Learning Variational Temporal Abstraction Embeddings in Option-Induced MDPs

Anonymous Author(s)
Affiliation
Address
email

Abstract

The option framework in hierarchical reinforcement learning has notably advanced the automatic discovery of temporally-extended actions from long-horizon tasks. However, existing methods often struggle with ineffective exploration and unstable updates when learning action and option policies simultaneously. Addressing these challenges, we introduce the Variational Markovian Option Critic (VMOC), an off-policy algorithm with provable convergence that employs variational inference to stabilize updates. VMOC naturally integrates maximum entropy as intrinsic rewards to promote the exploration of diverse and effective options. Furthermore, we adopt low-cost option embeddings instead of traditional, computationally expensive option triples, enhancing scalability and expressiveness. Extensive experiments in challenging Mujoco environments validate VMOC's superior performance over existing on-policy and off-policy methods, demonstrating its effectiveness in learning coherent and diverse option sets suitable for complex tasks.

1 Introduction

Recent advancements in deep reinforcement learning (DRL) have demonstrated significant successes across a variety of complex domains, such as mastering the human level of atari [36] and Go [44] games. These achievements underscore the potential of combining reinforcement learning (RL) with powerful function approximators like neural networks [5] to tackle intricate tasks that require nuanced control over extended periods. Despite these breakthroughs, Deep RL still faces substantial challenges, such as insufficient exploration in dynamic environments [18, 13, 42], inefficient learning associated with temporally extended actions [6, 9] and long horizon tasks [30, 4], and vast amounts of samples required for training proficient behaviors [16, 40, 15].

One promising area for addressing these challenges is the utilization of hierarchical reinforcement learning (HRL) [11, 2, 12], a diverse set of strategies that decompose complex tasks into simpler, hierarchical structures for more manageable learning. Among these strategies, the option framework [47], developed on the Semi-Markov Decision Process (SMDP), is particularly effective at segmenting non-stationary task stages into temporally-extended actions known as options. Options are typically learned through a maximum likelihood approach that aims to maximize the expected rewards across trajectories. In this framework, options act as temporally abstracted actions executed over variable time steps, controlled by a master policy that decides when each option should execute and terminate. This structuring not only simplifies the management of complex environments but also enables the systematic discovery and execution of temporal abstractions over long-horizon tasks [24, 23].

However, the underlying SMDP framework is frequently undermined by three key challenges: 1) Insufficient exploration and degradation [20, 37, 23]. As options are unevenly updated using conventional maximum likelihood methods [4, 10, 45, 25, 26], the policy is quickly saturated with early rewarding observations. This typically results in focusing on only low-entropy options that lead

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

to local optima rewards, causing a single option to either dominate the entire policy or switch every timestep. Such premature convergence limits option diversity significantly. 2) Sample Inefficiency. The semi-Markovian nature inherently leads to sample inefficiency [47, 29]: each policy update at the master level extends over multiple time steps, thus consuming a considerable volume of experience samples with relatively low informational gain. This inefficiency is further exacerbated by the prevalence of on-policy option learning algorithms [4, 52], which require new samples to be collected simultaneously from both high-level master policies and low-level action policies at each gradient step, and thus sample expensive. 3) Computationally expensive. Options are conventionally defined as triples [4] with intra-option policies and termination functions, often modeled using neural networks which are expensive to optimize. These challenges collectively limit the broader adoption and effectiveness of the option framework in real-world scenarios, particularly in complex continuous environments where scalability and stability are critical [14, 34, 26].

To address these challenges, we introduce the Variational Markovian Option Critic (VMOC), a novel off-policy algorithm that integrates the variational inference framework on option-induced MDPs [35]. We first formulate the optimal option-induced SMDP trajectory as a probabilistic inference problem, presenting a theoretical convergence proof of the variational distribution under the soft policy iteration framework [19]. Similar to prior variational methods [31], policy entropy terms naturally arise as intrinsic rewards during the inference procedure. As a result, VMOC not only seeks high-reward options but also maximizes entropy across the space, promoting extensive exploration and maintaining high diversity. We implements this inference procedure as an off-policy soft actor critic [19] algorithm, which allows reusing samples from replay buffer and enhances sample efficiency. Furthermore, to address the computational inefficiencies associated with conventional option triples, we follow [35] and employ low-cost option embeddings rather than complex neural network models. This not only simplifies the training process but also enhances the expressiveness of the model by allowing the agent to capture a more diverse set of environmental dynamics.

Our contributions can be summarized as follows:

- We propose a variational inference approach within the maximum entropy framework to enhance diverse and robust exploration of options.
- We implement an off-policy algorithm that improves sample efficiency.
- We introduce option embeddings into latent variable policies and enhance expressiveness and computational cost-effectiveness of option representations.
- We conduct extensive experiments in OpenAI Gym Mujoco [49] environments, demonstrating that VMOC significantly outperforms other option-based variants in terms of exploration capabilities, sample efficiency, and computational efficiency.

2 Preliminary

2.1 Control as Structured Variational Inference

Conventionally, the control as inference framework [19, 31, 19, 53] is derived using the maximum entropy objective. In this section, we present an alternative derivation from the perspective of structured variational inference. We demonstrate that this approach provides a more concise and intuitive pathway to the same theoretical results, where the maximum entropy principle naturally emerges through the direct application of variational inference techniques.

Traditional control methods focus on directly maximizing rewards, often resulting in suboptimal trade-offs between exploration and exploitation. By reinterpreting the control problem as a probabilistic inference problem, the control as inference framework incorporates both the reward structure and environmental uncertainty into decision-making, providing a more robust and flexible approach to policy optimization. In this framework, optimality is represented by a binary random variable $\mathcal{E} \in \{0, 1\}$[1]. The probability of optimality given a state-action pair $(\mathbf{s}, \mathbf{a})$ is denoted as $P(\mathcal{E} = 1 \mid \mathbf{s}, \mathbf{a}) = \exp(r(\mathbf{s}, \mathbf{a}))$, which is an exponential function of the conventional reward function $r(\mathbf{s}, \mathbf{a})$ that measures the desirability of an action in a specific state. Focusing on $\mathcal{E} = 1$ captures the occurrence of optimal events. For simplicity, we will use $\mathcal{E}$ instead of $\mathcal{E} = 1$ in the following text

[1] Conventionally, the optimality variable is denoted by $\mathcal{O}$. However, in this context, we use $\mathcal{E}$ to avoid conflict with notation used in the option framework.

to avoid cluttered notations. The joint distribution over trajectories $\tau = (\mathbf{s}_1, \mathbf{a}_1, \dots, \mathbf{s}_T, \mathbf{a}_T)$ given optimality is expressed as:

$$P(\tau|\mathcal{E}_{1:T}) \propto P(\tau, \mathcal{E}_{1:T}) = P(\mathbf{s}_1) \prod_{t=1}^{T-1} P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) P(\mathcal{E}_t|\mathbf{s}_t, \mathbf{a}_t)$$

where $P(\mathbf{s}_1)$ is the initial state distribution, $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ is the dynamics model. As explained in [19, 31], direct optimization of $P(\tau \mid \mathcal{E}_{1:T})$ can result in an optimistic policy that assumes a degree of control over the dynamics. One way to correct this risk-seeking behavior [31] is through structured variational inference. In our case, the goal is to approximate the optimal trajectory $P(\tau)$ with the variational distribution:

$$q(\tau) = P(\mathbf{s}_1) \prod_{t=1}^{T-1} P(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) q(\mathbf{a}_t \mid \mathbf{s}_t)$$

where the initial distribution $P(\mathbf{s}_1)$ and transition distribution $P(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$ is set to be the true environment dynamics from $P(\tau)$. The only variational term is the variational policy $q(\mathbf{a}_t \mid \mathbf{s}_t)$, which is used to approximate the optimal policy $P(\mathbf{a}_t \mid \mathbf{s}_t, \mathcal{E}_{1:T})$. Under this setting, the environment dynamics will be canceled out from the optimization objective between $P(\tau \mid \mathcal{E})$ and $q(\tau)$, thus explicitly disallowing the agent to influence its dynamics and correcting the risk-seeking behavior.

