On the Design of Variational RL Algorithms

Joseph Marino *[1] **Alexandre Piché** [2] **Yisong Yue** [1]

[1] California Institute of Technology (Caltech), Pasadena, CA, USA
[2] Mila, Université de Montréal, QC, CAN.

Abstract

We survey multiple recently proposed algorithms that formulate control, reinforcement learning, and planning as probabilistic inference. By viewing these algorithms through a common framework, we highlight the design choices implicit in each setting. This framework also allows us to identify several settings that have not yet been fully explored, and we discuss general directions for improving these algorithms. We present a preliminary set of experiments demonstrating one such improvement, using iterative amortized inference models (Marino et al., 2018b) to improve policy optimization, outperforming soft actor-critic on a continuous control task.

1 Introduction

Reinforcement learning (RL) has witnessed significant advances in recent years, largely as a result of improved function approximation in the form of deep networks (Mnih et al., 2013; Lillicrap et al., 2015). However, despite their success, these approaches often rely on heuristic techniques, such as ϵ-greedy exploration or entropy bonuses. In search of improved techniques, recent works have revisited the idea of framing RL as probabilistic inference (Dayan & Hinton, 1997; Attias, 2003; Toussaint & Storkey, 2006). These approaches result in maximum entropy (Ziebart, 2010; Levine, 2018) or relative entropy (Peters et al., 2010) constraints on policies, acting to regularize and stabilize training. Modern instantiations of these ideas, such as soft actor-critic (Haarnoja et al., 2018b) and relative entropy regularized policy iteration (Abdolmaleki et al., 2018a), have demonstrated empirical success. While these algorithms and others fall within the same framework, the algorithmic design choices underlying these approaches are rarely discussed in relation to each other.

The purpose of this paper is to discuss these design choices from the perspective of variational inference (Jordan et al., 1999). Specifically, we outline the overall framework of variational expectation maximization (Dempster et al., 1977; Neal & Hinton, 1998), stating its implications in the context of RL. We also review a major recent advancement in variational inference, *amortization* (Kingma & Welling, 2013; Rezende et al., 2014), drawing connections to RL. We hope this unified perspective serves to guide future investigations, including comparisons of existing design choices, as well as exploring sets of design choices that have been overlooked. As part of this process, we modify the soft actor-critic (SAC) algorithm (Haarnoja et al., 2018b) to use iterative amortized inference (Marino et al., 2018b), demonstrating improved performance in the continuous control domain.

[*] Correspondence to: Joseph Marino<jmarino@caltech.edu>

Deep RL workshop @ 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

2 Background

2.1 Reinforcement Learning

We consider the standard Markov decision process (MDP) formulation, where $\mathbf{s}_t \in \mathcal{S}$ and $\mathbf{a}_t \in \mathcal{A}$ are the state and action at time t, resulting in the reward $r_t = r(\mathbf{s}_t, \mathbf{a}_t)$. Environment state transitions are given by $\mathbf{s}_{t+1} \sim p_{\text{env}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$, and the agent is defined by a parametric policy distribution, $p_\theta(\mathbf{a}_t|\mathbf{s}_t)$, with parameters θ. The discounted sum of rewards is denoted as $\mathcal{R}(\tau) = \sum_t \gamma^t r_t$, where $\gamma \in (0, 1]$ is the discount factor. Defining a trajectory as $\tau = (s_1, a_1, \dots)$, the distribution over trajectories is

$$p(\tau) = \rho(\mathbf{s}_1) \prod_{t=1}^{T} p_{\text{env}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) p_\theta(\mathbf{a}_t|\mathbf{s}_t), \tag{1}$$

where the initial state is drawn from the distribution $\rho(\mathbf{s}_1)$. The standard RL objective consists of maximizing the expected discounted return, $\mathbb{E}_{p(\tau)}\left[\mathcal{R}(\tau)\right]$. In the following sections, we consider the undiscounted setting ($\gamma = 1$), later revisiting this concept.

2.2 Variational Inference

Variational inference (Jordan et al., 1998) is an approximate inference technique that recasts probabilistic inference as optimization. Consider a graphical model defined by the joint distribution $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})$, with observed variable, $\mathbf{x}$, latent variable, $\mathbf{z}$, and parameters, θ. Often, we are interested in 1) learning the model parameters to maximize the marginal log-likelihood of observations, i.e. $\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z}$, and 2) inferring the posterior distribution for a given observation, $p_\theta(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{x}, \mathbf{z})/p_\theta(\mathbf{x})$. However, evaluating $p_\theta(\mathbf{x})$ is typically intractable. Instead, variational inference introduces an approximate posterior, $q(\mathbf{z}|\mathbf{x})$, making use of the following:

$$\log p_\theta(\mathbf{x}) = \mathcal{L}(\mathbf{x}; q, \theta) + D_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x})), \tag{2}$$

where the evidence lower bound (ELBO) is defined as

$$\mathcal{L}(\mathbf{x}; q, \theta) \equiv \mathbb{E}_q\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - D_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z})). \tag{3}$$

