Journal of Artificial Intelligence Research 53 (2015) 375-438 Submitted 01/15; published 07/15

Approximate Value Iteration with Temporally Extended Actions

Timothy A. Mann MANN@EE.TECHNION.AC.IL
Shie Mannor SHIE@EE.TECHNION.AC.IL
Electrical Engineering
The Technion - Israel Institute of Technology,
Haifa, Israel

Doina Precup DPRECUP@CS.MCGILL.CA
School of Computer Science
McGill University,
Montreal, QC, H3A2A7, Canada

Abstract

Temporally extended actions have proven useful for reinforcement learning, but their duration also makes them valuable for efficient planning. The options framework provides a concrete way to implement and reason about temporally extended actions. Existing literature has demonstrated the value of planning with options empirically, but there is a lack of theoretical analysis formalizing when planning with options is more efficient than planning with primitive actions. We provide a general analysis of the convergence rate of a popular Approximate Value Iteration (AVI) algorithm called Fitted Value Iteration (FVI) with options. Our analysis reveals that longer duration options and a pessimistic estimate of the value function both lead to faster convergence. Furthermore, options can improve convergence even when they are suboptimal and sparsely distributed throughout the state-space. Next we consider the problem of generating useful options for planning based on a subset of landmark states. This suggests a new algorithm, Landmark-based AVI (LAVI), that represents the value function only at the landmark states. We analyze both FVI and LAVI using the proposed landmark-based options and compare the two algorithms. Our experimental results in three different domains demonstrate the key properties from the analysis. Our theoretical and experimental results demonstrate that options can play an important role in AVI by decreasing approximation error and inducing fast convergence.

1. Introduction

We consider the problem of planning in Markov Decision Processes (MDPs; Puterman, 1994, see Section 2) with large or even infinite state-spaces. In this setting, traditional planning algorithms, such as Value Iteration (VI) and Policy Iteration (PI), are intractable because the computational and memory complexities at each iteration scale (polynomially and linearly, respectively; Littman, Dean, & Kaelbling, 1995) with the number of states in the target MDP. Approximate Value Iteration (AVI) algorithms are more scalable than VI, because they compactly represent the value function (Bertsekas & Tsitsiklis, 1996). This allows AVI algorithms to achieve per iteration computational and memory complexities that are independent of the size of the state-space. However, there are many challenges to using AVI algorithms in practice. AVI and VI often need many iterations to solve the MDP (Munos & Szepesvári, 2008). It turns out that temporally extended actions can play an important role in reducing the number of iterations.

The options framework defines a unified abstraction for representing both temporally extended actions and primitive actions (Sutton, Precup, & Singh, 1999). When an option is initialized, it immediately selects a primitive action (or lower-level option) to execute but does not return control to the agent. Then, on each following timestep, the option tests whether it should return control to the agent that called the option or continue by selecting another primitive action (or lower-level option). Because they can represent temporally extended actions, options provide a valuable tool for efficient planning (Sutton et al., 1999; Silver & Ciosek, 2012). Under most analyses of AVI, one iteration corresponds to planning one additional timestep into the future. On the other hand, by performing a single iteration of AVI with temporally extended actions, one iteration could instead correspond to planning several timesteps into the future. We derive bounds that help us reason about when AVI with temporally extended actions converges faster than AVI with only primitive actions.

The options framework is appealing for investigating planning with temporally extended actions. For one thing, the class of options includes both primitive actions as well as a wide range of temporally extended actions, and many of the well-known properties of Markov Decision Processes generalize when arbitrary options are added (e.g., Value Iteration and Policy Iteration still converge, Precup & Sutton, 1997; Precup, Sutton, & Singh, 1998; Sutton et al., 1999). In addition, much effort has gone into algorithms that learn "good" options for exploration (Iba, 1989; Stolle & Precup, 2002; Mannor, Menache, Hoze, & Klein, 2004; Konidaris & Barto, 2007). These algorithms may produce options that are also useful for planning. Lastly, options allow for greater flexibility when modeling problems where actions do not have the same temporal resolution. For example, in inventory management problems where placing orders may not occur at regular intervals (Minner, 2003) or in the RoboCup Keepaway domain where agent's only make decisions when they have control of a soccer ball (Stone, Sutton, & Kuhlmann, 2005). Thus, options are an important candidate for investigating planning with temporally extended actions.

1.1 Motivation

Ultimately we care about the time it takes to solve an MDP. However, we focus on analyzing the convergence rate of AVI with options, because a faster convergence rate implies a solution with fewer iterations. Using the convergence rate we can determine the total computational cost of planning by bounding the computational cost at each iteration. If the total computational cost with options is smaller than with primitive actions, planning with options is faster than planning with primitive actions.

We focus on the convergence rate because it can provide valuable insight about when planning with options is faster than planning with primitive actions. However, we do not present the full computational complexity of planning with options because the computational cost per iteration is highly domain dependent. Therefore, the convergence rate of planning with options gives us insight about when planning with options may be faster than planning with primitive actions but without getting bogged down in domain specific details. For example, the computational cost of an iteration depends on (1) the computational complexity of simulating an option, (2) the computational complexity of the value function approximation method, and (3) the number primitive actions and temporally extended actions that can be initialized at each state.

For the sake of clarity, we discuss how each of these factors can impact the computational complexity of AVI.

1.1.1 The Cost of Simulating Actions and Options

The computational cost of simulating an option depends on the simulator. We assume for simplicity that all primitive actions can be simulated with approximately the same computational cost. The main question is: What is the compuational cost of simulating temporally extended actions compared to the cost of simulating a primitive actions?

In general, a simulator for primitive actions can be used to simulate options by executing the sequence of primitive actions prescribed by the option. The computational cost of this is equal to the cost of simulating the sequence of primitive actions. However, when the simulator is inexpensive, the simulation costs may be outweighed by the cost of fitting the data with a function approximator. This can be seen in our experiments in section 5.

In some cases, specialized simulators can be constructed so that temporally extended actions have approximately the same cost as primitive actions. For discrete state MDPs, this can be accomplished through a preprocessing step by composing options from primitive actions (Silver & Ciosek, 2012). In large- or continuous-state MDPs, the linear options framework enables the construction of option models by composing models for primitive actions (Sorg & Singh, 2010). In addition, some existing simulators are carefully designed for simulating actions at both long and short timescales (Chassin, Fuller, & Djilali, 2014).

1.1.2 The Cost of Value Function Approximation

The choice of function approximation architecture can have drastic implications on the computational cost of each iteration. Ridge Regression, LASSO, SVR, Neural Networks, etc. all have computational costs that scale with the number of features at varying rates. In some cases, the cost of training a suitable function approximation architecture may be significantly more expensive than the cost of querying the simulator. In these cases, decreasing the number of iterations can result in significant overall computational savings even if options require more queries to the simulator.

1.1.3 The Number of Actions and Options

At each iteration, AVI samples all actions from a collection of states. If there are a large (or even infinite) number of primitive actions, planning can be made both more computationally efficient and sample efficient by planning instead with a smaller number of options.

1.1.4 The Cost of Acquiring Options

One final consideration is the computational cost of acquiring options. If the options are designed in advance by experts, then there is no additional cost. However, if the options are discovered or generated, then this cost should be factored into the total cost of the algorithm. The landmark-based option generation approach proposed in section 5.3.2 has almost no overhead, given a set of "landmark" states. However, more computationally expensive methods for acquiring options (Simsek & Barto, 2004; Mannor et al., 2004) could be justified if the options are reused for planning in many tasks (Fernández & Veloso, 2006).

1.2 Contributions

The main contributions of this paper are the following:

- We propose the Options Fitted Value Iteration (OFVI) algorithm, which is a variant of the popular Fitted Value (or Q-) Iteration (FVI, Riedmiller, 2005; Munos & Szepesvári, 2008; Shantia, Begue, & Wiering, 2011) algorithm with samples generated by options.

- We analyze OFVI in Theorem 1, characterizing the asymptotic loss and the convergence behavior of planning with a given set of options. We give two corollaries specifying how the bound simplifies when: (1) all the options have a minimum duration $d > 1$ (Corollary 1) and (2) the option set contains some long duration options and primitive actions (Corollary 2).

- We introduce a novel method for generating options, based on "landmark" states. This suggests a new algorithm, Landmark-based Approximate Value Iteration (LAVI), that only needs to model the value of a finite set of states rather than the whole value function.

- We analyze the asymptotic loss and convergence behavior of LAVI in Theorem 2 and OFVI with landmark options in Theorem 3. Comparing the bounds of LAVI and OFVI suggests that LAVI may converge faster than OFVI. However, their asymptotic losses are not directly comparable.