With the variational distribution at hand, the conventional maximum entropy framework can be recovered through a direct application of standard structural variational inference [28]:

$$\begin{aligned}\log P(\mathcal{E}_{1:T}) &= \mathcal{L}(q(\tau), P(\tau, \mathcal{E}_{1:T})) + D_{\mathrm{KL}}(q(\tau) \parallel P(\tau|\mathcal{E}_{1:T})) \\ &= \underbrace{\mathbb{E}_{\tau\sim q(\tau)}[\sum_{t}^{T} r(\mathbf{s}_t, \mathbf{a}_t) + \mathcal{H}(q(\cdot|\mathbf{s}_t))]}_{\text{maximum entropy objective}} + D_{\mathrm{KL}}(q(\mathbf{a}_t|\mathbf{s}_t) \parallel P(\mathbf{a}_t|\mathbf{s}_t, \mathcal{E}_{1:T}))\end{aligned}$$

where $\mathcal{L}(q, P) = \mathbb{E}_q[\log \frac{P}{q}]$ is the Evidence Lower Bound (ELBO) [28]. The maximum entropy objective arises naturally as the environment dynamics in $P(\tau, \mathcal{E})$ and $q(\tau)$ cancel out. Under this formulation, the soft policy iteration theorem [19] has an elegant Expectation-Maximization (EM) algorithm [28] interpretation: the E-step corresponds to the policy evaluation of the maximum entropy objective $\mathcal{L}(q^{[k]}, P)$; while the M-step corresponds to the policy improvement of the D_{KL} term $q^{[k+1]} = \arg\max_q D_{\mathrm{KL}}(q^{[k]}(\tau) \parallel P(\tau \mid \mathcal{E}))$. Thus, soft policy iteration is an exact inference if both EM steps can be performed exactly.

Theorem 1 (Convergence Theorem for Soft Policy Iteration)**.** *Let τ be the latent variable and $\mathcal{E}$ be the observed variable. Define the variational distribution $q(\tau)$ and the log-likelihood $\log P(\mathcal{E})$. Let $M : q^{[k]} \to q^{[k+1]}$ represent the mapping defined by the EM steps inference update, so that $q^{[k+1]} = M(q^{[k]})$. The likelihood function increases at each iteration of the variational inference algorithm until convergence conditions are satisfied.*

Proof. See Appendix A.1. □

2.2 The Option Framework

In conventional SMDP-based Option Framework [47], an option is a triple $(\mathbb{I}_o, \pi_o, \beta_o) \in \mathcal{O}$, where $\mathcal{O}$ denotes the option set; $o \in \mathbb{O} = \{1, 2, \dots, K\}$ is a positive integer index which denotes the o-th triple where K is the number of options; $\mathbb{I}_o$ is an initiation set indicating where the option can be initiated; $\pi_o = P_o(\mathbf{a}|\mathbf{s}) : \mathbb{A} \times \mathbb{S} \to [0, 1]$ is the action policy of the oth option; $\beta_o = P_o(\mathbf{b} = 1|\mathbf{s}) : \mathbb{S} \to [0, 1]$ where $\mathbf{b} \in \{0, 1\}$ is a *termination function*. For clarity, we use $P_o(\mathbf{b} = 1|\mathbf{s})$ instead of β_o which is widely used in previous option literatures (e.g., Sutton et al. [47], Bacon et al. [4]). A *master policy* $\pi(\mathbf{o}|\mathbf{s}) = P(\mathbf{o}|\mathbf{s})$ where $\mathbf{o} \in \mathbb{O}$ is used to sample which option will be executed. Therefore, the dynamics (stochastic process) of the option framework is written as:

$$\begin{aligned} P(\tau) = P(\mathbf{s}_0, \mathbf{o}_0) \prod_{t=1}^{\infty} & P(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{a}_{t-1}) P_{o_t}(\mathbf{a}_t|\mathbf{s}_t) \\ & [P_{o_{t-1}}(\mathbf{b}_t = 0|\mathbf{s}_t) \mathbf{1}_{\mathbf{o}_t = o_{t-1}} + P_{o_{t-1}}(\mathbf{b}_t = 1|\mathbf{s}_t) P(\mathbf{o}_t|\mathbf{s}_t)], \end{aligned} \tag{1}$$

where $\tau = \{\mathbf{s}_0, \mathbf{o}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{o}_1, \mathbf{a}_1, \ldots\}$ denotes the trajectory of the option framework. $\mathbf{1}$ is an indicator function and is only true when $\mathbf{o}_t = o_{t-1}$ (notice that o_{t-1} is the realization at $\mathbf{o}_{t-1}$). Therefore, under this formulation the option framework is defined as a Semi-Markov process since the dependency on an activated option o can cross a variable amount of time [47]. Due to the nature of SMDP assumption, conventional option framework is unstable and computationally expensive to optimize. Li et al. [34, 35] proposed the Hidden Temporal Markovian Decision Process (HiT-MDP):

$$P(\tau) = P(\mathbf{s}_0, \mathbf{o}_0) \prod_{t=1}^{\infty} P(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{a}_{t-1}) P(\mathbf{a}_t|\mathbf{s}_t, \mathbf{o}_t) P(\mathbf{o}_t|\mathbf{s}_t, \mathbf{o}_{t-1}) \tag{2}$$

and theoretically proved that the option-induced HiT-MDP is homomorphically equivalent to the conventional SMDP-based option framework. Following RL conventions, we use $\pi^A = P(\mathbf{a}_t|\mathbf{s}_t, \mathbf{o}_t)$ to denote the action policy and $\pi^O = P(\mathbf{o}_t|\mathbf{s}_t, \mathbf{o}_{t-1})$ to denote the option policy respectively. In HiT-MDPs, options can be viewed as latent variables with a temporal structure $P(\mathbf{o}_t|\mathbf{s}_t, \mathbf{o}_{t-1})$, enabling options to be represented as dense latent embeddings rather than traditional option triples. They demonstrated that learning options as embeddings on HiT-MDPs offers significant advantages in performance, scalability, and stability by reducing variance. However, their work only derived an on-policy policy gradient algorithm for learning options on HiT-MDPs. In this work, we extend their approach to an off-policy algorithm under the variational inference framework, enhancing exploration and sample efficiency.

3 Methodology

In this section, we introduce the Variational Markovian Option Critic (VMOC) algorithm by extending the variational policy iteration (Theorem 1) to the option framework. In Section 3.1, we reformulate the optimal option trajectory and the variational distribution as probabilistic graphical models (PGMs), propose the corresponding variational objective, and present a provable exact inference procedure for these objectives in tabular settings. Section 3.2 extends this result by introducing VMOC, a practical off-policy option learning algorithm that uses neural networks as function approximators and proves the convergence of VMOC under approximate inference settings. Our approach differs from previous works [19, 33, 34] by leveraging structured variational inference directly, providing a more concise pathway to both theoretical results and practical algorithms.

3.1 PGM Formulations of The Option Framework

Formulating complex problems as probabilistic graphical models (PGMs) offers a consistent and flexible framework for deriving principled objectives, analyzing convergence, and devising practical algorithms. In this section, we first formulate the optimal trajectory of the conventional SMDP-based option framework (Eq. 1) as a PGM. We then use the HiT-MDPs as the variational distribution to approximate this optimal trajectory. With these PGMs, we can straightforwardly derive the variational objective, where maximum entropy terms arise naturally. This approach allows us to develop a stable algorithm for learning diversified options and preventing degeneracy. Specifically, we follow [31, 28]

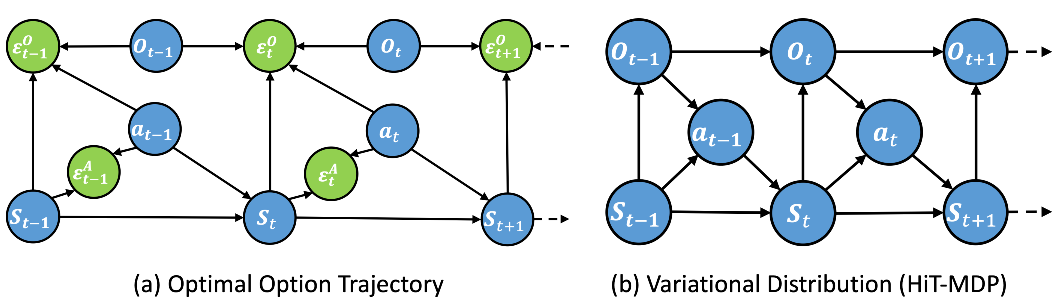

Figure 1: PGMs of the option framework.