From Eq. 2, note that $\mathcal{L}(\mathbf{x}; q) \leq \log p_\theta(\mathbf{x})$ because KL-divergence is non-negative, and this bound becomes tight when $q(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x})$. Thus, for a given prior, the optimal approximate posterior is given by the posterior distribution:

$$q^{\mathcal{B}}(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{x})} = \frac{p_\theta(\mathbf{z}) \exp(\log p_\theta(\mathbf{x}|\mathbf{z}))}{\int p_\theta(\mathbf{z}) \exp(\log p_\theta(\mathbf{x}|\mathbf{z})) d\mathbf{z}}, \tag{4}$$

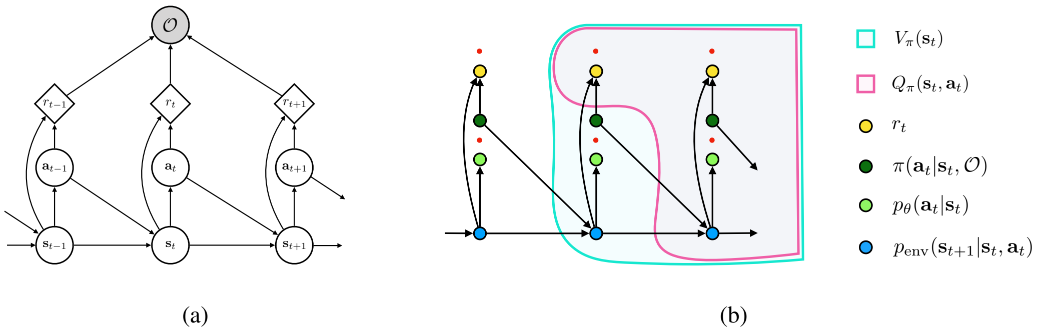
a Boltzmann distribution, which is *non-parametric*. However, to avoid the intractable integration, we can employ a *parametric* distribution family, such as a Gaussian, parameterizing $q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x})$ with distribution parameters $\boldsymbol{\lambda}$, e.g. mean and variance. In either case, we have an alternating optimization procedure, known as *variational expectation maximization* (EM) (Dempster et al., 1977; Neal & Hinton, 1998). In the E-step (inference), we maximize $\mathcal{L}(\mathbf{x}; q, \theta)$ w.r.t. q, either estimating $q^{\mathcal{B}}$ (Eq. 4) or optimizing w.r.t. $\boldsymbol{\lambda}$. In the M-step (learning), we maximize $\mathcal{L}(\mathbf{x}; q, \theta)$ w.r.t. θ. Both procedures can be performed using stochastic gradients (Ranganath et al., 2014; Hoffman et al., 2013).

Inference optimization is often computationally intensive. In the parametric case, we can improve efficiency by *amortizing* inference optimization, introducing a separate inference model (Dayan et al., 1995). The inference model parameters, ϕ, are optimized using stochastic gradient estimators (Mnih & Gregor, 2014; Kingma & Welling, 2013; Rezende et al., 2014). Inference models typically take the form $\boldsymbol{\lambda} \leftarrow f_\phi(\mathbf{x})$. However, direct models of this form tend to provide sub-optimal estimates of $\boldsymbol{\lambda}$ (Cremer et al., 2018). To overcome this gap, Marino et al. (2018b) proposed iterative inference models, which iteratively encode gradients, $\boldsymbol{\lambda} \leftarrow f_\phi(\boldsymbol{\lambda}, \nabla_{\boldsymbol{\lambda}} \mathcal{L})$. Empirically, iterative models tend to outperform direct models (Marino et al., 2018b,a).

3 Variational EM for RL

3.1 Set-Up

A number of works have formulated RL, planning, and control problems in terms of probabilistic inference (Dayan & Hinton, 1997; Attias, 2003; Toussaint & Storkey, 2006; Todorov, 2008; Botvinick


$\mathcal{O}$
r_{t-1}
r_t
r_{t+1}
$\mathbf{a}_{t-1}$
$\mathbf{a}_t$
$\mathbf{a}_{t+1}$
$\mathbf{s}_{t-1}$
$\mathbf{s}_t$
$\mathbf{s}_{t+1}$
$V_\pi(\mathbf{s}_t)$
$Q_\pi(\mathbf{s}_t, \mathbf{a}_t)$
r_t
$\pi(\mathbf{a}_t|\mathbf{s}_t, \mathcal{O})$
$p_\theta(\mathbf{a}_t|\mathbf{s}_t)$
$p_{\text{env}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$

(a)

(b)

Figure 1: **Graphical Representations**. **(a)** Graphical model for formulating reinforcement learning as probabilistic inference. Circles denote random variables, and diamonds denote deterministic variables. Optimality is an observed variable, while all other variables are latent. **(b)** Corresponding computation graph when using variational inference. Red dots denote terms in the variational objective, $\mathcal{J}$.

