

---

# Offline Reinforcement Learning with Reverse Model-based Imagination

---

Jianhao Wang<sup>\*1</sup>, Wenzhe Li<sup>\*1</sup>, Haozhe Jiang<sup>1</sup>, Guangxiang Zhu<sup>2†</sup>, Siyuan Li<sup>1</sup>,  
Chongjie Zhang<sup>1</sup>

<sup>1</sup>Institute for Interdisciplinary Information Sciences, Tsinghua University, China

<sup>2</sup>Baidu Inc., China

{wjh19, lwz21, jianghz20}@mails.tsinghua.edu.cn

guangxiangzhu@outlook.com

sy-li17@mails.tsinghua.edu.cn

chongjie@tsinghua.edu.cn

## Abstract

In offline reinforcement learning (offline RL), one of the main challenges is to deal with the distributional shift between the learning policy and the given dataset. To address this problem, recent offline RL methods attempt to introduce conservatism bias to encourage learning in high-confidence areas. Model-free approaches directly encode such bias into policy or value function learning using conservative regularizations or special network structures, but their constrained policy search limits the generalization beyond the offline dataset. Model-based approaches learn forward dynamics models with conservatism quantifications and then generate imaginary trajectories to extend the offline datasets. However, due to limited samples in offline datasets, conservatism quantifications often suffer from overgeneralization in out-of-support regions. The unreliable conservative measures will mislead forward model-based imaginations to undesired areas, leading to overaggressive behaviors. To encourage more conservatism, we propose a novel model-based offline RL framework, called *Reverse Offline Model-based Imagination* (ROMI). We learn a reverse dynamics model in conjunction with a novel reverse policy, which can generate rollouts leading to the target goal states within the offline dataset. These reverse imaginations provide informed data augmentation for model-free policy learning and enable conservative generalization beyond the offline dataset. ROMI can effectively combine with off-the-shelf model-free algorithms to enable model-based generalization with proper conservatism. Empirical results show that our method can generate more conservative behaviors and achieve state-of-the-art performance on offline RL benchmark tasks.

## 1 Introduction

Deep reinforcement learning (RL) has achieved tremendous successes in a range of domains [1–3] by utilizing a large number of interactions with the environment. However, in many real-world applications, collecting sufficient exploratory interactions is usually impractical, because online data collection can be costly or even dangerous, such as in healthcare [4] and autonomous driving [5]. To address this challenge, offline RL [6, 7] develops a new learning paradigm that trains RL agents only with pre-collected offline datasets and thus can abstract away from the cost of online exploration [8–17]. For such offline settings, recent studies demonstrate that directly applying

---

<sup>\*</sup>Equal contribution.

<sup>†</sup>Work done while Guangxiang was a Ph.D. student at Tsinghua University.

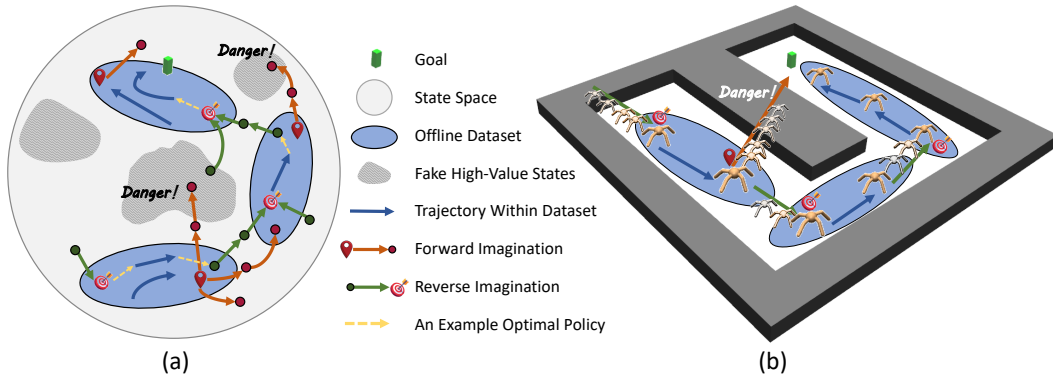


Figure 1: (a) Basic idea of ROMI. (b) A concrete RL task to demonstrate the superiority of ROMI.

the online RL algorithms can lead to poor performance [8, 9, 16]. This phenomenon is primarily attributed to *distributional shift* [7] between the learning policy and the behavior policy induced by the given dataset. With function approximation, the learning policy can overgeneralize the offline dataset and result in unexpected or dangerous behaviors. Thus developing techniques to handle distributional shift is becoming an active topic in the community of offline RL.

Recently, a variety of advanced offline RL algorithms have been proposed, which introduce conservatism bias and constrain the policy search in high-confidence regions induced by the offline dataset. Model-free offline RL methods [8–11] explicitly encode such bias into policy or value functions by using conservative regularizations or specially designed network structures. These methods often effectively address distributional shift issues, but their constrained policy search can limit the generalization beyond the offline dataset. In contrast, model-based offline RL [12–15] adopts more aggressive approaches. They first learn a forward dynamics model from the offline dataset with conservatism quantifications and then generate imaginary trajectories on high confidence regions to extend the offline dataset. Specifically, these methods may use a model-uncertainty quantification [12, 13], representation learning of a robust model [14], or a conservative estimation of value functions [15] to ensure the confidence of model-based rollouts. However, because samples in offline datasets are limited, conservatism quantifications often suffer from overgeneralization, especially in out-of-support regions. Thus, these unreliable measures can overestimate some unknown states and mislead forward model-based imaginations to undesired areas, leading to radicalism. In this paper, we will investigate a new direction in the context of model-based offline RL and propose reverse model imagination that enables effective conservative generalization.

We use Figure 1a to illustrate our basic ideas. When an offline dataset contains expert or nearly optimal trajectories, model-free offline RL methods [8–11] show promising performance. In other situations, such as in Figure 1a, when the optimal policy requires a composition of multiple trajectories ( $\rightarrow$ ) in the offline dataset, model-free offline RL usually fails because it may get stuck in isolated regions of the offline dataset ( $\circ$ ). In contrast, model-based approaches [12–15] have advantages of connecting trajectories in the offline dataset by generating bridging rollouts. When using forward dynamics models, model-based methods can generate aggressive rollouts from the dataset to outside areas ( $\rightarrow$ ). Such *forward imaginations* potentially discover a better policy outside the offline dataset, but may also lead to undesirable regions consisting of fake high-value states ( $\bullet$ ) due to overgeneralization errors. Things will be different if we reverse the model imagination. *Reverse imaginations* ( $\rightarrow$ ) generate possible traces leading to target goal states ( $\odot$ ) inside the offline dataset, which provides a conservative way of augmenting the offline dataset. We assume that the reverse dynamics model has similar accuracy to the forward model. In this way, reverse models not only maintain the generalization ability to interpolate between given trajectories and potential better policies but also avoid radical model imagination by enabling bidirectional search in conjunction with the existing forward real trajectories ( $\rightarrow$ ) in the offline dataset. The scenario illustrated in Figure 1a is common in RL tasks [18], where the agent may encounter new obstacles or unexpected dangers at any time (e.g., bumping into walls like in Figure 1b or falling). As many unexpected data are not recorded by an offline dataset, there are likely large generalization errors outside the support of the dataset. Thus, reverse models that can generate conservative imaginations are more appealing for real-world offline RL tasks.

Based on the above observation, we present a novel model-based offline RL framework, called *Reverse Offline Model-based Imagination* (ROMI). ROMI trains a reverse dynamics model and generates backward imaginary trajectories by a backtracking rollout policy. This rollout policy is learned by a conditional generative model to produce diverse reverse actions for each state and lead to the unknown data space with high probabilities. ROMI ensures start states of the reverse rollout trajectories (i.e., target goals of forward trajectories) are within the offline dataset, and thus naturally imposes conservative constraints on imaginations. With this conservative data augmentation, ROMI has the advantage of effectively combining with off-the-shelf model-free algorithms (e.g., BCQ [8] and CQL [11]) to further strengthen its generalization with proper conservatism. Taken together, the whole ROMI framework enables extensive interpolation of the dataset and potentially better performance (contributed by diverse policy) in a safe manner (contributed by reverse imagination). To our best knowledge, ROMI is the first offline RL approach that utilizes reverse model-based imaginations to induce conservatism bias with data augmentation. It provides a novel bidirectional learning paradigm for offline RL, which connects reverse imaginary trajectories with pre-collected forward trajectories in the offline dataset. Such a bidirectional learning paradigm shares similar motivations with humans’ bidirectional reasoning. Studies from Psychology [19] show that, during the decision-making process, humans not only consider the consequences of possible future actions from a forward view but also imagine possible traces leading to the ideal goal through backward inference.

We conduct extensive evaluations on the D4RL offline benchmark suite [18]. Empirical results show that ROMI significantly outperforms state-of-the-art model-free and model-based baselines. Our method achieves the best or comparable performance on 16 out of 24 tasks among all algorithms. Ablation studies verify that the reverse model imagination can effectively generate more conservative behaviors than forward model imagination. Videos of the experiments are available online<sup>3</sup>.

## 2 Preliminaries

We consider a Markov decision process (MDP) defined by a tuple  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, r, \mu_0, \gamma)$ , where  $\mathcal{S}$  and  $\mathcal{A}$  denote the state space and the action space, respectively.  $T(s'|s, a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$  denotes the transition distribution function,  $r(s, a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  denotes the reward function,  $\mu_0 : \mathcal{S} \rightarrow [0, 1]$  is the initial state distribution, and  $\gamma \in (0, 1)$  is the discount factor. Moreover, we denote the reverse transition distribution by  $T_r(s|s', a) = T^{-1} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ . The goal of an RL agent is to optimize a policy  $\pi(a|s) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  that maximizes the expected cumulative reward, i.e.,  $\mathcal{J}(\pi) = \mathbb{E}_{s_0 \sim \mu_0, s_{t+1} \sim T(\cdot|s_t, \pi(s_t))} [\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t))]$ .