- We provide a detailed experimental comparison of FVI with primitive actions, OFVI with hand-coded options, OFVI with landmark options, and LAVI. Our experiments in a domain with realistic pinball-like physics and a complex inventory management problem, demonstrate that LAVI achieves a favorable performance versus time trade-off.

The rest of this paper is organized as follows. Section 2 introduces background on Markov Decision Processes, Dynamic Programming, previous analysis of FVI, Semi-Markov Decision Processes, and options. Section 3 defines the Options Fitted Value Iteration (OFVI) algorithm and compares it to Primitive Actions Fitted Value Iteration (PFVI) considered in previous work. Section 3.2 provides a detailed discussion of the convergence properties of OFVI under different conditions. Section 4 introduces landmark options and explains how landmarks can be used to generate useful options for planning. This section also provides analyses the convergence rates of LAVI and OFVI with landmark-based options. Section 5 provides experiments and results comparing PFVI to OFVI in three different domains. Section 6 discusses the relationship between the results presented here and previous work, as well as, extensions and directions for future work.

2. Background

Let X be a subset of d-dimensional Euclidean space, $M(X)$ be the set of probability measures on X, and $f : X \rightarrow \mathbb{R}$ be a function from vectors in X to the real numbers. The max-norm $\|f\|_\infty = \sup_{x \in X} |f(x)|$. For $p \geq 1$ and $\mu \in M(X)$, the (p, μ)-norm is defined by $\|f\|_{p,\mu} = \left(\int \mu(x) \|f(x)\|^p dx\right)^{1/p}$.

An MDP is defined by a 5-tuple $\langle X, A, P, R, \gamma \rangle$ (Puterman, 1994) where X is a set of states, A is a set of primitive actions, P maps from state-action pairs to probability distributions over states, R is a mapping from state-action pairs to reward distributions bound to the interval $[-R_{\mathrm{MAX}}, R_{\mathrm{MAX}}]$, and $\gamma \in [0, 1)$ is a discount factor. Let $B(X; V_{\mathrm{MAX}})$ denote the set of functions with domain X and range bounded by $[-V_{\mathrm{MAX}}, V_{\mathrm{MAX}}]$ where $V_{\mathrm{MAX}} \leq \frac{R_{\mathrm{MAX}}}{1-\gamma}$. Throughout this paper we will consider MDPs where X is a bounded subset of a d-dimensional Euclidean space and A is a finite (non-empty) set of actions.

A policy $\pi : X \rightarrow A$ is a mapping from states to actions. We denote the set of deterministic, stationary Markov policies by Π. The standard objective of planning in an MDP is to derive a policy $\pi \in \Pi$ that maximizes

$$V^\pi(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t(x_t, \pi(x_t)) | x_0 = x, \pi\right] , \tag{1}$$

where x is the long-term value of following π starting in state x. The function V^π is called the value function with respect to policy π and it is well known that it can be written recursively as the solution of

$$\mathcal{T}^\pi V^\pi \triangleq \mathbb{E}\left[R(x, \pi(x))\right] + \gamma \int P(y|x, \pi(x)) V^\pi(y) dy, \tag{2}$$

where $\mathcal{T}^\pi$ is the Bellman operator with respect to π and V^π is its unique fixed point. Given a vector $V \in B(X; V_{\mathrm{MAX}})$, the greedy policy π with respect to V is defined by

$$\pi(x) = \arg\max_{a \in A} \mathbb{E}\left[R(x, a)\right] + \gamma \int P(y|x, \pi(x)) V(y) dy \ . \tag{3}$$

We denote the optimal value function by $V^* = \max_{\pi \in \Pi} V^\pi$.

Definition 1. *A policy π^* is* **optimal** *if its corresponding value function is V^*. A policy π is α-***optimal** *if $V^\pi(x) \geq V^*(x) - \alpha$ for all $x \in X$.*

The Bellman optimality operator $\mathcal{T}$ is defined by

$$(\mathcal{T}V)(x) = \max_{a \in A} \left(\mathbb{E}\left[R(x, a)\right] + \gamma \int_y P(y|x, a) V(y) dy \right) \ , \tag{4}$$

where $V \in B(X; V_{\mathrm{MAX}})$, which is known to have fixed point V^*. Value Iteration (VI), a popular planning algorithm for MDPs, is defined by repeatedly applying (4). The algorithm produces a series of value function estimates $V_0, V_1, V_2, \ldots, V_K$ and the greedy policy π_K is constructed based on V_K. Since VI converges only in the limit, the policy π_K may not be optimal. However, we would still like to measure the quality of π_K compared to π^*.

To measure the quality of a policy we need to define a notion of loss. The following defines loss of a policy with respect to a set of states and loss of a policy with respect to a probability distribution.

Definition 2. *Let $x \in X$. The* **subset-loss** *of a policy π with respect to a set of states $Y \subseteq X$ is defined by*

$$\mathcal{L}_Y(\pi) = \max_{x \in Y} \left(V^*(x) - V^\pi(x) \right) \ , \tag{5}$$

and we denote the special case where $Y \equiv X$ by $\mathcal{L}_\infty(\pi)$. Let $p \geq 1$ and $\mu \in M(X)$. The **loss** *of a policy with respect to a distribution over states μ is defined by*

$$\mathcal{L}_{p,\mu}(\pi) = \|V^* - V^\pi\|_{p,\mu} \ . \tag{6}$$

VI operates on the entire state space. This is how it is able to decrease the $\mathcal{L}_\infty$ error, but VI is computationally intractable in MDPs with extremely large or continuous state spaces. Thus approximate forms of VI generally seeks to decrease the loss with respect to a probability distribution over the state space.

2.1 Approximate Value Iteration (AVI)

Approximate Value Iteration (AVI) is the family of algorithms that estimate the optimal (action-)value function by iteratively applying an approximation of the Bellman optimality operator. There are many possible relaxations of VI. Which states are "backed up" according to $\mathcal{T}$, the representation of value function estimates, the number of times to sample from the simulator, etc. all impact the loss of the resulting policy.

One popular family of AVI algorithms are the Fitted Value Iteration (FVI) algorithms. These algorithms use a function approximator to represent value function estimates at each iteration. Primitive action Fitted Value Iteration (PFVI) is a generalization of VI to handle large or continuous state spaces. PFVI runs iteratively producing a sequence of $K \geq 1$ estimates $\{V_k\}_{k=1}^K$ of the optimal value function and returns a policy π_K that is greedy with respect to the final estimate V_K. During each iteration k, the algorithm computes a set of empirical estimates $\hat{V}_k$ of $\mathcal{T}V_{k-1}$ for n states, and then fits a function approximator to $\hat{V}_k$. To generate $\hat{V}_k$, n states $\{x_i\}_{i=1}^n$ are sampled from a distribution $\mu \in M(X)$. For each sampled state x_i and each primitive action $a \in A$, m next states $\{y_{i,j}^a\}_{j=1}^m$ and rewards $\{r_{i,j}^a\}_{j=1}^m$ are sampled from the MDP simulator $\mathbb{S}$. For the k^{th} iteration, the estimates of the Bellman backups are computed by

$$\hat{V}_k(x_i) = \max_{a \in A} \frac{1}{m} \sum_{j=1}^{m} \left(r_{i,j}^a + \gamma V_{k-1}(y_{i,j}^a) \right) \ , \tag{7}$$

where V_0 is the initial estimate of the optimal value function given as an argument to PFVI. The k^{th} estimate of the optimal value function is obtained by applying a supervised learning algorithm $\mathcal{A}$, that produces

$$V_k = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \left| f(x_i) - \hat{V}_k(x_i) \right|^p \ , \tag{8}$$

where $p \geq 1$ and $\mathcal{F} \subset B(X; V_{\text{MAX}})$ is the hypothesis space of the supervised learning algorithm.

The work of Munos and Szepesvári (2008) presented a full finite-sample, finite-iteration analysis of PFVI with guarantees dependent on the L_p-norm rather than the much more conservative infinity/max norm. This enabled analysis of instances of PFVI that use one of the many supervised learning algorithms minimizing L_1 or L_2 norm. A key assumption needed for their analysis is the notion of discounted-average concentrability of future state distributions.

Assumption 1. $[A1(\nu, \mu)]$ **[Discounted-Average Concentrability of Future-State Distributions] (Munos, 2005; Munos & Szepesvári, 2008)** *Given two distributions ν and μ defined over the state space X, $m \geq 1$, and m arbitrary policies $\pi_1, \pi_2, \ldots, \pi_m$, we assume that $\nu P^{\pi_1} P^{\pi_2} \ldots P^{\pi_m}$ is absolutely continuous with respect to μ implying that*

$$c(m) \overset{def}{=} \sup_{\pi_1, \pi_2, \ldots, \pi_m} \left\| \frac{d(\nu P^{\pi_1} P^{\pi_2} \ldots P^{\pi_m})}{d\mu} \right\|_\infty < +\infty \ , \tag{9}$$

and we assume that

$$C_{\nu,\mu} \overset{def}{=} (1-\gamma)^2 \sum_{m \geq 1} m \gamma^{m-1} c(m) < +\infty$$

is the discounted average concentrability coefficient, where P^π denotes the transition kernel induced by executing the action prescribed by the policy π.