& Toussaint, 2012; Levine, 2018). These approaches consider the agent-environment interaction as a graphical model, then convert reward maximization into a maximum likelihood estimation problem, learning and inferring a policy that results in maximal reward. This conversion is accomplished by introducing one or more binary observed variables (Cooper, 1988), denoted as $\mathcal{O}$, with

$$p(\mathcal{O} = 1|\tau) \propto \exp\big(\mathcal{R}(\tau)/\alpha\big),$$

where α is a temperature hyper-parameter. These new variables are often referred to as "optimality" variables (Levine, 2018). In this setting, we would like to infer latent variables, τ, and learn parameters, θ, that yield the maximum likelihood of optimality, i.e. $p(\mathcal{O} = 1)$. Evaluating this likelihood requires marginalizing the joint distribution over trajectories and optimality, $p(\tau, \mathcal{O})$:

$$p(\mathcal{O} = 1) = \int p(\tau, \mathcal{O} = 1) d\tau.$$

This corresponds to averaging over all trajectories, which is intractable in high-dimensional spaces. Instead, we can use variational inference to lower bound this objective, introducing a structured approximate posterior, which we denote as:

$$\pi(\tau|\mathcal{O}) = \prod_{t=1}^{T} p_{\text{env}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\pi(\mathbf{a}_t|\mathbf{s}_t, \mathcal{O}). \tag{5}$$

This provides the following lower bound on the objective:

$$\begin{aligned}\log p(\mathcal{O} = 1) &= \log \int p(\mathcal{O} = 1|\tau)p(\tau)d\tau \\ &\geq \int \pi(\tau|\mathcal{O})\left[\log p(\mathcal{O} = 1|\tau) + \log \frac{p(\tau)}{\pi(\tau|\mathcal{O})}\right] d\tau \\ &= \mathbb{E}_\pi[\mathcal{R}(\tau)/\alpha] - D_{\text{KL}}(\pi(\tau|\mathcal{O}) \| p(\tau)).\end{aligned}$$

Because we are optimizing this objective, we can multiply by α to obtain an equivalent result:

$$\mathcal{J}(\pi, \theta) = \mathbb{E}_\pi[\mathcal{R}(\tau)] - \alpha D_{\text{KL}}(\pi(\tau|\mathcal{O}) \| p(\tau)) \tag{6}$$

Thus, the variational objective consists of the expected return, i.e. the standard RL objective, and a KL-divergence between $\pi(\tau|\mathcal{O})$ and $p(\tau)$. In terms of states and actions, this objective is written as:

$$\mathcal{J}(\pi, \theta) = \mathbb{E}_{\substack{\mathbf{s}_t, r_t \sim p_{\text{env}} \\ \mathbf{a}_t \sim \pi}}\left[\sum_{t=1}^{T} r_t - \alpha \log \frac{\pi(\mathbf{a}_t|\mathbf{s}_t, \mathcal{O})}{p_\theta(\mathbf{a}_t|\mathbf{s}_t)}\right]. \tag{7}$$

Algorithm 1 Variational EM for RL

```
1: Initialize $\theta$
2: while $\theta$ not converged do
3:     for $i$ in $\{1, \dots, M\}$ do
4:         # E-step
5:         for $t$ in $\{1, \dots, T\}$ do
6:             infer $\pi(\mathbf{a}_t|\mathbf{s}_t, \mathcal{O})$
7:             $\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t, \mathcal{O})$.
8:             $\mathbf{s}_{t+1}, r_t \sim p_{\text{env}}(\cdot|\mathbf{s}_t, \mathbf{a}_t)$
9:         end for
10:        $\mathcal{J}_i := \sum_t r_t - \alpha \log \frac{\pi(\mathbf{a}_t|\mathbf{s}_t, \mathcal{O})}{p_\theta(\mathbf{a}_t|\mathbf{s}_t)}$
11:    end for
12:    # M-step
13:    $\theta \leftarrow \theta + \eta \nabla_\theta \frac{1}{M} \sum_i \mathcal{J}_i$
14: end while
```

One can also define analogous "soft" value functions for this set-up, corresponding to the expected future objective:

$$V_\pi(\mathbf{s}_t) = \mathbb{E}_{\substack{\mathbf{s}_{t'}, r_{t'} \sim p_{\text{env}} \\ \mathbf{a}_{t'} \sim \pi}} \left[\sum_{t'=t+1}^{T} r_{t'} - \alpha \log \frac{\pi(\mathbf{a}_{t'}|\mathbf{s}_{t'}, \mathcal{O})}{p_\theta(\mathbf{a}_{t'}|\mathbf{s}_{t'})} \right], \tag{8}$$

$$Q_\pi(\mathbf{s}_t, \mathbf{a}_t) = r_t + \mathbb{E}_{\substack{\mathbf{s}_{t'}, r_{t'} \sim p_{\text{env}} \\ \mathbf{a}_{t'} \sim \pi}} \left[\sum_{t'=t+1}^{T} r_{t'} - \alpha \log \frac{\pi(\mathbf{a}_{t'}|\mathbf{s}_{t'}, \mathcal{O})}{p_\theta(\mathbf{a}_{t'}|\mathbf{s}_{t'})} \right]. \tag{9}$$

