IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 October 2023, accepted 12 December 2023, date of publication 21 December 2023,
date of current version 28 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3345799

RESEARCH ARTICLE

Importance-Weighted Variational Inference Model Estimation for Offline Bayesian Model-Based Reinforcement Learning

TORU HISHINUMA AND KEI SENDA, (Member, IEEE)
Department of Aeronautics and Astronautics, Kyoto University, Kyoto 615-8540, Japan

Corresponding author: Toru Hishinuma (hishinuma.toru.43n@kyoto-u.jp)

This work was supported in part by the Collaborative Research for Evolutionary Mechanical System Technology with Mitsubishi Electric Corporation, and in part by the General Research Grant from the SECOM Science and Technology Foundation.

ABSTRACT This paper proposes a model estimation method in offline Bayesian model-based reinforcement learning (MBRL). Learning a Bayes-adaptive Markov decision process (BAMDP) model using standard variational inference often suffers from poor predictive performance due to covariate shift between offline data and future data distributions. To tackle this problem, this paper applies an importance-weighting technique for covariate shift to variational inference learning of a BAMDP model. Consequently, this paper uses a unified objective function to optimize both model and policy. The unified objective function can be seen as an importance-weighted variational objective function for model training. The unified objective function is also considered as the expected return for policy planning penalized by the model's error, which is a standard objective function in MBRL. This paper proposes an algorithm optimizing the unified objective function. The proposed algorithm performs better than algorithms using standard variational inference without importance-weighting. Numerical experiments demonstrate the effectiveness of the proposed algorithm.

INDEX TERMS Bayesian model-based reinforcement learning, decision-aware reinforcement learning, offline reinforcement learning.

I. INTRODUCTION

Reinforcement learning (RL) is a promising framework for autonomously learning a policy from interaction data [1]. Online model-free RL methods have succeeded in applications where the data can be obtained easily, such as games [2], [3]. However, such methods are often impractical for applications where the data collection is expensive, such as robotics or healthcare [4], [5]. Data-efficiency is one of the fundamental issues in RL.

There are several approaches for increasing data-efficiency in RL. One is model-based reinforcement learning (MBRL). In MBRL, the agent explicitly learns an environment model and utilizes it to improve a policy [6], [7], [8]. Bayesian MBRL is a subfield of MBRL in which the agent explicitly takes uncertainty about an environment model into account [9], [10]. Based on Bayes-optimal exploration/exploitation tradeoff in Bayesian MBRL, the data-efficiency can be further improved. Offline RL is also a data-efficient RL approach [11]. In offline RL, the agent learns a policy from previously collected data. Meta-RL is another approach for data-efficient RL [12]. In meta-RL, the agent learns a policy from data collected from multiple similar environments, assuming that each environment is drawn from some distribution every episode. Combining these data-efficient RL approaches has also been investigated.

Motivated by increasing data-efficiency, this paper discusses a Bayesian MBRL approach for offline meta-RL. A standard model in Bayesian MBRL is a Bayes-adaptive Markov Decision Process (BAMDP) [9], [10]. A task distribution to draw a task instance in meta-RL can be represented as a prior distribution over MDPs in a BAMDP.

The associate editor coordinating the review of this manuscript and approving it for publication was Tony Thomas.

A BAMDP is also reasonable for offline RL, as its goal is offline optimization of possible trial and error under its environment model and prior distribution. For these reasons, a BAMDP is a promising model for offline meta-RL.

Conventional Bayesian MBRL methods assume that a BAMDP is given in advance, implying that an environment is accurately represented by a likelihood function and a prior distribution specified in a BAMDP. This assumption is valid when using a flexible black-box model to infer from sufficient data from a current environment. However, this assumption is often difficult to hold when using a structured model with low-dimensional latent task representation to infer from few data from a current environment. If using an inaccurate model, Bayesian MBRL may not work for a real environment due to failing at belief update [13]. How to address a structured BAMDP is a question.

Recent meta-MBRL research has discussed learning latent variable models based on variational inference framework to obtain latent task representation in meta-RL [14], [15], [16], [17], [18]. A typical approach is to optimize an evidence lower bound that implicitly assumes that the data distribution does not change. Such implicit assumption can also be seen in meta-MBRL but also in MBRL, e.g., [8], [19], and [20]. However, in MBRL, the distribution of data previously collected to train a model differs from the distribution of data obtained in the future when applying a policy improved using the learned model. Such a situation is called covariate shift or distribution shift [11].

In the case of online MBRL, the effect of ignoring covariate shift is relatively mild. This is because the difference between the constantly updated data-collecting policy and the improved policy gradually becomes small in the online setting in which the policy is gradually improved and converged. Indeed, most of the above-mentioned meta-MBRL methods suppose online learning settings. However, in the case of offline MBRL, the difference between the data-collecting policy no longer updated and the improved policy is significant, and thus the effect of ignoring covariate shift is also significant. Prior work [17] addresses another issue that arises in offline meta-MBRL, whereas the issue of covariate shift is out of the discussion.

This paper discusses learning a BAMDP model considering covariate shift. This paper leverages the idea of learning a MDP model considering covariate shift [21]. The main idea of [21] is importance-weighted maximum likelihood estimation weighted by the ratio of the distributions to predict future data more accurately when applying an improved policy. The importance-weighted objective is also an estimate of the expected return in a MDP penalized by model error. The algorithm in [21] optimizes the importance-weighted objective with respect to both model and policy. This paper proposes to extend this idea from MDP model learning to BAMDP model learning. The outline of the discussion is similar to [21]. Firstly, this paper presents a unified objective function viewed as an importance-weighted variational objective function for training a model and as the expected return penalized by model's error for planning a policy. Secondly, this paper proposes an algorithm to optimize it with respect to both model and policy.

This paper and [21] are one of the decision-aware model learning approaches [22], [23]. Prior works [24], [25] are also similar approaches in that they consider importance-weighting with the distribution ratio. The difference is that this paper and [21] consider the data distribution in a simulation MDP model when applying a planned policy, not the data distribution in a real MDP as in other approaches. Using the data distribution in MDP model simulation has two advantages. Firstly, unlike in a real MDP, data when applying a newly planned policy in a simulation MDP model are accessible to the agent, and the importance-weight can be obtained in the standard framework of density ratio estimation [26]. Secondly, optimizing the importance-weighted variational objective with respect to policy takes the same form as standard BAMDP planning, and the proposed algorithm can use an existing BAMDP planning algorithm as a policy planning subroutine.

Sect. II describes the notations of MDP and BAMDP. Sect. III explains the problem setting of offline meta-MBRL in this paper and presents an importance-weighted variational objective. Sect. IV proposes an algorithm to optimize the importance-weighted variational objective. Sect. V demonstrates the effectiveness of the proposed algorithm in numerical experiments. Sect. VI concludes this paper.

II. PRELIMINARY

A. MDP

This paper considers a discounted infinite horizon MDP [27]. Let $\mathcal{S}$ be the state space. Let $\mathcal{A}$ be the action space. Let $\rho(s)$ be the initial state distribution. Let $\mathcal{P}(s'|s, a)$ be the transition probability function. Let $r(s, a)$ be the reward function. Let π be a policy. The state and state-action distributions are

$$d^{\pi}_{\mathcal{P}}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(s^t = s|\rho, \pi, \mathcal{P})$$
$$d^{\pi}_{\mathcal{P}}(s, a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(s^t = s, a^t = a|\rho, \pi, \mathcal{P}).$$

The expected return is

$$\eta^{\pi}_{\mathcal{P}} = \mathbb{E}_{(s^0,a^0,\cdots)\sim\Pr(\cdot|\rho,\pi,\mathcal{P})}\left[\sum_{t=0}^{\infty}\gamma^t r(s^t, a^t)\right]$$
$$= \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim d^{\pi}_{\mathcal{P}}}\left[r(s, a)\right].$$

B. BAMDP

A BAMDP is an augmented MDP whose augmented state is (b^t, s^t), where b^t is the agents' belief over MDPs at timestep t [9], [10]. For simplicity, this paper assumes that reward function r is known. In that case, the agent's belief is over transition probability function $\mathcal{P}$. The prior distribution, i.e., the agent's belief at timestep $t = 0$ is $b^0(\mathcal{P})$. The likelihood function is $l(\mathcal{P}; s^t, a^t, s^{t+1}) = \mathcal{P}(s^{t+1}|s^t, a^t)$. The posterior distribution, i.e., the agent's belief at $t \geq 1$ is updated using the Bayes rule,

$$\begin{aligned} b^{t+1}(\mathcal{P}) &= \Pr(\mathcal{P}|b^0, s^0, a^0, \cdots, s^t, a^t, s^{t+1}) \\ &\propto b^0(\mathcal{P}) \prod_{t'=0}^{t} \mathcal{P}(s^{t'+1}|s^{t'}, a^{t'}) \\ &\propto b^t(\mathcal{P})\mathcal{P}(s^{t+1}|s^t, a^t). \\ &= \Pr(\mathcal{P}|b^t, s^t, a^t, s^{t+1}) \end{aligned}$$