Intuitively, this assumption prevents too much transition probability mass from landing on a small number of states. The condition that $C_{\nu,\mu}$ is finite depends on $c(m)$ growing at most subexponentially. See the work of Munos (2005) for a more complete discussion of Assumption 1. We note that the work of Farahmand, Munos, and Szepesvári (2010) presents a refined analysis using the expectation in (9) rather than a supremum. This results in tighter bounds but the bounds are more difficult to interpret due to a blowup in notation.

The work of Munos and Szepesvári (2008) shows that given an MDP, if we select probability distributions $\mu, \nu \in M(X)$, a positive integer p, a supervised learning algorithm $\mathcal{A}$ over a bounded function space $\mathcal{F}$ that returns the function $f \in \mathcal{F}$ that minimizes the empirical p-norm error, $V_0 \in \mathcal{F}$ an initial estimate of the optimal value function, and $\varepsilon > 0$ and $\delta \in (0, 1]$. Then for any $K \geq 1$, with probability at least $1 - \delta$, there exist positive integers n, m, K such that the policy π_K returned by PFVI satisfies

$$\mathcal{L}_{p,\nu}(\pi_K) \leq \frac{2\gamma}{(1-\gamma)^2} C_{\nu,\mu}^{1/p} b_{p,\mu}(\mathcal{T}\mathcal{F}, \mathcal{F}) + \varepsilon + \left(\gamma^{K+1}\right)^{1/p} \left(\frac{2 \left\| V^* - V_0 \right\|_\infty}{(1-\gamma)^2} \right) \ , \tag{10}$$

where $b_{p,\mu}(\mathcal{T}\mathcal{F}, \mathcal{F}) = \sup_{f \in \mathcal{F}} \inf_{g \in \mathcal{F}} \|\mathcal{T} f - g\|_{p,\mu}$ is the inherent Bellman error of $\mathcal{F}$ with respect to Bellman operator $\mathcal{T}$.[1] The inherent Bellman error is a measure of how well the chosen hypothesis space $\mathcal{F}$ can represent $\hat{V}_k$ at each iteration. The first term in (10) is called the approximation error and corresponds to the error introduced by the inability of the supervised learning algorithm to exactly capture $\hat{V}_k$ at each iteration, while the second term, the estimation error, is due to using a finite number of samples to estimate $\hat{V}_k$. The last term is controlled by the number of iterations K of the algorithm. By increasing K the

[1] In this paper, we consider the multi-sample variant of PFVI that uses fresh samples at each iteration. The bound for the single-sample variant of PFVI, which uses the same batch of samples at each iteration, is almost identical to (10). See the work of Munos and Szepesvári (2008) for details.

last term shrinks exponentially fast. This last term characterizes the convergence rate of the algorithm. The size of the discount factor γ controls the rate of convergence. Convergence is faster when γ is smaller. Unfortunately, γ is part of the problem definition. However, because options execute for multiple timesteps, an option can have an effective discount factor that is smaller than γ.

2.2 Semi-Markov Decision Processes

Semi-Markov Decision Processes (SMDPs) are a generalization of the Markov Decision Process (MDP) model that incorporates temporally extended actions. Temporally extended actions have primarily been applied to direct exploration in reinforcement learning (Iba, 1989; Mannor et al., 2004; Konidaris & Barto, 2007; Jong & Stone, 2008). However, they may also play an important role in planning (Precup & Sutton, 1997; Precup et al., 1998; Sutton et al., 1999; Silver & Ciosek, 2012). For example, the popular dynamic programming algorithms VI and PI still converge when applied to SMDPs (Puterman, 1994). The work of Precup et al. (1998) shows that options and an MDP form an SMDP. The works of Sutton et al. (1999) and Silver and Ciosek (2012) provide experimental results demonstrating that options can speed up planning in finite state MDPs. However, these works did not apply options to tasks with continuous state spaces and there is no theoretical analysis of the convergence rate of planning with options compared to planning with primitive actions. We will use the SMDP framework to investigate planning with temporally extended actions.

An MDP paired with a set of temporally extended actions called options, denoted by $\mathcal{O}$, forms an SMDP.

Definition 3. *(Sutton et al., 1999) An* **option** *o is defined by a 3-tuple $\langle \mathcal{I}_o, \pi_o, \beta_o \rangle$ where $\mathcal{I}_o$ is the set of states that o can be initialized from, π_o is the stationary policy defined over primitive actions followed during the lifetime of o, and $\beta_o : X \to [0, 1]$ determines the probability that o will terminate while in a given state.*

For each state $x \in X$, we denote the set of options that can be initialized from x by $\mathcal{O}_x = \{o \in \mathcal{O} \mid x \in \mathcal{I}_o\}$. Options are a generalization of actions. In fact they encompass, not only primitive actions and temporally extended actions, but also stationary policies and other control structures. Here we take actions to be options that always terminate after only a finite number of timesteps. Policies on the other hand never terminate. For example, a stationary policy can be represented by an option by setting the termination probabilities to $\beta(x) = 0$ for all states.

For an option $o = \langle \mathcal{I}_o, \pi_o, \beta_o \rangle$, we denote the probability that o is initialized from a state x and terminates in a subset of states $Y \subseteq X$ in exactly t timesteps by $P^o_t(Y|x)$ and the discounted termination state probability distribution of o by $\widetilde{P}^o(Y|x) = \sum_{t=1}^{\infty} \gamma^t P^o_t(Y|x)$. For a state-option pair (x, o), the discounted cumulative reward distribution during the option's execution is denoted by $\widetilde{R}(x, o)$.

The objective of planning with options is to derive a policy $\varphi : X \to \mathcal{O}$ from states to options that maximizes

$$V^{\varphi}(x) = \mathbb{E}\left[\widetilde{R}(x, \varphi(x))\right] + \int \widetilde{P}^{\varphi(x)}(y|x) V^{\varphi}(y) dy \ . \tag{11}$$

The Bellman operator for an SMDP is defined by

$$(\mathbb{T}V)(x) = \max_{o \in \mathcal{O}_x} \left(\mathbb{E}\left[\widetilde{R}(x, o) \right] + \int \widetilde{P}^o(y|x) V(y) dy \right) \ , \tag{12}$$

where $\mathbb{T}$ is defined over the set of options $\mathcal{O}$ instead of primitive actions A. The differences between (4) and (12) could potentially lead to widely different results when embedded in the FVI algorithm.

3. Options Fitted Value Iteration

Algorithm 1 Options Fitted Value Iteration (OFVI)

```
Require: Collection of options $\mathcal{O}$, an SMDP simulator $\mathbb{S}$, state distribution $\mu$, function space $\mathcal{F}$, initial iterate $V_0 \in \mathcal{F}$, $n$ the number of states to sample, $m$ the number of samples to obtain from each state-option pair, $K$ the number of times to iterate before returning
for $k = 1, 2, \dots, K$ do {Generate $K$ iterates $V_1, V_2, \dots V_K$.}
  {Collect new batch of samples.}
  for $i = 1, 2, \dots, n$ do {Sample $N$ states.}
    $x \sim \mu$ {Sample a state from distribution $\mu$.}
    for $o \in \mathcal{O}_x$ do
      for $j = 1, 2, \dots, m$ do
        $\left(y^o_{i,j}, r^o_{i,j}, \tau^o_{i,j}\right) \sim \mathbb{S}(x, o)$ {Query the simulator for a terminal state, discounted cumulative reward, and duration of executing $(x, o)$.}
      end for
    end for
  end for
  {Estimate Bellman Backups.}
  $\hat{V} \leftarrow \frac{1}{m} \sum_{j=1}^{m} \left[ r^o_{i,j} + \gamma^{\tau^o_{i,j}} V_{k-1}(y^o_{i,j}) \right]$
  {Find the best fitting approximation to $\hat{V}$.}
  $V_k = \arg\inf_{f \in \mathcal{F}} \|f - \hat{V}\|_n$
end for
return $\varphi_K$ {Return the greedy policy wrt $V_K$.}
```