Thus,

$$V_\pi(\mathbf{s}_t) = \mathbb{E}_\pi \left[Q_\pi(\mathbf{s}_t, \mathbf{a}_t) \right] - \alpha D_{\text{KL}}(\pi(\mathbf{a}_t|\mathbf{s}_t, \mathcal{O}) || p_\theta(\mathbf{a}_t|\mathbf{s}_t)). \tag{10}$$

As discussed in Section 2.2, the optimal approximate posterior is a Boltzmann distribution. In the RL setting, we denote this as $\pi^{\mathcal{B}}(\tau|\mathcal{O})$, with the distribution at each time step given by:

$$\pi^{\mathcal{B}}(\mathbf{a}_t|\mathbf{s}_t, \mathcal{O}) = \frac{p_\theta(\mathbf{a}_t|\mathbf{s}_t) \exp(Q_\pi(\mathbf{s}_t, \mathbf{a}_t)/\alpha)}{\int p_\theta(\mathbf{a}_t|\mathbf{s}_t) \exp(Q_\pi(\mathbf{s}_t, \mathbf{a}_t)/\alpha) d\mathbf{a}_t}. \tag{11}$$

3.2 Variational EM

Applying variational EM to RL, we have an alternating optimization procedure on $\mathcal{J}(\pi, \theta)$, consisting of the following:

- **E-step**: infer/approximate the posterior, $\pi(\tau|\mathcal{O})$, by
  - *Non-Parametric*: estimating $\pi^{\mathcal{B}}(\tau|\mathcal{O})$,
  - *Parametric*: maximizing $\mathcal{J}$ w.r.t. $\pi(\tau|\mathcal{O})$.
- **M-step**: learn the the prior, $p_\theta(\tau)$, by maximizing $\mathcal{J}$ w.r.t. θ.

In practice, these procedures are computationally intractable, requiring various approximations. In particular, $\mathcal{J}$ consists of nested expectations over p_{env} and π, each of which can be estimated with samples (Hoffman et al., 2013; Ranganath et al., 2014). Algorithm 1 outlines variational EM in this sample-based setting, using a batch of M sampled trajectories to estimate $\nabla_\theta \mathcal{J}$. For each sampled trajectory, we must infer $\pi(\tau|\mathcal{O})$ (Line 7), which involves estimating the future objective, Q_π. Thus, we see that there are three main design choices that go into any variational RL algorithm: **1)** prior, $p_\theta(\mathbf{a}|\mathbf{s})$, **2)** approximate posterior, $\pi(\mathbf{a}|\mathbf{s}, \mathcal{O})$, and **3)** the conditional likelihood, i.e., value estimate, Q_π. In addition, one can select from various hyper-parameters and optimization procedures. We now discuss the design choices of existing algorithms from this perspective.

3.2.1 Prior

In the simplest case, the action prior is a uniform distribution over the action space. With a constant prior probability over all actions, we can drop this term, resulting in the popular maximum entropy formulation of RL (Ziebart, 2010):

$$\mathcal{J}(\pi, \theta) = \mathbb{E}_\pi[\mathcal{R}(\tau)] + \alpha\mathcal{H}(\pi(\tau|\mathcal{O})) \tag{12}$$

Empirically, this tends to yield robust policies (Eysenbach & Levine, 2019) that explore the state-space more efficiently (Fox et al., 2016). Indeed, similar techniques are often applied within standard RL for these reasons (Mnih et al., 2016). However, the optimal max-entropy approximate posterior, $\pi^{\mathcal{B}}$, is stochastic, whereas the optimal policy for a fully-observable MDP is deterministic. For this reason, previous works tend to report performance using the maximum (MAP) of the approximate posterior distribution (Haarnoja et al., 2018b), rather than sampling actions.

Learning the prior in the M-step can potentially improve upon the simplistic assumptions of max-entropy RL. The prior provides a mechanism for biasing actions toward previous rewarding trajectories, enabling further refinement and generalization (Galashov et al., 2019; Teh et al., 2017). This, again, assists in stabilizing learning, motivating methods like TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017b). Abdolmaleki et al. (2018b) show that optimizing the prior results in monotonic improvement in performance under certain assumptions. However, like other applications of the variational EM algorithm, this procedure may converge to local optima. Techniques from the optimization literature have been applied to help prevent premature convergence (Abdolmaleki et al., 2018a; Hämäläinen et al., 2018), however, this remains an active research area.

Beyond the stabilizing benefits, a learned prior can act as a model-free *rollout policy* (Silver et al., 2017), assisting in model-based policy optimization (Wang & Ba, 2019). Learning the prior also distills model-based planning into a model-free policy (Kurutach et al., 2018; Buesing et al., 2018; Marino & Yue, 2019), enabling efficient action selection after training.