The transition probability function in a BAMDP is

$$\Pr(b', s'|b, s, a) = \mathbb{E}_{\mathcal{P}\sim b}\left[\mathcal{P}(s'|s, a)\right] \delta\left(b' = \Pr(\cdot|b, s, a, s')\right).$$

By the assumption, the reward function in a BAMDP is $r(b, s, a) = r(s, a)$. The expected return in a BAMDP is

$$\begin{aligned} &\mathbb{E}_{b^0, s^0, a^0, \cdots\sim\Pr(\cdot|\rho,\pi,b^0)}\left[\sum_{t=0}^{\infty} \gamma^t r(b^t, s^t, a^t)\right] \\ &= \mathbb{E}_{\mathcal{P}\sim b^0}\left[\mathbb{E}_{(s^0,a^0,\cdots)\sim\Pr(\cdot|\rho,\pi,\mathcal{P})}\left[\sum_{t=0}^{\infty} \gamma^t r(s^t, a^t)\right]\right] \\ &= \frac{1}{1-\gamma}\mathbb{E}_{\mathcal{P}\sim b^0}\left[\mathbb{E}_{(s,a)\sim d^{\pi}_{\mathcal{P}}}\left[r(s, a)\right]\right] = \mathbb{E}_{\mathcal{P}\sim b^0}\left[\eta^{\pi}_{\mathcal{P}}\right]. \quad (1) \end{aligned}$$

A Bayes-optimal policy is a policy that maximizes (1). Since a BAMDP is an augmented MDP whose augmented state is (b, s), a Bayes-optimal policy is a function of (b, s). In principle, when given a BAMDP, i.e., when given likelihood function $l(\mathcal{P}; s, a, s') = \mathcal{P}(s'|s, a)$ and prior distribution $b^0(\mathcal{P})$, a Bayes-optimal policy can be planned offline, as (1) can be computed offline [10].

III. PROBLEM SETTING AND OJECTIVE FUNCTION

This paper assumes a meta-RL setting where a task represented by a MDP is drawn from a distribution. This paper considers optimizing the expected return averaged over MDPs as a reasonable criterion in meta-RL. For simplicity, this paper assumes that state space $\mathcal{S}$, action space $\mathcal{A}$, initial state distribution $\rho(s)$, and reward function $r(s, a)$ are the same between all MDPs. In that case, the expected return averaged over MDPs is $\mathbb{E}_{\mathcal{P}\sim b_0}\left[\eta^{\pi}_{\mathcal{P}}\right]$, which is the same as (1). That is, policy optimization in meta-RL in this setting can be seen as policy optimization in a BAMDP whose likelihood function and prior distribution are specified by $\mathcal{P}$ and b^0 (hereinafter called "the real BAMDP"). As described in Sect. I, this paper considers a setting where the real BAMDP is inaccessible, and only offline data are given. Even in principle, a Bayes-optimal policy cannot be planned offline in this setting, as the real BAMDP is not given. Throughout, this paper discusses a model-based approach to optimize (1) in this setting.

This paper assumes that offline data are collected from M real MDPs sampled from b^0. Let $\mathcal{D}^{ofl}_m = \{(s_{m,n}, a_{m,n}, s'_{m,n})\}_{n=1}^{N}$ be the offline data collected in the m-th real MDP, where $(s_{m,n}, a_{m,n}, s'_{m,n})$ is the n-th transition sample observed in the m-th real MDP. Let $\mathcal{D}^{ofl} = \{\mathcal{D}^{ofl}_m\}_{m=1}^{M}$ be the entire offline data. Let $\mathcal{P}_m$ be the m-th real MDP's transition probability function. Hereinafter, for notational shorthand, this paper uses $sa = (s, a)$, $sa_{m,n} = (s_{m,n}, a_{m,n})$, and $sas'_{m,n} = (s_{m,n}, a_{m,n}, s'_{m,n})$. Let $d^{ofl}_m(sa)$ be the underlying state-action distribution of $sa_{m,n}$.

To represent $\mathcal{P}_m(s'|sa)$ and $\mathcal{P}_m \sim b^0$, the agent uses a latent variable model denoted by $\hat{P}_{\theta,z}(s'|sa)$ and $z \sim \beta^0_{\phi}$, where θ is a model parameter vector shared between MDPs, z is a latent variable vector to specify one MDP, and β^0_{ϕ} is the prior distribution parameterized with ϕ. Hereinafter, this paper refers to a BAMDP model whose likelihood function and prior distribution are specified by $\hat{P}_{\theta,z}$ and β^0_{ϕ} as "the simulation BAMDP." Let β^t_{ϕ} be the agent's belief at timestep t. By the assumption, the reward function in the simulation BAMDP is $r(\beta_{\phi}, sa) = r(sa)$. In the MDP whose transition probability function is $\hat{P}_{\theta,z}$, let $\hat{\eta}^{\pi}_{\theta,z}$ be the expected return, let $\hat{d}^{\pi}_{\theta,z}(sa)$ be the state-action distribution, and let $\hat{\mathcal{D}}^{\pi}_{\theta,z}$ be simulated data collected using policy π.

The model-based meta-RL setting in this paper is summarized as

- the agent trains simulation BAMDP parameter (θ, ϕ) using the offline data obtained in the real BAMDP,
- the agent uses the trained simulation BAMDP to plan policy π to optimize the expected return in the real BAMDP, (1).

Below, this paper discusses how to train (θ, ϕ) and plan π. The first idea is to train (θ, ϕ) to optimize a standard latent variable model learning criterion and then plan π to optimize a standard MBRL criterion. This paper calls it "two-stage optimization." The second idea is to iterate between training (θ, ϕ) and planning π to optimize a unified objective function. This paper calls it "joint optimization." The former is a natural extension of existing methods, whereas the latter is what this paper proposes. Sections III-A-III-B describe objective functions for these ideas, respectively. Sections IV-A-IV-B show algorithms for these objective functions, respectively.

A. OBJECTIVE FUNCTION FOR TWO-STAGE OPTIMIZATION

1) FIRST STAGE: TRAINING (θ, ϕ)

The first stage is to train (θ, ϕ) based on variational inference for latent variable model learning. As a standard method, this paper uses variational autoencoder (VAE) [28]. Given $\mathcal{D}^{ofl}$, the log marginal likelihood function is

$$\begin{aligned} &\ln \Pr(\mathcal{D}^{ofl}|\theta) \\ &= \sum_{m=1}^{M} \ln \Pr(\mathcal{D}^{ofl}_m|\theta) = \sum_{m=1}^{M} \ln \mathbb{E}_{z\sim p}\left[\Pr(\mathcal{D}^{ofl}_m|\theta, z)\right] \\ &= \sum_{m=1}^{M} \ln \mathbb{E}_{z\sim p}\left[\prod_{n=1}^{N} \hat{\mathcal{P}}_{\theta,z}(s'_{m,n}|sa_{m,n})\right], \quad (2) \end{aligned}$$

where $p(z)$ is the prior distribution for VAE learning. Using Jensen's inequality, Equatoin (2) is bounded as

$$\begin{aligned}&\ln \Pr(\mathcal{D}^{ofl}|\theta)\\ &\geq \sum_{m=1}^{M} \mathbb{E}_{z\sim q_\phi(\cdot|\mathcal{D}_m^{ofl})}\left[\sum_{n=1}^{N} \ln \hat{\mathcal{P}}_{\theta,z}(s'_{m,n}|sa_{m,n}) - \ln \frac{q_\phi(z|\mathcal{D}_m^{ofl})}{p(z)}\right],\end{aligned} \tag{3}$$

where $q_\phi(z|\mathcal{D}_m^{ofl})$ is a variational distribution parameterized with ϕ. Let (θ^*, ϕ^*) denote parameters that maximize (3).

The initial belief in the simulation BAMDP is ideally the true latent variable distribution obtained after VAE learning. As a reasonable approximation, this paper uses $\beta^0(z) = \frac{1}{M}\sum_m q_{\phi^*}(z|\mathcal{D}_m^{ofl})$, which can be seen as a latent distribution learned from data and is called average encoding distribution [29] or aggregated posterior [30].

2) SECOND STAGE: PLANNING π

The second stage is to plan π using the simulation BAMDP represented by $\hat{\mathcal{P}}_{\theta^*,z}$ and $\beta^0_{\phi^*}$. The most naive idea is to optimize the expected return in the simulation BAMDP,

$$\mathbb{E}_{z\sim\beta_\phi^0}\left[\hat\eta^\pi_{\theta,z}\right] = \frac{1}{1-\gamma}\mathbb{E}_{z\sim\beta^0_\phi}\left[\mathbb{E}_{sa\sim\hat d^\pi_{\theta,z}}\left[r(sa)\right]\right], \tag{4}$$

with $(\phi,\theta) = (\phi^*,\theta^*)$. However, even in the case of MDP, this idea often only works for offline MBRL [20]. An improved idea is to optimize a penalized expected return in a MDP whose penalized reward function is $r(s,a) - \lambda u(s,a)$, where $u(s,a)$ is an estimate of model's error, and λ is the user-chosen penalty coefficient [20].