Algorithm 1 is a generalization of the multisample FVI algorithm to the case where samples are generated by options (with primitive actions as a special case). The algorithm, Options Fitted Value Iteration (OFVI), takes as arguments positive integers n, m, K, $\mu \in M(X)$, an initial value function estimate $V_0 \in \mathcal{F}$, and a simulator $\mathbb{S}$. At each iteration $k = 1, 2, \dots, K$, states $x_i \sim \mu$ for $i = 1, 2, \dots, n$ are sampled, and for each option $o \in \mathcal{O}_{x_i}$, m next states, rewards, and option execution times $\langle y^o_{i,j}, r^o_{i,j}, \tau^o_{i,j} \rangle \sim \mathbb{S}(x_i, o)$ are sampled for $j = 1, 2, \dots, m$. Then the update resulting from applying the Bellman operator to the previous iterate V_{k-1} is estimated by

$$\hat{V}_k(x_i) \leftarrow \max_{o \in \mathcal{O}_{x_i}} \frac{1}{m} \sum_{j=1}^{m} \left[r^o_{i,j} + \gamma^{\tau^o_{i,j}} V_{k-1}(y^o_{i,j}) \right] \ , \tag{13}$$

and we apply a supervised learning algorithm to obtain the best fit according to (8). The given simulator $\mathbb{S}$ differs from the simulator for PFVI. It returns the state where the option returned control to the agent, the total cumulative, discounted reward received during its execution, and the duration or number of timesteps that the option executed. This additional information is needed to compute (13). Otherwise the differences between PFVI and OFVI are minor and it is natural to ask if OFVI has similar finite-sample and convergence behavior compared to PFVI.

3.1 Simple Analysis

Notice that PFVI is a special case of OFVI where the given options contain only the primitive action set (i.e., $\mathcal{O} \equiv A$). Therefore, we cannot expect OFVI to always outperform PFVI. Instead, we aim to show that OFVI does not converge more slowly than PFVI and identify cases where OFVI converges more quickly than PFVI. The following proposition provides a general upper bound on the loss of the policy derived by OFVI when $\mathcal{O}$ contains A. This bound can be compared to bounds on the loss of the policy derived by PFVI.

Proposition 1. *For any $\varepsilon_S, \delta > 0$ and $K \geq 1$. Fix $p \geq 1$. Let $\mathcal{O}$ be a set of options that contains the set of primitive actions A. Given an initial state distribution $\nu \in M(X)$, a sampling distribution $\mu \in M(X)$, and $V_0 \in B(X, V_{\mathrm{MAX}})$, if $A1(\nu, \mu)$ (Assumption 1) holds, then there exists positive integers n and m such that when OFVI is executed,*

$$\mathcal{L}_{p,\nu}(\pi_K) \leq \frac{2\gamma}{(1-\gamma)^2} C_{\nu,\mu}^{1/p} b_{p,\mu}(\mathbb{T}\mathcal{F}, \mathcal{F}) + \varepsilon_S + \left(\gamma^{K+1}\right)^{1/p} \left(\frac{2\left\|V^* - V_0\right\|_\infty}{(1-\gamma)^2} \right) \tag{14}$$

holds with probability at least $1 - \delta$.

A proof of Proposition 1 as well as sufficient values for n and m are given in the appendix. Proposition 1 suggests that as long as $\mathcal{O}$ contains A, OFVI has performance at least comparable to PFVI (if not better). There are two main differences between the bound in Proposition 1 and in the work of Munos and Szepesvári (2008, Thm. 2). First, the inherent Bellman error in Proposition 1 may be larger than the inherent Bellman error with only primitive actions. Second, the convergence rate of OFVI tracks the convergence rate of the SMDP Bellman operator $\mathbb{T}$ rather than the MDP Bellman operator $\mathcal{T}$.

Proposition 1 implies that OFVI converges approximately as fast as PFVI when $\mathcal{O}$ contains A. However, the two algorithms may converge to different value functions due to the larger inherent Bellman error of OFVI.

Proposition 1 has two limitations. First it only considers the case where $\mathcal{O}$ contains A. Second, it does not describe when OFVI converges more quickly than PFVI. In the following section, we will investigate both of these possibilities.

3.2 General Analysis

There are two perspectives that explain how applying options to AVI can decrease the number of iterations needed to find a near-optimal policy.

In the first perspective, options increase *information flow* between otherwise temporally disparate states facilitating fast propagation of value throughout the state-space. For example, if it takes many primitive actions to transition from a state x to a state y, then planning with primitive actions will require many iterations before information can be propagated from y back to x. However, given an option that when initialized in state x, terminates in y, value from y is propagated back to x at every iteration.

In the second perspective, options with long duration can cause *rapid contraction* toward the optimal or a near-optimal value function. For the discounted and average reward objectives, the proof that VI converges is based on a contraction argument (for details, see Puterman, 1994). It turns out that options with long duration can induce a faster contraction than primitive actions (or faster than options with shorter durations).

How these options influence the convergence rate of AVI depends critically on the agent's objective. In this paper, we only analyze the discounted reward objective. However, to put our results into context, in this section, we comment on the finite horizon and average reward objectives as well.

- **Undiscounted, Finite Horizon:** The agent maximizes the sum of rewards received over $H \geq 1$ timesteps. Here options can short circuit the number of iterations needed to propagate reward back H steps. This effect is more naturally described as increasing the information flow between temporally disparate states.

- **Discounted Reward (our analysis):** Our analysis uses a contraction argument to show faster convergence and an information flow argument to show that fast contraction can occur even when the temporally extended actions are sparsely distributed in the state-space.

- **Average Reward:** The agent maximizes the average of an infinite sequence of rewards. While we only consider the discounted reward setting, similar contraction arguments could provably be applied to show how options can produce a closer approximation of the optimal value function with fewer iterations.

Our approach is based on a contraction mapping argument. By applying the MDP Bellman operator $\mathcal{T}$ to $V \in B(X, V_{\text{MAX}})$, we obtain the following contraction mapping

$$\|V^* - \mathcal{T}V\|_\infty \leq \gamma \|V^* - V\|_\infty \tag{15}$$

where γ (the discount factor) serves as the contraction coefficient. Since $\gamma < 1$, the left hand side is strictly smaller than $\|V^* - V\|_\infty$. Smaller values of γ imply a faster convergence rate, but the discount factor γ is part of the problem description and cannot be changed. However, if we apply the MDP Bellman operator $\mathcal{T}$, $\tau > 1$ times, then we obtain a contraction mapping where the contraction coefficient is $\gamma^\tau < \gamma$. Temporally extended options have a similar effect. Options can speed up the convergence rate of the SMDP Bellman operator $\mathbb{T}$ by inducing a smaller contraction coefficient that depends on the number of timesteps that the option executes for.

Intuitively, options with a long duration are desirable for planning because options that execute for many timesteps enable OFVI to look far into the future during a single iteration. However, the duration depends on both an option and the state where the option is initialized. The following definition makes the notion of an option's duration precise.

Definition 4. *Let $x \in X$ be a state and $o \in \mathcal{O}_x$ be an option. The* **duration** *of executing option o from state x is the number of timesteps that o executes before terminating (i.e., returning control to the option policy). We denote by $D^o_{x,Y}$ the random variable representing the duration of initializing option o from state x and terminating in $Y \subseteq X$. For a set of options $\mathcal{O}$, we define the* **minimum duration** *to be $d_{\min} = \min_{x \in X, o \times \mathcal{O}_x} \inf_{Y \subseteq X} \mathbb{E}\left[D^o_{x,Y}\right]$.*

First notice that the duration of an option is a random variable that depends on the state where the option was initialized. This complicates the analysis compared to assuming that all temporally extended actions terminate after a fixed number of timesteps, but it allows for much greater flexibility when selecting options to use for planning.

Similar to the analysis of PFVI, the analysis of OFVI depends on the concentrability of future state distributions. We introduce the following assumption on the future state distributions of MDPs with a set of options. The given set of options $\mathcal{O}$ may or may not contain the entire set of primitive actions A from the underlying MDP.

Assumption 2. $[A2(\nu, \mu)]$ **[Option-Policy Discounted-Average Concentrability of Future-State Distributions]** *Given two distributions ν and μ defined over the state space X, $m \geq 1, t \geq m$, and m arbitrary option policies $\varphi_1, \varphi_2, \ldots, \varphi_m$, we assume that $\nu P_t^{\varphi_1 \varphi_2 \cdots \varphi_m}$ is absolutely continuous with respect to μ implying that*

$$\hat{c}_t(m) = \sup_{\varphi_1, \varphi_2, \ldots, \varphi_m} \left\| \frac{d\left(P_t^{\varphi_1 \varphi_2 \cdots \varphi_m}\right)}{d\mu} \right\|_\infty < +\infty \ , \tag{16}$$

and we assume that

$$\mathbb{C}_{\nu,\mu} = (1-\gamma)^2 \sum_{t=1}^{\infty} t \gamma^{t-1} \max_{m \in \{1,2,\ldots,t\}} \hat{c}_t(m) < +\infty \tag{17}$$

is the option discounted average concentrability coefficient, where $\nu P_t^{\varphi_1 \varphi_2 \cdots \varphi_m}(y)$ assigns probability mass according to the event that a sequence of m options will terminate in a state y exactly t timesteps after an initial state is sampled from ν and a sequence of m options are executed where the i^{th} option in the sequence is chosen according to φ_i.