In the offline RL setting, the agent only has access to a static dataset  $\mathcal{D}_{\text{env}} = \{(s, a, r, s')\}$  and is not allowed to interact with the environment for additional online explorations. The data can be collected through multi-source logging policies and we denote the empirical distribution of behavior policy in a given dataset  $\mathcal{D}_{\text{env}}$  collected by  $\pi_D$ . Logging policies are not accessible in our setting.

Model-based RL methods aim at performing planning or policy searches based on a learned model of the environment. They usually learn a dynamics model  $\hat{T}$  and a reward model  $\hat{r}(s, a)$  from a collection of environmental data in a self-supervised manner. Most of the existing approaches use the forward model  $\hat{T}_f(s'|s, a)$  for dynamics learning, but we will show in Section 3 that the reverse dynamics model  $\hat{T}_r(s|s', a)$  can induce more conservatism and is critical for the offline RL tasks.

## 3 Reverse Offline Model-based Imagination

In the offline RL setting, the agent can only access a given dataset without additional online exploration. Model-based offline RL algorithms usually face three challenges in this setting. (i) The offline dataset has limited samples and generalization over the given data is required. (ii) There are a lot of uncertainties that are difficult to estimate out of the support of the dataset. (iii) The model-based rollouts cannot receive feedback from the real environment. Thus it is important to augment diverse data and keep conservative generalization at the same time. In this section, we will introduce the *Reverse Offline Model-based Imagination* (ROMI) framework to combine model-based imagina-

<sup>3</sup><https://sites.google.com/view/romi-offlinerl/>.

tion with model-free offline policy learning. This framework encourages diverse augmentation of model-based rollouts and enables conservative generalization of the generated imaginations.

We use a concrete example to show the superiority of our framework in Figure 1b. The agent navigates in a u-maze to reach the goal (■). We gather an offline dataset consisting of three data blocks (●). This dataset only contains the trajectories that do not hit the wall, thus the agent will be unaware of the walls during the offline learning process. In this situation, the conventional forward model may generate a greedy and radical imagination (→) that takes a shortcut to go straight to the goal (whose value function may be overestimated) and hit the wall. On the contrary, ROMI adopts backward imaginations and generates traces that can lead to the target goal states in the support of the dataset (→). Such imaginations can be further connected with the given forward trajectories (→) in the dataset to form an optimal policy. In this example, we assume that the model error is mainly due to the prediction of the OOD states (outside of the support of the data), then we can expect reverse models to help combat overestimation since they prevent rollout trajectories that end in the OOD states. A more detailed illustrative example is deferred to Appendix A.

Specifically, our framework consists of two main components: (i) a reverse model learned from the offline dataset, (ii) a diverse rollout policy to generate reverse actions that are close to the dataset. ROMI pre-generates the model-based trajectories under the rollout policy based on the reverse model. Then we use the mixture of the imaginary data and the original dataset to train a model-free offline agent. The whole algorithm is shown in Algorithm 1.

**Training the reverse model.** To backtrace the dynamics and reward function of the environment, we introduce a reverse model to estimate the reverse dynamics model  $\widehat{T}_r(s|s', a)$  and reward model  $\widehat{r}(s, a)$  from the offline dataset simultaneously. For simplicity, we unify the dynamics and reward function into our reverse model  $p(s, r|s', a)$ , i.e.,

$$p(s, r|s', a) = p(s|s', a)p(r|s', a, s) = T_r(s|s', a)p(r|s, a), \quad (1)$$

where we assume that the reward function only depends on the current state and action. This unified reverse model represents the probability of the current state and immediate reward conditioned on the next state and current action. We parameterize it by  $\phi$  and optimize it by minimizing the loss function  $\mathcal{L}_M(\phi)$  (which is equivalent to maximizing the log-likelihood):

$$\mathcal{L}_M(\phi) = \mathbb{E}_{(s, a, r, s') \sim \mathcal{D}_{\text{env}}} [-\log \widehat{p}_\phi(s, r|s', a)], \quad (2)$$

where  $\mathcal{D}_{\text{env}}$  is the offline dataset.

**Training the reverse rollout policy.** To encourage diversity for the reverse model-based imagination near the dataset, we train a generative model  $\widehat{G}_\theta(a|s')$ , which samples diverse reverse actions from the offline dataset using stochastic inference. Specifically, we use a conditional variational auto-encoder (CVAE) [20, 8] to represent the diverse rollout policy  $\widehat{G}_\theta(a|s')$ , which is parameterized by  $\theta$  and depends on the next state. The rollout policy  $\widehat{G}_\theta(a|s')$  contains two modules: (i) an action encoder  $\widehat{E}_\omega(s', a)$  that outputs a latent vector  $z$  under the gaussian distribution  $z \sim \widehat{E}_\omega(s', a)$ , and (ii) an action decoder  $\widehat{D}_\xi(s', z)$  whose input is the latent vector  $z$  and reconstructs the given action  $\widehat{a} = \widehat{D}_\xi(s', z)$ . The action encoder and decoder are parameterized by  $\omega$  and  $\xi$ , respectively. For simplicity, denote the parameters of the rollout policy  $\widehat{G}_\theta(a|s')$  by  $\theta = \{\omega, \xi\}$ . We defer the detailed discussion of CVAE to Appendix B.

We train the rollout policy  $\widehat{G}_\theta(a|s')$  by maximizing the variational lower bound  $\mathcal{L}_p(\theta)$ ,

$$\mathcal{L}_p(\theta) = \mathbb{E}_{(s, a, r, s') \sim \mathcal{D}_{\text{env}}, z \sim \widehat{E}_\omega(s', a)} \left[ \left( a - \widehat{D}_\xi(s', z) \right)^2 + D_{\text{KL}} \left( \widehat{E}_\omega(s', a) \parallel \mathcal{N}(0, \mathbf{I}) \right) \right], \quad (3)$$

where  $\mathbf{I}$  is an identity matrix. To optimize such loss function, we adopt similar optimization techniques as those used in the context of image generalization and video prediction [20, 8].

After the rollout policy  $\widehat{G}_\theta(a|s')$  is well trained, we can sample reverse actions  $\widehat{a}$  based on the policy. We first draw a latent vector from the multivariate normal distribution,  $\widehat{z} \sim \mathcal{N}(0, \mathbf{I})$ , and then utilize the action decoder to sample actions conditioned on the next state,  $\widehat{a} = \widehat{D}_\xi(s', \widehat{z})$ . To explore more possibilities, the rollout policy  $\widehat{G}_\theta(a|s')$  uses stochastic layers to generate a variety of reverse actions

for multiple times. In addition, if we need an extremely diverse policy in easy tasks, we can replace the generative model based policy with a uniformly random rollout policy.

**Combination with model-free algorithms.** Based on the learned reverse dynamics model and the reverse rollout policy, ROMI can generate reverse imaginations. We collect these rollouts to form a model-based buffer  $\mathcal{D}_{\text{model}}$  and further compose the total dataset with the offline dataset,  $\mathcal{D}_{\text{total}} = \mathcal{D}_{\text{env}} \cup \mathcal{D}_{\text{model}}$ . Since our rollout policy is agnostic to policy learning, such buffer can be obtained before the policy learning stage, i.e., ROMI can be combined with any model-free offline RL algorithm (e.g., BCQ [8] or CQL [11]). Specifically, during the model-based imagination, we sample the target state  $s_{t+1}$  from the dataset  $\mathcal{D}_{\text{env}}$ , and generate reverse imaginary trajectory  $\hat{\tau}$  with the rollout horizon  $h$  by the reverse model  $\hat{p}_\phi$  and rollout policy  $\hat{G}_\theta$ :

$$\hat{\tau} = \left\{ (s_{t-i}, a_{t-i}, r_{t-i}, s_{t+1-i}) \mid a_{t-i} \sim \hat{G}_\theta(\cdot | s_{t+1-i}) \text{ and } s_{t-i}, r_{t-i} \sim \hat{p}_\phi(\cdot | s_{t+1-i}, a_{t-i}) \right\}_{i=0}^{h-1}.$$

We gather trajectories  $\hat{\tau}$  to form the buffer  $\mathcal{D}_{\text{model}}$  and further combine it with  $\mathcal{D}_{\text{env}}$  to obtain  $\mathcal{D}_{\text{total}}$ . Then we will run the model-free offline policy learning algorithm on the total buffer to derive the final policy  $\pi_{\text{out}}$ . Compared to existing model-based approaches [12–15], ROMI provides informed data augmentation to extend the offline dataset. It is agnostic to policy optimization and thus can be regarded as an effective and flexible plug-in component to induce conservative model-based imaginations for offline RL.