Similarly, this paper considers a penalized version of the expected return in the simulation BAMDP. Writing the initial belief explicitly as $\beta^0_\phi(z) = \frac{1}{M}\sum_m q_\phi(z|\mathcal{D}_m^{ofl})$, this paper uses

$$\frac{1}{1-\gamma}\frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{z\sim q_\phi(\cdot|\mathcal{D}_m^{ofl})}\left[\mathbb{E}_{sa\sim\hat d^\pi_{\theta,z}}\left[r(sa)-\lambda u_{m,\theta,z}(sa)\right]\right], \tag{5}$$

with $(\phi,\theta) = (\phi^*,\theta^*)$ as the second stage objective function, where $u_{m,\theta,z}(sa)$ is an estimate of the model's error between $\mathcal{P}_m(\cdot|sa)$ and $\hat{\mathcal{P}}_{\theta,z}(\cdot|sa)$.

B. OBJECTIVE FUNCTION FOR JOINT OPTIMIZATION

In the joint-optimization, this paper gives the agent's belief at timestep $t = 0$ in the form of $\beta^0_\phi(z) = \frac{1}{M}\sum_{m=1}^{M} q_\phi(z|\mathcal{D}_m^{ofl})$, as in the two-stage optimization. This paper also approximates the expected return in the real BAMDP by $\mathbb{E}_{\mathcal{P}\sim b^0}\left[\eta^\pi_{\mathcal{P}}\right] \approx \frac{1}{M}\sum_{m=1}^{M}\eta^\pi_{\mathcal{P}_m}$. The difference between the expected return in the simulation BAMDP and the approximate expected return in the real BAMDP is bounded as

$$\left|\left(\frac{1}{M}\sum_{m=1}^{M}\eta^\pi_{\mathcal{P}_m}\right) - \mathbb{E}_{z\sim\beta^0_\phi}\left[\hat\eta^\pi_{\theta,z}\right]\right| \leq C\sqrt{L(\theta,\phi;\pi) - h_{\min}}, \tag{6}$$

where

$$\begin{aligned} L(\theta,\phi;\pi) &= \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{z\sim q_\phi(\cdot|\mathcal{D}_m^{ofl}),\, sa\sim\hat d^\pi_{\theta,z},\, s'\sim\mathcal{P}_m(\cdot|sa)}\Bigg[\\ &\qquad -\ln\hat{\mathcal{P}}_{\theta,z}(s'|sa) + \nu\ln\frac{q_\phi(z|\mathcal{D}_m^{ofl})}{p(z)}\Bigg]\\ C &= \frac{\gamma\max_{sa}|r(sa)|\sqrt{2}}{(1-\gamma)^2}\\ h_{\min} &= \min_{m,sa}\mathbb{E}_{s'\sim\mathcal{P}_m(\cdot|sa)}\left[-\ln\mathcal{P}_m(s'|sa)\right], \end{aligned}$$

and ν is a constant. For the derivation, see Appendix.

A lower bound of the approximate expected return in the real BAMDP is bounded as

$$\begin{aligned} &\mathbb{E}_{\mathcal{P}\sim b^0}\left[\eta^\pi_{\mathcal{P}}\right]\\ &\quad\approx \frac{1}{M}\sum_{m=1}^{M}\eta^\pi_{\mathcal{P}_m}\\ &\quad\geq \mathbb{E}_{z\sim\beta^0_\phi}\left[\hat\eta^\pi_{\theta,z}\right] - \left|\left(\frac{1}{M}\sum_{m=1}^{M}\eta^\pi_{\mathcal{P}_m}\right) - \mathbb{E}_{z\sim\beta^0_\phi}\left[\hat\eta^\pi_{\theta,z}\right]\right|\\ &\quad\geq \mathbb{E}_{z\sim\beta^0_\phi}\left[\hat\eta^\pi_{\theta,z}\right] - C\sqrt{L(\theta,\phi;\pi) - h_{\min}}. \end{aligned}$$

The first term is the expected return in the simulation BAMDP, and the second penalizes the policy evaluation error between the real and simulation BAMDPs.

Inspired by increasing the objective function by maximizing the lower bound, this paper defines a penalized objective function by

$$J(\theta,\phi,\pi) = \mathbb{E}_{z\sim\beta^0_\phi}\left[\hat\eta^\pi_{\theta,z}\right] - c\sqrt{L(\theta,\phi;\pi) - h_{\min}}, \tag{7}$$

where $c \in [0, C]$ is a user-chosen penalty coefficient. The main idea of the joint optimization is to iteratively optimize (θ,ϕ) and π based on an estimate of (7).

This paper uses the MM framework [31] to optimize (7). When updating from (θ_i,ϕ_i,π_i), the surrogate function is

$$\begin{aligned} &J_{\text{surr}}(\theta,\phi,\pi;\theta_i,\phi_i,\pi_i)\\ &= \mathbb{E}_{z\sim\beta^0_\phi}\left[\hat\eta^\pi_{\theta,z}\right] - \frac{c\left(L(\theta,\phi;\pi) + L(\theta_i,\phi_i;\pi_i) - 2\,h_{\min}\right)}{2\sqrt{L(\theta_i,\phi_i;\pi_i) - h_{\min}}}\\ &= \frac{1}{1-\gamma}\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{z\sim q_\phi(\cdot|\mathcal{D}_m^{ofl}),\, sa\sim\hat d^\pi_{\theta,z}}\Bigg[r(sa)\\ &\quad + \kappa\mathbb{E}_{s'\sim\mathcal{P}_m(\cdot|sa)}\left[\ln\hat{\mathcal{P}}_{\theta,z}(s'|sa)\right] - \kappa\,\nu\ln\frac{q_\phi(z|\mathcal{D}_m^{ofl})}{p(z)}\Bigg]\\ &\quad + \text{const}, \end{aligned} \tag{8}$$

where $\kappa = \frac{c(1-\gamma)}{2\sqrt{L(\theta_i,\phi_i;\pi_i) - h_{\min}}}$. Below, this paper omits the constant term.

1) ESTIMATED OBJECTIVE FUNCTION FOR TRAINING (θ, ϕ)

Equation (8) can be rewritten as

$$\frac{1}{1-\gamma}\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{z\sim q_\phi(\cdot|\mathcal{D}_m^{ofl})}\left[\mathbb{E}_{sa\sim d_m^{ofl}}\left[w_{m,\theta,z}^{\pi}(sa)\left(r(sa)\right.\right.\right.$$
$$\left.\left.\left.+\kappa\mathbb{E}_{s'\sim\mathcal{P}_m(\cdot|sa)}\left[\ln\hat{\mathcal{P}}_{\theta,z}(s'|sa)\right]-\kappa\nu\ln\frac{q_\phi(z|\mathcal{D}_m^{ofl})}{p(z)}\right)\right]\right],$$

where $w_{m,\theta,z}^{\pi}(sa)=\frac{\hat{d}_{\theta,z}^{\pi}(sa)}{d_m^{ofl}(sa)}$. This paper estimates the above equation by

$$\frac{\tilde{\kappa}}{1-\gamma}\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{z\sim q_\phi(\cdot|\mathcal{D}_m^{ofl})}$$
$$\times\left[\sum_{n=1}^{N}\frac{1}{N}\tilde{w}_{m,\theta,z}^{\pi}(sa_{m,n})\ell_{m,n}(\theta,z;\tilde{\kappa})-\nu\ln\frac{q_\phi(z|\mathcal{D}_m^{ofl})}{p(z)}\right], \tag{9}$$

where

$$\ell_{m,n}(\theta,z;\tilde{\kappa})=\ln\hat{\mathcal{P}}_{\theta,z}(s'_{m,n}|sa_{m,n})+\frac{r(sa_{m,n})}{\tilde{\kappa}},$$

$\tilde{w}_{m,\theta,z}^{\pi}$ is an estimate of $w_{m,\theta,z}^{\pi}$, and $\tilde{\kappa}$ is an estimate of κ. How to estimate them is described in Sect. IV-B.

Equation (9) can be interpreted as a kind of variational inference because (9) is similar to (3) in the following points. Firstly, $w_{m,\theta,z}^{\pi}(sa)$ is importance-weighting to address covariate shift between $d_m^{ofl}(sa)$ and $\hat{d}_{\theta,z}^{\pi}(sa)$. Secondly, $\ell_{m,n}(\theta,z;\tilde{\kappa})$ is a utility function modified from the log-likelihood function. Thirdly, ν scales the KL divergence regularization term in the same manner as β-VAE [32]. Based on the interpretation of (9) as a kind of variational inference, this paper uses it to update (θ, ϕ). This paper calls it "importance-weighted variational inference for BAMDP."

