Option2Vec: Learning Temporal-State Abstraction Embeddings on MDPs

Anonymous Author(s)
Affiliation
Address
email

Abstract

The option framework develops options on a sample inefficient Semi-Markov Decision Process (SMDP) and represents each option computational expensively as two distributions and one initiation set. In this paper, we present the option-induced SMDP as a simple Hidden-Markov-Model-style Probabilistic Graphical Model (PGM), which enables representing each option as efficient as one embedding vector (hidden variable) and combines temporal abstraction together with state abstraction. We derive policy gradient theorems based on this PGM and prove that it can be solved efficiently by employing the Double Actor-Critic (DAC) algorithm. For learning option embeddings, we implement Option2Vec (O2V), a simple yet effective Attention based Encoder-Decoder architecture. Empirical studies on challenging locomotion environments demonstrate O2V's efficiency: under widely used configuration, with merely 15.8% parameters, O2V achieves SOTA-level performance on all finite horizon and transfer learning environments. Moreover, O2V significantly outperforms all baselines on infinite horizon environments while exhibiting smaller variance, faster convergence and interpretability.

1 Introduction

The option framework [27] is one of the most promising framework to extend RL methods to lifelong learning agents [20] and has been proved beneficial in speeding learning [2], improving exploration [11], and facilitating transfer learning [31]. However, conventional options tend to over-complicate methods in ways that are less suited to leveraging computation [2].

The first deficiency is that the option framework is developed on Semi-Markov Decision Process, we refer to this as *SMDP-Option*. In *SMDP-Option*, an option is a temporally abstracted action whose execution cross a various amount of time steps. *SMDP-Option* has been identified [31] as sample inefficient and unstable to optimize. Since RL is notoriously sample expensive and hyper-parameters sensitive [10], the SMDP formulation severely impair options' applicability in broader context [14].

We address this issue by first proving that *SMDP-Option* has a sample efficient Markov Decision Process equivalence (*bisimulation relation* [9]), we refer to this as *MDP-Option*. In RL [19, 31] and Imitation Learning [12, 18, 25, 26] areas, similar formulations have been employed as "one-step option" [12, 31], such approximations either drop the dependency on $\mathbf{o}_{t-1}$ (option executed from last steps) thus lose the temporal abstraction functionality, or use an inaccurate value function to update policies. Instead, we are the first identify the issue that the conventional *Value Function* $V[\mathbf{s}_t]$ no longer yields the Bellman equation [27] under the one-step setting, and preserve the temporal abstraction by proposing a novel *Markovian Option-Value Function* $\bar{V}[\mathrm{s}_t, \mathbf{o}_{t-1}]$, which is an unbiased estimation of $V[\mathrm{s}_t]$ and its variance is up-bounded by $V[\mathrm{s}_t]$, and derive the novel Bellman equation for *MDP-Option*. Based on the Bellman equation, we not only prove the equivalence to *SMDP-Option*, but also derive policy gradient theorems for learning *MDP-Option*. As a result, *MDP-Option*

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

is a general-purpose MDP which can be combined with any MDP-style [31] policy optimization algorithms (such as PPO [30]) off-the-shelf.

The second deficiency of *SMDP-Option* is that it is extremely expensive to learn and scale up. Each option is represented as a triple containing three components: one *intra-option policy*, one *termination function*, and one initiation set. Learning options is amenable to learning local representations [2] on a statistical manifold [1]. As pointed out by Bacon [2] (Chapter 3.6), local representations do a poor job at representing knowledge compactly and require more samples than distributed representations.

In this paper, we make the first attempt to learn options with embeddings (distributed representations [13]). Distributed representations have proved its efficacy in representing entities and played a central role in recent advances of large-scale frameworks in both CV [8, 17] and NLP [5, 6, 28] areas. As shown in Section 4, *MDP-Option* naturally gives rise to representing each option as a single embedding vector and the option space as an ambient space of the state space. Therefore, options defined in *MDP-Option* combine temporal abstraction together with state abstraction [16]. To learn option embeddings, we propose *Option2Vec* (O2V) architecture, a simple yet effective Attention [28] based Encoder-Decoder architecture. Complexities of learning option distributions and classification hyperplanes on statistical manifold are simplified as an efficient clustering mechanism over option embedding centroids on a homeomorphic parametric space [1]. We illustrate this difference between *SMDP-Option* and *MDP-Option* in Figure 1.