Assumption 2 is analogous to Assumption 1. Despite the fact that options are a more general framework than the set of primitive actions, Assumption 2 results in a smaller concentrability coefficient than Assumption 1.

Lemma 1. *Let $\nu, \mu \in M(X)$. Assumption $A1(\nu, \mu)$ implies Assumption $A2(\nu, \mu)$ (i.e., $\mathbb{C}_{\nu,\mu} \leq C_{\nu,\mu}$).*

Proof. First notice that since any t timestep sequence of actions generated by a sequence $\varphi_1, \varphi_2, \ldots, \varphi_m$ of $m \leq t$ option policies can be expressed by a sequence of t primitive policies $\pi_1, \pi_2, \ldots, \pi_t$. Thus

$$\begin{aligned} \hat{c}_t(m) &= \sup_{\varphi_1, \varphi_2, \ldots, \varphi_m} \left\| \frac{d\left(P_t^{\varphi_1 \varphi_2 \cdots \varphi_m}\right)}{d\mu} \right\|_\infty \\ &\leq \sup_{\pi_1, \pi_2, \ldots, \pi_t} \left\| \frac{d(P^{\pi_1} P^{\pi_2} \ldots P^{\pi_t})}{d\mu} \right\|_\infty \\ &= c(t) \ . \end{aligned}$$

Since $\hat{c}_t(m) \leq c(t)$ for all $m \geq 1$ and $t \geq m$, then

$$\begin{array}{rcl} \mathbb{C}_{\nu,\mu} & = & (1-\gamma)^2 \sum_{t=1}^{\infty} t\gamma^{t-1} \max_{m \in \{1,2,\ldots,t\}} \hat{c}_t(m) \\ & \leq & (1-\gamma)^2 \sum_{t=1}^{\infty} t\gamma^{t-1} c(t) \\ & = & C_{\nu,\mu} \ . \end{array}$$

□

Lemma 1 implies that Assumption 2 holds for any set of options whenever Assumption 1 holds. The main reason is because a sequence of m options that executes for t timesteps has fewer degrees of freedom than a sequence of t primitive policies. Furthermore, the proof of Lemma 1 tells us that the important property of the discounted concentrability of future states is not the number of options executed in a sequence but the number of timesteps that the sequence of options executes for.

In our analysis of the convergence rates of OFVI, we will report bounds containing the coefficient $\mathbb{C}_{\nu,\mu}$ rather than $C_{\nu,\mu}$. This is because, in cases where the option set contains mostly temporally extended actions, $\mathbb{C}_{\nu,\mu}$ may be smaller than $C_{\nu,\mu}$. However, Lemma 1 tells us that we can replace $\mathbb{C}_{\nu,\mu}$ in the bounds with $C_{\nu,\mu}$ for the purposes of comparing with (10) and (14).

The important properties of temporally extended actions that cause faster convergence are (1) the quality of the policy they follow, and (2) how long the action executes for (or its duration). The following definition describes the set of states where there exists an option that follows a near-optimal policy and has sufficient duration.

Definition 5. *Let X be the set of states in an MDP with option set $\mathcal{O}$, $\alpha \geq 0$, and $d \geq d_{\min}$. The (α, d)*-**omega set** *defined by*

$$\omega_{\alpha,d} \equiv \left\{ x \in X \mid \exists_{o \in \mathcal{O}_x} \ s.t. \ \inf_{Y \subseteq X} \mathbb{E}\left[D^o_{x,Y}\right] \geq d \wedge Q^{\Phi^*}(x,o) \geq V^{\Phi^*}(x) - \alpha \right\} \ , \tag{18}$$

is the set of states where there is an α-optimal temporally extended action with duration longer than d and where Φ^* is the optimal option policy.

The states in $\omega_{\alpha,d}$ have particularly long duration and follow an α-optimal policy. However, the states outside of $\omega_{\alpha,d}$ do not. At these other states, either the available options are not sufficiently temporally extended or they follow a suboptimal policy. To obtain faster convergence, we need a way of connecting the convergence rates of the states outside of $\omega_{\alpha,d}$ with the convergence rates of the states in $\omega_{\alpha,d}$.

Assumption 3. *[$A3(\alpha, d, \psi, \nu, j)$] Let $\alpha, \psi, j \geq 0$, $d \geq d_{\min}$, and $\nu \in M(X)$. For any $m \geq 0$ option policies $\varphi_1, \varphi_2, \ldots, \varphi_m$, let $\rho = \nu P^{\varphi_1} P^{\varphi_2} \ldots P^{\varphi_m}$. There exists an α-optimal option policy $\hat{\varphi}$ such that either (1) $\Pr_{x \sim \rho}[x \in \omega_{\alpha,d}] \geq 1 - \psi$ or (2) $\exists_{i \in \{1,2,\ldots,j\}} \Pr_{y \sim \eta_i}[y \in \omega_{\alpha,d}] \geq 1 - \psi$ where $\eta_i = \nu P^{\varphi_1} P^{\varphi_2} \ldots P^{\varphi_m} \left(P^{\hat{\varphi}}\right)^i$ for $i = 1, 2, \ldots, j$.*

Assumption 3 points out three key features that impact planning performance with options:

1. Quality of the option set controlled by α,

2. Duration of options specified by d, and

3. Sparsity of $\omega_{\alpha,d}$ in the state-space characterized by j and ψ.

We refer to the policy $\hat{\varphi}$ as the "bridge" policy, because it bridges the gap between the states in $\omega_{\alpha,d}$ and other states. Notice that we do not assume that the planner has any knowledge of $\hat{\varphi}$. It is enough that such a policy exists. Assumption 3 says that no matter what policies are followed, either (1) the agent will end up in $\omega_{\alpha,d}$ with high probability or (2) there exists a near-optimal option policy that will transport the agent to $\omega_{\alpha,d}$ in at most j timesteps with high probability. This enables us to account for problems where only a few states have temporally extended actions, but these states can be reached quickly without following a policy that is too suboptimal.

The following theorem provides a comprehensive description of the convergence behavior of OFVI (with PFVI as a special case where $\mathcal{O} = A$).

Theorem 1. *Let $\varepsilon_S, \delta > 0$, $\alpha, \psi, j \geq 0$, $K, p \geq 1$, $d \geq d_{\min}$, $0 \leq Z \leq K$, and $\nu, \mu \in M(X)$. Suppose that $A2(\nu, \mu)$ (Assumption 2) and $A3(\alpha, d, \psi, \nu, j)$ (Assumption 3) hold. Given $V_0 \in B(X, V_{\mathrm{MAX}})$, if the first Z iterates $\{V_k\}_{k=0}^{Z}$ produced by the algorithm are pessimistic (i.e., $V_k(x) \leq V^{\Phi^*}(x)$ for all $x \in X$), then there exists positive integers n and m such that when OFVI is executed,*

$$\mathcal{L}_{p,\nu}(\varphi_K) \leq \mathcal{L}_{p,\nu}(\Phi^*) + \frac{2\gamma^{d_{\min}}}{(1-\gamma)^2}\mathbb{C}_{\nu,\mu}^{1/p}\left(b_{p,\mu}(\mathbb{T}\mathcal{F}, \mathcal{F}) + \alpha\right) + \varepsilon_S + \left(\gamma^{d_{\min}(K+1)+(1-\psi)(d-d_{\min})\lfloor Z/\hat{j}\rfloor}\right)^{1/p}\left(\frac{2\left\|V^{\Phi^*} - V_0\right\|_\infty}{(1-\gamma)^2}\right) \quad (19)$$

holds with probability at least $1-\delta$ where Φ^* is the optimal option policy with respect to the given options $\mathcal{O}$ and $\hat{j} = j + 1$.

Theorem 1 bounds the loss of the option policy φ_K returned after performing $K \geq 1$ iterations of value iteration with respect to a (p, ν)-norm. The distribution ν can be thought of as an initial state distribution. It places more probability mass on the regions of the state space where we want the policy φ_K to have the best performance. The value of $p \geq 1$ is generally determined by the function approximation procedure. For $p = 1$, the function approximation procedure minimizes the L_1-norm and for $p = 2$, the function approximation procedure minimizes the L_2-norm.

