a}'), \tilde{a}' \sim \pi_{\theta}(s')$ 
     $\phi_{1,2} \leftarrow \operatorname{argmin}_{\phi_{1,2}} \frac{1}{|B|} \sum_B \left( Q_{\phi_{1,2}}(s, a) - y \right)^2$ 
     $\theta \leftarrow \operatorname{argmax}_{\theta} \frac{1}{|B|} \sum_B \min_{\phi_{1,2}} Q_{\phi_{1,2}}(s, \tilde{a})$ 
     $-\beta \left( E_{\omega}(s, \tilde{a}) / T + \log \pi_{\theta}(\tilde{a} | s) \right), \tilde{a} \sim \pi_{\theta}(s)$ 
     $\bar{\phi}_{1,2} \leftarrow (1 - \tau)\bar{\phi}_{1,2} + \tau\phi_{1,2}$ 
End

```

Algorithm 1.

```

Initialise: Initialise energy networks  $E_{\omega}$ 
and replay buffer  $\mathcal{D}$ 
Setting hyperparameter  $N^{\text{neg}}$ 
for  $i = 1$  to  $\text{max\_steps}$  do
    Sample batch  $B = \{s_i, a_i\}_{i=1}^N$  from dataset  $\mathcal{D}$ 
    Sample  $N^{\text{neg}}$  negative actions  $\tilde{a}_i$  for each
    state  $s_i$  from  $U(a_{\min}, a_{\max})$ 
    Calculate  $\mathcal{L}_{EBM}$  loss using Equation (8)
     $\omega \leftarrow \operatorname{argmin}_{\omega} \mathcal{L}_{EBM}$ 
End

```

Algorithm 2. Learning EBN

Experiments

In this section, we initially benchmark the performance of IPC against established offline baselines, including BC, best trajectories BC (10%BC), DT [35], RvS-R [48], OneStep [49], CQL [25], and TD3BC [15]. Behavior cloning serves as a standard baseline since the objective of offline RL is to produce a target policy that surpasses the performance of basic behavioral imitation. The 10%BC baseline applies behavior cloning exclusively to the top 10% of trajectories within the datasets, which markedly improves performance relative to imitating an entirely suboptimal dataset. DT formulates reinforcement learning as a sequence modeling task and resolves it using a Transformer network architecture, creating a return-conditioned autoregressive model. The RvS-R algorithm operates on a similar concept to DT but employs straightforward MLP networks. The OneStep method circumvents value error exploitation stemming from iterative off-policy evaluation by computing the Q-value only a single time. CQL enforces conservative value estimations to prevent encountering OOD actions, while TD3BC constrains the policy by appending a behavior cloning loss term. Following these evaluations, we perform ablation studies to evaluate how varying the intensity of implicit policy constraints impacts performance. We also examine alternative training paradigms for the EBM, comparing independent training against joint optimization with the policy. Lastly, we analyze the capability of the trained EBM to reliably detect OOD state-actions and predict imminent failures within the evaluation environment.

Performance on offline RL benchmarks

The D4RL [50] Gym benchmarks encompass three distinct environments: 'halfcheetah', 'hopper', and 'walker2d', as illustrated in **Figure 1**. Each environment provides several types of datasets: 'medium' datasets are compiled by capturing 1 million samples from a partially trained SAC policy; 'random' datasets are generated using a randomly initialized policy; 'medium-replay' datasets consist of the replay buffer states recorded during training until the policy achieves 'medium' level proficiency; and 'medium-expert' datasets combine equal proportions of expert demonstrations and suboptimal data.