by introducing the concept of "Optimality" [48] into the conventional SMDP-based option framework (Equation equation 1). This allows us to define the probability of an option trajectory being optimal as a probabilistic graphical model (PGM), as illustrated in Figure 1 (a):

$$P(\tau, \mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}) = P(\mathbf{s}_0, \mathbf{o}_0) \prod_{t=1}^{T} P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) P(\mathcal{E}^A_t = 1|\mathbf{s}_t, \mathbf{a}_t) P(\mathcal{E}^O_t = 1|\mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_t, \mathbf{o}_{t-1}) P(\mathbf{o}_t) P(\mathbf{a}_t)$$
$$\propto \underbrace{P(\mathbf{s}_0) \prod_{t=1}^{T} P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)}_{\text{Environment Dynamics}} \underbrace{\prod_{t=1}^{T} P(\mathcal{E}^A_t = 1|\mathbf{s}_t, \mathbf{a}_t) P(\mathcal{E}^O_t = 1|\mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_t, \mathbf{o}_{t-1})}_{\text{Optimality Likelihood}}, \quad (3)$$

where $\mathcal{E} \in \{0, 1\}$ are observable binary "optimal random variables" [31], $\tau = \{\mathbf{s}_0, \mathbf{o}_0, \mathbf{a}_0, \mathbf{s}_1 \dots\}$ denotes the trajectory of the option framework. The agent is *optimal* at time step t when $P(\mathcal{E}^A_t = 1|\mathbf{s}_t, \mathbf{a}_t)$ and $P(\mathcal{E}^O_t = 1|\mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_t, \mathbf{o}_{t-1})$. We will use $\mathcal{E}$ instead of $\mathcal{E} = 1$ in the following text to avoid cluttered notations. To simplify the derivation, priors $P(\mathbf{o})$ and $P(\mathbf{a})$ can be assumed to be uniform distributions without loss of generality [31]. Note that Eq. 3 shares the same environment dynamics with Eq. 1 and Eq. 2. With the optimal random variables $\mathcal{E}^O$ and $\mathcal{E}^A$, the likelihood of a state-action $\{\mathbf{s}_t, \mathbf{a}_t\}$ pair that is optimal is defined as:

$$P(\mathcal{E}^A_t|\mathbf{s}_t, \mathbf{a}_t) = \exp(r(\mathbf{s}_t, \mathbf{a}_t)), \quad (4)$$

as this specific design facilitates recovering the value function at the latter structural variational inference stage. Based on the same motivation, the likelihood of an option-state-action $\{\mathbf{o}_t, \mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_{t-1}\}$ pair that is optimal is defined as,

$$P(\mathcal{E}^O_t|\mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_t, \mathbf{o}_{t-1}) = \exp(f(\mathbf{o}_t, \mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_{t-1})), \quad (5)$$

where $f(\cdot)$ is an arbitrary non-positive function which measures the preferable of selecting an option given state-action pair $[\mathbf{s}_t, \mathbf{a}_t]$ and the previous executed option $\mathbf{o}_{t-1}$. In this work, we choose f to be the mutual-information $f = I[\mathbf{o}_t|\mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_{t-1}]$ as a fact that when the uniform prior assumption of $P(\mathbf{o})$ is relaxed the optimization introduces a mutual-information as a regularizer [35].

As explained in Section 2.1, direct optimization of Eq. 3 results in optimistic policies that assumes a degree of control over the dynamics. We correct this risk-seeking behavior [31] through approximating the optimal trajectory $P(\tau)$ with the variational distribution:

$$q(\tau) = P(\mathbf{s}_0, \mathbf{o}_0) \prod_{t=1}^{T-1} P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) q(\mathbf{a}_t|\mathbf{s}_t, \mathbf{o}_t) q(\mathbf{o}_t|\mathbf{s}_t, \mathbf{o}_{t-1}) \quad (6)$$

where the initial distribution $P(\mathbf{s}_0, \mathbf{o}_0)$ and transition distribution $P(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$ is set to be the true environment dynamics from $P(\tau)$. The variational distribution turns out to be the HiT-MDP, where the action policy $q(\mathbf{a}_t \mid \mathbf{s}_t)$ and the option policy $q(\mathbf{o}_t|\mathbf{s}_t, \mathbf{o}_{t-1})$ are used to approximate the optimal policy $P(\mathbf{a}_t|\mathbf{s}_t, \mathbf{o}_t, \mathcal{E}^A_{1:T})$ and $P(\mathbf{o}_t|\mathbf{s}_t, \mathbf{o}_{t-1}, \mathcal{E}^O_{1:T})$. The Evidence Lower Bound (ELBO) [28] of the log-likelihood optimal trajectory (Eq. 3) can be derived as (see Appendix A.3):

$$\begin{aligned}\mathcal{L}(q(\tau), P(\tau, \mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T})) &= \mathbb{E}_{q(\tau)}[\log P(\tau, \mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}) - \log q(\tau)] \\ &= \mathbb{E}_{q(\tau)}[r(\mathbf{s}_t, \mathbf{a}_t) + f(\cdot) - \log q(\mathbf{a}_t|\mathbf{s}_t, \mathbf{o}_t) - \log q(\mathbf{o}_t|\mathbf{s}_t, \mathbf{o}_{t-1})] \\ &= \mathbb{E}_{q(\tau)}\left[r(\mathbf{s}_t, \mathbf{a}_t) + f(\cdot) + \mathcal{H}[\pi^A] + \mathcal{H}[\pi^O]\right] \end{aligned} \quad (7)$$

where line 2 is substituting Eq. 3 and Eq. 6 into the ELBO. As a result, the maximum entropy objective naturally arises in Eq. 7. Optimizing the ELBO not only seeks high-reward options but also maximizes entropy across the space, promoting extensive exploration and maintaining high diversity.

Given the ELBO, we now define soft value functions of the option framework following the Bellman Backup Functions along the trajectory $q(\tau)$ as bellow:

$$Q^{soft}_O[\mathbf{s}_t, \mathbf{o}_t] = f(\cdot) + \mathbb{E}_{\mathbf{a}_t \sim \pi^A}\left[Q^{soft}_A[\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t]\right] + H[\pi^A], \quad (8)$$

$$Q^{soft}_A[\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t] = r(s, a) + \mathbb{E}_{\mathbf{s}_{t+1} \sim P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)}\left[\mathbb{E}_{\mathbf{o}_{t+1} \sim \pi^O}\left[Q^{soft}_O[\mathbf{s}_{t+1}, \mathbf{o}_{t+1}]\right] + H[\pi^O]\right] \quad (9)$$

Assuming policies $\pi^A, \pi^O \in \Pi$ where Π is an arbitrary feasible set, under a tabular setting where the inference on $\mathcal{L}$ can be done exactly, we have the following theorem holds:

Theorem 2 (Soft Option Policy Iteration Theorem)**.** *Repeated optimizing $\mathcal{L}$ and D_{KL} defined in Eq. 10 from any $\pi_0^A, \pi_0^O \in \Pi$ converges to optimal policies π^{A*}, π^{O*} such that $Q_O^{soft*}[\mathbf{s}_t, \mathbf{o}_t] \geq Q_O^{soft}[\mathbf{s}_t, \mathbf{o}_t]$ and $Q_A^{soft*}[\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t] \geq Q_A^{soft}[\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t]$, for all $\pi_0^A, \pi_0^O \in \Pi$ and $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_t) \in \mathcal{S} \times \mathcal{A} \times \mathcal{O}$, assuming under tabular settings where $|\mathcal{S}| < \infty,\ |\mathcal{O}| < \infty,\ |\mathcal{A}| < \infty$.*

Proof. See Appendix A.2. □

Theorem 2 guarantees finding the optimal solution only when the inference can be done exactly under tabular settings. However, real-world applications often involve large continuous domains and employ neural networks as function approximators. In these cases, inference procedures can only be done approximately. This necessitate a practical approximation algorithm which we present below.