3.2.2 Approximate Posterior

Approximate posteriors can either be parametric or non-parametric, corresponding respectively to policy-based and value-based methods in standard RL. Multiple works have investigated non-parametric approximate posteriors in the settings of discrete (Fox et al., 2016) and continuous (Haarnoja et al., 2017; Abdolmaleki et al., 2018b; Piché et al., 2019) actions. Non-parametric approximations are able to capture complex, free-form distributions. However, sampling such distributions is difficult in continuous action spaces, accurate estimation requires drawing multiple action samples, and their flexibility may make them prone to overfitting to value estimation errors.

Other works have opted for parametric distributions (Schulman et al., 2017a; Haarnoja et al., 2018b). While the fixed form of continuous parametric distributions can be restrictive, e.g. Gaussian, these approaches often perform on-par with their non-parametric counterparts (Abdolmaleki et al., 2018b,a). Parametric distributions can be made more flexible with additional latent variables (Hausman et al., 2018; Gupta et al., 2018; Haarnoja et al., 2018a; Tirumala et al., 2019) or normalizing flows (Tang & Agrawal, 2018). For efficiency, parametric approximate posteriors are typically implemented with direct amortization (Dayan et al., 1995; Kingma & Welling, 2013; Rezende et al., 2014), e.g., (Schulman et al., 2017a; Haarnoja et al., 2018b). However, as we show in Section 4, distribution parameters can be optimized separately for each state.

3.2.3 Conditional Likelihood (Value Estimate)

The variational RL objective (Eq. 6) contains expectations over the approximate posterior and environment dynamics, which we must evaluate to perform inference. However, this objective presents a number of issues: 1) it has high variance, as it factors on multiple steps of interaction, 2) it is generally non-differentiable, and 3) it is expensive, as it requires sampling many full trajectories from the environment. The RL community has developed three general approaches to building surrogate objectives: i) learning a transition model of the environment, ii) learning a parameterized expectation function, and iii) a combination of a learned model and expectation function. These surrogates are differentiable, can be decomposed per time step, and have a lower variance than the original objective. However, bias in these these surrogates can reduce the accuracy of policy optimization (inference).

Another difference is the use of the discount factor, γ. This factor does not naturally appear in the variational RL formulation, but as noted by Schulman et al. (2017a) and others, this factor can help reduce variance in estimates of $Q_\pi(\mathbf{s}_t, \mathbf{a}_t)$. In practice, most implementations of variational RL algorithms use $\gamma < 1$.

			Approximate Posterior	
			Parametric	**Non-parametric**
RL	**Uniform Prior**	**Model-Free**	SAC, ERPG	SQL
		Model-Based	-	SMCP
	Learned Prior	**Model-Free**	MPO, IADP, HDP	MPO, RERPI, V-MPO, Distral
		Model-Based	-	-

Table 1: **Design Choices.** Soft Q-Learning (SQL) (Haarnoja et al., 2017), Soft Actor-Critic (SAC) (Haarnoja et al., 2018b), Entropy Regularized Policy Gradient (ERPG) (Schulman et al., 2017a), Sequential Monte Carlo Planning (SMCP) (Piché et al., 2019), Maximum a Posteriori Policy Optimization (MPO) (Abdolmaleki et al., 2018b), Relative Entropy Regularized Policy Iteration (RERPI) (Abdolmaleki et al., 2018a), V-Maximum a Posterior Policy Optimization (V-MPO) (Song et al., 2019), Information Asymmetry Default Policy (IADP) (Galashov et al., 2019), Hierarchical Default Policy (HDP) (Tirumala et al., 2019), Distill & Transfer Learning (Distral) (Teh et al., 2017).

3.2.4 Other Design Choices

Beyond the prior, approximate posterior, and value estimate, there are a variety of other design choices that go into any particular algorithm. For instance, we must choose the optimization algorithms used for inference and learning, typically amortized inference and stochastic gradient descent learning. Other design choices include the use of off-policy or on-policy interactions to estimate the objective as well as the sampling procedure for (continuous) non-parametric approximate posteriors. For example, soft Q-learning (Haarnoja et al., 2017) uses Stein variational gradient descent (Liu & Wang, 2016), whereas maximium a posteriori policy optimization (MPO) (Abdolmaleki et al., 2018b) samples from the action prior.

We summarize the main design choices of many recent algorithms in Table 1[2]. For simplicity, we chose to only include algorithms formulated within the variational framework, however, many standard RL algorithms, such as A3C (Mnih et al., 2016), DDPG (Lillicrap et al., 2015), PPO (Schulman et al., 2017b), and others could also be interpreted through this perspective. Although Table 1 is not exhaustive, there are noticeable gaps, particularly for model-based value estimation. Future work in model-based RL may benefit from the variational perspective, particularly with regard to the interplay between model-based value estimation (planning) and the model-free action prior.