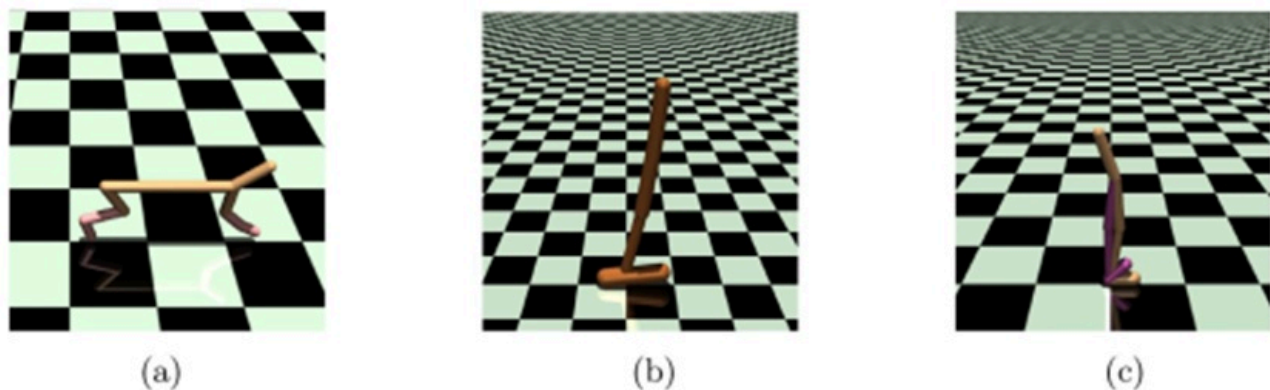


Figure 1. The D4RL Gym environments. (a) Halfcheetah. (b) Hopper (c) Walker2d.

Compared to the behavior cloning-based policy constraint framework TD3BC, the IPC algorithm delivers superior performance across the majority of tasks, showing particularly strong gains on the 'halfcheetah-random', 'hopper-medium', and 'hopper-medium-replay' datasets. Our approach also outperforms the conservative value-based framework CQL, illustrating the capability of policy constraint methods when constraints are effectively structured.

Experimental details

To evaluate performance, we execute 5 independent random seeds for both the TD3BC and the IPC frameworks, presenting the resulting mean normalized scores [50] in **Table 1**. Performance metrics for BC, 10%BC, DT, OneStep, and RvS-R are extracted directly from the RvS literature, whereas the data for IQL and CQL are sourced from their respective

original publications. The IPC model is trained over 1 million gradient steps, with evaluations conducted every 5000 steps across an average of 10 episodes. Training for the EBM spans 500 thousand steps. We employ a temperature hyperparameter $T = 1$ and perform a grid search for the hyperparameter β across the set $\{0.1, 0.5, 1.0, 5.0\}$. Ultimately, $\beta = 0.5$ is selected for all experimental tasks.

Table 1. Performance comparison between Implicit policy constraint and previous offline reinforcement learning baselines on the D4RL gym datasets; averaged normalised scores are reported.

Benchmark Task (-v2)	IPC	TD3+BC	Conservative Q-Learning (CQL)	OneStep RL	RvS-R	Decision Transformer (DT)	10% BC	sBehavioral Cloning (BC)
halfcheetah-random	20.9 ± 1.3	11.7 ± 1.2	18.6	6.9	3.9	2.2	2.0	2.3
hopper-random	8.1 ± 1.8	8.6 ± 0.4	9.1	7.8	0.2	7.5	4.1	4.8
walker2d-random	8.5 ± 2.6	0.9 ± 1.0	2.5	6.1	7.7	2.0	1.7	1.7
halfcheetah-medium	49.0 ± 0.1	48.2 ± 0.2	49.1	55.6	41.6	42.6	42.5	42.6
hopper-medium	73.1 ± 5.7	57.7 ± 1.5	64.6	83.3	60.2	67.6	56.9	52.9
walker2d-medium	85.3 ± 1.7	83.2 ± 0.7	82.9	85.6	71.7	74.0	75.0	75.3
halfcheetah-medium-replay	45.3 ± 0.3	44.6 ± 0.3	47.3	42.4	38.0	36.6	40.6	36.6
hopper-medium-replay	101.6 ± 1.3	67.4 ± 5.5	97.8	71.0	73.5	82.7	75.9	18.1
walker2d-medium-replay	92.2 ± 4.3	83.7 ± 4.7	86.1	71.6	60.6	66.6	62.5	26.0
halfcheetah-medium-expert	94.7 ± 0.5	90.7 ± 1.6	85.8	93.5	92.2	86.8	92.9	55.2
hopper-medium-expert	107.4 ± 9.1	102.0 ± 6.8	102.1	106.1	101.7	107.6	110.9	52.5
walker2d-medium-expert	109.1 ± 0.6	110.1 ± 0.2	109.5	110.9	106.0	108.1	109.0	107.5
Aggregate Score	795.2	712.9	755.5	736.8	657.3	684.3	674.0	475.5

• Note: Bold indicates the highest score. The variances of TD3BC and IPC are also reported.