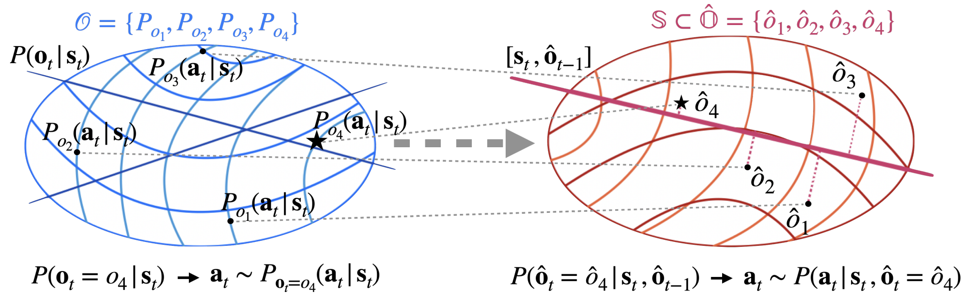

Figure 1: Illustrations of *SMDP-Option* Classification on Statistical Manifold v.s. *MDP-Option* Clustering on Parametric Space. On Statistical Manifold, for M options there are M action policies to learn. Selecting options is analogously learning classification hyperplanes. On Parametric Space, for M options there are M embedding centroids to learn yet all embeddings share a single action policy (decoder). Selecting options is analogously assigning the closest centroid to the hyperplane $[\mathbf{s}_t, \mathbf{o}_{t-1}]$.

It is worth to point out that, due to space limitation we have to solely focus on proposing *MDP-Option* and O2V, and designing experiments to address that O2V achieves at least the same performance as *SMDP-Option*. Since *MDP-Option* is equivalent to *SMDP-Option*, it still shares many identified limitations such as "the dominant skill problem" [29, 31] as identified in Section **??**. As briefly discussed in Appendix **??**, *MDP-Option* actually gives rise to efficient solutions to many identified limitations of *SMDP-Option* yet have to be deferred to our future works. Our main contributions are: (1) proposing the *MDP-Option*, a MDP equivalence of *SMDP-Option*, and developing the bellman equation and gradient theorems; (2) proposing option embedding vectors, the first distributed representations of options; (3) proposing the first Attention [28] based Encoder-Decoder architecture, the Option2Vec (O2V) architecture, for learning options with much better scalability, smaller variance and faster convergence; (4) demonstrating option embeddings are interpretable, which is a key property for developing real-world RL applications (e.g. ensuring safety for human).

2 Background

Markov Decision Process: A Markov Decision Process [22] $M = \{\mathbb{S}, \mathbb{A}, R, P, \gamma\}$ consists of a state space $\mathbb{S}$, an action space $\mathbb{A}$, a state transition function $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) : \mathbb{S} \times \mathbb{A} \to \mathbb{S}$, a discount factor $\gamma \in \mathbb{R}$, and a reward function $R(\mathbf{s}, \mathbf{a}) = \mathbb{E}[r|\mathbf{s}, \mathbf{a}] : \mathbb{S} \times \mathbb{A} \to \mathbb{R}$ which is the expectation of the reward $r_{t+1} \in \mathbb{R}$ received from the environment after executing action $\mathbf{a}_t$ at state $\mathbf{s}_t$. A policy $\pi = P(\mathbf{a}|\mathbf{s}) : \mathbb{A} \times \mathbb{S} \to [0, 1]$ is a probability distribution defined over actions conditioning on states. A discounted return is defined as $G_t = \sum_k^N \gamma^k r_{t+k+1}$, where $\gamma \in (0, 1)$ is a discounting factor. The value function $V[\mathbf{s}_t] = \mathbb{E}_{\tau \sim \pi}[G_t|\mathbf{s}_t]$ is the expected return starting at state $\mathbf{s}_t$ and the trajectory $\tau = \{\mathbf{s}_t, \mathbf{a}_t, r_{t+1}, \mathbf{s}_{t+1}, \dots\}$ follows policy π thereafter. The action-value function is defined as $Q[\mathbf{s}_t, \mathbf{a}_t] = \mathbb{E}_{\tau \sim \pi}[G_t|\mathbf{s}_t, \mathbf{a}_t]$.

Bisimulation Relation: Given two processes $M = \{\mathbb{S}, \mathbb{A}, R, P, \gamma\}$ with the trajectory τ and $\tilde{M} = \{\tilde{\mathbb{S}}, \mathbb{A}, \tilde{R}, \tilde{P}, \tilde{\gamma}\}$ with the trajectory $\tilde{\tau}$. Assume both M and $\tilde{M}$ share the same action space $\mathbb{A}$. The equivalence relation between M and $\tilde{M}$ is defined by Givan et al. [9]. An equivalence relation $\tilde{B} : \tilde{\mathbb{S}} \to \mathbb{S}$ is a *Bisimulation Relation* if 1) for any state $\mathbf{s}$, there exists an *one-to-one correspondence* equivalent state $\tilde{\mathbf{s}}$ that $\mathbf{s}/\tilde{B} = \tilde{\mathbf{s}}/\tilde{B}$, or denoted as $\tilde{B}(\tilde{\mathbf{s}}) = \mathbf{s}$, 2) and the following conditions hold:

1. $P(\tau/\tilde{B}) \equiv P(\tilde{\tau}/\tilde{B}), \quad$ and $\tilde{B}$ is a *bijection*,

2. $V[\tau/\tilde{B}] \equiv V[\tilde{\tau}/\tilde{B}]$

In this paper, we follow this definition to prove the equivalence relationship.