4 Experiments

We present initial experiments demonstrating improvements on top of a state-of-the-art algorithm, soft actor-critic (SAC) (Haarnoja et al., 2018b,c), which uses a parametric approximate posterior, uniform prior, and a learned conditional likelihood for inference. Specifically, we replace the direct amortized inference procedure in SAC with an iterative amortized inference procedure (Marino et al., 2018b). We demonstrate improved empirical performance on the `HalfCheetah-v2` MuJoCo environment from OpenAI gym (Brockman et al., 2016; Todorov et al., 2012).

4.1 Set-Up

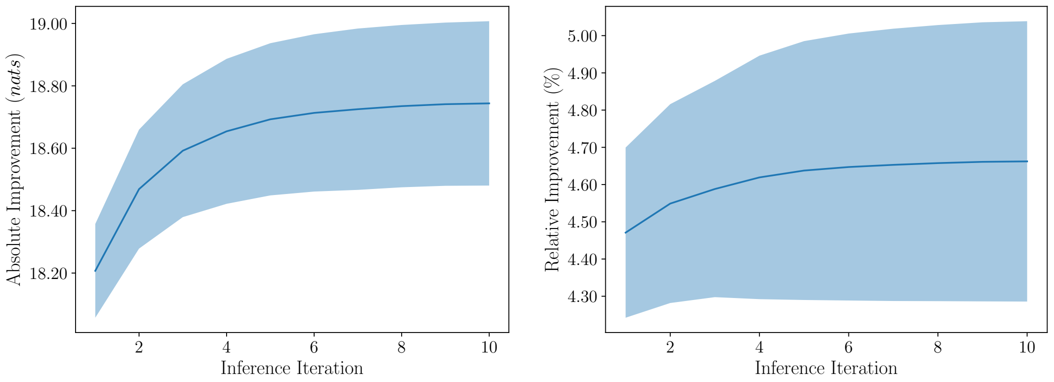
SAC uses a direct amortized approximate posterior of the form

$$\mathbf{u}_t \sim \mathcal{N}(\mathbf{u}_t; \boldsymbol{\mu}_{\mathbf{u}}, \text{diag}(\boldsymbol{\sigma}^2_{\mathbf{u}})) \tag{13}$$

$$\mathbf{a}_t = \tanh(\mathbf{u_t}), \tag{14}$$

where $\boldsymbol{\mu}_{\mathbf{u}} = \boldsymbol{\mu}_\phi(\mathbf{s}_t)$ and $\boldsymbol{\sigma}_{\mathbf{u}} = \boldsymbol{\sigma}_\phi(\mathbf{s}_t)$ are parameterized by a neural network, and the non-linearity $\tanh(\cdot)$ ensures that $\mathbf{a}_t \in [-1, +1]$. The network parameters, ϕ, are trained with the reparameterization gradient estimator (Kingma & Welling, 2013; Rezende et al., 2014), using another neural network, $Q_\psi(\mathbf{s}_t, \mathbf{a}_t)$, to estimate the Q-value. Thus, $\boldsymbol{\mu}_\phi(\mathbf{s}_t)$ and $\boldsymbol{\sigma}_\phi(\mathbf{s}_t)$ are trained to output the Gaussian the maximizes $Q_\psi(\mathbf{s}_t, \mathbf{a}_t)$, subject to the action prior. We replace this direct estimation procedure with an iterative procedure. Following Marino et al. (2018b), we use a gated update

[2] The methods from Schulman et al. (2017a); Galashov et al. (2019); Tirumala et al. (2019) did not provide names for their methods. We created names for these methods for the purposes of the table.



(a)

(b)

Figure 2: **Inference Improvement**. Average improvement over 10 inference iterations in **(a)** absolute and **(b)** relative terms. Solid lines denote the average over 5 episodes in the `HalfCheetah-v2` environment, with shaded regions representing one standard deviation. Improvement diminishes over successive inference iterations, yet remains stable.

function, e.g.

$$\boldsymbol{\mu}_{\mathbf{u}} \leftarrow \boldsymbol{\omega}_\phi \odot \boldsymbol{\mu}_{\mathbf{u}} + (1 - \boldsymbol{\omega}_\phi) \odot f_\phi(\mathbf{s}_t, \nabla_{\boldsymbol{\mu}_{\mathbf{u}}} \tilde{\mathcal{J}}), \tag{15}$$

where $\boldsymbol{\omega}_\phi = \boldsymbol{\omega}_\phi(\mathbf{s}_t, \nabla_{\boldsymbol{\mu}_{\mathbf{u}}} \tilde{\mathcal{J}}) \in [0, 1]$ is a gating function and the objective, $\mathcal{J}$, is approximated as

$$\tilde{\mathcal{J}} = \mathbb{E}_{\pi_\phi} [Q_\psi(\mathbf{s}_t, \mathbf{a}_t)] - \alpha D_{\text{KL}}(\pi_\phi(\mathbf{a}_t|\mathbf{s}_t, \mathcal{O})||p_\theta(\mathbf{a}_t|\mathbf{s}_t)). \tag{16}$$