---

**Algorithm 1** ROMI: Reverse Offline Model-based Imagination

---

- 1: **Require:** Offline dataset  $\mathcal{D}_{\text{env}}$ , rollout horizon  $h$ , the number of iterations  $C_\phi, C_\theta, T$ , learning rates  $\alpha_\phi, \alpha_\theta$ , model-free offline RL algorithm (i.e., BCQ or CQL)
  - 2: Randomly initialize reverse model parameters  $\phi$
  - 3: **for**  $i = 0 \dots C_\phi - 1$  **do** ▷ Learning a reverse dynamics model  $\hat{p}_\phi$
  - 4:     Compute  $\mathcal{L}_M$  using the dataset  $\mathcal{D}_{\text{env}}$
  - 5:     Update  $\phi \leftarrow \phi - \alpha_\phi \nabla_\phi \mathcal{L}_M$
  - 6: Randomly initialize rollout policy parameters  $\theta$
  - 7: **for**  $i = 0 \dots C_\theta - 1$  **do** ▷ Learning a diverse rollout policy  $\hat{G}_\theta$
  - 8:     Compute  $\mathcal{L}_p$  using the dataset  $\mathcal{D}_{\text{env}}$
  - 9:     Update  $\theta \leftarrow \theta - \alpha_\theta \nabla_\theta \mathcal{L}_p$
  - 10: Initialize the replay buffer  $\mathcal{D}_{\text{model}} \leftarrow \emptyset$
  - 11: **for**  $i = 0 \dots T - 1$  **do** ▷ Collecting the replay buffer  $\mathcal{D}_{\text{model}}$
  - 12:     Sample target state  $s_{t+1}$  from the dataset  $\mathcal{D}_{\text{env}}$
  - 13:     Generate reverse model rollout  $\hat{\tau} = \{(s_{t-i}, a_{t-i}, r_{t-i}, s_{t+1-i})\}_{i=0}^{h-1}$  from  $s_{t+1}$  by drawing samples from the dynamics model  $\hat{p}_\phi$  and rollout policy  $\hat{G}_\theta$
  - 14:     Add model rollouts to replay buffer,  $\mathcal{D}_{\text{model}} \leftarrow \mathcal{D}_{\text{model}} \cup \{(s_{t-i}, a_{t-i}, r_{t-i}, s_{t+1-i})\}_{i=0}^{h-1}$
  - 15: Compose the final dataset  $\mathcal{D}_{\text{total}} \leftarrow \mathcal{D}_{\text{env}} \cup \mathcal{D}_{\text{model}}$
  - 16: Combine model-free offline RL algorithms to derive the final policy  $\pi_{\text{out}}$  using the dataset  $\mathcal{D}_{\text{total}}$
  - 17: **Return:**  $\pi_{\text{out}}$
- 

## 4 Experiments

In this section, we conduct a bunch of experiments in the offline RL benchmark [18] to answer the following questions: (i) Does ROMI outperform the state-of-the-art offline RL baselines (see Table 1 and 2)? (ii) Does ROMI achieve excellent performance because of the reverse model-based imagination (see Section 4.3)? (iii) Is CVAE-based rollout policy critical for ROMI (see Table 4)? (iv) Compared with the forward imagination, does ROMI trigger more conservative and effective behaviors (see Figure 3)?

### 4.1 Evaluation Environments

We evaluate ROMI on a wide range of domains in the D4RL benchmark [18], including the Maze2D domain, the Gym-MuJoCo tasks, and the AntMaze domain. Figure 2 shows the snapshots of nine environments used in our experiments. We defer the quantification of ROMI’s model accuracy in

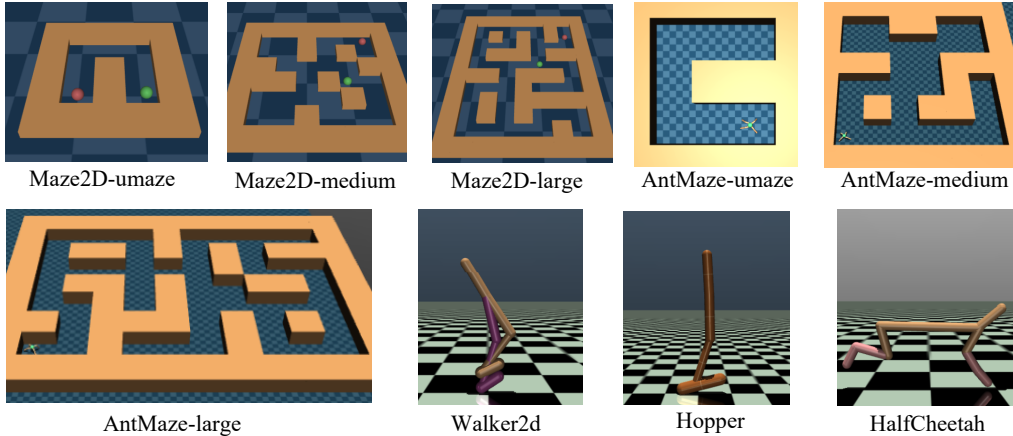


Figure 2: Experimental environments.

these domains to Appendix C and empirical evaluations show that reverse models have comparable accuracy, if not worse, than forward models.

**Maze2D.** The *maze2d* domain requires a 2D agent to learn to navigate in the maze to reach a fixed target goal and stay there. As shown in Figure 2, there are three maze layouts (i.e., *umaze*, *medium*, and *large*) and two dataset types (i.e., *sparse* and *dense* reward signal) in this domain. The dataset of each layout is generated by a planner moving between randomly sampled waypoints. From the detailed discussion of the D4RL benchmark [18], we found that the agents in the *maze2d* dataset are always moving on the clearing and will not stay in place. We will visualize the dataset of *mazed2d-umaze* in Section 4.5.

**Gym-MuJoCo.** The Gym-MuJoCo tasks consist of three different environments (i.e., *walker2d*, *hopper*, and *halfcheetah*), and four types of datasets (i.e., *random*, *medium*, *medium-replay*, and *medium-expert*). *Random* dataset contains experiences selected by a random policy. *Medium* dataset contains experiences from an early-stopped SAC policy. *Medium-replay* dataset records the samples in the replay buffer during the training of the "medium" SAC policy. *Medium-expert* dataset is mixed with suboptimal samples and samples generated from an expert policy.

**AntMaze.** The *antmaze* domain combines challenges of the previous two domains. The policy needs to learn to control the robot and navigate to the goal simultaneously. This domain also contains three different layouts (i.e., *umaze*, *medium*, and *large*) shown in Figure 2. D4RL benchmark [18] introduces three flavors of datasets (i.e., *fixed*, *diverse*, and *play*) in this setting, which commands the ant from different types of start locations to various types of goals.

## 4.2 Overall Results

In this subsection, the experimental results are presented in Table 1 and 2, which are evaluated in the D4RL benchmark tasks [18] illustrated in Figure 2. We compare ROMI with 11 state-of-the-art baselines: *MF* denotes the best performance from offline model-free algorithms, including BCQ [8], BEAR [10], BRAC-v, BRAC-p [9], BAIL [17], and CQL [11]; *MB* denotes the best performance from offline model-based algorithms, including MOPO [13], MOREL [12], Repb-SDE [14], and COMBO [15]; *BC* denotes the popular behavior cloning from the dataset in the imitation learning. The implementation details of ROMI and these algorithms are deferred to Appendix D.2. Towards fair evaluation, all experimental results are illustrated with the averaged performance with  $\pm$  standard deviation over three random seeds.

We evaluate ROMI in nine D4RL benchmark domains with 24 tasks. Our experiments show that ROMI significantly outperforms baselines and achieves the best or comparable performance on 16 out of 24 continuous control tasks. Table 1 and 2 only contain the best performance of *MF* and *MB* categories<sup>4</sup>. We defer the pairwise comparison of ROMI and each baseline to Appendix E, which

<sup>4</sup>In Table 2, each score of COMBO or MOREL is the better one between the score our reproduction and their reported score.

Table 1: Performance of ROMI and best performance of prior methods on the *maze* and *antmaze* domains, on the normalized return metric proposed by D4RL benchmark [18]. Scores roughly range from 0 to 100, where 0 corresponds to a random policy performance and 100 corresponds to an expert policy performance. *med* is short for *medium*.

Environment	BC	ROMI-BCQ	MF	MB
sparse-maze2d-umaze	-3.2	<b>139.5</b> $\pm$ 3.6	65.7 $\pm$ 6.9 <sup>BEAR</sup>	76.4 $\pm$ 19.2 <sup>COMBO</sup>
sparse-maze2d-med	-0.5	<b>82.4</b> $\pm$ 15.2	70.6 $\pm$ 34.3 <sup>BRAC-v</sup>	68.5 $\pm$ 83.6 <sup>COMBO</sup>
sparse-maze2d-large	-1.7	<b>83.1</b> $\pm$ 22.1	<b>81.0</b> $\pm$ 65.3 <sup>BEAR</sup>	14.1 $\pm$ 10.7 <sup>COMBO</sup>
dense-maze2d-umaze	-6.9	<b>98.3</b> $\pm$ 2.5	51.5 $\pm$ 8.2 <sup>BRAC-p</sup>	94.3 $\pm$ 13.6 <sup>Repb-SDE</sup>
dense-maze2d-med	2.7	<b>102.6</b> $\pm$ 32.4	41.7 $\pm$ 2.0 <sup>BAIL</sup>	84.2 $\pm$ 9.5 <sup>COMBO</sup>
dense-maze2d-large	-0.3	<b>124</b> $\pm$ 1.3	<b>133.0</b> $\pm$ 25.5 <sup>BEAR</sup>	36.8 $\pm$ 12.4 <sup>MOPO</sup>
fixed-antmaze-umaze	82.0	68.7 $\pm$ 2.7	75.3 $\pm$ 13.7 <sup>BCQ</sup>	<b>80.3</b> $\pm$ 18.5 <sup>COMBO</sup>
play-antmaze-med	0.0	<b>35.3</b> $\pm$ 1.3	1.7 $\pm$ 1.0 <sup>BAIL</sup>	0.0
play-antmaze-large	0.0	<b>20.2</b> $\pm$ 14.8	2.2 $\pm$ 1.3 <sup>BAIL</sup>	0.0
diverse-antmaze-umaze	47.0	<b>61.2</b> $\pm$ 3.3	54.0 $\pm$ 15.0 <sup>BAIL</sup>	57.3 $\pm$ 33.6 <sup>COMBO</sup>
diverse-antmaze-med	0.0	27.3 $\pm$ 3.9	<b>61.5</b> $\pm$ 10.0 <sup>CQL</sup>	0.0
diverse-antmaze-large	0.0	<b>41.2</b> $\pm$ 4.2	1.0 $\pm$ 0.9 <sup>BAIL</sup>	0.0

Table 2: Performance of ROMI and best performance of prior methods on Gym-MuJoCo tasks.