The SMDP-based Option Framework: In *SMDP-Option* [2, 27], an option is a triple $(\mathbb{I}_o, \pi_o, \beta_o) \in \mathcal{O}$, where $\mathcal{O}$ denotes the option set; the subscript $o \in \mathbb{O} = \{1, 2, \ldots, K\}$ is a positive integer index which denotes the oth 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 reasons, we use $P_o(\mathbf{b} = 1|\mathbf{s})$ instead of β_o which is widely used in previous option literatures (e.g. [3, 27]).

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. Note that we use the bold-case $\mathbf{o}$ to denote unrealized random variables and the light-italic-case o to denote a realized instantiation. Conventionally, the execution of an option employs the call-and-return model [27]: at time step t, an agent either continues the previously executed option $\mathbf{o}_{t-1} = o$ with probability $P_o(\mathbf{b} = 0|\mathbf{s})$ and sets $\mathbf{o}_t = \mathbf{o}_{t-1} = o$, or terminates o with probability $P_o(\mathbf{b} = 1|\mathbf{s})$ and samples a new option $\mathbf{o}_t$ from the master policy $P(\mathbf{o}_t|\mathbf{s}_t)$. Therefore, the dynamics (stochastic process) of the option framework is written as:

$$
\begin{aligned}
P(\tau) = P(\mathbf{s}_0)P(\mathbf{o}_0)P_{o_0}(\mathbf{a}_0|\mathbf{s}_0) \prod_{t=1}^{\infty} P(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{a}_{t-1}) P_{o_t}(\mathbf{a}_t|\mathbf{s}_t) \\
[P_{o_{t-1}}(\mathbf{b}_t = 0|\mathbf{s}_t)\mathbf{1}_{\mathbf{o}_t = o_{t-1}} + P_{o_{t-1}}(\mathbf{b}_t = 1|\mathbf{s}_t)P(\mathbf{o}_t|\mathbf{s}_t)]. \qquad (1)
\end{aligned}
$$

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 [27].

3 MDP Equivalences of the SMDP-based Option Framework

We prove that *MDP-Option* is equivalent to *SMDP-Option* under the definition of *bisimulation* [9]. To derive Bellman equation for *MDP-Option*, we develop a novel *Markovian skill-value function* $\bar{V}[\mathbf{s}_t, \mathbf{o}_t]$, which is an unbiased estimation of the conventional value function $V[\mathbf{s}_t]$ and the variance of $\bar{V}[\mathbf{s}_t, \mathbf{o}_t]$ is up-bounded by $V[\mathbf{s}_t]$. Based on Bellman equation, policy gradient theorems for *MDP-Option* are then derived. As a result, *MDP-Option* is a general-purpose MDP which can be combined with any policy optimization algorithm off-the-shelf.

In this section, we propose *MDP-Option*, a simple yet effective option-induced MDP and prove its equivalence (as shown in Figure 2) to *SMDP-Option*. For clarity, in Section 3.1 we first prove an intermediate equivalence *MDP-Mixture* to bridge the equivalence between *SMDP-Option* and *MDP-Option*. Based on *MDP-Mixture*, in Section 3.2 we propose the *MDP-Option*, a marginalized variation of the *SMDP-Option*. *MDP-Option* uses the *skill policy* (Eq. 5), which is a marginal distribution, to replace the *master policy* and *termination function*. In order to derive *MDP-Option*'s Bellman equation, we propose the novel *Markovian skill-value function* (Eq. 6) and prove that it is an unbiased estimation of the conventional value function and its variance is up-bounded by the conventional value function. Policy gradient theorems for *MDP-Option* are then derived basing on the Bellman equation. In Section 4, we propose O2V, which is an implementation of the *MDP-Option* by employing the Embedding and Attention [28] techniques.

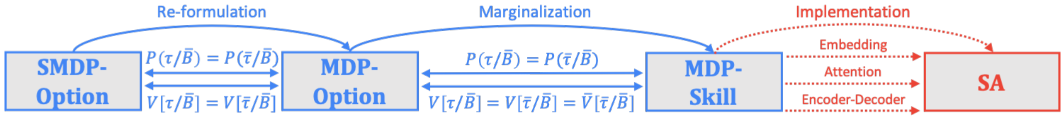

Figure 2: Paper Structure?. Blue terms are Decision Processes. Solid arrows indicate that their equivalences are theoretically justified. O2V is an Architecture which implements the MDP-Option by employing Embedding, Attention, and Encoder-Decoder techniques. Dashed connections indicate they are architecture choices.