Ablation experiments

Hyperparameter ablation experiments

This section examines how the intensity of policy constraints, regulated by the hyperparameter β , affects overall performance. The respective learning trajectories under varying values of β are illustrated in **Figure 2**. When $\beta = 0$, no constraint is applied to the policy; consequently, the IPC framework reverts to a standard off-policy RL approach and fails to succeed in the majority of tasks. Comparable outcomes are observed when setting β to either 0.5 or 1.0. However, escalating β further to 5.0 causes a decline in performance, demonstrating that the model becomes excessively restricted under this condition.

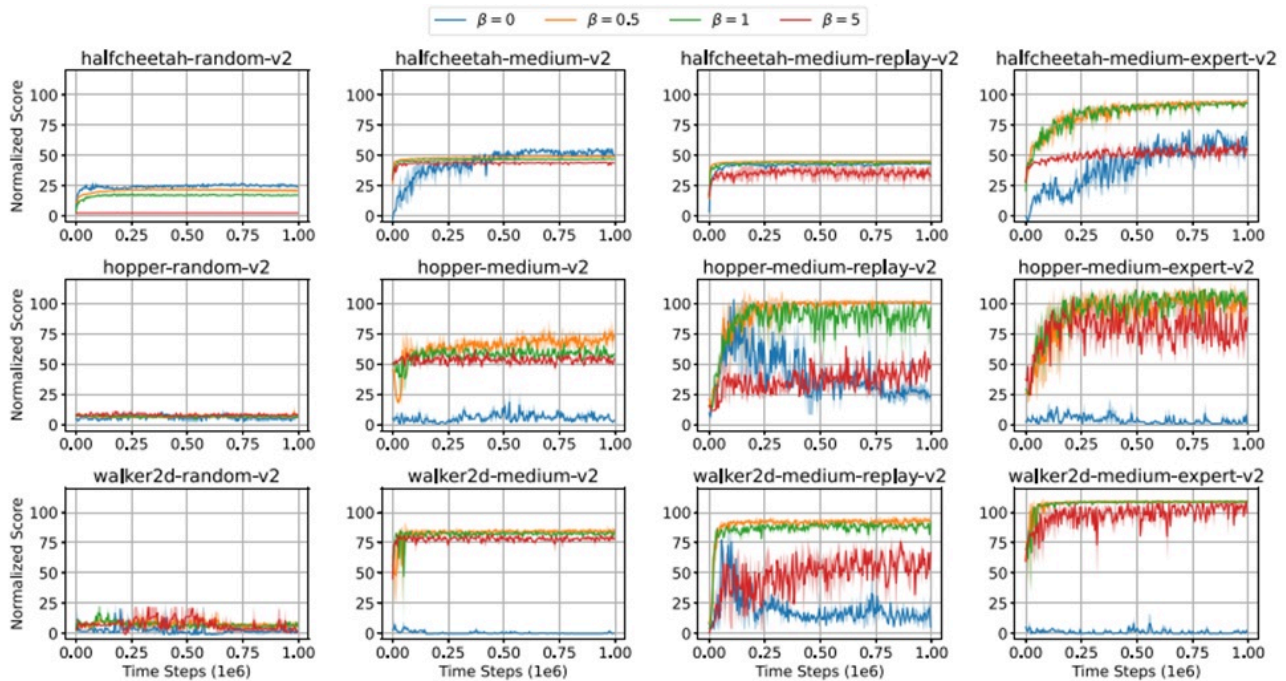


Figure 2. Learning curves of the implicit policy constraint algorithm under different strengths of policy constraints on the D4RL Gym datasets. The greater β is, the stronger the policy constraint. Each curve is averaged over 3 random seeds and the standard deviations are indicated by shaded areas.