3.2 Variational Markovian Option Critic Algorithm

Formulating complex problems as probabilistic graphical models (PGMs) allowing us to leverage established methods from PGM literature to address the associated inference and learning challenges in real-world applications. To this end, we utilizes the structured variational inference treatment for optimizing the log-likelihood of optimal trajectory and prove its convergence under approximate inference settings. Specifically, using the variational distribution $q(\tau)$ (Eq. 6) as an approximator, the ELBO can be derived as (see Appendix A.3):

$$\mathcal{L}(q(\tau), P(\tau, \mathcal{E}_{1:T}^A, \mathcal{E}_{1:T}^O)) = -D_{\mathrm{KL}}(q(\tau)||P(\tau|\mathcal{E}_{1:T}^A, \mathcal{E}_{1:T}^O)) + \log P(\mathcal{E}_{1:T}^A, \mathcal{E}_{1:T}^O) \tag{10}$$

where D_{KL} is the KL-Divergence between the trajectory following variational policies $q(\tau)$ and optimal policies $P(\tau|\mathcal{E}_{1:T}^A, \mathcal{E}_{1:T}^O)$. Under the structural variational inference [28] perspective, convergence to the optimal policy can be achieved by optimizing the ELBO with respect to the the variational policy repeatedly:

Theorem 3 (Convergence Theorem for Variational Markovian Option Policy Iteration)**.** *Let τ be the latent variable and $\mathcal{E}^A, \mathcal{E}^O$ be the ground-truth optimality variables. Define the variational distribution $q(\tau)$ and the true log-likelihood of optimality $\log P(\mathcal{E}^A, \mathcal{E}^O)$. iterates according to the update rule $q^{k+1} = \arg\max_q \mathcal{L}(q(\tau), P(\tau, \mathcal{E}_{1:T}^A, \mathcal{E}_{1:T}^O))$ converges to the maximum value bounded by the true log-likelihood of optimality.*

Proof. See Appendix A.4. □

We further implements a practical algorithm, the Variational Markovian Option Critic (VMOC) algorithm, which is suitable for complex continuous domains. Specifically, we employ parameterized neural networks as function approximators for both the Q-functions ($Q_{\psi^A}^{soft}$, $Q_{\psi^O}^{soft}$) and the policies (π_{θ^A}, π_{θ^O}). Instead of running evaluation and improvement to full convergence using Theorem 2, we can optimize the variational distribution by taking stochastic gradient descent following Theorem 3 with respect to the ELBO (Eq. 7) directly. Share the same motivation with Haarnoja et al. [19] of reducing the variance during the optimization procedure, we derive an option critic framework by optimizing the maximum entropy objectives between the action Eq. 9 and the option Eq. 8 alternatively. The Bellman residual for the action critic is:

$$J_{Q^A}(\psi_i^A) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) \sim D}\Bigg[\bigg(\min_{i=1,2} Q_{\psi_i^A}(\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t) - \\ \big(r(\mathbf{s}_t, \mathbf{a}_t) + \mathbb{E}_{\mathbf{o}_{t+1} \sim \pi^O}\Big[Q_O^{soft}[\mathbf{s}_{t+1}, \mathbf{o}_{t+1}]\Big] + \alpha^O H[\pi^O]\big)\bigg)^2\Bigg]$$

where α^O is the temperature hyper-parameter and the expectation over option random variable $\mathbb{E}_{\mathbf{o}_{t+1} \sim \pi^O}$ can be evaluated exactly since π^O is a discrete distribution. The Bellman residual for the option critic is:

$$J_{Q^O}(\psi_i^O) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{o}_t) \sim D}\Bigg[\bigg(\min_{i=1,2} Q_{\psi_i^O}^O(\mathbf{s}_t, \mathbf{o}_t) - \\ \big(f(\cdot) + \mathbb{E}_{\mathbf{a}_t \sim \pi^A}\Big[Q_A^{soft}[\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t] - \alpha^A \log q(\mathbf{a}_t|\mathbf{s}_t, \mathbf{o}_t)\Big]\big)\bigg)^2\Bigg]$$

α^A is the temperature hyper-parameter. Unlike $\mathbb{E}_{\mathbf{o}_{t+1}\sim\pi^O}$ can be trivially evaluated, evaluating $\mathbb{E}_{\mathbf{a}_t\sim\pi^A}$ is typically intractable. Therefore, in implementation we use $\mathbf{a}_t$ sampled from the replay buffer to estimate the expectation over π^A.

Following Theorem 3, the policy gradients can be derived by directly taking gradient with respect to the ELBOs defined for the action Eq. 9 and the option Eq. 8 policies respectively. The action policy objective is given by:

$$J_{\pi^A}(\theta^A) = -\mathbb{E}_{(\mathbf{s}_t,\mathbf{o}_t)\sim D}\left[\min_{i=1,2} Q_{\psi_i^A}(\mathbf{s}_t,\mathbf{o}_t,\tilde{\mathbf{a}}_t) - \alpha^A \log q(\tilde{\mathbf{a}}_t|\mathbf{s}_t,\mathbf{o}_t)\right],\ \tilde{\mathbf{a}}_t \sim q(\cdot|\mathbf{s}_t,\mathbf{o}_t)$$

where in practice the action policy is often sampled by using the re-parameterization trick introduced in [19]. The option objective is given by:

$$J_{\pi^O}(\theta^O) = -\mathbb{E}_{(\mathbf{s}_t,\mathbf{o}_{t-1})\sim D}\left[\min_{i=1,2} Q_{\psi_i^O}(\mathbf{s}_t,\mathbf{o}_t) + \alpha^O \mathcal{H}[\pi^O]\right]$$

The variational distribution $q(\tau)$ defined in Eq. 6 allows us to learn options as embeddings [34, 35] with a learnable embedding matrix $\mathbf{W} \in \mathbb{R}^{\text{num\_options}\times\text{embedding\_dim}}$. Under this setting, the embedding matrix $\mathbf{W}$ can be absorbed into the parameter vector θ^O. This integration into VMOC ensures that options are represented as embeddings without any additional complications, thereby enhancing the expressiveness and scalability of the model.

The temperature hyper-parameters can also be adjusted by minimizing the following objective:

$$J(\alpha^A) = -\mathbb{E}_{\tilde{\mathbf{a}}_t\sim\pi^A}\left[\alpha^A(\log \pi^A(\tilde{\mathbf{a}}_t \mid \mathbf{s}_t,\mathbf{o}_t) + \overline{\mathcal{H}})\right]$$

for the action policy temperature α^A, where $\overline{\mathcal{H}}$ is a target entropy. Similarly, the option policy temperature α^O can be adjusted by:

$$J(\alpha^O) = -\mathbb{E}_{\mathbf{o}_t\sim\pi^O}\left[\alpha^O(\log \pi^O(\mathbf{o}_t \mid \mathbf{s}_t,\mathbf{o}_{t-1}) + \overline{\mathcal{H}})\right]$$

where $\overline{\mathcal{H}}$ is also a target entropy for the option policy. In both cases, the temperatures α^A and α^O are updated using gradient descent, ensuring that the entropy regularization terms dynamically adapt to maintain a desired level of exploration. This approach aligns with the methodology proposed in SAC [19]. By adjusting the temperature parameters, the VMOC algorithm ensures a balanced trade-off between exploration and exploitation, which is crucial for achieving optimal performance in complex continuous control tasks. We summarize the VMOC algorithm in Appendix B.

4 Experiments

In this section, we design experiments on the challenging single task OpenAI Gym MuJoCo [7] environments (10 environments) to test Variational Markovian Option Critic (VMOC)'s performance over other option variants and non-option baselines.

For VMOC in all environments, we fix the temperature rate for both α^O and α^A to 0.05; we add an exploration noise $\mathcal{N}(\mu = 0, \sigma = 0.2)$ during exploration. For all baselines, we follow DAC [52]'s open source implementations and compare our algorithm with six baselines, five of which are option variants, *i.e.*, MOPG [35], DAC+PPO, AHP+PPO [32], IOPG [45], PPOC [27], OC [4] and PPO [41]. All baselines' parameters used by DAC remain unchanged over 1 million environment steps to converge. Figures are plotted following DAC's style: curves are averaged over 10 independent runs and smoothed by a sliding window of size 20. Shaded regions indicate standard deviations. All experiments are run on an Intel® Core™ i9-9900X CPU @ 3.50GHz with a single thread and process. Our implementation details are summarized in Appendix C. For a fair comparison, we follow option literature conventions and use four options in all implementations. Our code is available in supplemental materials.