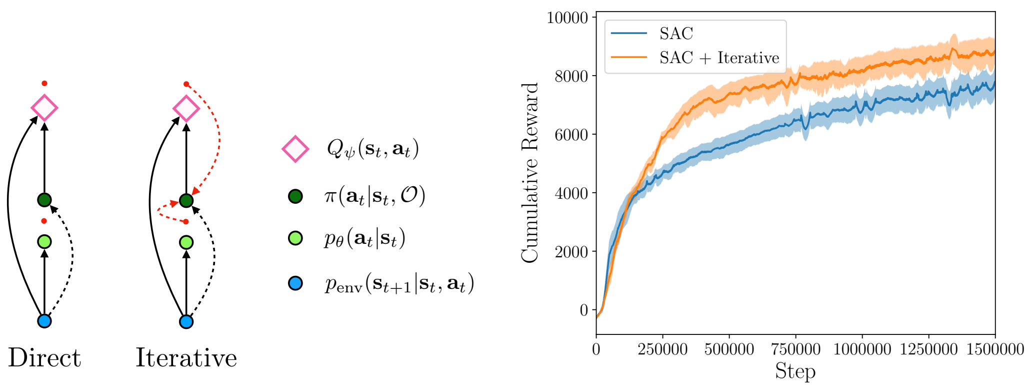
The same gating procedure is used for updating $\boldsymbol{\sigma}_{\mathbf{u}}$. Unlike the direct inference procedure, the procedure in Eq. 15 can be run for multiple iterations. For computational efficiency and purposes of comparison, we use 2 iterations in our experiments. We use the same architectures as Haarnoja et al. (2018b) for all networks, with the addition of $4 \cdot |\mathcal{A}|$ inputs and outputs for the amortized inference network, where $|\mathcal{A}|$ denotes the dimensionality of the action space. Inference diagrams for direct and amortized inference are shown in Figure 3a. SAC uses a uniform action prior, i.e. $p_\theta(\mathbf{a}_t|\mathbf{s}_t) = \text{Uniform}(-1, 1)$. We replaced this distribution with a constant Tanh-Gaussian distribution (Eq. 13 & 14), as we found it gave improved inference performance.

We compare the performance of SAC with these modifications on the `HalfCheetah-v2` MuJoCo environment from OpenAI gym (Brockman et al., 2016; Todorov et al., 2012). Our baseline implementation is based on the version of SAC from Haarnoja et al. (2018c) and Pong (2018). Our modifications are built on the same codebase, which will soon be made publicly available. For our experiments, we made two changes from the original set-up of Haarnoja et al. (2018c): 1) we use a batch size of 128 rather than 256 to improve experiment runtime, and 2) we use 200 action samples rather than 1 to estimate Q during inference, providing low-variance estimates of $\nabla_{\boldsymbol{\mu}_{\mathbf{u}}} \tilde{\mathcal{J}}$ and $\nabla_{\boldsymbol{\sigma}_{\mathbf{u}}} \tilde{\mathcal{J}}$. We use the procedure proposed by Haarnoja et al. (2018c) for automatically tuning α.

4.2 Results

We first demonstrate the inference optimization capabilities of iterative inference models. In Figure 2, we plot the average performance improvement over inference iterations for an agent trained with 10 inference iterations. We plot this both in terms of absolute (Figure 2a) and relative (Figure 2b) improvement from inference initialization. Consistent with Marino et al. (2018b,a), we see that performance increases across inference iterations, with diminishing improvement at each iteration.

Performance results are shown in Figure 3b. Results are averaged over 5 trials for each set-up, with shaded regions denoting one standard deviation. Iterative inference, as a drop-in replacement for direct inference in SAC, improves performance. However, it should be noted that iterative inference is more computationally costly, requiring additional wall-clock time. Likewise, the iterative inference models used here contain additional inputs and outputs beyond the direct inference model (see above), increasing the number of parameters.



(a) (b)

Figure 3: **Inference Diagrams & Results**. **(a)** Diagrams for direct (left) and iterative (right) amortized inference in SAC. Red dots denote terms in the objective. Dotted lines denote inference computations, and red dotted lines denote inference gradients. **(b)** Cumulative reward on `HalfCheetah-v2`, averaged over 5 trials.

5 Discussion

We have discussed a common variational EM framework that encompasses many recently proposed algorithms. While these ideas have been in the literature for quite some time, we have attempted to connect these ideas more closely to the terminology of variational inference, drawing connections to concepts like amortized variational inference. By using this framing, we have shown that recent algorithms can be interpreted as making different design choices within the same family of algorithms. The research community should work toward quantifying the theoretical and empirical trade-offs of these design choices. We have specifically demonstrated the viability of replacing direct amortized inference methods with iterative inference methods.

The overall approach presented in this paper, following previous work, uses a fully-observable environment and simplistic agent. However, the formalism of probabilistic graphical models allows us to easily extend these components to more structured forms. This may involve incorporating partial observability in the environment or additional latent variables within the agent (Hausman et al., 2018; Gupta et al., 2018; Haarnoja et al., 2018a; Tirumala et al., 2019; Marino & Yue, 2019). The variational RL perspective may be helpful in scaling algorithms to these more complex settings.