Separately training versus jointly training

The training procedure for IPC is decoupled into two distinct phases: initially optimizing the EBM on the offline data, and subsequently integrating it into the policy optimization objective function. Throughout the initial EBM training phase, negative samples are drawn uniformly from a distribution. In the subsequent phase, the EBM network parameters are kept fixed and receive no further updates. An alternative strategy involves optimizing the EBM and the policy concurrently. Under this joint configuration, negative samples are drawn from the current policy, updating the EBM simultaneously alongside the actor and critic networks. Because the active policy evolves continuously, the negative sample distribution shifts throughout the training process. This dynamic is comparable to the OOD sampling strategy utilized in CQL, where OOD samples are generated from the active policy. The learning curves contrasting the decoupled and simultaneous training strategies are depicted in **Figure 3**. Both approaches yield comparable performance, demonstrating that the IPC framework remains highly robust to the underlying EBM configurations. Decoupled training is computationally more efficient because uniform distribution sampling is considerably faster than policy-based sampling. Nevertheless, joint

training offers a prospective benefit in high-dimensional action spaces, where sampling directly from the active policy can prove more efficient.

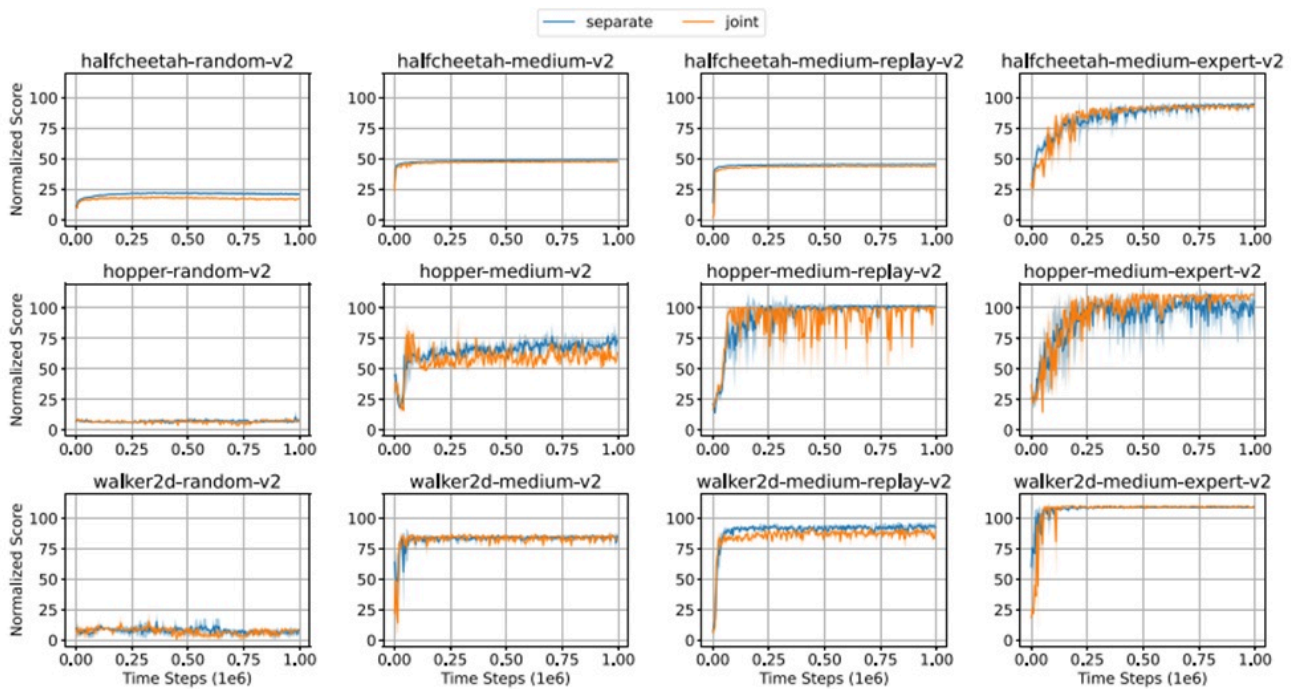


Figure 3. Learning curves of the implicit policy constraint algorithm with separately and jointly trained energy-based model. Each curve is averaged over 3 random seeds and the standard deviations are indicated by shaded areas.