5 Experiments

We evaluate the performance of VMOC against six option-based baselines (MOPG [35], DAC+PPO [52], AHP+PPO [32], IOPG [45], PPOC [27], and OC [4]) as well as the hierarchy-free

PPO algorithm [41]. Previous studies [27, 45, 20, 52] have suggested that option-based algorithms do not exhibit significant advantages over hierarchy-free algorithms in single-task environments. Nonetheless, our results demonstrate that VMOC significantly outperforms all baselines in terms of episodic return, convergence speed, step variance, and variance across 10 runs, as illustrated in Figure 2. The only exception is the relatively simple InvertedDoublePendulum environment, which

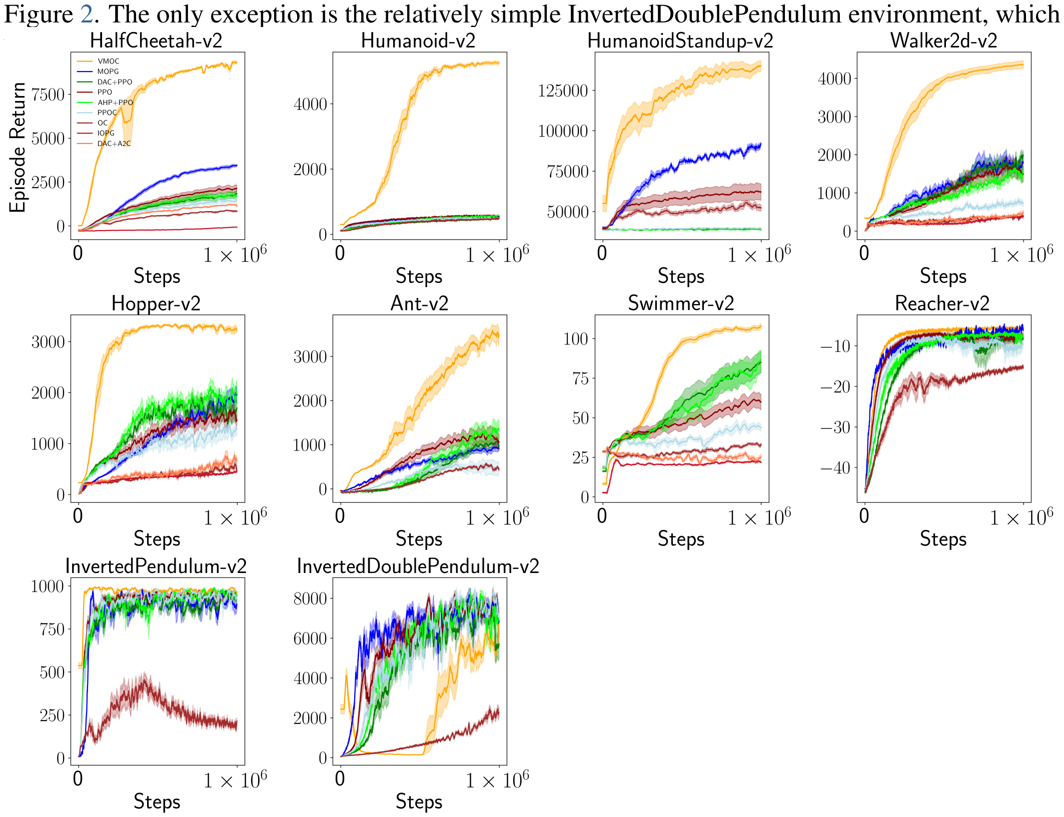

Figure 2: Experiments on Mujoco Environments. Curves are averaged over 10 independent runs with different random seeds and smoothed by a sliding window of size 20. Shaded regions indicate standard deviations.

Notably, VMOC exhibits superior performance on the Humanoid-v2 and HumanoidStandup-v2 environments. These environments are characterized by a large state space ($\mathcal{S} \in \mathbb{R}^{376}$) and action space ($\mathcal{A} \in \mathbb{R}^{17}$), whereas other environments typically have state dimensions less than 20 and action dimensions less than 5. The enhanced performance of VMOC in these environments can be attributed to its maximum entropy capability: in large state-action spaces, the agent must maximize rewards while exploring a diverse set of state-action pairs. Maximum likelihood methods tend to quickly saturate with early rewarding observations, leading to the selection of low-entropy options that converge to local optima.

A particularly relevant comparison is with the Markovian Option Policy Gradient (MOPG) [35], as both VMOC and MOPG are developed based on HiT-MDPs and employ option embeddings. Despite being derived under the maximum entropy framework, MOPG utilizes an on-policy gradient descent approach. Our experimental results show that VMOC's performance surpasses that of MOPG, highlighting the limitations of on-policy methods, which suffer from shortsighted rollout lengths and quickly saturate to early high-reward observations. In contrast, VMOC's variational off-policy approach effectively utilizes the maximum entropy framework by ensuring better exploration and stability across the learning process. Additionally, the off-policy nature of VMOC allows it to reuse samples from a replay buffer, enhancing sample efficiency and promoting greater diversity in the learned policies. This capability leads to more robust learning, as the algorithm can leverage a broader range of experiences to improve policy optimization.

6 Related Work

The VMOC incorporates three key ingredients: the option framework, a structural variational inference based off-policy algorithm and latent variable policies. We review prior works that draw on some of these ideas in this section. The options framework [47] offers a promising approach for discovering and reusing temporal abstractions, with options representing temporally abstract skills. Conventional option frameworks [39], typically developed under the maximum likelihood (MLE) framework with few constraints on options behavior, often suffer from the option degradation problem [32, 4]. This problem occurs when options quickly saturate with early rewarding observations, causing a single option to dominate the entire policy, or when options switch every timestep, maximizing policy at the expense of skill reuse across tasks. On-policy option learning algorithms [4, 3, 52, 34, 35] aim to maximize expected return by adjusting policy parameters to increase the likelihood of high-reward option trajectories, which often leads to focusing on low-entropy options. Several techniques [20, 21, 23] have been proposed to enhance on-policy algorithms with entropy-like extrinsic rewards as regularizers, but these often result in biased optimal trajectories. In contrast, the maximum entropy term in VMOC arises naturally within the variational framework and provably converges to the optimal trajectory.

Although several off-policy option learning algorithms have been proposed [10, 43, 45, 50], these typically focus on improving sample efficiency by leveraging the control as inference framework. Recent works [45] aim to enhance sample efficiency by inferring and marginalizing over options, allowing all options to be learned simultaneously. Wulfmeier et al. [50] propose off-policy learning of all options across every experience in hindsight, further boosting sample efficiency. However, these approaches generally lack constraints on options behavior. A closely related work [33] also derives a variational approach under the option framework; however, it is based on probabilistic graphical model that we believe are incorrect, potentially leading to convergence issues. Additionally, our algorithm enables learning options as latent embeddings, a feature not present in their approach.

Recently, several studies have extended the maximum entropy reinforcement learning framework to discover skills by incorporating additional latent variables. One class of methods [22, 17] maintains latent variables constant over the duration of an episode, providing a time-correlated exploration signal. Other works [19, 51] focus on discovering multi-level action abstractions that are suitable for repurposing by promoting skill distinguishability, but they do not incorporate temporal abstractions. Studies such as [38, 1, 8] aim to discover temporally abstract skills essential for exploration, but they predefine their temporal resolution. In contrast, VMOC learns temporal abstractions as embeddings in an end-to-end data-driven approach with minimal prior knowledge encoded in the framework.

7 Conclusion

In this paper, we have introduced the Variational Markovian Option Critic (VMOC), a novel off-policy algorithm designed to address the challenges of ineffective exploration, sample inefficiency, and computational complexity inherent in the conventional option framework for hierarchical reinforcement learning. By integrating a variational inference framework, VMOC leverages maximum entropy as intrinsic rewards to promote the discovery of diverse and effective options. Additionally, by employing low-cost option embeddings instead of traditional, computationally expensive option triples, VMOC enhances both scalability and expressiveness. Extensive experiments in challenging Mujoco environments demonstrate that VMOC significantly outperforms existing on-policy and off-policy option variants, validating its effectiveness in learning coherent and diverse option sets suitable for complex tasks. This work advances the field of hierarchical reinforcement learning by providing a robust, scalable, and efficient method for learning temporally extended actions.