2) ESTIMATED OBJECTIVE FUNCTION FOR PLANNING π

Equation (8) can also be rewritten as

$$\frac{1}{1-\gamma}\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{z\sim q_\phi(\cdot|\mathcal{D}_m^{ofl})}\left[\mathbb{E}_{sa\sim\hat{d}_{\theta,z}^{\pi}}\left[r(sa)\right.\right.$$
$$\left.\left.+\kappa\mathbb{E}_{s'\sim\mathcal{P}_m(\cdot|sa)}\left[\ln\hat{\mathcal{P}}_{\theta,z}(s'|sa)\right]\right]-\kappa\nu\ln\frac{q_\phi(z|\mathcal{D}_m^{ofl})}{p(z)}\right].$$

The KL divergence term is constant in terms of π. If the penalty function is $u_{m,\theta,z}(sa)=-\mathbb{E}_{s'\sim\mathcal{P}_m(\cdot|sa)}\left[\ln\hat{\mathcal{P}}_{\theta,z}(s'|sa)\right]$, then optimizing the above equation with respect to π is equivalent to optimizing (5), In practice, however, $\mathbb{E}_{s'\sim\mathcal{P}_m(\cdot|sa)}\left[\ln\hat{\mathcal{P}}_{\theta,z}(s'|sa)\right]$ is inaccessible. This paper replaces the penalty function with a model trained to estimate it, denoted by $\hat{u}_{m,\theta,z}(sa)$. How to train $\hat{u}_{m,\theta,z}(sa)$ is described in Sect. IV-B. Replacing κ with $\tilde{\kappa}$ as in Sect. III-B1, the resulting estimated objective function is

$$\frac{1}{1-\gamma}\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{z\sim q_\phi(\cdot|\mathcal{D}_m^{ofl}),sa\sim\hat{d}_{\theta,z}^{\pi}}\left[r(sa)-\tilde{\kappa}\hat{u}_{m,\theta,z}(sa)\right], \tag{10}$$

which is a penalized version of the expected return in the simulation BAMDP.

Algorithm 1 Two-Stage Optimization

```
Input: $(\mathcal{D}^{ofl}, \lambda)$.
Train $(\theta, \phi)$, variational inference using Equation (3) given $\mathcal{D}^{ofl}$.
Plan $\pi$, planning in simulation BAMDP using Equation (5) given $(\theta, \phi, \lambda)$.
```

a: COMPARISON TO TWO-STAGE OPTIMIZATION

The objective function for planning π is essentially the same for the joint optimization and the two-stage optimization, comparing (10) and (5). In the joint optimization, the objective function for training (θ, ϕ) is relevant to the one for planing π, as (9) and (10) are both estimates of (8). However, in the two-stage optimization, the objective function for training (θ, ϕ) is different from the one for planning π, comparing (3) and (5). In other words, for one objective, the joint optimization optimizes it with respect to both (θ, ϕ) and π, whereas the two-stage optimization does it with respect to only π. As a result, the joint optimization is better than the two-stage optimization in terms of optimizing one objective.

b: ADVANTAGE OF USING $\frac{\hat{D}_{\theta,Z}^{\pi}(SA)}{D_M^{OFL}(SA)}$ AS IMPORTANCE-WEIGHT

It is also possible to consider importance-weighting with $\frac{d_m^{\pi}(sa)}{d_m^{ofl}(sa)}$, becase another bound similar to (6) can also be derived by replacing $\hat{d}_{\theta,z}^{\pi}$ in $L(\theta, \phi; \pi)$ with d_m^{π}, see Sect.IV of [21]. However, in that case, the resulting variant of (10) does not have the same form as the objective function of a BAMDP planning problem. One advantage of using $\frac{\hat{d}_{\theta,z}^{\pi}(sa)}{d_m^{ofl}(sa)}$ is that (10) is a BAMDP planning objective function and can be optimized using an existing BAMDP planning algorithm. Another advantage is that, since the agent cannot access data sampled from $d_m^{\pi}(sa)$ in the real BAMDP but can generate data sampled from $\hat{d}_{\theta,z}^{\pi}(sa)$ in the simulation BAMDP, $\frac{\hat{d}_{\theta,z}^{\pi}(sa)}{d_m^{ofl}(sa)}$ can be obtained in the standard framework of density ratio estimation [26], which is a simpler setting.

IV. ALGORITHM

A. ALGORITHM FOR TWO-STAGE OPTIMIZATION

The main idea of the two-stage optimization is to train BAMDP parameter (θ, ϕ) and subsequently plan policy π. Algorithm 1 shows the outline.

Algorithm 2 Joint Optimization

```
1: Input: $(\mathcal{D}^{ofl}, \nu, c)$.
2: for $i = 1, 2, \cdots$ do
3:   if $i = 1$ then
4:     Train $(\theta, \phi)$, variational inference using Equation (3) given $\mathcal{D}^{ofl}$.
5:   else
6:     Train $(\theta, \phi)$, importance-weighted variational inference using Equation (9) given $(\mathcal{D}^{ofl}, \nu, c, \pi)$, see Algorithm 3.
7:   end if
8:   Plan $\pi$, planning in simulation BAMDP using Equation (10) given $(\theta, \phi, \nu, c)$.
9: end for
```

1) FIRST STAGE: TRAINING (θ, ϕ)

Line 2 in Algorithm 1 optimizes (3), which is variational inference for latent variable model learning. To represent q_ϕ, this paper uses permutation-invariant amortized inference networks [33],

$$q_\phi(z|\mathcal{D}_m^{ofl}) = \mathcal{N}\left(\mu_\phi\left(\sum_{n=1}^{N} f_\phi(sas'_{m,n})\right), \sigma_\phi\left(\sum_{n=1}^{N} f_\phi(sas'_{m,n})\right)\right). \tag{11}$$

Below, for notational shorthand, this paper uses $\mu_{\phi,m} = \mu_\phi\left(\sum f_\phi(sas'_{m,n})\right)$ and $\sigma_{\phi,m} = \sigma_\phi\left(\sum f_\phi(sas'_{m,n})\right)$.

2) SECOND STAGE: PLANNING π

Line 3 in Algorithm 1 optimizes (5), which is policy planning. Inspired by VariBAD [16], this paper approximately gives an augmented state in the BAMDP by a pair of a state and a variational approximation of the belief. To reduce computational efforts, as the prior for the variational approximation of the belief, this paper uses a variational distribution that minimizes the KL divergence from $\beta^0(z) = \frac{1}{M}\sum_m q_{\phi^*}(z|\mathcal{D}_m^{ofl})$. As the likelihood function for the variational approximation of the belief, this paper uses $\hat{\mathcal{P}}_{\theta,z}$, the decoder trained as in Line 2. This paper trains $u_{m,\theta,z}(sa)$ in (5) using input data $\{(sa_{n,m}, z, \mu_{\phi,m}, \ln\sigma_{\phi,m})\}_{n,m}$ and output data $\{-\ln\hat{\mathcal{P}}_{\theta,z}(s'_{n,m}|sa_{n,m})\}_{n,m}$.

B. ALGORITHM FOR JOINT OPTIMIZATION

The main idea of the joint optimization is to iterate between training (θ, ϕ) and planning π. Algorithm 2 shows the outline.

1) TRAINING (θ, ϕ)

At the first iteration, where π remains an initial value, importance-weighting depending on π is not reasonable. Line 4 in Algorithm 2 optimizes (3), as with the two-stage optimization. At the subsequent iterations, Line 6 in Algorithm 2 optimizes (9). Below, this paper discusses how to execute Line 6 concretely.

This paper considers gradient-based optimization of (θ, ϕ). When given $\tilde{\kappa}$ and $\tilde{w}^\pi_{m,\theta,z}(sa)$, the gradient of (9) with respect to ϕ can be estimated using the reparameterization trick [28]. The gradient of (9) with respect to θ is

$$\begin{aligned}&\nabla_\theta\,\{\text{Equation (9)}\}\\ &= \frac{\tilde{\kappa}}{1-\gamma}\frac{1}{MN}\sum_m\sum_n \mathbb{E}_{z\sim q_\phi(\cdot|\mathcal{D}_m^{ofl})}\Bigg[\tilde{w}^\pi_{m,\theta,z}(sa_{m,n})\\ &\quad\times\left(\nabla_\theta \ln\hat{\mathcal{P}}_{\theta,z}(s'_{m,n}|sa_{m,n}) + \ell_{m,n}(\theta, z; \tilde{\kappa})\, v^\pi_{\theta,z}(s_{m,n})\right)\Bigg],\end{aligned} \tag{12}$$

where $v^\pi_{\theta,z}(s) = \nabla_\theta \ln \hat{d}^\pi_{\theta,z}(s)$. Below, this paper describes how to use it approximately.