OOD indication in test environment

Because policies in offline RL are optimized entirely using static datasets, keeping the generated trajectories within the support of the training distribution is vital. When deploying the trained policy within a test environment, it might encounter OOD states and execute sub-optimal actions, leading to a reduction in performance. Given that EBM's possess inherent capabilities for OOD detection [18], the optimized EBM can serve as an indicator to monitor whether the agent enters OOD states or remains within the training data distribution during test-time deployment.

Figure 4 illustrates the energy profiles of 5 randomized trajectories within the 'walker2d-medium' environment, where the policy achieves a mean score of approximately 75.0 via the IPC framework. The maximum horizon for the 'walker2d-medium' environment is capped at 1000 steps; here, the agent is rewarded for forward locomotion and suffers termination upon falling. Three specific trajectories (represented by the red, blue, and green curves) exhibit elevated energy levels during intermediate timesteps and terminate prematurely, resulting in diminished scores. Conversely, the remaining trajectories maintain consistently low energy profiles, survive for the full duration of the episode, and achieve superior scores relative to the prematurely terminated runs. Evaluations of a policy optimized to peak convergence demonstrate that all produced trajectories maintain uniformly low energy levels, confirming that the trained policy remains strictly constrained.

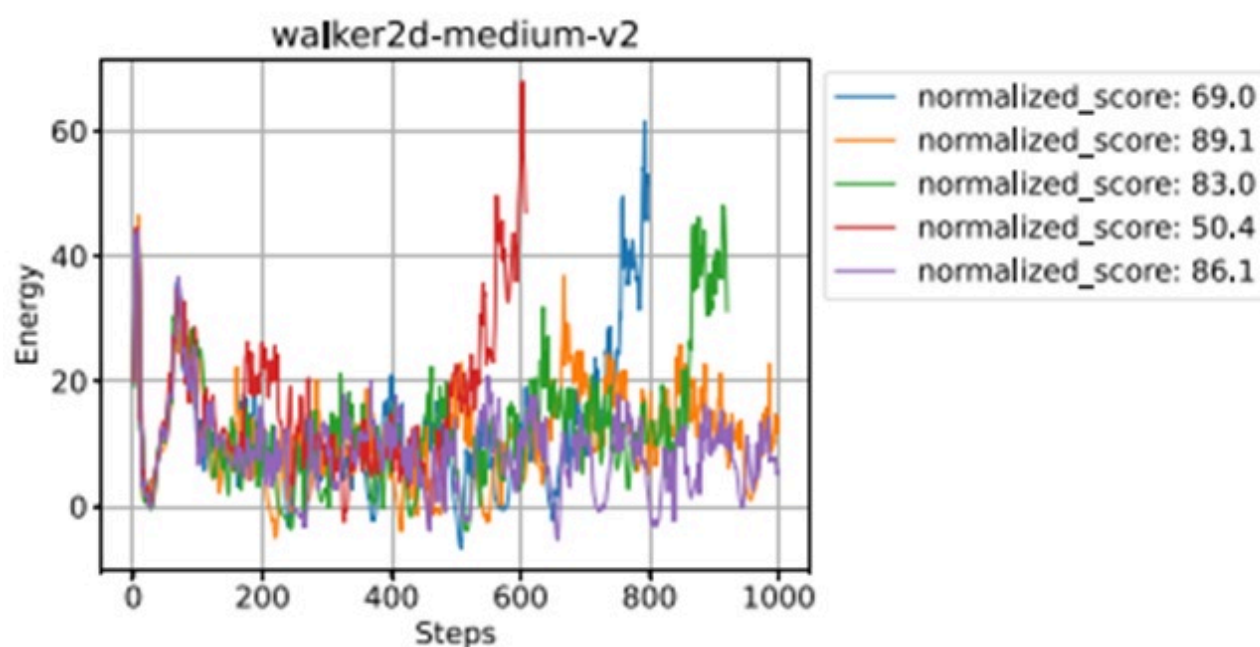


Figure 4. Energy of trajectories and the corresponding final scores in the walker2d test environment. Energy values are calculated by the trained energy-based model.