8 Limitations

Due to limited computing resources, we did not conduct an ablation study of VMOC. Additionally, the temperature parameter was fixed in our experiments, whereas an automatically tuned parameter could potentially enhance performance (see SAC [19]). While our baselines focus on option variants, a thorough comparison to other off-policy algorithms is also worth investigating. It is particularly important to explore whether VMOC exhibits performance improvements in scalability when the number of option embeddings is significantly increased. These investigations are left for future work.

References

[1] Ajay, A., Kumar, A., Agrawal, P., Levine, S., and Nachum, O. Opal: Offline primitive discovery for accelerating offline reinforcement learning. *arXiv preprint arXiv:2010.13611*, 2020.

[2] Araujo, E. G. and Grupen, R. A. Learning control composition in a complex environment. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pp. 333–342, 1996.

[3] Bacon, P.-L. *Temporal Representation Learning*. PhD thesis, McGill University Libraries, 2018.

[4] Bacon, P.-L., Harb, J., and Precup, D. The option-critic architecture. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[5] Bertsekas, D. and Tsitsiklis, J. N. *Neuro-dynamic programming*. Athena Scientific, 1996.

[6] Brockett, R. W. Hybrid models for motion control systems. In *Essays on Control: Perspectives in the Theory and its Applications*, pp. 29–53. Springer, 1993.

[7] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[8] Co-Reyes, J., Liu, Y., Gupta, A., Eysenbach, B., Abbeel, P., and Levine, S. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. In *International conference on machine learning*, pp. 1009–1018. PMLR, 2018.

[9] Colombetti, M., Dorigo, M., and Borghi, G. Behavior analysis and training-a methodology for behavior engineering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(3):365–380, 1996.

[10] Daniel, C., Van Hoof, H., Peters, J., and Neumann, G. Probabilistic inference for determining options in reinforcement learning. *Machine Learning*, 104(2-3):337–357, 2016.

[11] Dayan, P. and Hinton, G. E. Feudal reinforcement learning. *Advances in Neural Information Processing Systems*, pp. 271–278, 1993.

[12] Dietterich, T. G. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

[13] Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[14] Fujimoto, S., Van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

[15] Goyal, A., Islam, R., Strouse, D., Ahmed, Z., Botvinick, M., Larochelle, H., Bengio, Y., and Levine, S. Infobot: Transfer and exploration via the information bottleneck. *arXiv preprint arXiv:1901.10902*, 2019.

[16] Guo, Z., Thomas, P. S., and Brunskill, E. Using options and covariance testing for long horizon off-policy policy evaluation. In *Advances in Neural Information Processing Systems*, pp. 2492–2501, 2017.

[17] Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

[18] Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pp. 1352–1361. PMLR, 2017.

[19] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[20] Harb, J., Bacon, P.-L., Klissarov, M., and Precup, D. When waiting is not an option: Learning options with a deliberation cost. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[21] Harutyunyan, A., Dabney, W., Borsa, D., Heess, N., Munos, R., and Precup, D. The termination critic. *arXiv preprint arXiv:1902.09996*, 2019.

[22] Hausman, K., Springenberg, J. T., Wang, Z., Heess, N., and Riedmiller, M. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018.

[23] Kamat, A. and Precup, D. Diversity-enriched option-critic. *arXiv*, 2020.

[24] Khetarpal, K. and Precup, D. Learning options with interest functions. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 1–2, 2019.

[25] Khetarpal, K., Klissarov, M., Chevalier-Boisvert, M., Bacon, P.-L., and Precup, D. Options of interest: Temporal abstraction with interest functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4,444–4,451, 2020.

[26] Klissarov, M. and Precup, D. Flexible option learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4632–4646. Curran Associates, 2021.

[27] Klissarov, M., Bacon, P.-L., Harb, J., and Precup, D. Learnings options end-to-end for continuous action tasks. *arXiv preprint arXiv:1712.00004*, 2017.

[28] Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[29] Kolobov, A., Weld, D. S., et al. Discovering hidden structure in factored mdps. *Artificial Intelligence*, 189:19–47, 2012.

[30] Konidaris, G. and Barto, A. G. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in neural information processing systems*, pp. 1015–1023, 2009.

[31] Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

[32] Levy, K. Y. and Shimkin, N. Unified inter and intra options learning using policy gradient methods. In *European Workshop on Reinforcement Learning*, pp. 153–164. Springer, 2011.

[33] Li, C., Ma, X., Zhang, C., Yang, J., Xia, L., and Zhao, Q. Soac: The soft option actor-critic architecture. *arXiv preprint arXiv:2006.14363*, 2020.

[34] Li, C., Song, D., and Tao, D. The skill-action architecture: Learning abstract action embeddings for reinforcement learning. 2020.

[35] Li, C., Song, D., and Tao, D. Hit-mdp: learning the smdp option framework on mdps with hidden temporal embeddings. In *The Eleventh International Conference on Learning Representations*, 2022.

[36] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[37] Osa, T., Tangkaratt, V., and Sugiyama, M. Hierarchical reinforcement learning via advantage-weighted information maximization. *arXiv preprint arXiv:1901.01365*, 2019.

[38] Pertsch, K., Rybkin, O., Ebert, F., Finn, C., Jayaraman, D., and Levine, S. Long-horizon visual planning with goal-conditioned hierarchical predictors. *NeurIPS*, 2020.

[39] Precup, D. *Temporal abstraction in reinforcement learning*. University of Massachusetts Amherst, 2000.

[40] Schulman, J., Chen, X., and Abbeel, P. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017.

[41] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[42] Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

[43] Shiarlis, K., Wulfmeier, M., Salter, S., Whiteson, S., and Posner, I. Taco: Learning task decomposition via temporal alignment for control. In *International Conference on Machine Learning*, pp. 4654–4663. PMLR, 2018.

[44] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[45] Smith, M., Hoof, H., and Pineau, J. An inference-based policy gradient method for learning options. In *International Conference on Machine Learning*, pp. 4,703–4,712, 2018.

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

[47] Sutton, R. S., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.

[48] Todorov, E. Linearly-solvable markov decision problems. *Advances in neural information processing systems*, 19, 2006.

[49] Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

[50] Wulfmeier, M., Rao, D., Hafner, R., Lampe, T., Abdolmaleki, A., Hertweck, T., Neunert, M., Tirumala, D., Siegel, N., Heess, N., et al. Data-efficient hindsight off-policy option learning. *arXiv preprint arXiv:2007.15588*, 2020.

[51] Zhang, D., Courville, A., Bengio, Y., Zheng, Q., Zhang, A., and Chen, R. T. Latent state marginalization as a low-cost approach for improving exploration. *arXiv preprint arXiv:2210.00999*, 2022.

[52] Zhang, S. and Whiteson, S. DAC: The double actor-critic architecture for learning options. In *Advances in Neural Information Processing Systems*, pp. 2,012–2,022, 2019.

[53] Ziebart, B. D., Bagnell, J. A., and Dey, A. K. Modeling interaction via the principle of maximum causal entropy. In *ICML*, 2010.

A Proofs

A.1 Theorem 1

Theorem 1 (Convergence Theorem for Structured Variational Policy Iteration). *Let τ be the latent variable and $\mathcal{E}$ be the observed variable. Define the variational distribution $q(\tau)$ and the log-likelihood $\log P(\mathcal{E})$. Let $M : q^{[k]} \to q^{[k+1]}$ represent the mapping defined by the EM steps inference update, so that $q^{[k+1]} = M(q^{[k]})$. The likelihood function increases at each iteration of the variational inference algorithm until the conditions for equality are satisfied and a fixed point of the iteration is reached:*

$$\log P(\mathcal{E} \mid q^{[k+1]}) \geq \log P(\mathcal{E} \mid q^{[k]}), \text{ with equality if and only if}$$

$$\mathcal{L}(q^{[k+1]}, P) = \mathcal{L}(q^{[k]}, P)$$

and

$$D_{KL}(q^{[k+1]}(\tau) \parallel P(\tau \mid \mathcal{E})) = D_{KL}(q^{[k]}(\tau) \parallel P(\tau \mid \mathcal{E})).$$

Proof. Let τ be the latent variable and $\mathcal{E}$ be the observed variable. Define the evidence lower bound (ELBO) as $\mathcal{L}(q, P)$ and the Kullback-Leibler divergence as $\mathrm{D}_{\mathrm{KL}}(q \parallel P)$, where $q(\tau)$ approximates the posterior distribution and $P(\mathcal{E} \mid \tau)$ is the likelihood.