Environment	BC	ROMI-CQL	MF	MB
random-walker2d	0.0	7.5 $\pm$ 20.0	<b>11.1</b> $\pm$ 8.8 <sup>BEAR</sup>	7.0 <sup>COMBO</sup>
random-hopper	0.9	<b>30.2</b> $\pm$ 4.4	<b>31.4</b> $\pm$ 0.1 <sup>CQL</sup>	<b>31.7</b> $\pm$ 0.1 <sup>Repb-SDE</sup>
random-halfcheetah	-0.1	24.5 $\pm$ 0.7	19.6 $\pm$ 1.2 <sup>CQL</sup>	<b>38.8</b> <sup>COMBO</sup>
medium-walker2d	41.7	<b>84.3</b> $\pm$ 1.1	<b>83.8</b> $\pm$ 0.2 <sup>CQL</sup>	<b>85.3</b> $\pm$ 2.2 <sup>Repb-SDE</sup>
medium-hopper	40.0	72.3 $\pm$ 17.5	66.6 $\pm$ 4.1 <sup>CQL</sup>	<b>95.4</b> <sup>MOReL</sup>
medium-halfcheetah	39.2	49.1 $\pm$ 0.8	49.0 $\pm$ 0.4 <sup>CQL</sup>	<b>69.5</b> $\pm$ 0.0 <sup>MOPO</sup>
medium-replay-walker2d	2.2	<b>109.7</b> $\pm$ 9.8	88.4 $\pm$ 1.1 <sup>CQL</sup>	83.8 $\pm$ 7.6 <sup>Repb-SDE</sup>
medium-replay-hopper	8.1	<b>98.1</b> $\pm$ 2.6	<b>97.0</b> $\pm$ 0.8 <sup>CQL</sup>	93.6 <sup>MOReL</sup>
medium-replay-halfcheetah	25.6	47.0 $\pm$ 0.7	46.4 $\pm$ 0.3 <sup>CQL</sup>	<b>68.2</b> $\pm$ 3.2 <sup>MOPO</sup>
medium-expert-walker2d	73.4	<b>109.7</b> $\pm$ 5.3	<b>109.5</b> $\pm$ 0.1 <sup>CQL</sup>	<b>111.2</b> $\pm$ 0.2 <sup>Repb-SDE</sup>
medium-expert-hopper	36.0	<b>111.4</b> $\pm$ 5.6	<b>106.8</b> $\pm$ 2.9 <sup>CQL</sup>	<b>111.1</b> <sup>COMBO</sup>
medium-expert-halfcheetah	39.7	86.8 $\pm$ 19.7	90.8 $\pm$ 5.6 <sup>CQL</sup>	<b>95.6</b> <sup>MOReL</sup>

can demonstrate that ROMI in conjunction with off-the-shelf model-free methods (i.e., BCQ [8] and CQL [11]) can outperform all offline RL baselines. Specifically, we denote the methods with ROMI as ROMI-BCQ, and ROMI-CQL, respectively. The suffix *-BCQ* or *-CQL* indicates that ROMI adopts BCQ [8] or CQL [11] as base learning algorithms for policy optimization.

Table 1 shows that ROMI-BCQ is the best performer on 10 out of 12 tasks in the *maze2d* and *antmaze* domains. In these tasks, model-free methods can achieve reasonable performance, while current model-based algorithms with forward imagination cannot perform well, especially in *antmaze*. This may be because that messy walls (see Figure 2) are unknown from the given datasets, and the forward imagination may lead RL agent to bump into walls and lose the game. In contrast, reverse model-based imagination can avoid directing agents towards unconfident area, which makes a safe way to imagine in the complex domains. Table 2 shows that ROMI-CQL achieves the best performer on six out of 12 tasks in *gym* domain. CQL is the best performer among all model-free methods, and ROMI-CQL can further outperform CQL on four challenging tasks. In this domain, current model-based methods perform pretty well. We argue that similar to forward imagination, reverse direction can also generalize beyond offline datasets for better performance in the relatively safe tasks, i.e., there is no obstacle around, as shown in Figure 2.

### 4.3 Ablation Study with Model-based Imagination

In this subsection, we conduct an ablation study to investigate whether ROMI works due to the reverse model-based imagination. Specifically, we replace the reverse imagination with the forward direction in ROMI, which is denoted as *Forward rROMI (FOMI)*. In this subsection, we study the performance of ROMI and FOMI in *maze2d* and *antmaze* domains and defer the ablation study in *gym* domain to Appendix F. Towards fair comparison, we integrate FOMI with BCQ [8] in these settings, called *FOMI-BCQ*. Table 3 shows that ROMI-BCQ significantly outperforms FOMI-BCQ and the base model-free method BCQ [8], which implies that reverse model-based imagination is critical for ROMI in the offline RL settings.

In the *maze2d* domain, compared with the base model-free algorithm BCQ, ROMI-BCQ outperforms all settings, while FOMI-BCQ achieves the superior performance in *maze2d-medium* but performs poorly in the *umaze* and *large* layouts. As illustrated in Figure 2, *maze2d-medium* enjoys less obstacles on the diagonal to the goal than *maze2d-umaze* and *maze2d-large*. In this case, the conservative reverse model-based imagination can enable safe generalization in all layouts. A detailed case study of *maze2d-umaze* will be provided in Section 4.5. Moreover, in the *antmaze* domain, ROMI-BCQ achieve the best performance in the *medium* and *large* layouts, while FOMI-BCQ and BCQ perform well in *antmaze-umaze*. From Figure 2, we find that the *medium* and *large* layouts of *antmaze* have larger mazes with narrower passages, which may frustrate forward imagination and make reverse imagination more effective.

Table 3: Ablation study about ROMI with model-based imagination. Delta equals the improvement of ROMI-BCQ over BCQ on the normalized return metric.

Dataset type	Environment	ROMI-BCQ (ours)	FOMI-BCQ	BCQ (base)	Delta
sparse	maze2d-umaze	<b>139.5</b> ± 3.6	8.1 ± 15.5	41.1 ± 7.6	98.4
sparse	maze2d-medium	82.4 ± 15.2	<b>93.6</b> ± 41.3	9.7 ± 14.2	72.7
sparse	maze2d-large	<b>83.1</b> ± 22.1	-2.5 ± 0.0	38.3 ± 10.4	44.8
dense	maze2d-umaze	<b>98.3</b> ± 2.5	30.7 ± 0.9	37.0 ± 5.3	61.3
dense	maze2d-medium	<b>102.6</b> ± 32.4	64.7 ± 37.0	37.9 ± 4.5	64.7
dense	maze2d-large	<b>124.0</b> ± 1.3	-0.7 ± 7.1	79.8 ± 12.2	44.2
fixed	antmaze-umaze	68.7 ± 2.7	<b>79.5</b> ± 2.5	75.3 ± 13.7	-6.6
play	antmaze-medium	<b>35.3</b> ± 1.3	26.2 ± 5.5	0.0	35.3
play	antmaze-large	<b>20.2</b> ± 14.8	12.0 ± 3.3	0.0	20.2
diverse	antmaze-umaze	61.2 ± 3.3	<b>66.8</b> ± 3.5	49.3 ± 9.9	11.9
diverse	antmaze-medium	<b>27.3</b> ± 3.9	12.3 ± 2.1	0.0	27.3
diverse	antmaze-large	<b>41.2</b> ± 4.2	17.8 ± 2.1	0.0	41.2

### 4.4 Ablation Study with Different Rollout Policies

In this subsection, we conduct an ablation study to investigate the effect of ROMI’s different rollout policies. To compare with a CVAE-based policy, we propose a new rollout policy (i.e., reverse behavior cloning) for ROMI-BCQ, denoted by *ROMI-RBC-BCQ*. To realize the reverse behavior cloning method, we train a stochastic policy  $\hat{\pi}_\varphi(a|s')$ , which is parameterized by  $\varphi$  and can sample current action depended on the next state. During training  $\hat{\pi}_\varphi(a|s')$ , the objective  $\mathcal{L}_{rbc}(\varphi)$  is formalized by

$$\mathcal{L}_{rbc}(\varphi) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}_{env}} [-\log \hat{\pi}_\varphi(a|s')], \tag{4}$$

where  $\mathcal{D}_{env}$  is the given offline dataset, and minimizing the loss function  $\mathcal{L}_{rbc}(\varphi)$  is equivalent to maximizing the log-likelihood of probability.

We illustrate the performance of ROMI with different rollout policies in Table 4, where ROMI-BCQ achieves the best performance and ROMI-RBC-BCQ also outperforms BCQ. As suggested by prior generative models [21, 20], in comparison to the deterministic layers (e.g., fully connected layers), CVAE-based methods can generate more diverse and realistic structured samples using stochastic inference. We argue that, the rollout policy implemented by CVAE is critical for ROMI to rollout more diverse trajectories for proposed generalization, and BC-based implementation is also effective in reverse model-based imagination.



Table 4: Ablation study about ROMI with different rollout policies.

Dataset type	Environment	ROMI-BCQ (ours)	ROMI-RBC-BCQ	BCQ (base)
fixed	antmaze-umaze	68.7 $\pm$ 2.7	62.2 $\pm$ 5.6	<b>75.3</b> $\pm$ 13.7
play	antmaze-medium	<b>35.3</b> $\pm$ 1.3	<b>33.8</b> $\pm$ 6.2	0.0
play	antmaze-large	<b>20.2</b> $\pm$ 14.8	13.3 $\pm$ 16.1	0.0
diverse	antmaze-umaze	<b>61.2</b> $\pm$ 3.3	43.8 $\pm$ 13.3	49.3 $\pm$ 9.9
diverse	antmaze-medium	<b>27.3</b> $\pm$ 3.9	20.8 $\pm$ 15.5	0.0
diverse	antmaze-large	<b>41.2</b> $\pm$ 4.2	14.2 $\pm$ 9.8	0.0

#### 4.5 A Case Study in *maze2d-umaze*

To dive deeper into how ROMI triggers more conservative and effective behaviors, we provide a detailed visual demonstration of one particular task in D4RL benchmark [18]: *maze2d-umaze-sparse*.

As mentioned in Section 4.1, experiences in *maze2d* domain are generated by a planner moving between randomly sampled waypoints on the clearing. Figure 3a shows the movement of the agent from randomly sampled trajectories. To earn high returns, the agent not only needs to learn how to direct to the goal, but also how to stay in the high reward region — the latter behavior is not in the dataset yet. Model-free offline RL algorithms constrain their policy "close" to the dataset, thus it is hard to learn such behaviors out of the support. To see this, Figure 3d shows the behavior of a trained BCQ policy during execution. After reaching the goal, the agent will still oscillate between the high-reward and low-reward regions. This motivates us to use model imagination to generalize beyond the dataset.