a: ESTIMATING κ

This paper estimates κ by

$$\begin{aligned}\tilde{\kappa} &= \frac{c(1-\gamma)}{2\sqrt{L(\theta_i, \phi_i; \pi_i) - \tilde{h}_{\min}}}\\ \tilde{h}_{\min} &= \min_{n,m}\left[-\ln\hat{\mathcal{P}}_{\theta_i, z_m}(s'_{m,n}|sa_{m,n})\right]\Bigg|_{z_m=\mu_{\phi_i}(\mathcal{D}_m^{ofl})}.\end{aligned} \tag{13}$$

b: IGNORING $V^\pi_{\theta,Z}(S)$

In principle, $v^\pi_{\theta,z}(s)$ may be estimated by a meta-RL extension of LSDG [34]. However, in practice, estimating $v^\pi_{\theta,z}(s)$ is computationally unrealistic if θ is high-dimensional. Specifically, LSDG in a single MDP RL setting requires estimating the same number of value functions as the dimension of model parameters, and $v^\pi_{\theta,z}(s)$ additionally needs its meta-RL version. In the case of MDP, the numerical experiments in [21] observe that importance-weighted model estimation ignoring this term can also perform better than unweighted model estimation. Assuming that this also holds for a BAMDP, this paper ignores $v^\pi_{\theta,z}(s)$ in the gradient-based optimization.

c: ESTIMATING $W^\pi_{M,\theta,Z}$

Estimating importance-weight $w^\pi_{m,\theta,z}(sa) = \frac{\hat{d}^\pi_{\theta,z}(sa)}{d^{ofl}_m(sa)}$ is meta-learning of density ratio where the source datasets are $\mathcal{D}_m^{ofl}$, the target datasets are $\hat{D}^\pi_{\theta_j,z}$. This paper estimates $w^\pi_{m,\theta,z}$ using neural networks that take $(sa, z, \mu_{\phi,m})$ as input, denoted by $\hat{w}^\pi_{m,\theta,z}$. Since $\mu_{\phi,m}$ encodes data from the m-th MDP, it contains the information of the source distribution. Since z specifies a simulation MDP, it captures the characteristics of the target distribution. Adding latent representations of both source and target distributions to input is inspired by [35].

d: LOCALLY UPDATING θ WHILE FIXING $\hat{W}^\pi_{M,\theta,Z}$

Computational efforts to obtain $\tilde{w}^\pi_{m,\theta,z}$, i.e., meta-learning of density ratio, are not negligible. Instead of computing $\tilde{w}^\pi_{m,\theta,z}$ every updating θ, this paper considers fixing $\tilde{w}^\pi_{m,\theta,z}$ during

Algorithm 3 Importance-Weighted Variational Inference for BAMDP

```
Input: $(\mathcal{D}^{ofl}, \nu, c, \pi)$ and $\theta$.
for $j = 0, 1, \cdots$ do
    $(\theta_j, \phi_j, \pi_j) \leftarrow (\theta, \phi, \pi)$.
    Compute $\tilde{\kappa}$, computing Equation (13).
    Generate $\{\hat{D}^{\pi}_{\theta_j,z}\}_z$, rollout in simulation BAMDP given $(\theta_j, \phi_j, \pi)$.
    Estimate $\tilde{w}^{\pi}_{m,\theta_j,z}$, meta-learning of density ratio given $\mathcal{D}^{ofl}$ and $\{\hat{D}^{\pi}_{\theta_j,z}\}_z$.
    Update $(\theta, \phi)$, optimizing Equation (14) given $(\mathcal{D}^{ofl}, \tilde{\kappa}, \tilde{w}^{\pi}_{m,\theta_j,z}, \nu)$.
end for
```

locally updating θ. When locally updating from θ_j, this paper uses a local approximate objective function defined by

$$\frac{\tilde{\kappa}}{1-\gamma}\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{z\sim q_{\phi}(\cdot|\mathcal{D}_m^{ofl})}\Bigg[-\nu\ln\frac{q_{\phi}(z|\mathcal{D}_m^{ofl})}{p_m(z)} + \sum_{n=1}^{N}\frac{1}{N}\tilde{w}^{\pi}_{m,\theta_j,z}(sa_{m,n})\ell_{m,n}(\theta, z; \tilde{\kappa})\Bigg], \quad (14)$$

where $v^{\pi}_{\theta,z}(s)$ is ignored as described above. The gradient of (14) with respect to θ is

$$\frac{\tilde{\kappa}}{1-\gamma}\frac{1}{MN}\sum_{m}\sum_{n}\mathbb{E}_{z\sim q_{\phi}(\cdot|\mathcal{D}_m^{ofl})} \times\Big[\tilde{w}^{\pi}_{m,\theta_j,z}(sa_{m,n})\nabla_{\theta}\ln\hat{\mathcal{P}}_{\theta,z}(s'_{m,n}|sa_{m,n})\Big].$$

Importance-weighting with $\tilde{w}^{\pi}_{m,\theta_j,z}(sa_{m,n})$ is consistent but can be unstable in practice [36]. To stabilize importance-weighted model learning, this paper replaces $\tilde{w}^{\pi}_{m,\theta_j,z}(sa_{m,n})$ in (14) with $\{\tilde{w}^{\pi}_{m,\theta_j,z}(sa_{m,n})\}^{\alpha}$, as in [21].

Algorithm 3 summarizes how to update (θ, ϕ) described above.

2) OPTIMIZING π

Line 8 in Algorithm 2 optimizes (10) using the same method in Sect. IV-A2.

V. NUMERICAL EXPERIMENTS

A. POLICY EVALUATION

Firstly, to illustrate the effectiveness of importance-weighted variational inference for BAMDP, this paper discusses the problem of predicting behaviors of a given target policy. This problem can be seen as a policy evaluation problem, as the expected return is computed from the predicted behavior. This paper compares predicting behaviors of standard variational inference and importance-weighted variational inference when training BAMDP models expressed using the same NN model.

This paper considers an inverted pendulum task, where state s is a pair of angle and angular velocity, and action a is torque input. The environmental variation in meta-RL is that the viscosity coefficient of the equation of motion behind a real MDP changes every episode. The offline data is collected using a random policy in 100 sampled real MDPs. The target policy is a controller that swings up and stabilizes the pendulum to $(0, 0)$ in the real MDP, whose viscosity coefficient is zero. For more details, see Appendix.

The outline of variational inference is as follows. The agent considers a one-dimensional dimension latent variable z. The agent represents each model by neural networks. For learning each model, the agent uses the data obtained from 80 real MDPs for training and the rest for validation. For regularizing importance-weighting, the agent uses $\alpha = 0.2$. The number of iterations of Algorithm 3 is five. For more details, see Appendix.

Fig. 1 illustrates the predicted behavior when using standard variational inference, i.e., optimizing (3). The 100 subplots correspond to the 100 sampled real MDPs. In each subplot, the horizontal and vertical axes stand for angle and angular velocity, respectively. The black lines show real future data when applying the target policy from initial state $(\pi, 0)$ in each real MDP, which is the ground truth behavior the agent wants to predict. Multiple black line patterns show that the target policy planned for zero viscosity coefficient swings more weakly than expected as the viscosity coefficient increases, finally failing to swing up. The colored markers show simulated future data when applying the target policy from the same initial state in each simulation MDP, whose latent variable is the encoding of the offline data collected in the real MDP in the same subplot. That is, this is the prediction that the agent obtains using the trained model. Note that since the state transition model is estimated as a probabilistic model, there are variations in the predicted behavior, which are drawn in different colors. The red markers indicate $(0, 0)$. The top 20 subplots and the bottom 80 subplots are the real MDPs where the offline data for validation and training are collected, respectively. There is a big difference between the black lines and the colored markers, meaning that the simulation BAMDP trained using standard variational inference does not capture the behavior of the target policy.

Fig. 2 illustrates the predicted behavior when using importance-weighted variational inference for BAMDP. Note that the black lines, i.e., real future data, are the same as Fig. 1. The difference between the black lines and the colored markers in Fig. 2 is small compared to Fig. 1. Thus, the simulation BAMDP trained using importance-weighted variational inference for BAMDP captures the behavior of the target policy more accurately compared to standard variational inference.

Fig. 3 shows the offline data colored based on the logarithm of the importance-weights at the fifth iteration of Algorithm 3. This figure also shows the same black lines as Fig. 1 for reference. Roughly speaking, data points close to the black line are colored brightly, assigning large importance-weighting. Such importance-weighting is

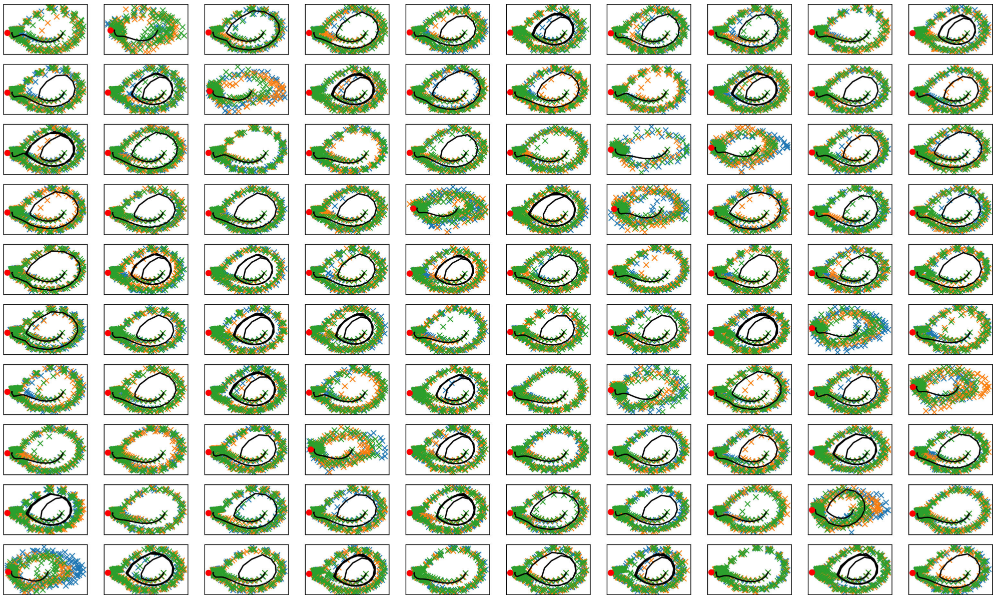

FIGURE 1. Behaviors in real and simulation BAMDPs when using standard variational inference (policy evaluation).