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Jonas Degrave, Steven Bohez, Yuval Tassa, Dan Belov, Nicolas Heess, and Martin Riedmiller. Relative entropy regularized policy iteration. *arXiv preprint arXiv:1812.02256*, 2018a.

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018b.

Hagai Attias. Planning by probabilistic inference. In *AISTATS*. Citeseer, 2003.

Matthew Botvinick and Marc Toussaint. Planning as inference. *Trends in cognitive sciences*, 16(10): 485–488, 2012.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Lars Buesing, Theophane Weber, Sébastien Racanière, S. M. Ali Eslami, Danilo Jimenez Rezende, David P. Reichert, Fabio Viola, Frederic Besse, Karol Gregor, Demis Hassabis, and Daan Wierstra. Learning and querying fast generative models for reinforcement learning. *CoRR*, abs/1802.03006, 2018. URL `http://arxiv.org/abs/1802.03006`.

Gregory F Cooper. A method for using belief networks as influence diagrams. In *Fourth Workshop on Uncertainty in Artificial Intelligence.*, 1988.

Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pp. 1086–1094, 2018.

Peter Dayan and Geoffrey E Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.

Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

Benjamin Eysenbach and Sergey Levine. If maxent rl is the answer, what is the question? *arXiv preprint arXiv:1910.01913*, 2019.

Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 202–211. AUAI Press, 2016.

Alexandre Galashov, Siddhant M Jayakumar, Leonard Hasenclever, Dhruva Tirumala, Jonathan Schwarz, Guillaume Desjardins, Wojciech M Czarnecki, Yee Whye Teh, Razvan Pascanu, and Nicolas Heess. Information asymmetry in kl-regularized rl. *arXiv preprint arXiv:1905.01240*, 2019.

Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems*, pp. 5307–5316, 2018.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1352–1361. JMLR. org, 2017.

Tuomas Haarnoja, Kristian Hartikainen, Pieter Abbeel, and Sergey Levine. Latent space policies for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pp. 1846–1855, 2018a.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1856–1865, 2018b.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018c.

Perttu Hämäläinen, Amin Babadi, Xiaoxiao Ma, and Jaakko Lehtinen. Ppo-cma: proximal policy optimization with covariance matrix adaptation. *arXiv preprint arXiv:1810.02541*, 2018.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *NATO ASI SERIES D BEHAVIOURAL AND SOCIAL SCIENCES*, 1998.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart Russell, and Pieter Abbeel. Learning plannable representations with causal infogan. *arXiv preprint arXiv:1807.09341*, 2018.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

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.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in neural information processing systems*, pp. 2378–2386, 2016.

Joseph Marino and Yisong Yue. An inference perspective on model-based reinforcement learning. *ICML Workshop on Generative Modeling and Model-Based Reasoning for Robotics and AI*, 2019.

Joseph Marino, Milan Cvitkovic, and Yisong Yue. A general method for amortizing variational filtering. In *Advances in Neural Information Processing Systems*, 2018a.

Joseph Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. *arXiv preprint arXiv:1807.09356*, 2018b.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pp. 1791–1799, 2014.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pp. 355–368. Springer, 1998.

Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

Alexandre Piché, Valentin Thomas, Cyril Ibrahim, Yoshua Bengio, and Chris Pal. Probabilistic planning with sequential monte carlo methods. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=ByetGn0cYX`.

Vitchyr Pong. rlkit. `https://github.com/vitchyr/rlkit/`, 2018.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, 2014.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

H Francis Song, Abbas Abdolmaleki, Jost Tobias Springenberg, Aidan Clark, Hubert Soyer, Jack W Rae, Seb Noury, Arun Ahuja, Siqi Liu, Dhruva Tirumala, et al. V-mpo: On-policy maximum a posteriori policy optimization for discrete and continuous control. *arXiv preprint arXiv:1909.12238*, 2019.

Yunhao Tang and Shipra Agrawal. Boosting trust region policy optimization by normalizing flows policy. *arXiv preprint arXiv:1809.10326*, 2018.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 4496–4506, 2017.

Dhruva Tirumala, Hyeonwoo Noh, Alexandre Galashov, Leonard Hasenclever, Arun Ahuja, Greg Wayne, Razvan Pascanu, Yee Whye Teh, and Nicolas Heess. Exploiting hierarchy for learning and transfer in kl-regularized rl. *arXiv preprint arXiv:1903.07438*, 2019.

Emanuel Todorov. General duality between optimal control and estimation. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pp. 4286–4292. IEEE, 2008.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 5026–5033. IEEE, 2012.

Marc Toussaint and Amos Storkey. Probabilistic inference for solving discrete and continuous state markov decision processes. In *Proceedings of the 23rd international conference on Machine learning*, pp. 945–952. ACM, 2006.

Tingwu Wang and Jimmy Ba. Exploring model-based planning with policy networks. *arXiv preprint arXiv:1906.08649*, 2019.

Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD thesis, CMU, 2010.