3.1 The MDP Equivalence (MDP-Mixture) of the Option Framework

With the definitions of *SMDP-Option* in hand, we now show how to reformulate it into an MDP-based equivalence *MDP-Mixture*. The first reformulation is that we follow Bishop [4]'s formulation of mixture distributions and redefine the option random variable $\mathbf{o} \in \mathbb{O} = \{1, 2, \ldots, K\}$, which was originally defined as an integer index, but now as a K-dimensional one-hot vector $\bar{\mathbf{o}} \in \bar{\mathbb{O}} = \{0, 1\}^K$ where K is the number of options. The second reformulation is that we exploit the one-hot vector to reformulate the *termination function* and action function of each option into two mixture distributions by introducing extra dependencies on $\bar{\mathbf{o}}$:

$$P(\mathbf{a}_t|\mathbf{s}_t, \bar{\mathbf{o}}_t) = \prod_{o \in \bar{\mathbf{o}}_t} P_o(\mathbf{a}_t|\mathbf{s}_t)^o, \qquad P(\mathbf{b}_t|\mathbf{s}_t, \bar{\mathbf{o}}_{t-1}) = \prod_{o \in \bar{\mathbf{o}}_{t-1}} P_o(\mathrm{b}_t|\mathbf{s}_t)^o \tag{2}$$

Since the option random variable $\bar{\mathbf{o}}$ is now a one-hot vector, for $\bar{\mathbf{o}}_t = o_t$, by definition only the entry $o_t = 1$ and all the other entry $o \in \mathbb{O} - \{o_t\} = 0$. Therefore, we have $P_{o_t}(\mathbf{a}_t|\mathbf{s}_t) = P(\mathbf{a}_t|\mathbf{s}_t, \bar{\mathbf{o}}_t = o_t)$ and $\beta_{o_{t-1}} = P_{o_{t-1}}(\mathbf{b}_t = 1|\mathbf{s}_t) = P(\mathbf{b}_t = 1|\mathbf{s}_t, \bar{\mathbf{o}}_{t-1} = o_{t-1})$.

The third reformulation is that we propose a novel *MDP mixture master policy* $P(\bar{\mathbf{o}}_t|\mathbf{s}_t, \mathbf{b}_t, \bar{\mathbf{o}}_{t-1})$, which is a mixture distribution containing the *SMDP master policy* and a degenerate probability as mixture components by adding two extra dependencies on $\mathbf{b}_t$ and $\bar{\mathbf{o}}_{t-1}$:

$$P(\bar{\mathbf{o}}_t|\mathbf{s}_t, \mathbf{b}_t, \bar{\mathbf{o}}_{t-1}) = P(\bar{\mathbf{o}}_t|\mathbf{s}_t)^{\mathrm{b}_t} P(\bar{\mathbf{o}}_t|\bar{\mathbf{o}}_{t-1})^{1-\mathrm{b}_t}, \tag{3}$$

where the indicator function $\mathbf{1}_{\mathbf{o}_t = o_{t-1}}$ used in Eq.1 is now redefined as a degenerate probability distribution [22]:

$$P(\bar{\mathbf{o}}_t|\bar{\mathbf{o}}_{t-1}) = \begin{cases} 1 & \text{if } \bar{\mathbf{o}}_t = \bar{\mathbf{o}}_{t-1}, \\ 0 & \text{if } \bar{\mathbf{o}}_t \neq \bar{\mathbf{o}}_{t-1}. \end{cases}$$

We define a function $\bar{B}(\bar{\mathbf{o}}) = \bar{\mathbf{o}} \cdot \mathbf{d}^T : \bar{\mathbb{O}} \to \mathbb{O}$ which maps $\bar{\mathbf{o}}$ to $\mathbf{o}$, where $\mathbf{d} = [1, 2, \ldots, K]^T$ is a K-dimensional constant integer vector and hence $\bar{B}(\bar{\mathbf{o}}) = \mathbf{o}$. Note that $\bar{B}$ is a *Bijection* since it is a linear function defined on a finite integer space. Therefore, by following the definition of *Bisimulation Relation*, the dynamics of the *SMDP-Option* in Eq.1 under the Bijection $\bar{B}$ can be reformulated as:

$$\begin{aligned} P(\tau/\bar{B}) = P(\bar{\tau}/\bar{B}) = P(\mathbf{s}_0)P(\bar{\mathbf{o}}_0)P(\mathbf{a}_0|\mathbf{s}_0, \bar{\mathbf{o}}_0) \prod_{t=1}^{\infty} P(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{a}_{t-1})P(\mathbf{a}_t|\mathbf{s}_t, \bar{\mathbf{o}}_t) \\ \sum_{\mathbf{b}_t} P(\mathbf{b}_t|\mathbf{s}_t, \bar{\mathbf{o}}_{t-1})P(\bar{\mathbf{o}}_t|\mathbf{b}_t, \mathbf{s}_t, \bar{\mathbf{o}}_{t-1}) \end{aligned} \tag{4}$$

where $\bar{\tau} = \{\mathbf{s}_0, \bar{\mathbf{o}}_0, \mathbf{a}_0, \mathbf{s}_1, \bar{\mathbf{o}}_1, \mathbf{a}_1, \ldots\}$ is the trajectory of the *MDP-Mixture*.