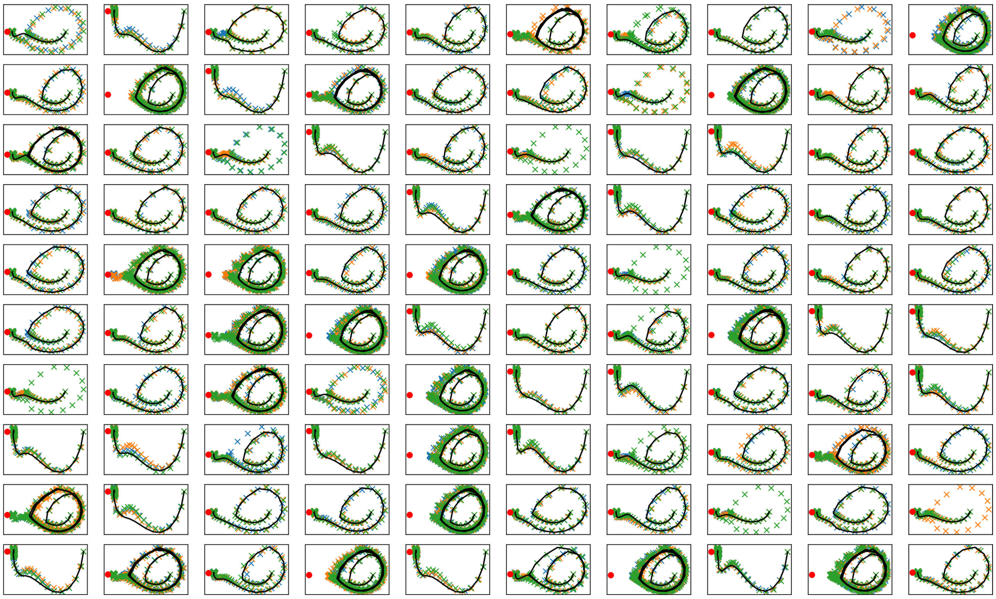

FIGURE 2. Behaviors in real and simulation BAMDPs when using standard variational inference (policy evaluation).

effective for more accurately predicting behaviors of the target policy.

Fig. 4 illustrates the relationship between the real MDP parameter and the simulation MDP latent variable when using standard variational inference. The horizontal axis stands for the viscosity coefficient, which is the real MDP parameter and is inaccessible to the agent. The vertical axis indicates the one-dimensional latent variable mean of the approximate belief, which encodes the offline data collected in the same real MDP and is accessible to the agent. The orange and blue markers are the results of the real MDPs where the offline data for validation and training are collected, respectively. This figure also shows that the simulation BAMDP learned using standard variational inference is not very accurate.

Fig. 5 illustrates the relationship between the real MDP parameter and the simulation MDP latent variable when using importance-weighted variational inference for BAMDP. The magnitude relation of the one-dimensional latent variable accessible to the agent roughly captures the magnitude relation of the viscosity coefficient inaccessible to the agent. This figure also shows that the simulation BAMDP learned using importance-weighted variational inference for BAMDP is more accurate. Note that, for the few subplots that do not capture the ground truth behaviors, the viscosity coefficient

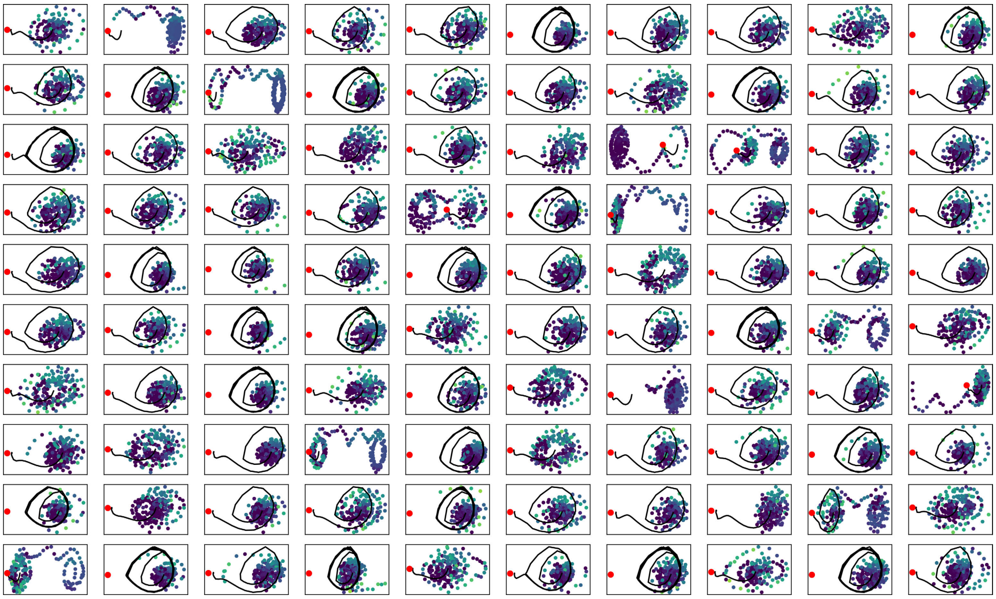

FIGURE 3. Importance-weighting of offline data (policy evaluation).

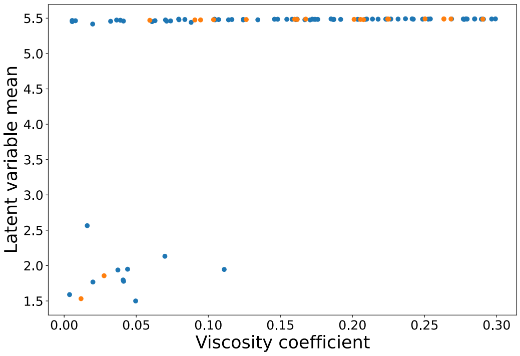

FIGURE 4. Real MDP parameter and simulation mpd latent variable when using standard variational inference (policy evaluation).

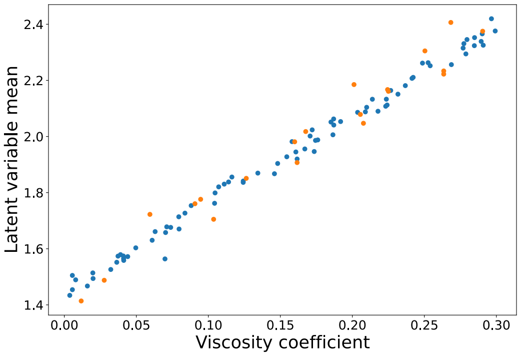

FIGURE 5. Real MDP parameter and simulation mpd latent variable when using standard variational inference (policy evaluation).

is close to the critical point where the target policy cannot swing up.

B. POLICY OPTIMIZATION

Next, this paper discusses policy optimization experiments to demonstrate the effectiveness of the proposed algorithm. This paper presents the results of the inverted pendulum task described in Sect. V-A and a cartpole swing-up task. For the cartpole task, the environmental variation in meta-RL is that the pole mass and the pole length of the equation of motion behind a real MDP change every episode. Similar to the inverted pendulum task, the offline data is collected using a random policy in 100 sampled real MDPs. For more details, see Appendix.

The outline of the two-stage optimization and the joint optimization is as follows. The agent considers a one-dimensional latent variable in the inverted pendulum task and a two-dimensional one in the cartpole swing-up task. The agent uses a decoder with 48 hidden units in the inverted pendulum task and one with 64 hidden units in the cartpole swing-up task. The others are the same between the inverted pendulum and cartpole swing-up tasks. For regularizing importance-weighting, the agent uses $\alpha = 0.2$. The number of iterations of Algorithm 3 is five. The number of iterations of Algorithm 2 is two. The agent uses SAC [37] as a policy planning subroutine to learn an augment-state-dependent policy in the simulation BAMDP. For more details, see Appendix.