Tracking trajectory energy levels offers a practical metric for calibrating hyperparameters in offline RL frameworks. If the deployed policy generates high-energy trajectories, it indicates that the policy frequently encounters OOD samples, signifying that a more rigorous or conservative penalty is required to suppress OOD actions. Additionally, during real-time deployment in test settings, spikes in energy values signify that the agent has drifted into unfamiliar states where it may execute erratic actions. Under such conditions, the system can pivot to an alternative safety policy or request human intervention to mitigate unpredictable behavior.

Conclusion

In this work, we introduce IPC, an energy-based offline RL framework utilizing implicit policy constraints. The behavior policy is modeled implicitly via an energy function, providing a robust mechanism for capturing complex, multi-modal distributions. Furthermore, the EBM can differentiate between in-distribution data and OOD instances, allowing for a more flexible application of behavioral policy constraints. Compared to explicit constraint strategies like TD3BC, our approach yields substantial performance improvements across the D4RL Gym benchmarks. Concurrently, the optimized energy function effectively identifies OOD deviations and flags impending failures during test-time execution, which is highly advantageous for safety-critical applications. Although the EBM serves as an effective generative framework, this study limits its application to policy regularization. Future research will explore generating actions directly from the learned EBM to establish a more expressive policy representation, which naturally incorporates the IPC mechanism.

Acknowledgements

None

Conflict of interest

None

Financial support

This work was supported by the National Natural Science Foundation of China under Grant U19A2083.

Ethics statement

None

Received: 26 Feb 2023

Revised: 17 Mar 2023

Accepted: 12 Apr 2023

Published online: 15 September 2023

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

- Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. *Nature*. 2015;518(7540):529-33.
<https://doi.org/10.1038/nature14236>.
- Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*. 2016;529(7587):484-9.
<https://doi.org/10.1038/nature16961>.
- Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, et al. Mastering the game of Go without human knowledge. *Nature*. 2017;550(7676):354-9.
<https://doi.org/10.1038/nature24270>.
- Hwangbo J, Lee J, Dosovitskiy A, Bellicoso D, Tsounis V, Koltun V, et al. Learning agile and dynamic motor skills for legged robots. *Sci Robot*. 2019;4(26):eaau5872.
<https://doi.org/10.1126/scirobotics.aau5872>.
- OpenAI, Andrychowicz M, Baker B, Chociej M, Jozefowicz R, McGrew B, et al. Learning dexterous in-hand manipulation. *Int J Robot Res*. 2020;39(1):3-20.
<https://doi.org/10.1177/0278364919887447>.
- Fan T, Cheng X, Pan J, Manocha D, Yang R. CrowdMove: Autonomous mapless navigation in crowded scenarios. *arXiv [Preprint]*. 2018. arXiv:1807.07870.
<https://doi.org/10.48550/arXiv.1807.07870>.
- Bain M, Sammut C. A framework for behavioural cloning. *Mach Intell*. 1995;15:103-29.
- Pomerleau DA. ALVINN: An autonomous land vehicle in a neural network. *Adv Neural Inf Process Syst*. 1988;1:305-13.

Bojarski M, Del Testa D, Dworakowski D, Firner B, Flepp B, Goyal P, et al. End to end learning for self-driving cars. arXiv [Preprint]. 2016. arXiv:1604.07316.
<https://doi.org/10.48550/arXiv.1604.07316>.

Abbeel P, Coates A, Ng AY. Autonomous helicopter aerobatics through apprenticeship learning. *Int J Robot Res*. 2010;29(13):1608-39.
<https://doi.org/10.1177/0278364910371999>.

Osa T, Sugita N, Mitsuishi M. Online trajectory planning and force control for automation of surgical tasks. *IEEE Trans Autom Sci Eng*. 2018;15(2):675-91.
<https://doi.org/10.1109/TASE.2017.2676018>.