As shown in Table 1, ROMI solves this task but previous model-based methods have poor performance, sometimes even worse than model-free algorithms. To better understand this counter-intuitive phenomenon, we compare the rollout trajectories and the learned policy of ROMI-BCQ, FOMI-BCQ (mentioned in Section 4.3), and MOPO [13]. Figure 3(g-i) shows the imagined trajectories in the learning process of the three methods. Figure 3(b,c,e) shows the learned policy behavior at the execution phase. While all model-based imagination will leave the dataset for better generalization, forward model rollout naturally takes some risks as it directs the agent to unknown areas. Undesired forward model imagination will ruin the policy learning (e.g., FOMI-BCQ in Figure 3e and Table 3) or mislead the policy optimization to the suboptimal solution (e.g., MOPO in Figure 3c). Moreover, as shown in Figure 3f, the regularization penalty based on model uncertainty also failed, which is also pointed out in the literature [15]. On the other hand, reverse model imagination inherits the conservatism of the dataset, as it is always ended in the real experience points. Figure 3(b,g) shows that ROMI induces the conservative and optimal behaviors: ROMI will stay around the goal point, and will stick to the data points for higher expected returns. We aim to quantify the aggressiveness of each learned policy and thus define the following trajectory-based discrepancy to measure the distance between learning policy and dataset:

**Definition 1** (Average Trajectory Discrepancy). *Given a dataset  $\mathcal{D} = \{(s, a, r, s')\}$  and a trajectory  $\tau = (s_0, a_0, \dots, s_H, a_H)$ , the discrepancy between  $\mathcal{D}$  and  $\tau$  is defined as:*

$$\mathbb{D}(\mathcal{D}, \tau) = \frac{1}{H+1} \sum_{t=0}^H \min_{(s, a, r, s') \in \mathcal{D}} \|s_t - s\|_2. \quad (5)$$

We report the average trajectory discrepancy for policies during execution in Figure 3j and defer results for other maze environments in Appendix G. The results implicate that ROMI is as conservative as model-free method BCQ, but forward model-based offline RL policy (i.e., ROMI and MOPO) tends to deviate from the dataset and will touch the undesirable walls.

## 5 Related Work

**Offline RL.** In the literature, offline RL methods can be primarily divided into model-free and model-based algorithms. Prior model-free offline RL methods constrain their policy search in the offline dataset. They can realize their bias by either estimating uncertainty quantification to the value function [7, 9, 10, 16], using importance sampling based algorithms [22–28], explicitly constraining

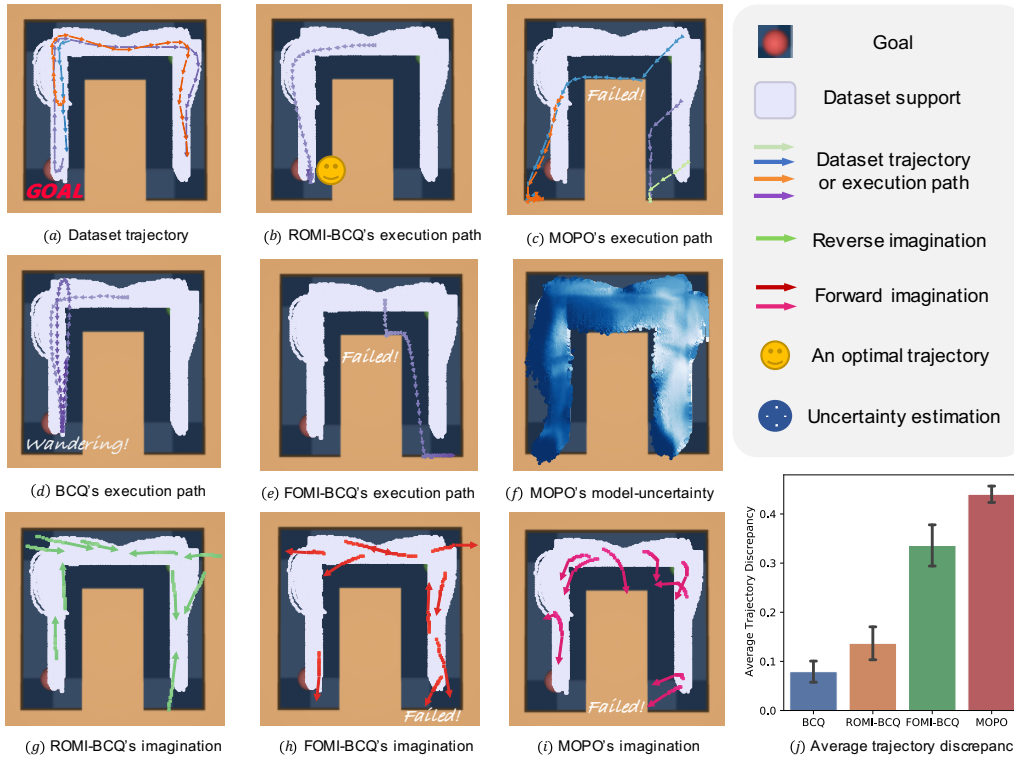


Figure 3: A case study in *maze2d-umaze*. Note that in figure (f), the penalty is reflected by the color, where blue represents large penalty while white represents small penalty.

the learning policy to be close to the dataset [8, 29], learning the conservative value function [11], and using KL divergence [30, 9, 31] or MMD [10]. On the other hand, prior model-based offline RL methods [12–14, 32–38] have studied in model-uncertainty quantification [12, 13, 39], representation learning [14], constraining the policy to imitate the behavioral policy [35], and using conservative estimation of value function [15]. Different from these works, we propose ROMI to investigate a new direction in the model-based offline RL, which will provide natural conservatism bias with maintaining superior model-based generalization benefits.

**Reverse model-based RL.** The idea of learning a reverse dynamics model to imagine reversal samples from goal states first emerges in the literature of online RL algorithms [19, 40–43], and shows the potential of speeding up learning and improving sample efficiency by aimed exploration. Similar backward model shows benefit in planning for credit assignment [44, 45] and robustness [46]. Lai et al. (2020) [42] utilizes a backward model to reduce the reliance on accuracy in forward model predictions. In contrast to the backward model in online RL, we show that reverse imagination inherently incorporates the conservatism into rollout trajectories in the offline RL, which can lead rollouts towards target states in dataset. To our best knowledge, ROMI is the first offline RL method with reverse model-based imagination to induce conservatism bias with data augmentation.

## 6 Conclusion

This paper introduces ROMI, a novel model-based framework for offline RL. To enable conservative generalization, it adopts reverse dynamics model to imagine possible trajectories that can lead to the states within the offline dataset. We demonstrate ROMI can effectively combine with off-the-shelf model-free algorithms and achieve state-of-the-art performance on offline RL benchmark tasks. ROMI leverages good trajectories (e.g., reaching goal states or with high rewards) in an offline dataset to generate effective augmentation data. Although most practical offline datasets contain such trajectories, it is challenging for ROMI to work in other rare cases. One potential future direction is to extend ROMI with forward models, generating bolder imaginations and reaching goal states outside a given dataset. Theoretical formalization of the reverse imagination in offline RL is also an interesting future direction.

## Acknowledgments and Disclosure of Funding

The authors would like to thank the anonymous reviewers, Zhizhou Ren, Kun Xu, and Hang Su for valuable and insightful discussions and helpful suggestions. This work is supported in part by Science and Technology Innovation 2030 – “New Generation Artificial Intelligence” Major Project (No. 2018AAA0100904), a grant from the Institute of Guo Qiang, Tsinghua University, and a grant from Turing AI Institute of Nanjing.

## References

- [1] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.
- [2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- [3] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [4] Omer Gottesman, Fredrik Johansson, Matthieu Komorowski, Aldo Faisal, David Sontag, Finale Doshi-Velez, and Leo Anthony Celi. Guidelines for reinforcement learning in healthcare. *Nature medicine*, 25(1):16–18, 2019.
- [5] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018.
- [6] Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning*, pages 45–73. Springer, 2012.
- [7] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.
- [8] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062. PMLR, 2019.
- [9] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.
- [10] Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *arXiv preprint arXiv:1906.00949*, 2019.
- [11] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.
- [12] Rahul Kidambi and Aravind Rajeswaran. Morel: Model-based offline reinforcement learning. *Neurips*, 2020.
- [13] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*, 2020.
- [14] Byung-Jun Lee, Jongmin Lee, and Kee-Eung Kim. Representation balancing offline model-based reinforcement learning. In *International Conference on Learning Representations*, 2021.
- [15] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *arXiv preprint arXiv:2102.08363*, 2021.