Table 1 shows the result of the two-stage optimization and the joint optimization. Note that the two-stage optimization is an existing method, and the joint optimization is the proposed algorithm, as described in Sect. III. For each task, Table 1 reports the score averaged over five runs with different random seeds. For each run, this paper estimates the expected return by averaging the return in 100 sampled real MDPs. For both tasks, the joint optimization achieves better performance.

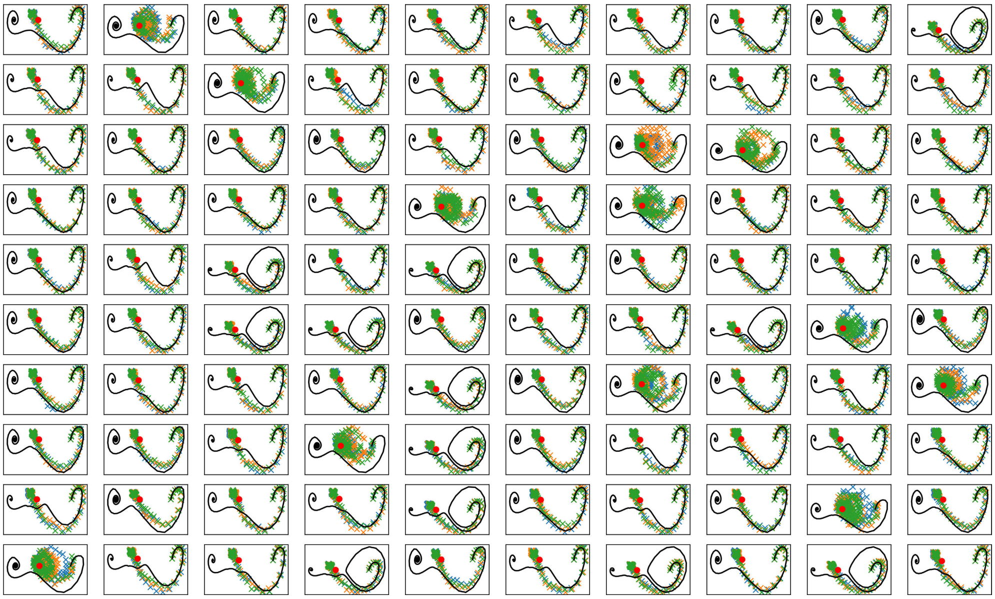

FIGURE 6. Behaviors in real and simulation BAMDPs when planned using two-stage optimization (inverted pendulum policy optimization).

TABLE 1. Performance comparison.

Task	Two-stage optimization	Joint optimization
Inverted pendulum	-53.99	-11.26
Cartpole swing-up	-58.49	-17.77

Figs. 6-7 show the behaviors in the real BAMDP when planned using the two-stage optimization and the joint optimization, respectively. The policy planned using the two-stage optimization cannot stabilize the pendulum around $(0, 0)$ as shown in Fig. 6, leading to the worse performance shown in Table 1. This is because the simulation BAMDP trained by the two-stage optimization cannot accurately represent transitions around $(0, 0)$. The joint optimization trains the simulation BAMDP by assigning larger importance-weighting to data around $(0, 0)$. As a result, the policy planned using the joint optimization can stabilize the pendulum around $(0, 0)$ as shown in Fig. 7, resulting in the better performance shown in Table 1.

VI. CONCLUSION AND FUTURE DIRECTIONS

This paper discusses importance-weighted variational inference to train a BAMDP model in offline Bayesian MBRL. The proposed algorithm optimizes a unified objective function that is an importance-weighted variational objective function for training a model and is a penalized expected return for planning a policy. In theory, since a method using standard variational inference without importance-weighting optimizes an objective function of interest only with respect to a policy, the proposed algorithm is better in terms of optimizing one objective function. In practice, numerical experiments demonstrate that the proposed algorithm can perform better.

Future directions to improve the proposed algorithm will be as follows. Firstly, this paper considers the case where the number of real MDPs collected in offline data, M, is not large. To address a large number of real MDPs, the average encoding distribution, $\beta^0(z) = \frac{1}{M}\sum_m q_{\phi^*}(z|\mathcal{D}_m^{ofl})$, needs to be approximated by a mixture of variational posteriors with pseudo-inputs [38] or a similar technique. Secondly, applying to large-scale tasks is an important challenge. One of the bottlenecks is density ratio estimation in high-dimensional settings, as this is itself a research topic [39], [40]. It is necessary to incorporate recent developments. Thirdly, improving variational inference of BAMDP as a latent variable model is essential for both unweighted and importance-weighted settings.

APPENDIX A
DERIVING POLICY EVALUATION ERROR BOUND

The policy evaluation error between the real and simulation MDPs is bounded as follows (see Sect. IV-A of [21]).

$$|\eta_m^\pi - \hat{\eta}_{\theta,z}^\pi| \le C\sqrt{\xi(\theta, \phi, z; \pi) - h_{\min}^m}, \tag{15}$$

where $\xi(\theta, \phi, z; \pi) = \mathbb{E}_{sa\sim\hat{d}_{\theta,z}^\pi, s'\sim\mathcal{P}_m(\cdot|sa)}\left[-\ln\hat{\mathcal{P}}_{\theta,z}(s'|sa)\right]$ and $h_{\min}^m = \min_{sa}\mathbb{E}_{s'\sim\mathcal{P}_m(\cdot|sa)}\left[-\ln\mathcal{P}_m(s'|sa)\right]$.

Equation (6) is obtained as follows,

$$\begin{aligned}
&\left|\left(\frac{1}{M}\sum_m \eta_{\mathcal{P}_m}^\pi\right) - \mathbb{E}_{z\sim\beta_\phi^0}\left[\hat{\eta}_{\theta,z}^\pi\right]\right| \\
&= \left|\frac{1}{M}\sum_m \left(\eta_{\mathcal{P}_m}^\pi - \mathbb{E}_{z\sim q_m}\left[\hat{\eta}_{\theta,z}^\pi\right]\right)\right| \\
&\le \frac{1}{M}\sum_m \mathbb{E}_{z\sim q_m}\left[|\eta_m^\pi - \hat{\eta}_{\theta,z}^\pi|\right]
\end{aligned}$$

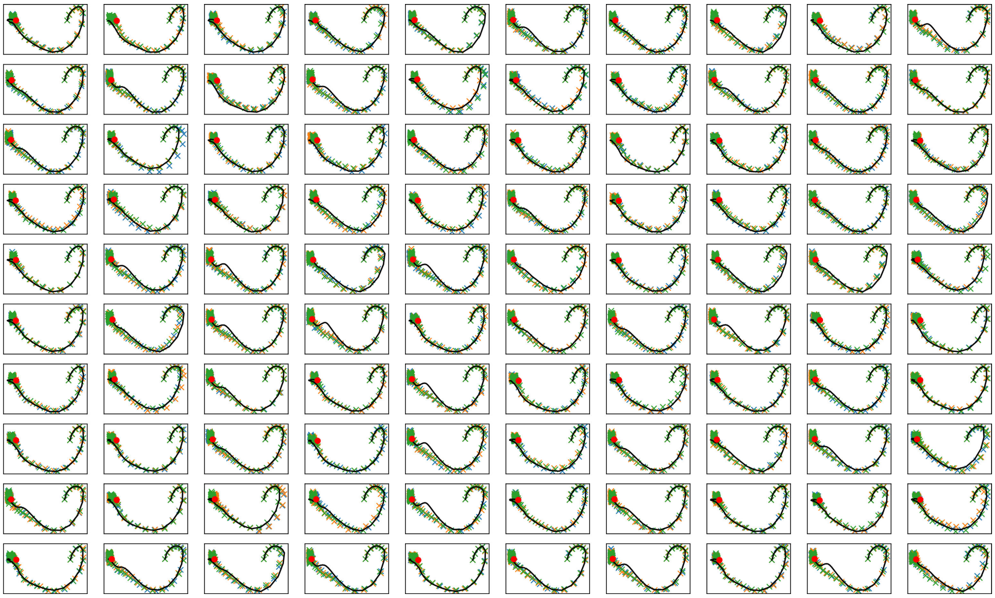

FIGURE 7. Behaviors in real and simulation BAMDPs when planned using joint optimization (inverted pendulum policy optimization).

$$
\begin{aligned}
&\leq \frac{1}{M}\sum_{m}\mathbb{E}_{z\sim q_m}\left[C\sqrt{\xi(\theta,\phi,z;\pi)-h^m_{\min}}\right]\\
&\leq C\sqrt{\frac{1}{M}\sum_{m}\left\{\mathbb{E}_{z\sim q_m}\left[\xi(\theta,\phi,z;\pi)\right]-h^m_{\min}\right\}}\\
&\leq C\sqrt{\frac{1}{M}\sum_{m}\mathbb{E}_{z\sim q_m}\left[\xi(\theta,\phi,z;\pi)\right]-h_{\min}}\\
&\leq C\sqrt{\frac{1}{M}\sum_{m}\mathbb{E}_{z\sim q_m}\left[\xi(\theta,\phi,z;\pi)+\nu\ln\frac{q_m(z)}{p(z)}\right]-h_{\min}}\\
&= C\sqrt{L(\theta,\phi;\pi)-h_{\min}},
\end{aligned}
$$

where $q_m(z) = q_\phi(z|\mathcal{D}^{ofl}_m)$ for notational shorthand. The first inequality uses $|\mathbb{E}[\cdot]| \leq \mathbb{E}[|\cdot|]$, the second uses (15), the third uses Jensen's inequality, the forth uses $\frac{1}{M}\sum_m h^m_{\min} \geq \min_m h^m_{\min} = h_{\min}$, and the last uses the non-negativity of KL divergence.