The log-likelihood function $\log P(\mathcal{E})$ can be decomposed as:

$$\log P(\mathcal{E}) = \mathcal{L}(q, P) + \mathrm{D}_{\mathrm{KL}}(q(\tau) \parallel P(\tau \mid \mathcal{E})),$$

where

$$\mathcal{L}(q, P) = \mathbb{E}_{q(\tau)} \left[\log P(\mathcal{E}, \tau) - \log q(\tau)\right]$$

and

$$\mathrm{D}_{\mathrm{KL}}(q(\tau) \parallel P(\tau \mid \mathcal{E})) = \mathbb{E}_{q(\tau)} \left[\log \frac{q(\tau)}{P(\tau \mid \mathcal{E})}\right].$$

Let $M : q^{[k]} \to q^{[k+1]}$ represent the mapping defined by the variational inference update, so that $q^{[k+1]} = M(q^{[k]})$. If q^* is a variational distribution that maximizes the ELBO, so that $\log P(\mathcal{E} \mid q^*) \geq \log P(\mathcal{E} \mid q)$ for all q, then $\log P(\mathcal{E} \mid M(q^*)) = \log P(\mathcal{E} \mid q^*)$. In other words, the maximizing distributions are fixed points of the variational inference algorithm. Since the likelihood function is bounded (for distributions of practical interest), the sequence of variational distributions $q^{[0]}, q^{[1]}, \ldots, q^{[k]}$ yields a bounded nondecreasing sequence $\log P(\mathcal{E} \mid q^{[0]}) \leq \log P(\mathcal{E} \mid q^{[1]}) \leq \cdots \leq \log P(\mathcal{E} \mid q^{[k]}) \leq \log P(\mathcal{E} \mid q^{[k]})$ which must converge as $k \to \infty$.

□

A.2 Theorem 2

Theorem 2 (Soft Option Policy Iteration Theorem). *Repeated optimizing $\mathcal{L}$ and D_{KL} defined in Eq. 10 from any $\pi_0^A, \pi_0^O \in \Pi$ converges to optimal policies π^{A*}, π^{O*} such that $Q_O^{soft*}[\mathbf{s}_t, \mathbf{o}_t] \geq Q_O^{soft}[\mathbf{s}_t, \mathbf{o}_t]$ and $Q_A^{soft*}[\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t] \geq Q_A^{soft}[\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t]$, for all $\pi_0^A, \pi_0^O \in \Pi$ and $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_t) \in \mathcal{S} \times \mathcal{A} \times \mathcal{O}$, assuming $|\mathcal{S}| < \infty,\ |\mathcal{O}| < \infty,\ |\mathcal{A}| < \infty$.*

Proof. Define the entropy augmented reward as $r^{soft}(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \mathcal{H}[\pi^A]$ and $f^{soft}(\mathbf{o}_t, \mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_{t-1}) = f(\mathbf{o}_t, \mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_{t-1}) + \mathcal{H}[\pi^O]$ and rewrite Bellman Backup functions as,

$$\begin{aligned} Q_O[\mathbf{s}_t, \mathbf{o}_t] &= f^{soft}(\cdot) + \mathbb{E}_{\mathbf{a}_t \sim \pi^A} \left[Q_A[\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t]\right], \\ Q_A[\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t] &= r^{soft}(s, a) + \mathbb{E}_{\mathbf{s}_{t+1} \sim P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)} \left[\mathbb{E}_{\mathbf{o}_{t+1} \sim \pi^O} \left[Q_O[\mathbf{s}_{t+1}, \mathbf{o}_{t+1}]\right]\right] \end{aligned}$$

We start with proving the convergence of soft option policy evaluation. As with the standard Q-function and value function, we can relate the Q-function at a future state via a *Bellman Operator* $\mathcal{T}^{soft}$. The option-action value function satisfies the Bellman Operator $\mathcal{T}^{soft}$

$$\begin{aligned} \mathcal{T}^{soft} Q_A[\mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t] &= \mathbb{E}[G_t | \mathbf{s}_t, \mathbf{o}_t, \mathbf{a}_t] \\ &= r^{soft}(s, a) + \gamma \sum_{\mathbf{s}_{t+1}} P(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t) Q_O[\mathbf{s}_{t+1}, \mathbf{o}_t], \end{aligned}$$

As with the standard convergence results for policy evaluation [46], by the definition of $\mathcal{T}^{soft}$ (Eq. 11) the option-action value function $Q_A^{\pi_A}$ is a fixed point.

To prove the $\mathcal{T}^{soft}$ is a contraction, define a norm on V-values functions V and U

$$\|V - U\|_\infty \triangleq \max_{\bar{s} \in \bar{S}} |V(\bar{s}) - U(\bar{s})|. \tag{11}$$

where $\bar{s} = \{s, o\}$.

By recurssively apply the Hidden Temporal Bellman Operator $\mathcal{T}^{soft}$, we have:

$$
\begin{aligned}
Q_O[\mathbf{s}_t, \mathbf{o}_{t-1}] = \mathbb{E}[G_t|\mathbf{s}_t, \mathbf{o}_{t-1}] &= \sum_{\mathbf{o}_t} P(\mathbf{o}_t|\mathbf{s}_t, \mathbf{o}_{t-1}) Q_O[\mathbf{s}_t, \mathbf{o}_t] \\
&= \sum_{\mathbf{o}_t} P(\mathbf{o}_t|\mathbf{s}_t, \mathbf{o}_{t-1}) \sum_{\mathbf{a}_t} P(\mathbf{a}_t|\mathbf{s}_t, \mathbf{o}_t) \Big[r(s,a) + \gamma \sum_{\mathbf{s}_{t+1}} P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) Q_O[\mathbf{s}_{t+1}, \mathbf{o}_t] \Big] \\
&= r(s,a) + \gamma \sum_{\mathbf{o}_t} P(\mathbf{o}_t|\mathbf{s}_t, \mathbf{o}_{t-1}) \sum_{\mathbf{a}_t} P(\mathbf{a}_t|\mathbf{s}_t, \mathbf{o}_t) \sum_{\mathbf{s}_{t+1}} P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) Q_O[\mathbf{s}_{t+1}, \mathbf{o}_t] \\
&= r(s,a) + \gamma \sum_{\mathbf{o}_t, \mathbf{s}_{t+1}} P(\mathbf{s}_{t+1}, \mathbf{o}_t|\mathbf{s}_t, \mathbf{o}_{t-1}) Q_O[\mathbf{s}_{t+1}, \mathbf{o}_t] \\
&= r(s,a) + \gamma E_{\mathbf{s}_{t+1}, \mathbf{o}_t} \Big[Q_O[\mathbf{s}_{t+1}, \mathbf{o}_t] \Big] \qquad (12)
\end{aligned}
$$

Therefore, by applying Eq. 12 to V and U we have:

$$
\begin{aligned}
&\|T^\pi V - T^\pi U\|_\infty \\
&= \max_{\bar{s} \in \bar{S}} \Big| \gamma E_{\mathbf{s}_{t+1}, \mathbf{o}_t} \Big[Q_O[\mathbf{s}_{t+1}, \mathbf{o}_t] \Big] - \gamma E_{\mathbf{s}_{t+1}, \mathbf{o}_t} \Big[U[\mathbf{s}_{t+1}, \mathbf{o}_t] \Big] \Big| \\
&= \gamma \max_{\bar{s} \in \bar{S}} E_{\mathbf{s}_{t+1}, \mathbf{o}_t} \Big[\Big| Q_O[\mathbf{s}_{t+1}, \mathbf{o}_t] - U[\mathbf{s}_{t+1}, \mathbf{o}_t] \Big| \Big] \\
&\leq \gamma \max_{\bar{s} \in \bar{S}} E_{\mathbf{s}_{t+1}, \mathbf{o}_t} \Big[\gamma \max_{\bar{s} \in \bar{S}} \Big| Q_O[\mathbf{s}_{t+1}, \mathbf{o}_t] - U[\mathbf{s}_{t+1}, \mathbf{o}_t] \Big| \Big] \\
&\leq \gamma \max_{\bar{s} \in \bar{S}} |V[\bar{s}] - U[\bar{s}]| \\
&= \gamma \|V - U\|_\infty \qquad (13)
\end{aligned}
$$

Therefore, $\mathcal{T}^{soft}$ is a contraction. By the fixed point theorem, assuming that throughout our computation the $Q_A[\cdot, \cdot]$ and $Q_O[\cdot]$ are bounded and $\mathbb{A} < \infty$, the sequence Q_A^k defined by $Q_A^{k+1} = \mathcal{T}^{soft} Q_A^k$ will converge to the option-action value function $Q_A^{\pi_A}$ as $k \to \infty$.