- [16] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114. PMLR, 2020.
- [17] Xinyue Chen, Zijian Zhou, Zheng Wang, Che Wang, Yanqiu Wu, and Keith Ross. Bail: Best-action imitation learning for batch deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.
- [18] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.
- [19] Keith J Holyoak and Dan Simon. Bidirectional reasoning in decision making by constraint satisfaction. *Journal of Experimental Psychology: General*, 128(1):3, 1999.
- [20] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.
- [21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [22] Doina Precup, Richard S Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *ICML*, pages 417–424, 2001.
- [23] Richard S Sutton, A Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 17(1):2603–2631, 2016.
- [24] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Off-policy policy gradient with stationary distribution correction. In *Uncertainty in Artificial Intelligence*, pages 1180–1190. PMLR, 2020.
- [25] Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research*, 16(1):1731–1755, 2015.
- [26] Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.
- [27] Avi Singh, Albert Yu, Jonathan Yang, Jesse Zhang, Aviral Kumar, and Sergey Levine. Cog: Connecting new skills to past experience with offline reinforcement learning. *arXiv preprint arXiv:2010.14500*, 2020.
- [28] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.
- [29] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Provably good batch reinforcement learning without great exploration. *arXiv preprint arXiv:2007.08202*, 2020.
- [30] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.
- [31] Wenxuan Zhou, Sujay Bajracharya, and David Held. Plas: Latent action space for offline reinforcement learning. *arXiv preprint arXiv:2011.07213*, 2020.
- [32] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.
- [33] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

- [34] Gregory Kahn, Adam Villaflor, Pieter Abbeel, and Sergey Levine. Composable action-conditioned predictors: Flexible off-policy learning for robot navigation. In *Conference on Robot Learning*, pages 806–816. PMLR, 2018.
- [35] Tatsuya Matsushima, Hiroki Furuta, Yutaka Matsuo, Ofir Nachum, and Shixiang Gu. Deployment-efficient reinforcement learning via model-based offline optimization. *arXiv preprint arXiv:2006.03647*, 2020.
- [36] Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. *arXiv preprint arXiv:2008.05556*, 2020.
- [37] Phillip Swazinna, Steffen Udluft, and Thomas Runkler. Overcoming model bias for robust offline deep reinforcement learning. *arXiv preprint arXiv:2008.05533*, 2020.
- [38] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. *arXiv preprint arXiv:2012.11547*, 2020.
- [39] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.
- [40] Anirudh Goyal, Philemon Brakel, William Fedus, Soumye Singhal, Timothy Lillicrap, Sergey Levine, Hugo Larochelle, and Yoshua Bengio. Recall traces: Backtracking models for efficient reinforcement learning. *arXiv preprint arXiv:1804.00379*, 2018.
- [41] Ashley D Edwards, Laura Downs, and James C Davidson. Forward-backward reinforcement learning. *arXiv preprint arXiv:1803.10227*, 2018.
- [42] Hang Lai, Jian Shen, Weinan Zhang, and Yong Yu. Bidirectional model-based policy optimization. In *International Conference on Machine Learning*, pages 5618–5627. PMLR, 2020.
- [43] Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 5757–5766. PMLR, 2020.
- [44] Hado P van Hasselt, Matteo Hessel, and John Aslanides. When to use parametric models in reinforcement learning? *Advances in Neural Information Processing Systems*, 32:14322–14333, 2019.
- [45] Veronica Chelu, Doina Precup, and Hado van Hasselt. Forethought and hindsight in credit assignment. *arXiv preprint arXiv:2010.13685*, 2020.
- [46] Taher Jafferjee, Ehsan Imani, Erin Talvitie, Martha White, and Micheal Bowling. Hallucinating value: A pitfall of dyna-style planning with imperfect environment models. *arXiv preprint arXiv:2006.04363*, 2020.
- [47] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *arXiv preprint arXiv:1906.08253*, 2019.

## A An Illustrative Example for ROMI

Assume state  $s_{\text{in}}$  has a trajectory leading to the goal (denoted by  $s_{\text{goal}}$ ) in the dataset. Considered below are two five-step trajectories from forward imagination and reverse imagination, which visit the same state sequence, but in the opposite direction (hence the actions are not the same). Let the forward imaginary trajectory be from  $s_{\text{in}}$  to  $s_5$ , where  $s_5$  is the end state of the trajectory. Similarly, let the reverse imaginary trajectory be from  $s_5$  to  $s_{\text{in}}$ . Note that the rollout orders of these two trajectories are the same, starting at  $s_{\text{in}}$  and ending at  $s_5$ . We formalize these two imaginary trajectories as follows:

**Forward rollout:**  $\langle s_{\text{in}}, a_1^f, s_1, a_2^f, s_2, a_3^f, s_3, a_4^f, s_4, a_5^f, s_5 \rangle$ ,

**Reverse rollout:**  $\langle s_5, a_5^r, s_4, a_4^r, s_3, a_3^r, s_2, a_2^r, s_1, a_1^r, s_{\text{in}} \rangle$ ,

where  $s_{1:5}$  denotes the imaginary state sequence,  $a_{1:5}^f$  denotes the action sequence for forward imagination, and  $a_{1:5}^r$  denotes the action sequence for reverse imagination.

During the training process, the reverse rollout will expand the trajectory from  $\langle s_{\text{in}}, \dots, s_{\text{goal}} \rangle$  to  $\langle s_5, \dots, s_{\text{in}}, \dots, s_{\text{goal}} \rangle$ , i.e.,  $s_{1:5}$  can now reach the goal. Through the reverse imagination of  $s_{1:5}$ , the reverse rollout can benefit the policy learning in this task. For the forward rollout, there are three cases for the state  $s_5$ :

1. Consider  $s_5$  is outside the dataset and the value of  $s_5$  (i.e.,  $Q(s_5, \cdot)$ ) is overestimated, which is a common challenge for function approximation in offline RL [8]. The forward rollout can mislead the learning policy from  $s_{\text{in}}$  to  $s_5$  and harm the learning performance. In contrast, the reverse rollout does not have such negative impact. It is because that  $s_{\text{in}}$  does not have an action sequence to reach  $s_5$  in the reversed data augmentation.
2. If  $s_5$  is outside the dataset and the value of  $s_5$  is not overestimated, the forward rollout does not affect the policy learning. It is because that the execution policy will not go to the state  $s_5$  with a lower  $Q(s_5, \cdot)$ . In contrast, as discussed above, the reverse rollout would benefit the learning in this case due to its effective trajectory expansion with the opposite direction of imagination.
3. Consider  $s_5$  is inside the dataset. When  $s_5$  has a trajectory leading to the goal in the dataset, we believe the forward rollout can improve the learning by selecting the better of the two successful trajectories,  $\langle s_{\text{in}}, \dots, s_{\text{goal}} \rangle$  and  $\langle s_{\text{in}}, \dots, s_5, \dots, s_{\text{goal}} \rangle$ . In the reversed data augmentation, there is only one successful trajectory from the state  $s_{\text{in}}$ , i.e.,  $\langle s_{\text{in}}, \dots, s_{\text{goal}} \rangle$ , but this could naturally be compensated by generating a reverse rollout from state  $s_5$ . Note that, in this case, both forward and reverse methods do not need to deal with the conservatism issue, because  $s_5$  is in the original offline dataset.

In summary, we would like to highlight that reverse imagination is more conservative for imagined states outside the dataset and may even have overly estimated values. This explains why it works well in offline RL.

## B Omitted Background in Section 3

**Conditional Variational Auto-Encoder** A variational auto-encoder (VAE) [21] is a generative model of  $p_\theta(X) = \prod_{i=1}^n p_\theta(x_i)$ , where  $X = \{x_1, \dots, x_n\}$  denotes the given dataset and  $\theta$  denotes the parameters of approximate ML or MAP estimation. It leverages the latent variable model structure of  $p_\theta(X) = \int p_\theta(X|z)p(z)dz$ , where  $z$  is the latent variable with prior distribution  $p(z)$ . To perform tractable maximum log-likelihood, VAE optimizes the following variational lower bound:

$$\log p_\theta(X) \geq \mathbb{E}_{z \sim q_\xi(\cdot|X)} [\log p_\theta(X|z)] - D_{\text{KL}}(q_\xi(z|X) \| p(z)), \quad (6)$$

where  $q_\xi(z|X) = \mathcal{N}(z | \mu_\xi(X), \Sigma_\xi(X))$  is the encoder parameterized by  $\xi$  that represents a multivariate Gaussian distribution with mean  $\mu_\xi$  and variance  $\Sigma_\xi$ , and  $p_\theta(X|z)$  is the decoder parameterized by  $\theta$ . With reparametrization trick [21], VAE performs gradient descent on the variational lower bound. For inference,  $z$  is sampled from a Gaussian distribution and passed through the decoder to generate diverse samples  $x$ .

Conditional VAE (CVAE) [20] is a variant of VAE, which aims to model  $p_\theta(Y|X)$ . Similar to variational lower bound, CVAE optimizes the following lower bound:

$$\log p_\theta(Y|X) \geq \mathbb{E}_{z \sim q_\xi(\cdot|Y, X)} [\log p_\theta(Y|z, X)] - D_{\text{KL}}(q_\xi(z|Y, X) \| p(z|X)), \quad (7)$$

where  $q_\xi(z|Y, X)$ ,  $p_\theta(Y|z, X)$  also denote the encoder and decoder parameterized by  $\xi, \theta$  with the evidence  $X$ , respectively.

## C Omitted Quantification of Model Accuracy in Section 4.1

To compare the model accuracy of forward and reverse imagination, we estimate the model error of forward and reverse models. To be fair, we utilize the same neural network architecture to implement these models. We quantify the one-step model error as mean squared error (MSE) on the validation set. We evaluate the forward and reverse model error in five different robots of D4RL benchmark suite [18] illustrated in Table 5. For each robot, we report the MSE across different types of configurations. Table 5 shows that reverse models have comparable accuracy to, if not worse than, forward models.

Table 5: MSE of forward model and reverse model.