APPENDIX B
NUMERICAL EXPERIMENT SETTINGS

The inverted pendulum task and the cartpole swing-up tasks are modifications of OpenAI Gym [41]. The modified parts are as follows. For the inverted pendulum task, the time discretization width is 0.1, the mass is 0.5, the viscosity coefficient is uniformly sampled from $[0, 0.3]$ as task variation, the initial angle and angular velocity uniformly are sampled from $[-0.75\pi, 0.75\pi]$ and $[-5, 5]$, and the cost function is $1 - \exp(-0.5 \times \text{angle}^2)$. For the cartpole swing-up task, the goal is changed from balancing to swing-up, the time discretization width is 0.05, the pole mass and length are uniformly sampled from $[0.05, 0.3]$ and $[0.4, 0.5]$ as task variation, the initial angle is uniformly sampled from $[-\pi, \pi]$, the initial values of the other state variables are uniformly sampled from $[-0.5, 0.5]$, and the cost function is $1 - \exp(-0.5 \times \text{angle}^2)$.

The details of model training are as follows. For the encoder, f_ϕ and $[\mu_\phi, \log\sigma_\phi]$ are four-layer neural networks with ReLU activation with 32 hidden units. The decoder, $\hat{\mathcal{P}}_{\theta,z}$, is two-layer neural networks with ReLU activation with 48 hidden units in the inverted pendulum task and with 64 hidden units in the cartpole swing-up task. The encoder and the decoder are trained using standard variational inference or importance-weighted variational inference for BAMDP. The importance-weight model, $\hat{w}^\pi_{m,\theta,z}$, is four-layer neural networks with tanh activation with 32 hidden units and learned using a logistic regression loss and $\alpha = 0.2$. The penalty model, $\hat{u}_{m,\theta,z}$, is four-layer neural networks with tanh activation with 16 hidden units and learned using a regression loss.

The discount factor is $\gamma = 0.99$. The constant scaling the KL divergence regularization term is $\nu = 1$. The penalty coefficient for importance-weighted variational inference for BAMDP is $c = 0.1$. The penalty coefficient for standard variational inference is $\lambda = \tilde{\kappa}$, to compare with importance-weighted variational inference for BAMDP under the same condition.

The code is avalable at https://github.com/numahha/iwvi.git.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, vol. 135. Cambridge, MA, USA: MIT Press, 1998.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[4] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, Sep. 2013.

[5] C. Yu, J. Liu, S. Nemati, and G. Yin, "Reinforcement learning in healthcare: A survey," *ACM Comput. Surveys*, vol. 55, no. 1, pp. 1–36, Jan. 2023.

[6] C. G. Atkeson and J. C. Santamaria, "A comparison of direct and model-based reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1997, pp. 3557–3564.

[7] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 408–423, Feb. 2015.

[8] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 4759–4770.

[9] M. O. Duff, "Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes," Ph.D. dissertation, Dept. Comput. Sci., Univ. Massachusetts, Amherst, MA, USA, 2002.

[10] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar, "Bayesian reinforcement learning: A survey," *Found. Trends Mach. Learn.*, vol. 8, nos. 5–6, pp. 359–483, 2015.

[11] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," 2020, *arXiv:2005.01643*.

[12] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[13] K. Senda, T. Hishinuma, and Y. Tani, "Approximate Bayesian reinforcement learning based on estimation of plant," *Auto. Robots*, vol. 44, no. 5, pp. 845–857, May 2020.

[14] S. Saemundsson, K. Hofmann, and M. P. Deisenroth, "Meta reinforcement learning with latent variable Gaussian processes," in *Proc. Conf. Uncertainty Artif. Intell.*, 2018, pp. 642–652.

[15] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen, "Efficient off-policy meta-reinforcement learning via probabilistic context variables," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5331–5340.

[16] L. Zintgraf, S. Schulze, C. Lu, L. Feng, M. Igl, K. Shiarlis, Y. Gal, K. Hofmann, and S. Whiteson, "VariBAD: Variational Bayes-adaptive deep RL via meta-learning," *J. Mach. Learn. Res.*, vol. 22, no. 289, pp. 1–39, 2021.

[17] R. Dorfman, I. Shenfeld, and A. Tamar, "Offline meta learning of exploration," in *Proc. Neural Inf. Process. Syst.*, 2021, pp. 1–22.

[18] T. Imagawa, T. Hiraoka, and Y. Tsuruoka, "Off-policy meta-reinforcement learning with belief-based task inference," *IEEE Access*, vol. 10, pp. 49494–49507, 2022.

[19] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 1–12.

[20] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma, "MOPO: Model-based offline policy optimization," in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 14129–14142.

[21] T. Hishinuma and K. Senda, "Weighted model estimation for offline model-based reinforcement learning," in *Proc. Neural Inf. Process. Syst.*, 2021, pp. 1–18.

[22] A.-M. Farahmand, A. Barreto, and D. Nikovski, "Value-aware loss function for model-based reinforcement learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1486–1494.

[23] R. Abachi, M. Ghavamzadeh, and A.-M. Farahmand, "Policy-aware model learning for policy gradient methods," 2020, *arXiv:2003.00030*.

[24] P. D'Oro, A. M. Metelli, A. Tirinzoni, M. Papini, and M. Restelli, "Gradient-aware model-based policy search," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 3801–3808.

[25] S. Yang, S. Zhang, Y. Feng, and M. Zhou, "A unified framework for alternating offline model training and policy learning," in *Proc. Neural Inf. Process. Syst.*, 2022, pp. 1–17.

[26] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density Ratio Estimation in Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[27] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 1994.

[28] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.

[29] M. D. Hoffman and M. J. Johnson, "ELBO surgery: Yet another way to carve up the variational evidence lower bound," in *Proc. Workshop Adv. Approx. Bayesian Inference (NIPS)*, vol. 1, no. 2, 2016, pp. 1–4.

[30] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow, "Adversarial autoencoders," in *Proc. Int. Conf. Learn. Represent., Workshop*, 2016, pp. 1–16.

[31] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *J. Comput. Graph. Statist.*, vol. 9, no. 1, pp. 1–20, Mar. 2000.

[32] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–22.

[33] E. Iakovleva, J. Verbeek, and K. Alahari, "Meta-learning with shared amortized variational inference," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4572–4582.

[34] T. Morimura, E. Uchibe, J. Yoshimoto, J. Peters, and K. Doya, "Derivatives of logarithmic stationary distributions for policy gradient reinforcement learning," *Neural Comput.*, vol. 22, no. 2, pp. 342–376, Feb. 2010.

[35] A. Kumagai, T. Iwata, and Y. Fujiwara, "Meta-learning for relative density-ratio estimation," in *Proc. Neural Inf. Process. Syst.*, 2021, pp. 1–13.

[36] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Stat. Planning Inference*, vol. 90, no. 2, pp. 227–244, Oct. 2000.

[37] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[38] J. Tomczak and M. Welling, "VAE with a VampPrior," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2018, pp. 1214–1223.

[39] B. Rhodes, K. Xu, and M. U. Gutmann, "Telescoping density-ratio estimation," in *Proc. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 4905–4916.

[40] K. Choi, C. Meng, Y. Song, and S. Ermon, "Density ratio estimation via infinitesimal classification," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2022, pp. 2552–2573.

[41] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI gym," 2016, *arXiv:1606.01540*.

TORU HISHINUMA is currently a Graduate Student with the Department of Aeronautics and Astronautics, Graduate School of Engineering, Kyoto University. His research interests include the intelligence of robots and reinforcement learning.

KEI SENDA (Member, IEEE) received the M.S. degree in aeronautical engineering and the Ph.D. degree in engineering from Osaka Prefecture University, in 1988 and 1993, respectively. From 1988 to 2008, he was with Osaka Prefecture University and Kanazawa University. Since 2008, has been a Professor with the Department of Aeronautics and Astronautics, Graduate School of Engineering, Kyoto University. His research activities have included more than 100 articles on the dynamics and control of aerospace systems, the intelligence and autonomy of mechanical systems, and the motion intelligence of animals. He received the Best Presented Paper Award of the AIAA Guidance, Navigation, and Control Conference, in 1992; the Finalist of the Best Paper of the IEEE International Conference on Systems, Man, and Cybernetics, in 2011; the Best Paper Award of the IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, in 2012; and the Best Paper Award of the SICE Systems and Information Division, in 2017.