With $P(\tau/\bar{B}) = P(\bar{\tau}/\bar{B})$ in hand, to prove the equivalence between the *SMDP-Option* and *MDP-Mixture*, we move on to prove both of them share the same expected reward. This is non-trivial since compared to the *SMDP-Option*, the MDP formulation introduces extra dependencies on $\bar{\mathbf{o}}$ and $\mathbf{b}$ in Eq.4 as described above. However, in Appendix **??**, by exploiting conditional independencies we prove that they do have the same expected return under the Bijection $\bar{B}$. Therefore, the SMDP-based option framework has an MDP-based equivalence:

Theorem 3.1. *By the definition of Bisimulation Relation, the SMDP-based option framework, which employs Markovian options, has an underlying MDP equivalence because:*

1. $P(\tau/\bar{B}) = P(\bar{\tau}/\bar{B})$ (Eq. 4) and $\bar{B}$ is a Bijection.

2. $V[\tau/\bar{B}] = V[\bar{\tau}/\bar{B}]$ (Proofs in Appendix **??**).

3.2 The Option2Vec induced Markov Decision Problem (MDP-Skill)

In this section, we define the option-induced MDP, *MDP-Option*, prove the equivalence between *MDP-Option* and *SMDP-Option*, and derive policy gradient theorems for *MDP-Option*. Although the mixture master policy in Eq.4 is MDP-formulated, the master policy $P(\bar{\mathbf{o}}_t|\mathbf{s}_t)$ as a mixture component in it is still SMDP-formulated hence cannot be updated by MDP-based algorithms. The beauty of *MDP-Option* is that it addresses this issue in a natural and simple way: notice the marginalization over the termination variable $\mathbf{b}_t$ in Eq. 4: $\sum_{\mathbf{b}_t} P(\mathbf{b}_t|\mathbf{s}_t, \bar{\mathbf{o}}_{t-1})P(\bar{\mathbf{o}}_t|\mathbf{b}_t, \mathbf{s}_t, \bar{\mathbf{o}}_{t-1})$, *MDP-Option* uses the *skill policy* $P(\bar{\mathbf{o}}_t|\mathbf{s}_t, \bar{\mathbf{o}}_{t-1})$ to model this marginal distribution explicitly:

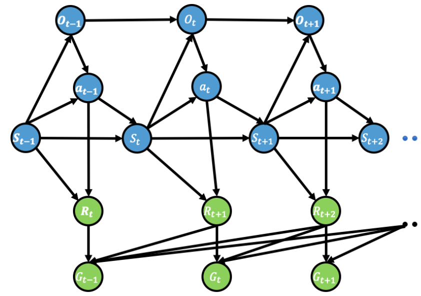

Figure 3: Graphical Model of *MDP-Option*

$$P(\bar{\tau}) = P(\mathbf{s}_0)P(\bar{\mathbf{o}}_0)P(\mathbf{a}_0|\mathbf{s}_0, \bar{\mathbf{o}}_0)\prod_{t=1}^{\infty} P(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{a}_{t-1})P(\mathbf{a}_t|\mathbf{s}_t, \bar{\mathbf{o}}_t)P(\bar{\mathbf{o}}_t|\mathbf{s}_t, \bar{\mathbf{o}}_{t-1}) \tag{5}$$

Therefore, *MDP-Option* shares the same trajectory with the *MDP-Mixture* while the *skill policy* can be updated by any MDP-based algorithms. It is natural to ask that how *MDP-Option* temporally extends an option without the *termination function*. In fact, the *skill policy* captures temporal relationships between options explicitly: it selects the new option $\bar{\mathbf{o}}_t$ by choosing the one which has the closest distance to the current state $\mathbf{s}_t$, while has a tendency to continue the executed option o from the last time step $\bar{\mathbf{o}}_{t-1} = o$. We defer this topic to Section 4 and focus this section on proposing *MDP-Option*.

Since the *skill policy* $P(\bar{\mathbf{o}}_t|\mathbf{s}_t, \bar{\mathbf{o}}_{t-1})$ introduces one extra dependency on $\bar{\mathbf{o}}_{t-1}$, conventional Bellman equation which is derived by following the conventional value function $V[\mathbf{s}_t]$ no longer applies to *MDP-Option*. In order to derive the Bellman equation of *MDP-Option*, we propose the novel *Markovian skill-value function*, value functions with Markov dependencies (such as $\bar{\mathbf{o}}_{t-1}$). Specifically, rather than use the conventional value function $V[\mathbf{s}_t]$, we define the *Markovian skill-value function* as $\bar{V}[\mathbf{s}_t, \bar{\mathbf{o}}_{t-1}]$ (derivations in Appendix **??**):