Dataset type	Environment	Forward model	Reverse model
sparse	maze2d-umaze	0.0062 $\pm$ 0.0019	0.0061 $\pm$ 0.0019
sparse	maze2d-medium	0.0087 $\pm$ 0.0017	0.0107 $\pm$ 0.0011
sparse	maze2d-large	0.0195 $\pm$ 0.0083	0.0242 $\pm$ 0.0053
dense	maze2d-umaze	0.0014 $\pm$ 0.0005	0.0014 $\pm$ 0.0004
dense	maze2d-medium	0.0008 $\pm$ 0.0003	0.0007 $\pm$ 0.0003
dense	maze2d-large	0.0011 $\pm$ 0.0004	0.0021 $\pm$ 0.0001
fixed	antmaze-umaze	0.1460 $\pm$ 0.0470	0.2027 $\pm$ 0.0086
play	antmaze-medium	0.1225 $\pm$ 0.0212	0.1947 $\pm$ 0.0053
play	antmaze-large	0.1201 $\pm$ 0.0272	0.1875 $\pm$ 0.0058
diverse	antmaze-umaze	0.1509 $\pm$ 0.0494	0.1374 $\pm$ 0.0043
diverse	antmaze-medium	0.1303 $\pm$ 0.0206	0.1743 $\pm$ 0.0139
diverse	antmaze-large	0.1167 $\pm$ 0.0236	0.1813 $\pm$ 0.0031
random	mujoco-walker2d	0.5141 $\pm$ 0.0083	0.7102 $\pm$ 0.0720
random	mujoco-hopper	0.0010 $\pm$ 0.0004	0.0012 $\pm$ 0.0006
random	mujoco-halfcheetah	0.1268 $\pm$ 0.0111	1.4220 $\pm$ 0.1144
medium	mujoco-walker2d	0.2647 $\pm$ 0.0680	0.2312 $\pm$ 0.0048
medium	mujoco-hopper	0.0020 $\pm$ 0.0002	0.0024 $\pm$ 0.0002
medium	mujoco-halfcheetah	0.2437 $\pm$ 0.0156	0.5865 $\pm$ 0.0247
medium-replay	mujoco-walker2d	0.3572 $\pm$ 0.0556	0.6123 $\pm$ 0.0913
medium-replay	mujoco-hopper	0.0037 $\pm$ 0.0005	0.0045 $\pm$ 0.0004
medium-replay	mujoco-halfcheetah	0.4684 $\pm$ 0.0156	1.4916 $\pm$ 0.1705
medium-expert	mujoco-walker2d	0.1447 $\pm$ 0.0157	0.1687 $\pm$ 0.0107
medium-expert	mujoco-hopper	0.0022 $\pm$ 0.0004	0.0019 $\pm$ 0.0005
medium-expert	mujoco-halfcheetah	0.1586 $\pm$ 0.0093	0.5161 $\pm$ 0.0389

## D Experiment Settings and Implementation Details

### D.1 Evaluated Settings of D4RL Benchmark

We consider a wide range of domains in the popular offline RL benchmark, D4RL [18]. D4RL is specifically designed for offline RL setting, which contains many elaborate datasets with key properties related to the practical offline RL applications. As discussed in Section 4.1, we use three D4RL benchmark domains with 24 datasets by controlling five different robots. We adopt the normalized scores metric suggested by D4RL benchmark [18], where 0 indicates a random policy performance and 100 corresponds to an expert. As suggested by (Fu et al., 2020) [18], we illustrate the reference min and max scores of our evaluated D4RL settings in Table 6, where score  $C$  is

normalized to  $\tilde{C}$  by

$$\tilde{C} = \frac{C - \text{reference min score}}{\text{reference max score} - \text{reference min score}}. \quad (8)$$

In the *maze2d* domain, different types of datasets have different reference scores. Moreover, in the *antmaze*, *walker2d hopper*, and *halfcheetah* domains, all domains share the same reference min and max scores. We assume that the termination conditions of tasks are known, which is common in other model-based offline RL baselines [13, 14]. Note that we evaluate ROMI with all baselines in the \*-v2 datasets of *gym* domain (i.e., *walker2d*, *hopper*, and *halfcheetah*). Table 6 also shows the episode length of each task, in which the algorithms will time-out when the execution phase exceeds the time limit of each episode.

Table 6: Reference minmax scores and episode lengths of evaluated D4RL settings.

Dataset type	Environment	Reference min score	Reference max score	Ep. length
sparse	maze2d-umaze	23.85	161.86	300
sparse	maze2d-medium	13.13	277.39	600
sparse	maze2d-large	6.7	273.99	800
dense	maze2d-umaze	68.54	193.66	300
dense	maze2d-medium	44.26	297.46	600
dense	maze2d-large	30.57	303.49	800
—	antmaze-umaze	0.0	1.0	700
—	antmaze-medium	0.0	1.0	1000
—	antmaze-large	0.0	1.0	1000
—	walker2d	1.63	4592.3	1000
—	hopper	-20.27	3234.3	1000
—	halfcheetah	-280.18	12135.0	1000

## D.2 Implementation Details

We train a probabilistic neural network to represent the approximated reverse dynamics and reward model, which takes the next state and current action as input and outputs a diagonal multivariate Gaussian distribution predicting state difference and reward:

$$\hat{p}_\phi(s, r|s', a) = \mathcal{N}(\boldsymbol{\mu}_\phi(s', a), \boldsymbol{\Sigma}_\phi(s', a)), \quad (9)$$

where  $\boldsymbol{\mu}_\phi$  and  $\boldsymbol{\Sigma}_\phi$  denote the mean and variance of reverse model  $\hat{p}_\phi$ , respectively. To boost the model accuracy, following Janner et al. (2019) [47], we train an ensemble of seven such models and pick the best five models according to the validated performance on a hold-out set of 1000 transitions in the offline dataset. During the model-based imagination, we randomly select a model from the best five model candidates to rollout trajectories per step. The neural network architecture of each model has four feedforward layers with 200 hidden units. Each intermediate layer has a *swish* activation, and the inputs and outputs of our models are normalized across the given dataset.

The rollout policy  $\hat{G}_\theta(a|s')$  implemented by a conditional VAE has two components: an encoder  $\hat{E}_\omega(s', a)$  and a decoder  $\hat{D}_\xi(s', z)$ . Both the encoder and the decoder contain two intermediate layers with 750 hidden units and *relu* activations. Besides the common implementation of conditional VAE [20, 8], our rollout policy  $\hat{G}_\theta(a|s')$  takes the next state as input and can sample reverse action using stochastic inference from an underlying latent space. To generate diverse imaginary trajectories in complex tasks, we adopt such conditional VAE as a rollout policy introduced in Section 3. Moreover, we observe that ROMI’s reverse model is quite accurate in the *maze2d* and *hopper* domains, in which ROMI can also use a uniform policy to launch diverse trajectories instead. We introduce the hyperparameters of ROMI in Table 7. As suggested by the evaluated reverse model error in Table 5, we perform short-horizon model rollouts in the 24 tasks of D4RL benchmark, in which ROMI rollouts five steps in domains with higher model accuracy and performs one step imagination in *walker2d* and *halfcheetah*. For *antmaze* with sparse rewards, we prioritize the start point of model imagination, in which the prioritized weights are similarly introduced in BAIL [17].



We combine ROMI with prior offline RL learning algorithms (i.e., BCQ [8] or CQL [11]) to conduct extensive evaluations on D4RL benchmark suite. As discussed in Section 3, ROMI combines the given dataset  $\mathcal{D}_{\text{env}}$  and model-based buffer  $\mathcal{D}_{\text{model}}$  to obtain the total dataset  $\mathcal{D}_{\text{total}}$  for model-free learning methods. During policy training, we collect the training minibatches from two sources,  $\mathcal{D}_{\text{env}}$  and  $\mathcal{D}_{\text{model}}$ , and define the ratio of data collected from the model-based buffer  $\mathcal{D}_{\text{model}}$  as  $\eta \in [0, 1]$ . Therefore, the ratio of data collected from  $\mathcal{D}_{\text{env}}$  is naturally derived by  $1 - \eta$ . We list the hyperparameters  $\eta$  in Table 7, where we search over  $\eta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$  and choose the best one for each domain. We found that data ratio  $\eta = 0.1$  is generally effective for ROMI. The only exception is the *maze2d* domain, in which other ratios were found to work better. Moreover, ROMI’s training time on an NVIDIA RTX 2080TI GPU of each task is about eight hours to 20 hours with the BCQ or CQL learning algorithms.

We compare ROMI with eleven state-of-the-art offline RL baselines: BC (behavior cloning), BCQ [8], BEAR [10], BRAC-v, BRAC-p [9], BAIL [17], CQL [11], MOPO [13], MOREL [12], Repb-SDE [14], and COMBO [15]. Our implementation of these baseline algorithms refers to their public source codes. In particular, we utilize the implementation of BC introduced by the Repb-SDE repository and the implementation of MOREL and COMBO is referred to <https://github.com/SwapnilPande/MOREL/> and <https://github.com/takuseno/d3r1py/>, respectively.

Table 7: Hyperparameters of ROMI used in the D4RL datasets.

Environment	Rollout length	Data ratio $\eta$	Rollout policy
maze2d-umaze	5	0.7	uniform
maze2d-medium	5	0.5	uniform
maze2d-large	5	0.3	uniform
antmaze-umaze	5	0.1	conditional VAE
antmaze-medium	5	0.1	conditional VAE
antmaze-large	5	0.1	conditional VAE
walker2d	1	0.1	conditional VAE
hopper	5	0.1	uniform
halfcheetah	1	0.1	conditional VAE

## E Omitted Tables in Section 4.2

See Table 8.

Table 8: Performance of ROMI with eleven offline RL baselines on the *maze*, *antmaze*, *walker2d*, *hopper*, *halfcheetah* domains with the normalized score metric proposed by D4RL benchmark [18], averaged over three random seeds with  $\pm$  standard deviation. Scores roughly range from 0 to 100, where 0 corresponds to a random policy performance and 100 indicates an expert.