The convergence results of and the Soft Option Policy Improvement Theorem then follows conventional Soft Policy Improvement Theorem Theorem 1. Consequently, the Soft Option Policy Iteration Theorem follows directly from these results.

□

A.3 Derivation of Eq. 10

$$
\begin{aligned}
\mathcal{L}(q(\tau), P(\tau, \mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T})) &= \mathbb{E}_{q(\tau)}[\log P(\tau, \mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}) - \log q(\tau)] \\
&= \mathbb{E}_{q(\tau)}[\log P(\tau|\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}) + \log P(\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}) - \log q(\tau)] \\
&= \mathbb{E}_{q(\tau)}[\log P(\tau|\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}) - \log q(\tau)] + \mathbb{E}_{q(\tau)} \log P(\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}) \\
&= \mathbb{E}_{q(\tau)}[\frac{\log P(\tau|\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T})}{\log q(\tau)}] + \log P(\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}) \\
&= -D_{\mathrm{KL}}(\log q(\tau) \parallel \log P(\tau|\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T})) + \log P(\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T})
\end{aligned}
$$

A.4 Theorem 3

Theorem 3 (Convergence Theorem for Variational Markovian Option Policy Iteration). *Let τ be the latent variable and $\mathcal{E}^A, \mathcal{E}^O$ be the ground-truth optimality variables. Define the variational distribution $q(\tau)$ and the true log-likelihood of optimality $\log P(\mathcal{E}^A, \mathcal{E}^O)$. iterates according to the update rule $q^{k+1} = \arg\max_q \mathcal{L}(q(\tau), P(\tau, \mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}))$ converges to the maximum value bounded by the data log-likelihood.*

Proof. The objective is to maximize the ELBO with respect to the policy q. Formally, this can be written as:

$$q^{k+1} = \arg\max_q \mathcal{L}(q, P).$$

Suppose we q is a neural network function approximator, assuming the continuity and differentiability of q with respect to its parameters. Using stochastic gradient descent (SGD) to optimize the parameters guarantees that the ELBO increases, such that $\mathcal{L}(q^{k+1}, P) \geq \mathcal{L}(q^k, P)$.

Rearranging Eq. 10 we get:

$$\begin{aligned}
D_{\mathrm{KL}}(q^{k+1}(\tau)||P(\tau|\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T})) &= -L(q^{k+1}(\tau), P(\tau, \mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T})) + \log P(\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}) \\
&\leq -L(q^{k}(\tau), P(\tau, \mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T})) + \log P(\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}) \\
&= D_{\mathrm{KL}}(q^{k}(\tau)||P(\tau|\mathcal{E}^A_{1:T}, \mathcal{E}^O_{1:T}))
\end{aligned}$$

Thus, each SGD update not only potentially increases the ELBO but also decreases the KL divergence, moving q closer to P. Given the properties of SGD and assuming appropriate learning rates and sufficiently expressive neural network architectures, the sequence $\{q^k\}$ converges to a policy q^* that minimizes the KL divergence to the true posterior. $\square$

B VMOC Algorithm

Algorithm 1 VMOC Algorithm

1: Initialize parameter vectors ψ^A, ψ^O, θ^O, θ^A
2: **for** each epoch **do**
3: Collect trajectories $\{\mathbf{o}_{t-1}, \mathbf{s}_t, \mathbf{a}_t, \mathbf{o}_t\}$ into the replay buffer
4: **for** each gradient step **do**
5: Update $Q^{soft}_{\psi^A_i}$: $\psi^A_i \leftarrow \psi^A_i - \eta_{Q^A} \nabla J_{Q^{soft}_{\psi^A_i}}$ for $i \in \{1, 2\}$
6: Update $Q^{soft}_{\psi^O_i}$: $\psi^O_i \leftarrow \psi^O_i - \eta_{Q^O} \nabla J_{Q^{soft}_{\psi^O_i}}$ for $i \in \{1, 2\}$
7: Update $\pi^O_{\theta^O}$: $\theta^O \leftarrow \theta^O - \eta_{\pi^O} \nabla J_{\pi^O}$
8: Update $\pi^A_{\theta^A}$: $\theta^A \leftarrow \theta^A - \eta_{\pi^A} \nabla J_{\pi^A}$
9: Update target networks: $\bar{\psi}^A \leftarrow \sigma\psi^A + (1-\sigma)\bar{\psi}^A$, $\bar{\psi}^O \leftarrow \sigma\psi^O + (1-\sigma)\bar{\psi}^O$
10: Update temperature factors: $\alpha^O \leftarrow \alpha^O - \eta_{\alpha^O} \nabla J_{\alpha^O}$, $\alpha^A \leftarrow \alpha^A - \eta_{\alpha^A} \nabla J_{\alpha^A}$
11: **end for**
12: **end for**

C Implementation Details

C.1 Hyperparameters

In this section we summarize our implementation details. For a fair comparison, all baselines: MOPG [35], DAC+PPO [52], AHP+PPO [32], PPOC [27], OC [4] and PPO [41] are from DAC's open source Github repo: `https://github.com/ShangtongZhang/DeepRL/tree/DAC`. Hyper-parameters used in DAC [52] for all these baselines are kept unchanged.

VMOC Network Architecture: We use Pytorch to build neural networks. Specifically, for option embeddings, we use an embedding matrix $\boldsymbol{W}_S \in \mathbb{R}^{4\times 40}$ which has 4 options (4 rows) and an embedding size of 40 (40 columns). For layer normalization we use Pytorch's built-in function LayerNorm [2]. For Feed Forward Networks (FNN), we use a 2 layer FNN with ReLu function as activation function with input size of state-size, hidden size of $[256, 256]$, and output size of action-dim neurons. For Linear layer, we use built-in Linear function[3] to map FFN's outputs to 4 dimension.

[2] https://pytorch.org/docs/stable/generated/torch.nn.LayerNorm.html
[3] https://pytorch.org/docs/stable/generated/torch.nn.Linear.html

Each dimension acts like a logit for each skill and is used as density in Categorical distribution[4]. For both action policy and critic module, FFNs are of the same size as the one used in the skill policy.

Preprocessing: States are normalized by a running estimation of mean and std.

Hyperparameters for all on-policy option variants: For a fair comparison, we use exactly the same parameters of PPO as DAC . Specifically:

- Optimizer: Adam with $\epsilon = 10^{-5}$ and an initial learning rate 3×10^{-4}
- Discount ratio γ: 0.99
- GAE coefficient: 0.95
- Gradient clip by norm: 0.5
- Rollout length: 2048 environment steps
- Optimization epochs: 10
- Optimization batch size: 64
- Action probability ratio clip: 0.2

Computing Infrastructure: We conducted our experiments on an Intel® Core™ i9-9900X CPU @ 3.50GHz with a single thread and process with PyTorch.

[4]https://github.com/pytorch/pytorch/blob/master/torch/distributions/categorical.py

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the claims and findings of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of the study are discussed in the discussion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper provides a full derivation of assumptions and proofs of the theoretical result (convergence of the evidence lower bound)

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes. Our code is provided in supplementary materials. Full details of the experimental setup, model architectures are provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper provides open access to the code and data.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Justification: All details are provided in the main content and the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All gym env experiments are run with 10 different random seeds. Performance are reported by 1 sigma shaded area over all 10 runs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computational details are provided in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research was conducted in accordance with the NeurIPs Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The work in the paper has no potential for societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The only applicable assets are the code which are credited and distributed under a Creative Commons Attribution License.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: New assets include the code required to run the experiments described in the paper. Documentation is provided along with the code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.