$$\bar{V}[\mathbf{s}_t, \bar{\mathbf{o}}_{t-1}] = \mathbb{E}[G_t|\mathbf{s}_t, \bar{\mathbf{o}}_{t-1}] = \sum_{\bar{\mathbf{o}}_t} P(\bar{\mathbf{o}}_t|\mathbf{s}_t, \bar{\mathbf{o}}_{t-1})Q_O[\mathbf{s}_t, \bar{\mathbf{o}}_t]. \tag{6}$$

where the *skill value function* $Q_O[\mathbf{s}_t, \bar{\mathbf{o}}_t]$ can then be derived as (derivations in Appendix **??**):

$$Q_O[\mathbf{s}_t, \bar{\mathbf{o}}_t] = \mathbb{E}[G_t|\mathbf{s}_t, \bar{\mathbf{o}}_t] = \sum_{\mathbf{a}_t} P(\mathbf{a}_t|\mathbf{s}_t, \bar{\mathbf{o}}_t)Q_A[\mathbf{s}_t, \bar{\mathbf{o}}_t, \mathbf{a}_t], \tag{7}$$

where the *skill-action value function* $Q_A[\mathbf{s}_t, \bar{\mathbf{o}}_t, \mathbf{a}_t]$ can then be derived as (Appendix **??**):

$$\begin{aligned} Q_A[\mathbf{s}_t, \bar{\mathbf{o}}_t, \mathbf{a}_t] &= \mathbb{E}[G_t|\mathbf{s}_t, \bar{\mathbf{o}}_t, \mathbf{a}_t] \\ &= r(s, a) + \gamma \sum_{\mathbf{s}_{t+1}} P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\bar{V}[\mathbf{s}_{t+1}, \bar{\mathbf{o}}_t], \end{aligned} \tag{8}$$

Expanding $\bar{V}[\mathbf{s}_{t+1}, \bar{\mathbf{o}}_t]$ in Eq. 8 through Eq. 6 to 8 gives *MDP-Option*'s Bellman equation. As in Section 3.1, we can now move on to prove the equivalence between the *MDP-Option* and *MDP-Mixture* hence to *SMDP-Option*. Specifically, in Appendix **??** we prove that:

Proposition 3.2. *$\bar{V}[\mathbf{s}_t, \bar{\mathbf{o}}_{t-1}]$ is an unbiased estimation of $V[\mathbf{s}_t]$.*

Therefore, the SMDP-based option framework and *MDP-Option* are equivalent under the Bijection $\bar{B}$:

Theorem 3.3. *By the definition of Bisimulation Relation,* MDP-Option *is equivalent to the SMDP-based option framework because:*

1. $P(\tau/\bar{B}) = P(\bar{\tau}/\bar{B})$ (Eq. 4 and 5),

2. $V[\tau/\bar{B}] = V[\bar{\tau}/\bar{B}] \equiv \bar{V}[\bar{\tau}/\bar{B}]$ (follows directly from Theorem 3.1 and Proposition 3.2).

Other than an unbiased estimation of $V[\mathbf{s}_t]$, in Appendix **??** we also prove that

Proposition 3.4. *The variance of $\bar{V}[\mathbf{s}_t, \bar{\mathbf{o}}_{t-1}]$ is up-bounded by $V[\mathbf{s}_t]$.*

which means that $\bar{V}[\mathbf{s}_t, \bar{\mathbf{o}}_{t-1}]$ also has a variance-reduction effect compared to the conventional value function. This property is empirically witnessed in Section **??** and further discussed in Appendix **??**.

With the Bellman equation, we now are able to derive policy gradient theorems for *MDP-Option*. To keep notations uncluttered, we use $\theta_{\bar{o}}$ to denote *skill policy*'s parameters $P(\bar{\mathbf{o}}_t|\mathbf{s}, \bar{\mathbf{o}}_{t-1}; \theta_{\bar{o}})$ and θ_a to denote action policy's parameters $P(\mathbf{a}_t|\mathbf{s}_t, \bar{\mathbf{o}}_t; \theta_a)$. Policy gradient theorems of *MDP-Option* are:

Theorem 3.5. ***Skill Policy Gradient Theorem:*** *Given a stochastic* skill policy *differentiable in its parameter vector $\theta_{\bar{o}}$, the gradient of the expected discounted return with respect to $\theta_{\bar{o}}$ is:*

$$\frac{\partial \bar{V}[\mathbf{s}_t, \bar{\mathbf{o}}_{t-1}]}{\partial \theta_{\bar{o}}} = \mathbb{E}[\ \frac{\partial P(\bar{\mathbf{o}}'|\mathbf{s}', \bar{\mathbf{o}})}{\partial \theta_{\bar{o}}} Q_O[\mathbf{s}', \bar{\mathbf{o}}'] \mid \mathbf{s}_t, \bar{\mathbf{o}}_{t-1}], \tag{9}$$

where $\bar{\mathbf{o}}'$ is one time step later than $\bar{\mathbf{o}}$.