Dataset type	Environment	BC	ROMI-BCQ (ours)	ROMI-CQL (ours)	BCQ	CQL	BEAR	BRAC-p	BRAC-v	BAIL	MOPO	MORel	Reph-SDE	COMBO
sparse	maze2d-umaze	-3.2	139.5 $\pm$ 3.6	54.3 $\pm$ 12.2	49.1	18.9	65.7	-9.2	8.7	44.7	-14.8	-14.4	-12.3	76.4
sparse	maze2d-med	-0.5	82.4 $\pm$ 15.2	13.3 $\pm$ 10.0	17.1	14.6	25.0	70.0	70.6	19.7	37.4	-4.8	-4.7	68.5
sparse	maze2d-large	-1.7	83.1 $\pm$ 22.1	10.2 $\pm$ 5.2	30.8	16.0	81.0	0.2	-2.2	4.4	-0.8	-2.3	-2.2	14.1
dense	maze2d-umaze	-6.9	98.3 $\pm$ 2.5	39.4 $\pm$ 11.8	48.4	14.4	32.6	51.5	17.6	43.3	29.9	36.9	94.3	86.2
dense	maze2d-med	2.7	102.6 $\pm$ 32.4	37.0 $\pm$ 4.1	41.1	30.5	19.1	73.0	24.4	41.7	70.5	-1.9	22.7	84.2
dense	maze2d-large	-0.3	124.0 $\pm$ 1.3	44.0 $\pm$ 1.9	75.2	46.9	133.8	-4.7	-7.0	58.2	36.8	-3.2	7.9	100.2
fixed	antmaze-umaze	82.0	68.7 $\pm$ 2.7	68.1 $\pm$ 5.2	79.5	66.9	0.0	0.0	0.0	58.0	0.0	0.0	0.0	80.3
play	antmaze-med	0.0	35.3 $\pm$ 1.3	40.7 $\pm$ 14.7	0.0	0.0	0.0	0.0	0.0	1.7	0.0	0.0	0.0	0.0
play	antmaze-large	0.0	20.2 $\pm$ 14.8	23.6 $\pm$ 14.7	0.0	0.0	0.0	0.0	0.0	2.2	0.0	0.0	0.0	0.0
diverse	antmaze-umaze	47.0	61.2 $\pm$ 3.3	63.6 $\pm$ 9.5	48.0	6.3	37.7	0.0	0.0	52.0	0.0	0.0	0.0	57.3
diverse	antmaze-med	0.0	27.3 $\pm$ 3.9	20.9 $\pm$ 4.6	0.0	61.5	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0
diverse	antmaze-large	0.0	41.2 $\pm$ 4.2	31.4 $\pm$ 12.1	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
random	mujoco-walker2d	36.0	5.0 $\pm$ 0.1	7.5 $\pm$ 20.0	1.1	2.4	11.1	0.0	0.0	2.4	4.2	37.3	-0.0	7.0
random	mujoco-hopper	0.9	8.0 $\pm$ 0.8	30.2 $\pm$ 4.4	7.9	6.7	31.4	10.0	7.1	8.0	4.1	53.6	31.7	17.9
random	mujoco-halfcheetah	2.2	2.3 $\pm$ 0.0	24.5 $\pm$ 0.7	2.3	19.6	2.3	2.2	2.2	2.2	35.4	25.6	29.1	38.8
med	mujoco-walker2d	41.7	72.4 $\pm$ 0.7	84.3 $\pm$ 1.1	45.2	83.8	-0.3	0.1	-0.3	68.8	-0.2	77.8	85.3	75.5
med	mujoco-hopper	40.0	50.0 $\pm$ 5.7	72.3 $\pm$ 17.5	53.4	66.6	50.8	1.0	0.7	57.5	48.0	95.4	11.2	94.9
med	mujoco-halfcheetah	39.2	46.2 $\pm$ 0.3	49.1 $\pm$ 0.8	47.1	49.0	43.1	0.9	1.7	43.3	69.5	42.1	67.4	54.2
med-replay	mujoco-walker2d	2.2	49.5 $\pm$ 1.6	109.7 $\pm$ 9.8	41.8	88.2	-2.4	1.7	4.3	51.4	69.4	49.8	83.8	56.0
med-replay	mujoco-hopper	8.1	39.1 $\pm$ 2.2	98.1 $\pm$ 2.0	53.2	97.0	30.6	1.2	6.1	54.7	39.1	93.6	5.7	73.1
med-replay	mujoco-halfcheetah	25.6	43.3 $\pm$ 0.3	46.4 $\pm$ 0.7	41.1	47.1	35.7	2.2	2.2	38.9	68.2	40.2	65.5	55.1
med-expert	mujoco-walker2d	73.4	110.7 $\pm$ 0.2	109.7 $\pm$ 9.8	110.7	109.4	-0.2	-0.1	1.1	75.7	-0.3	95.6	111.2	96.1
med-expert	mujoco-hopper	36.0	77.2 $\pm$ 0.6	111.4 $\pm$ 0.4	76.4	106.8	40.5	7.9	0.7	106.4	3.3	108.7	69.8	111.1
med-expert	mujoco-halfcheetah	39.7	89.5 $\pm$ 1.0	86.8 $\pm$ 19.7	91.0	90.8	50.6	2.2	2.3	92.6	72.7	95.6	77.5	90.0

## F Comparison between ROMI-CQL and FOMI-CQL

Similar to Table 3, we compare ROMI-CQL with FOMI-CQL on the *gym* tasks. From the results shown in Table 9, we can see that reverse imagination consistently outperforms forward imagination and that forward rollout sometimes even hurts the policy learning (e.g., in *hopper-medium*). These empirical results are consistent with those on *maze2d* and *antmaze*. Together with Table 2 we find that ROMI-CQL does not achieve the best performance in *halfcheetah*, which may be due to large model error.

Table 9: Performance of ROMI-CQL and FOMI-CQL on Gym-MuJoCo tasks.

Dataset type	Environment	ROMI-CQL (ours)	FOMI-CQL
random	walker2d	7.5 ± 20.0	2.1 ± 0.5
random	hopper	30.2 ± 4.4	8.5 ± 0.3
random	halfcheetah	24.5 ± 0.7	23.0 ± 0.8
medium	walker2d	84.3 ± 1.1	84.2 ± 2.0
medium	hopper	72.3 ± 17.5	1.9 ± 0.1
medium	halfcheetah	49.1 ± 0.8	48.4 ± 1.1
medium-replay	walker2d	109.7 ± 9.8	83.7 ± 4.1
medium-replay	hopper	98.1 ± 2.6	94.2 ± 4.4
medium-replay	halfcheetah	47.0 ± 0.7	46.6 ± 0.4
medium-expert	walker2d	109.7 ± 5.3	109.6 ± 0.3
medium-expert	hopper	111.4 ± 5.6	111.4 ± 1.2
medium-expert	halfcheetah	86.8 ± 19.7	61.2 ± 44.4

## G Average Trajectory Discrepancy for *maze2d-medium-sparse* and *maze2d-large-sparse*

We show the average trajectory discrepancy defined by Eq. (5) in *maze2d-medium-sparse* and *maze2d-large-sparse* environments in Figure 4. The result in *maze2d-medium-sparse* is consistent with that in *maze2d-umaze-sparse*, showing that the reverse imagination is more conservative than the forward counterpart. Note that the discrepancy of ROMI-BCQ is comparable with FOMI-BCQ and MOPO in *maze2d-large-sparse* environment, as the maze is with relatively narrow paths to the goal.

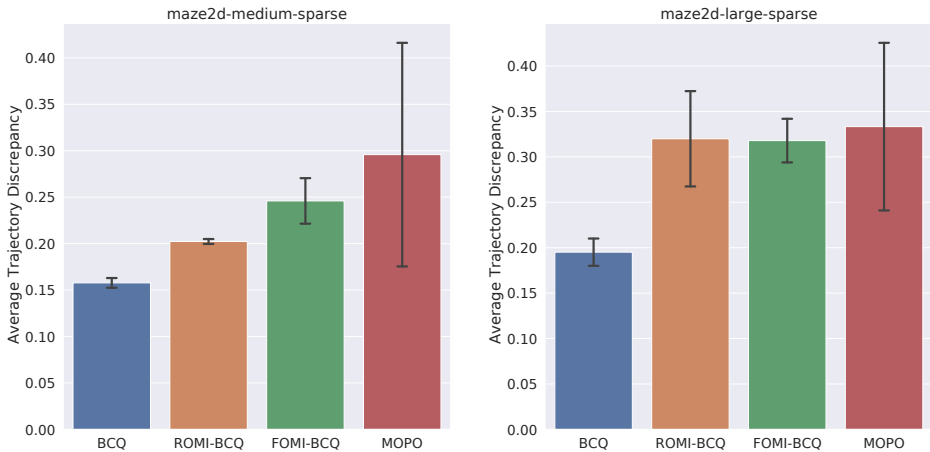


Figure 4: Average Trajectory Discrepancy for *maze2d-medium-sparse* and *maze2d-large-sparse*.

## H Ablations on Rollout Length

To investigate the effect of the rollout length more clearly, we conduct an ablation study by varying this hyperparameter on the *maze2d* and *antmaze* environments. The results are listed in Table 10 and 11. We find that on *maze2d*, ROMI performs equally well or even better when increasing the rollout length to 20, because the model is accurate enough for longer horizon imagination. However, increasing ROMI’s rollout length has a negative impact on *antmaze*, where model cannot predict future states accurately after multi-step rollout.

Table 10: Ablations on rollout length on *maze2d*. \* indicates the hyperparameter used in the main paper.

Environment	len=1	len=5*	len=10	len=20
sparse-maze2d-umaze	82.0 ± 31.5	139.5 ± 3.6	134.4 ± 17.2	135.6 ± 7.3
sparse-maze2d-medium	70.4 ± 28.6	82.4 ± 15.2	82.9 ± 6.2	104.5 ± 54.7
sparse-maze2d-large	120.3 ± 18.3	83.1 ± 22.1	116.2 ± 49.3	76.8 ± 17.2
dense-maze2d-umaze	68.0 ± 11.5	98.3 ± 2.5	113.6 ± 7.0	99.2 ± 10.3
dense-maze2d-medium	88.5 ± 20.9	102.6 ± 32.4	93.2 ± 16.3	83.0 ± 10.1
dense-maze2d-large	120.2 ± 4.8	124.0 ± 1.3	113.6 ± 17.9	97.5 ± 40.7

Table 11: Ablations on rollout length on *antmaze*. \* indicates the hyperparameter used in the main paper.

Environment	len=1	len=5*	len=7	len=10
fixed-antmaze-umaze	69.7 ± 6.5	68.7 ± 2.7	44.2 ± 29.6	54.8 ± 25.4
play-antmaze-medium	36.8 ± 6.8	35.3 ± 1.3	33.5 ± 14.9	28.0 ± 10.4
play-antmaze-large	37.5 ± 6.1	20.2 ± 14.8	7.3 ± 12.7	0
diverse-antmaze-umaze	12.8 ± 19.7	61.2 ± 3.3	58.3 ± 18.5	42.8 ± 14.3
diverse-antmaze-medium	25.3 ± 4.9	27.3 ± 3.9	18.2 ± 9.3	13.5 ± 4.6
diverse-antmaze-large	31.2 ± 2.4	41.2 ± 4.2	32.8 ± 5.1	25.0 ± 5.8