The right hand side of (19) contains four terms.

1. The first term bounds the abstraction loss, which is the loss between the optimal policy over primitive actions and the optimal option policy.

2. The second term bounds the approximation error, which is the error caused by the inability of the function approximation architecture to exactly fit $\hat{V}_k(x_i)$ during each iteration and α which shows up in this term is due to bootstrapping off options that follow α-optimal policies to gain faster convergence. Notice that $\frac{\gamma^{d_{\min}}}{(1-\gamma)^2}$ shrinks as $d_{\min}$ grows. Thus option sets with longer minimum duration shrink the approximation error.

3. The third term ε_S is the sample error, which is controlled by the number of samples taken at each iteration.

4. The last term controls the convergence error. Notice that γ, the discount factor, is in $[0, 1)$ and therefore the last term shrinks rapidly as its exponent grows. While OFVI does not actually converge in the sense that the loss may never go to zero, this last term goes to zero as $K \to \infty$. In the worst case, the convergence rate is controlled by $\gamma^{d_{\min}(K+1)}$, but the convergence rate can be significantly faster if Z and d are large and j is small.

An iterate $\hat{V} : X \to \mathbb{R}$ is pessimistic if

$$\forall_{y \in X} \;\; \hat{V}(y) \leq V^{\Phi^*}(y) \;\; ,$$

where Φ^* is the optimal policy defined over option set $\mathcal{O}$. Whether iterates are pessimistic (or not) has a critical impact on the convergence rate of OFVI. To understand why, suppose that $q \in \mathcal{O}_x$ is an option that can be initialized from a state $x \in X$ where q is α-optimal with respect to Φ^* (i.e., $Q^{\Phi^*}(x, q) \geq V^{\Phi^*}(x) - \alpha$) and has a long duration (at least d timesteps). If $\hat{V}$ is pessimistic, then the Bellman optimality operator performs an update

$$\begin{aligned}
(\mathbb{T}\hat{V})(x) =& \max_{o \in \mathcal{O}_x} \left(\mathbb{E}\left[\widetilde{R}(x, o)\right] + \int \widetilde{P}^o(y|x)\hat{V}(y)dy \right) \;\; , && \text{By definition (12).} \\
\geq & \mathbb{E}\left[\widetilde{R}(x, q)\right] + \int \widetilde{P}^q(y|x)\hat{V}(y)dy \;\; . && \text{The update with option } q.
\end{aligned}$$

Since $\mathbb{T}$ is known to be a monotone operator, $V^{\Phi^*}(x) \geq (\mathbb{T}\hat{V})(x)$. Taken together, these facts imply that even if an option other than q was selected for the update, $(\mathbb{T}\hat{V})(x)$ is at least as close to $V^{\Phi^*}(x)$ as if q was selected. This allows us to prove that when the iterates are pessimistic, the convergence rate of OFVI is rapid (depending on d). Unfortunately, when the iterates are not pessimistic, this reasoning no longer holds and convergence may depend on options with duration $d_{\min}$ instead.

On each of the Z iterations where the estimate of the value function is pessimistic, OFVI exploits the options with duration d rather than $d_{\min}$. However, it can only get samples of these options from states in $\omega_{\alpha,d}$. The states outside of $\omega_{\alpha,d}$ can benefit from the rapid convergence of the states in $\omega_{\alpha,d}$ but only after j additional iterations. The main reason is because it can take j steps to propagate value from states in $\omega_{\alpha,d}$ back to the other states.

Using Theorem 1 it is possible to consider the convergence rates of OFVI on a wide range of planning problems. In the following subsections, we examine special cases of Theorem 1. First, we consider what happens when $d_{\min}$ is greater than 1 ignoring the possibility of exploiting options with longer duration. Second, we consider what happens when we mix primitive actions with temporally extended actions.

3.2.1 Abstraction

An important case involves planning where only temporally extended actions are available. The main advantage in this case is that we can guarantee an upper bound on the convergence rate of the algorithm is strictly faster than the upper bound for PFVI. However, the solution that OFVI converges to may be inferior to the solution converged to by PFVI if the best policy with respect to the given set of options is poor.

Corollary 1. *Let $\varepsilon_S, \delta > 0$, $K, p, d_{\min} \geq 1$, and $\nu, \mu \in M(X)$. Given $V_0 \in B(X, V_{\mathrm{MAX}})$, if $A2(\nu, \mu)$ (Assumption 2) and $A3(\alpha = 0, d \geq d_{\min}, \psi = 0, \nu, j = 0)$ (Assumption 3) hold, then there exist positive integers n and m such that when OFVI is executed*

$$\mathcal{L}_{p,\nu}(\varphi_K) \leq \mathcal{L}_{p,\nu}(\Phi^*) + \frac{2\gamma^{d_{\min}}}{(1-\gamma)^2}\mathbb{C}_{\nu,\mu}^{1/p} b_{p,\mu}(\mathbb{T}\mathcal{F}, \mathcal{F}) + \varepsilon_S + \left(\gamma^{d_{\min}(K+1)}\right)^{1/p}\left(\frac{2\left\|V^{\Phi^*} - V_0\right\|_\infty}{(1-\gamma)^2}\right) \tag{20}$$

holds with probability at least $1 - \delta$, where Φ^* is the optimal option policy with respect to the given options $\mathcal{O}$.

First notice that in Corollary 1, the upper bound is with respect to the loss of an optimal policy π^* (over primitive actions). The bound on the loss in Corollary 1 depends on four terms,

1. the first term is controlled by the error between the optimal policy π^* and the best policy Φ^* with respect to the given options $\mathcal{O}$,
2. the second term is controlled by the option policy future state concentrability coefficient $\mathbb{C}_{\nu,\mu}$ and inherent bellman error $b_{p,\mu}(\mathbb{T}\mathcal{F}, \mathcal{F})$,
3. the third term is simply the estimation error term ε (which is controlled by the amount of sampling done by OFVI), and
4. the last term is the convergence error controlled by $\left(\gamma^{d_{\min}(K+1)}\right)$.

When $d_{\min} > 1$ the convergence rate of OFVI can be significantly faster than PFVI, but the loss term $\mathcal{L}_{p,\nu}(\Phi^*)$ may be large if the given option set $\mathcal{O}$ cannot represent a sufficiently good policy.

Although the abstraction setting has a fast convergence rate, the quality of the policies produced depends on the best possible option policy derived from the given set of options. If this policy is poor, then the policy produced by OFVI will also be poor. In the next subsection, we try to overcome this limitation by augmenting the set of primitive actions instead of discarding them.

3.2.2 Augmentation with Sparsely Scattered Temporally Extended Actions

Experimental results have demonstrated that a few well placed temporally extended actions often improve the convergence rate of planning (Precup et al., 1998). We would like to describe conditions where sparsely scattered temporally extended actions cause faster convergence.

The following theorem gives a bound for OFVI in environments with sparsely distributed temporally extended actions.

Corollary 2. *Let $\varepsilon_S, \delta > 0$, $\alpha, \psi \geq 0$, $K, p, d, j \geq 1$, $0 \leq Z \leq K$, and $\nu, \mu \in M(X)$. Suppose that $A2(\nu, \mu)$ (Assumption 2) and $A3(\alpha, d, \psi, \nu, j)$ (Assumption 3) hold. Given $V_0 \in B(X, V_{\mathrm{MAX}})$, if the first Z iterates $\{V_k\}_{k=0}^{Z}$ produced by the algorithm are pessimistic (i.e., $V_k(x) \leq V^*(x)$ for all $x \in X$), then there exist positive integers n and m such that when OFVI is executed,*

$$\mathcal{L}_{p,\nu}(\varphi_K) \leq \frac{2\gamma}{(1-\gamma)^2} C_{\nu,\mu}^{1/p} \left(b_{p,\mu}(\mathbb{T}\mathcal{F}, \mathcal{F}) + \alpha \right) + \varepsilon_S + \left(\gamma^{K+1+(1-\psi)(d-1)\lfloor Z/\hat{j} \rfloor} \right)^{1/p} \left(\frac{2\left\| V^* - V_0 \right\|_\infty}{(1-\gamma)^2} \right) \quad (21)$$

holds with probability at least $1-\delta$ and $\hat{j} = j + 1$.