Theorem 3.6. ***Action Policy Gradient Theorem:*** *Given a stochastic action policy differentiable in its parameter vector θ_a, the gradient of the expected discounted return with respect to θ_a is:*

$$\frac{\partial Q_O[\mathbf{s}_t, \bar{\mathbf{o}}_t]}{\partial \theta_a} = \mathbb{E}[\ \frac{\partial P(\mathbf{a}|\mathbf{s}, \bar{\mathbf{o}})}{\partial \theta_a} Q_A[\mathbf{s}, \bar{\mathbf{o}}, \mathbf{a}] \mid \mathbf{s}_t, \bar{\mathbf{o}}_t]. \tag{10}$$

Proof. See Appendix **??** □

4 The Option2Vec Architecture (O2V)

To overcome these difficulties, we significantly improves *SMDP-Option*'s scalability by redefining options as distributed representations. In order to achieve this, we first need to propose a novel option-induced Markov Decision Process (MDP), the *MDP-Option*. The *MDP-Option* is equivalent to *SMDP-Option* under the definition of *bisimulation* [9], and compatible with any sample efficient MDP-style policy optimization algorithms (e.g. PPO [30]). The *MDP-Option* enables simultaneously maintaining the equivalence to *SMDP-Option* while encoding all local information (where to initiate, what actions to emit and when to terminate) of an option altogether into an option embedding $\hat{o}$ (distributed representation). Option embeddings are essentially clustering centroids on an Euclidean parametric space [1] which is homeomorphic to the statistical manifold. Because distances between real vectors (embeddings) are trivial to calculate, complexities of learning options and classification hyperplanes are simplified as a clustering problem over option embedding centroids.

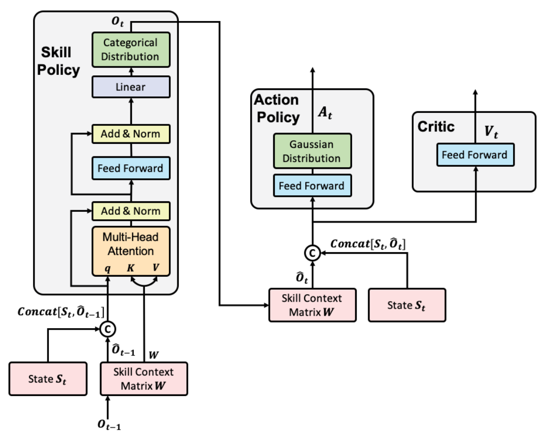

Figure 4: The Option2Vec Architecture

We propose the *Option2Vec* (O2V) architecture, a simple yet effective Attention [28] based Encoder-Decoder architecture to implement such mechanism. Specifically, an *option policy* $P(\hat{\mathbf{o}}_t|\mathbf{s}_t, \hat{\mathbf{o}}_{t-1} = \hat{o})$ is employed as an encoder: given a hyperplane $[\mathbf{s}_t, \hat{o}]$, the *option policy* simply calculates whichever cluster centroid $\hat{o}^*$ is closest to the hyperplane and assigns it to $\hat{\mathbf{o}}_t = \hat{o}^*$. Since a vector is closest to itself, the *option policy* has a natural tendency to temporally extend the option $\hat{o}$ executed from last step. Under this formulation, option embeddings combine advantages of both temporal abstraction [27] and state abstraction [16] on the ambient space of state space $\mathbb{S}$ and option space $\mathbb{O}$. All *option embeddings* share a single decoder, the *action policy* $P(\mathbf{a}_t|\mathbf{s}_t, \hat{\mathbf{o}}_t)$, which decodes an embedding vector $\hat{\mathbf{o}}_t$ into concrete actions $\mathbf{a}_t$. As a result, adding a new option in O2V is as cheap as adding an embedding vector, and regardless the number of options are learned, O2V only needs to approximate two distributions (option policy and action policy). Empirical studies on challenging locomotion environments demonstrate that the *MDP-Option* and O2V exhibit better scalability, smaller variance, faster convergence and interpretability.

With the *MDP-Option* in hand, we can finally move on to propose the Option2Vec Architecture (O2V). As mentioned in Section 3.2, one major change *MDP-Option* has been made is that it marginalizes the *mixture master policy* and *termination function* away, and models their marginal distribution, the *skill policy*, directly. In this section, we demonstrate how O2V temporally extends skills in the absence of *termination function* by implementing *MDP-Option* with much more scalable and effective Embedding and the phenomenal Attention mechanisms [28].

First, one-hot to embedding. one-hot waste, embedding nlp good. by to embedding, we can: representation good; option is by two neural nets, scalability 2 nets -> 1 vector. One decoder to decode all. Decoder can be trained all-time, more sample efficiency.

1. How options encode temporal abstraction (what is temporal abstraction), why is this bad?

Integer Index: integer categories. For example, the identity of option i out of |O| options can be represented by the

How to describe option's architecture in one term? 1 option 2 nets

2. What is distributed representation? How is skill embedded in distributed representation?

3. Why embedding good?

gain even more generality and expressivity through a distributional shift: viewing the identity of an option as property of the pattern of activation in its vector-valued representation.