Levine S, Kumar A, Tucker G, Fu J. Offline reinforcement learning: tutorial, review, and perspectives on open problems. arXiv [Preprint]. 2020. arXiv:2005.01643.
<https://doi.org/10.48550/arXiv.2005.01643>.

Fujimoto S, Meger D, Precup D. Off-policy deep reinforcement learning without exploration. In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR; 2019. p. 2052-62.

Kumar A, Fu J, Soh M, Tucker G, Levine S. Stabilizing off-policy Q-learning via bootstrapping error reduction. *Adv Neural Inf Process Syst*. 2019;32:11761-71.

Fujimoto S, Gu SS. A minimalist approach to offline reinforcement learning. *Adv Neural Inf Process Syst*. 2021;34:20132-45.

Wu Y, Tucker G, Nachum O. Behavior regularized offline reinforcement learning. arXiv [Preprint]. 2019. arXiv:1911.11361.
<https://doi.org/10.48550/arXiv.1911.11361>.

LeCun Y, Chopra S, Hadsell R, Ranzato MA, Huang FJ. A tutorial on energy-based learning. In: Bakir G, Hofmann T, Schölkopf B, Smola AJ, Taskar B, editors. *Predicting structured data*. Cambridge (MA): MIT Press; 2006. p. 191-246.

Liu W, Wang X, Owens JD, Li Y. Energy-based out-of-distribution detection. *Adv Neural Inf Process Syst*. 2020;33:21464-75.

Elflein S. Out-of-distribution detection with energy-based models. arXiv [Preprint]. 2023. arXiv:2302.12002.
<https://doi.org/10.48550/arXiv.2302.12002>.

Prudencio RF, Maximo MROA, Colombini EL. A survey on offline reinforcement learning: taxonomy, review, and open problems. *IEEE Trans Neural Netw Learn Syst*. 2024;35(8):10237-57.
<https://doi.org/10.1109/TNNLS.2023.3250269>.

Kingma DP, Welling M. Auto-encoding variational Bayes. arXiv [Preprint]. 2013. arXiv:1312.6114.
<https://doi.org/10.48550/arXiv.1312.6114>.

Sohn K, Lee H, Yan X. Learning structured output representation using deep conditional generative models. *Adv Neural Inf Process Syst*. 2015;28:3483-91.

Peng Z, Han C, Liu Y, Zhou Z. Weighted policy constraints for offline reinforcement learning. *Proc AAAI Conf Artif Intell*. 2023;37(8):9435-43.
<https://doi.org/10.1609/aaai.v37i8.26130>.

Kostrikov I, Fergus R, Tompson J, Nachum O. Offline reinforcement learning with Fisher divergence critic regularization. In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR; 2021. p. 5774-83.

Kumar A, Zhou A, Tucker G, Levine S. Conservative Q-learning for offline reinforcement learning. *Adv Neural Inf Process Syst*. 2020;33:1179-91.

Lyu J, Ma X, Li X, Lu Z. Mildly conservative Q-learning for offline reinforcement learning. *Adv Neural Inf Process Syst*. 2022;35:1711-24.

An G, Moon S, Kim JH, Song HO. Uncertainty-based offline reinforcement learning with diversified Q-ensemble. *Adv Neural Inf Process Syst*. 2021;34:7436-47.

Bai C, Wang L, Yang Z, Deng ZH, Garg A, Liu P, et al. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In: *International Conference on Learning Representations*. 2022.

Yang R, Bai C, Ma X, Wang Z, Zhang C, Han L. RORL: robust offline reinforcement learning via conservative smoothing. *Adv Neural Inf Process Syst*. 2022;35:23851-66.
<https://doi.org/10.52202/068431-1732>.

Peng Z, Liu Y, Chen H, Zhou Z. Conservative network for offline reinforcement learning. *Knowl Based Syst*. 2023;282:111101.
<https://doi.org/10.1016/j.knosys.2023.111101>.

Kostrikov I, Nair A, Levine S. Offline reinforcement learning with implicit Q-learning. In: *International Conference on Learning Representations*. 2022.