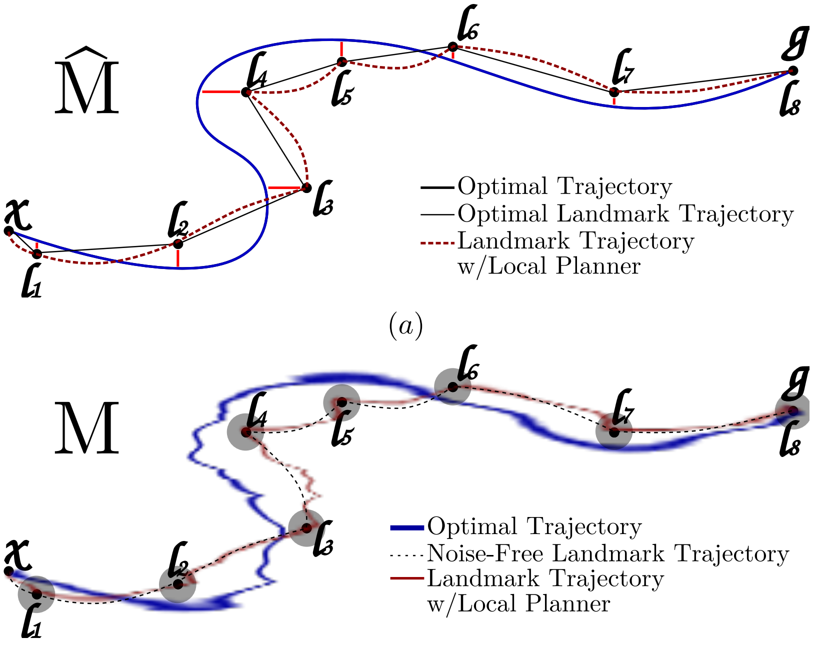
Notice that the abstraction loss $\mathcal{L}_{p,\nu}(\Phi^*)$ disappears because in this special case $V_M^* = V_M^{\Phi^*}$. The improvement in convergence rate is only on the first Z pessimistic iterates and a small penalty α appears in the approximation error term due to our exploitation of α-optimal temporally extended actions. The convergence rate of Corollary 2 is driven by $\gamma^{K+1+(1-\psi)(d-1)\lfloor Z/j \rfloor} \leq \gamma^{K+1}$, demonstrating that OFVI can converge faster than PFVI. Notice that when j is large, meaning that it can take more timesteps to visit $\omega_{\alpha,d}$, the convergence rate is slower than when j is small. This means that convergence improvement may be less dramatic when the temporally extended actions are too sparse. When $\hat{j} = 1$, the convergence rate is controlled by $\gamma^{K+1+(1-\psi)(d-1)Z} \leq \gamma^{K+1}$.

4. Generating Options via Landmarks

One limitation of planning with options is that options typically need to be designed by an expert. In this section, we consider one approach to generating options automatically. Our approach is similar in spirit to the successful FF-Replan algorithm (Yoon, Fern, & Givan, 2007), which plans on a deterministic projection of the target MDP. The algorithm replans whenever the agent enters a state that is not part of the current plan. However, unlike FF-Replan, our approach is more scalable as it does not plan globally over the entire system.

More specifically, we assume access to a simulator $\mathbb{S}_M$ for the target MDP $M = \langle X, A, P, R, \gamma \rangle$ and a simulator $\mathbb{S}_{\widehat{M}}$ for a "relaxed" MDP $\widehat{M} = \langle X, A, \widehat{P}, R, \gamma \rangle$ with deterministic transition dynamics. Given a state x and option $o \in \mathcal{O}_x$, a simulator $\mathbb{S}$ returns the discounted cumulative reward $\tilde{R}$ of executing the option, the duration of the option's execution τ, and the termination state y.

Since $\widehat{M}$ has deterministic transition probabilities, its dynamics are captured by a directed graph $G = \langle X, \widehat{P} \rangle$. Furthermore, R specifies the reward associated with each edge. If two or more actions transition from a state x to the same state y, the maximum reward of these actions is associated with the edge (x, y) in G. We denote a maximum reward path from $x \in X$ to $g \in X$ by $p_G^*(x, g)$ and the length of the maximum reward path by $|p_G^*(x, g)|$. Throughout this section we will assume that the rewards are all non-positive (i.e. bound to $[-R_{\mathrm{MAX}}, 0]$). The reason for this is because stochastic shortest path problems are undefined when the MDP contains positive reward cycles, however, in our experiments we relax this assumption.



(a)

(b)

Figure 1: A trajectory from state x to state g. Error is introduced by planning a policy over landmark options rather than following the optimal policy. (a) In the deterministic relaxation $\widehat{M}$, errors are caused by the landmarks not being exactly on the optimal trajectory and the local planner not taking the maximum reward path from one landmark to another. (b) In the stochastic target problem M, errors are introduced by noise, which causes the agent to only reach states "nearby" the landmark states on its path to the goal.

The purpose of introducing $\widehat{M}$ is that computationally efficient planning algorithms are known for minimum cost path problems (Dijkstra, 1959; Hart, Nilsson, & Raphael, 1968), which are equivalent to maximum reward path planning problems provided there are no positive reward cycles. Thus, if $\widehat{M}$ is a reasonable approximation for M, then we can dynamically generate options for M by planning on $\widehat{M}$. We assume an efficient local planner $\mathcal{P}$ exists for G.

However, in very large directed graphs, even so called "efficient" algorithms can be computationally expensive. Thus, we assume that $\mathcal{P}$ has a given maximum planning horizon $d^+ \geq 1$.

Recent work on finding minimum cost paths in very large graphs has shown that paths can be found more efficiently by introducing "landmarks" (Sanders & Schultes, 2005), an intuition that has been used in robotic control for a long time (Lazanas & Latombe, 1992).

Definition 6. *A* **landmark set** $\mathbb{L}$ *is a finite, non-empty subset of the state-space.*

Each landmark is a single state, and a landmark set induces a directed graph over the state-space. Obtaining a provably good landmark set is generally a hard problem (Peleg & Schäffer, 1989). Here we assume that the landmark states are given, but they could be acquired through analyzing the dynamics of $\widehat{M}$ as in the work of Simsek and Barto (2004) or from demonstrations as in the work of Konidaris, Kuindersma, Barto, and Grupen (2010).

Given a set of landmarks, we can define a corresponding set of options, as follows.

Definition 7. *Let* $\eta \geq 0$ *and* σ *be a metric over the state-space. For landmark set* $\mathbb{L}$ *and local planner* $\mathcal{P}$*, the set of* **landmark options***, denoted by* $\mathcal{O}$*, contains one option* $o_l = \langle I_l, \pi_l, \beta_l \rangle$ *for each landmark state* $l \in \mathbb{L}$*, where*

1. $I_l = \{x \in X \mid |\mathcal{P}(x, l)| \leq d^+\}$ *is the initialization set,*
2. $\pi_l(x) = \mathcal{P}(x, l)$ *is the policy for* $x \in X$*, and*
3. $\beta_l(x) = \begin{cases} 1 & \text{if } \sigma(x, l) \leq \eta \vee x \notin I_l \\ 0 & \text{otherwise} \end{cases}$ *defines termination probabilities for each state* $x \in X$*.*

In other words, landmark options result from planning on the deterministic MDP $\widehat{M}$, and they terminate once the vicinity of the landmark has been reached. A landmark l's option can only be executed from states where reaching l in the graph would take less than d^+ timesteps. Once discovered, these options will be executed in the target MDP M. We denote the number of valid landmark-option pairs by L. Note that in principle, some landmarks might not be reachable within the given planning horizon.

The idea is to plan using only the set of landmark options. To achieve this, we require that the local planner can derive a path to at least one landmark state from every state in X:

Assumption 4. *For all* $x \in X$ *there exists* $l \in \mathbb{L}$ *such that* $|\mathcal{P}(x, l)| \leq d^+$*.*

In a deterministic MDP, starting in a landmark state, it is possible to avoid visiting non-landmark states (Figure 1a). In this case, it is possible to ignore the other states and plan entirely based on the landmark states. However, in stochastic MDPs, landmark options will not always terminate in landmark states. We solve this problem by allowing landmark options to terminate near landmark states (Figure 1b).

Landmark-based options can be used directly with OFVI or alternatively can be used to create a new AVI algorithm that only maintains estimates of the value function at the finite number of landmark states and therefore avoids explicit function approximation. In this section, we discuss both of these approaches and analyze their convergence properties.

4.1 Landmark-Based Approximate Value Iteration

Landmark-based Approximate Value Iteration (LAVI), Algorithm 2, belongs to the family of AVI algorithms. It takes as arguments: (1) K the number of iterations to perform, (2) a landmark set $\mathbb{L}$, (3) an initial guess V_0 of the value function V^{Φ^*} for states in $\mathbb{L}$, (4) the number of times to sample each landmark-option pair during updates m, and (5) a simulator $\mathbb{S}$. As output, the algorithm produces value estimates for the landmark states $\mathbb{L}$.