4. How to temporal extension a skill? Why MHA good? MHA »> termination why?

5. WHy Encoder-Decoder? Save parameters parameter sharing.

5 Related Works

As a result, most option variants compromise to only two-level SMDP because the number of options grows exponentially with levels [23]; the initiation set is widely ignored because of difficulties in learning it from data [15]. Moreover, learning switching between options is analogously learning classification hyperplanes on the statistical manifold, in a linear case, for N hyperplanes, theoretically there are 2^N options to be learned with [20].

We must appreciate that Bacon [2] (Chapter 3.5 and 3.6) first conceptually discussed the possibility of introducing the skill policy and distributed representations into the option framework. However, to the best of our knowledge, this is the first concrete work that discovers and proves the MDP equivalence of the SMDP-Option, and enables learning options as distributed representations. Although sharing similar formulations with [2], our work is motivated by causal reinforcement learning [7, 21] and capsule networks [24] (more details in Appendix **??**) and is developed independently from [2].

6 Conclusions

In this paper, we presented a novel MDP equivalence of the SMDP formulated option framework, from which an MDP implementation of the option framework, i.e., the Option2Vec architecture, was derived. We theoretically proved that O2V has lower variance than conventional RL models and provided policy gradient theorems for updating O2V. Our empirical studies on challenging infinite horizon robot simulation environments demonstrated that O2V not only outperforms all baselines by a large margin, but also exhibits smaller variance, faster convergence, and good interpretability. On transfer learning, O2V also outperforms the other models in 5 out of 6 environments and shows its advantages in knowledge reuse tasks.

The final and most important contribution of O2V is hierarchically learning explicit abstract actions' representations with "skill context vectors". This design significantly improves the scalability and interpretability of O2V. It is straightforward to extend O2V to deeper and wider (Appendix **??**) architectures, which gives rise to a large-scale pre-training and transfer learning architecture in the reinforcement learning area.

References

[1] Amari, S.-i. Differential geometrical theory of statistics. *Amari et al. ABNK+87*, pp. 19–94, 1987.

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

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

[4] Bishop, C. M. *Pattern recognition and machine learning*. springer, 2006.

[5] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[6] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Doshi-Velez, F. and Konidaris, G. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *IJCAI: proceedings of the conference*, volume 2016, pp. 1432. NIH Public Access, 2016.

[8] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Givan, R., Dean, T., and Greig, M. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 147(1-2):163–223, 2003.

[10] 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.

[11] 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.

[12] Henderson, P., Chang, W.-D., Bacon, P.-L., Meger, D., Pineau, J., and Precup, D. Optiongan: Learning joint reward-policy options using generative adversarial inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[13] Hinton, G. E. et al. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, pp. 12. Amherst, MA, 1986.

[14] Jong, N. K., Hester, T., and Stone, P. The utility of temporal abstraction in reinforcement learning. In *AAMAS (1)*, pp. 299–306. Citeseer, 2008.

[15] 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. 4444–4451, 2020.

[16] Knoblock, C. A. Learning abstraction hierarchies for problem solving. In *AAAI*, pp. 923–928, 1990.

[17] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[18] Lee, S.-H. and Seo, S.-W. Learning compound tasks without task-specific knowledge via imitation and self-supervised learning. In *International Conference on Machine Learning*, pp. 5747–5756. PMLR, 2020.

[19] 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.

[20] Mankowitz, D. J., Mann, T. A., and Mannor, S. Adaptive skills, adaptive partitions (asap). *arXiv preprint arXiv:1602.03351*, 2016.

[21] Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pp. 947–1012, 2016.

[22] Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

[23] Riemer, M., Liu, M., and Tesauro, G. Learning abstract options. In *Advances in Neural Information Processing Systems*, pp. 10424–10434, 2018.

[24] Sabour, S., Frosst, N., and Hinton, G. E. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pp. 3856–3866, 2017.

[25] Shankar, T. and Gupta, A. Learning robot skills with temporal variational inference. In *International Conference on Machine Learning*, pp. 8624–8633. PMLR, 2020.

[26] Sharma, A., Sharma, M., Rhinehart, N., and Kitani, K. M. Directed-info gail: Learning hierarchical policies from unsegmented demonstrations using directed information. *arXiv preprint arXiv:1810.01266*, 2018.

[27] 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.

[28] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[29] Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pp. 3540–3549. PMLR, 2017.

[30] Witoonchart, P. and Chongstitvatana, P. Application of structured support vector machine backpropagation to a convolutional neural network for human pose estimation. *Neural Networks*, 92:39–46, 2017.

[31] Zhang, S. and Whiteson, S. Dac: The double actor-critic architecture for learning options. In *Advances in Neural Information Processing Systems*, pp. 2012–2022, 2019.