Lyu J, Gong A, Wan L, Lu Z, Li X. State advantage weighting for offline RL. *arXiv [Preprint]*. 2022. arXiv:2210.04251.
<https://doi.org/10.48550/arXiv.2210.04251>.

Edwards AD, Sahni H, Liu R, Hung J, Jain A, Wang R, et al. Estimating $Q(s,s')$ with deep deterministic dynamics gradients. In: *Proceedings of the 37th International Conference on Machine Learning*. PMLR; 2020. p. 2825-35.

Ma X, Yang Y, Hu H, Liu Q, Yang J, Zhang C, et al. Offline reinforcement learning with value-based episodic memory. *arXiv [Preprint]*. 2021. arXiv:2110.09796.
<https://doi.org/10.48550/arXiv.2110.09796>.

Chen L, Lu K, Rajeswaran A, Lee K, Grover A, Laskin M, et al. Decision transformer: reinforcement learning via sequence modeling. *Adv Neural Inf Process Syst*. 2021;34:15084-97.

Janner M, Li Q, Levine S. Offline reinforcement learning as one big sequence modeling problem. *Adv Neural Inf Process Syst*. 2021;34:1273-86.

Janner M, Du Y, Tenenbaum JB, Levine S. Planning with diffusion for flexible behavior synthesis. In: *Proceedings of the 39th International Conference on Machine Learning*. PMLR; 2022. p. 9902-15.

Wang Z, Hunt JJ, Zhou M. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv [Preprint]*. 2022. arXiv:2208.06193.
<https://doi.org/10.48550/arXiv.2208.06193>.

Wang J, Li W, Jiang H, Zhu G, Li S, Zhang C. Offline reinforcement learning with reverse model-based imagination. *Adv Neural Inf Process Syst*. 2021;34:29420-32.

Lyu J, Li X, Lu Z. Double check your state before trusting it: confidence-aware bidirectional offline model-based imagination. *Adv Neural Inf Process Syst*. 2022;35:38218-31.
<https://doi.org/10.52202/068431-2769>.

Zhang J, Lyu J, Ma X, Yan J, Yang J, Wan L, et al. Uncertainty-driven trajectory truncation for data augmentation in offline reinforcement learning. *arXiv [Preprint]*. 2023. arXiv:2304.04660.
<https://doi.org/10.48550/arXiv.2304.04660>.

Song Y, Kingma DP. How to train your energy-based models. arXiv [Preprint]. 2021. arXiv:2101.03288.
<https://doi.org/10.48550/arXiv.2101.03288>.

Haarnoja T, Tang H, Abbeel P, Levine S. Reinforcement learning with deep energy-based policies. In: Proceedings of the 34th International Conference on Machine Learning. PMLR; 2017. p. 1352-61.

Grathwohl W, Wang KC, Jacobsen JH, Duvenaud D, Norouzi M, Swersky K. Your classifier is secretly an energy based model and you should treat it like one. In: International Conference on Learning Representations. 2020.

Florence P, Lynch C, Zeng A, Ramirez OA, Wahid A, Downs L, et al. Implicit behavioral cloning. In: Proceedings of the 5th Conference on Robot Learning. PMLR; 2022. p. 158-68.

van den Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding. arXiv [Preprint]. 2018. arXiv:1807.03748.
<https://doi.org/10.48550/arXiv.1807.03748>.

Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proceedings of the 35th International Conference on Machine Learning. PMLR; 2018. p. 1861-70.

Emmons S, Eysenbach B, Kostrikov I, Levine S. RvS: what is essential for offline RL via supervised learning? In: International Conference on Learning Representations. 2022.

Brandfonbrener D, Whitney WF, Ranganath R, Bruna J. Offline RL without off-policy evaluation. Adv Neural Inf Process Syst. 2021;34:4933-46.

Fu J, Kumar A, Nachum O, Tucker G, Levine S. D4RL: datasets for deep data-driven reinforcement learning. arXiv [Preprint]. 2020. arXiv:2004.07219.
<https://doi.org/10.48550/arXiv.2004.07219>.