Algorithm 2 Landmark-based AVI

```
Require: $K$, $\mathbb{L}$, $V_0$, $m$, $\mathbb{S}$
for $k = 1, 2, \ldots, K$ do
    for $l \in \mathbb{L}$ do
        for $o_l \in \mathcal{O}_l$ do
            $(\tilde{R}^{(j)}_{l,o}, \tau^{(j)}, y^{(j)}) \sim \mathbb{S}(l, o_l)$ for $j = 1, 2, \ldots, m$
        end for
        $V_k(l) \leftarrow \max_{o \in \mathcal{O}_l} \frac{1}{m} \sum_{j=1}^{m} \tilde{R}^{(j)}_{l,o} + \gamma^{\tau^{(j)}} \Delta(V_{k-1}, y^{(j)})$
    end for
end for
return $V_K$
```

Unlike basic VI, LAVI scales to large or infinite MDPs because it only estimates values for the landmark states, while at the same time avoiding the use of complicated function approximation algorithms.

If the target MDP M has deterministic dynamics, then we can ensure that options will always terminate in landmark states. So we can construct backups directly with $V(y)$ where $y \in \mathbb{L}$. However, when M has stochastic dynamics, it may be impossible to guarantee that all options terminate in landmark states. A less restrictive requirement is to assume that options terminate "near" a landmark state with high probability. This notion of closeness requires that we have a metric $\sigma : X \times X \to [0, \infty)$. For some small positive constant η and a state $x \in X$, we define

$$\mathbb{L}_\eta(x) = \{l \in \mathbb{L} \mid \sigma(x, l) < \eta\} \tag{22}$$

to be the set of landmark states that are closer than η to $x \in X$. The function

$$\Delta(V, x) = \begin{cases} \max_{l \in \mathbb{L}_\eta(x)} V(l) & \text{if } \mathbb{L}_\eta(x) \neq \emptyset \ , \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

takes into consideration the fact that options do not necessarily terminate in landmark states. If the distance between the termination state and some landmark l is less than η, then we plug in $V(l)$. Otherwise, we assume a value of 0.

After Algorithm 2 returns its K^{th} estimate of the landmark values V_K, we define the "greedy" policy for LAVI to be

$$\varphi_K(x) = \arg\max_{o \in \mathcal{O}_x} \left(\tilde{R}_x^o + \sum_{t=1}^{\infty} \int \gamma^t P_t^o(y|x) \Delta(V_K, y) dy \right) \ . \tag{24}$$

4.1.1 Analysis

We provide a theoretical analysis of LAVI along two dimensions. (1) We bound the loss associated with policies returned by LAVI compared to the optimal policy over primitive actions, and (2) we analyze the convergence rate of LAVI. To save space, the proofs are deferred to Appendix C.

For deterministic MDPs, $M = \widehat{M}$. Thus, no error is introduced by stochasticity in the environment. However, the selection of landmark states and the local planner can both introduce error.

Definition 8. *(Landmark Error) Given a landmark set $\mathbb{L}$, the smallest $\varepsilon_{\mathbb{L}}$ such that for all $x \in X$ and some $l \in \{l' \in \mathbb{L} \mid \widehat{d}_{\min} \leq |\mathcal{P}(x, l')| \leq d_{\max}\}$*

$$V_{\widehat{M}}^*(x) - \left(\tilde{R}_{p_G^*(x,l)} + \gamma^{|p_G^*(x,l)|} V_{\widehat{M}}^*(l) \right) \leq \varepsilon_{\mathbb{L}} \ , \tag{25}$$

is called the **landmark error** *where $V_{\widehat{M}}^*$ is the optimal value function for $\widehat{M}$ and $\tilde{R}_{p_G^*(x,l)}$ is the discounted reward of the optimal path from x to l in $\widehat{M}$.*

The landmark error quantifies how well the chosen landmark states preserve maximum reward paths. In Figure 1a, the landmark error is represented by the distance of the landmarks from the optimal trajectory. This definition assumes that our local planner is optimal, however, it may be convenient to use a suboptimal local planner.

Definition 9. *(Local Planning Error) Given a local planner $\mathcal{P}$ and landmark set $\mathbb{L}$, the smallest $\varepsilon_{\mathcal{P}}$ such that for all $x \in X$ and $l \in \mathbb{L}$ where $\mathcal{P}(x, l) < d^+$, the path $\mathcal{P}(x, l)$ generated by $\mathcal{P}$ satisfies*

$$\left(\tilde{R}_{p_G^*(x,l)} + \gamma^{|p_G^*(x,l)|} V_{\widehat{M}}^*(l) \right) - \left(\tilde{R}_{\mathcal{P}(x,l)} + \gamma^{|\mathcal{P}(x,l)|} V_{\widehat{M}}^*(l) \right) \leq \varepsilon_{\mathcal{P}} \ , \tag{26}$$

is called the **local planning error**.

The local planning error quantifies the loss due to using the local planner $\mathcal{P}$ instead of a planner that returns the maximum reward path from x to a nearby landmark state. In Figure 1a, the local planning error is represented by the trajectory (dashed line) that transitions from landmark to landmark, but does not follow the shortest path between landmarks.

So far, we have only considered factors that impact planning error when the environment is deterministic. When M contains stochastic dynamics, we need a way to bound the error

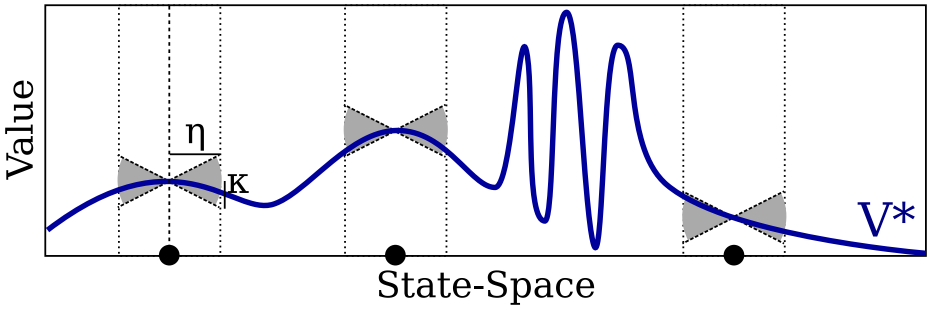


Figure 2: Optimal value function over a one dimensional state space with landmarks depicted by black circles. The gray hourglass shapes around the landmarks depict landmark error. Assumption 5 only requires V_M^* to change slowly around the landmarks. The value function may change rapidly at regions with no landmarks.

of following a policy in M planned in $\widehat{M}$. When M is stochastic, a landmark option may have trouble reaching a particular state. Thus, we need to relax the condition that an option always terminates at a landmark state.

Assumption 5. *(Locally Lipschitz around Landmarks) We are given a metric σ over the state-space X such that for all $l \in \mathbb{L}$ and $x \in X$, if $\sigma(x,l) < \eta$, then $V_M^*(l) \geq V_M^*(x) - \kappa\sigma(x,l)$ for some $\kappa \geq 0$.*

Assumption 5 says that the optimal value function V_M^* does not change too dramatically for states that are close to landmark states. If this assumption is violated, options terminating arbitrarily close to a landmark state may have unboundedly lower value with respect to V_M^*. This would lead to unboundedly suboptimal landmark policies. Thus, Assumption 5 is critical to obtain meaningful bounds on the quality of landmark policies. Notice, however, that Assumption 5 only applies near the landmarks. Figure 2 depicts a value function for a one-dimensional state-space that illustrates the fact that V_M^* can change rapidly in regions of the state-space that are not too close to a landmark.

Assumption 5 allows us to treat the hypersphere with radius η around a landmark state all as one state. However, treating all of these states the same introduces the following error.

Definition 10. *(Local Lipschitz Error) We define the* **local Lipschitz error bound** *by $\varepsilon_H = \kappa\eta$.*

The local Lipschitz error is the largest possible difference between the value at a landmark l and the value at a state within the η-radius hypersphere centered at l. This is essentially the error introduced by allowing landmark options to terminate at any state that is within a distance of η of l.

Now we need to define the error caused by following a policy whose options were planned on $\widehat{M}$ in M.

Definition 11. *(Stochastic Plan Failure) Let σ and η be as in Assumption 5. The* **stochastic planning failure** *ψ is the smallest value such that*

$$\Pr_{(\hat{R},\hat{\tau},\hat{y})\sim \mathbb{S}_M(x,o_l)} [\sigma(\hat{y}, l) > \eta] \leq \psi$$

for all $x \in X$ and $o_l \in \mathcal{O}_x$ where l is the landmark associated with o_l.

The stochastic plan failure bounds the probability that a path to a landmark state planned on $\widehat{M}$ will terminate in a state that is far from the desired landmark state when executed in M.

We also need a way of characterizing how good of an approximation $\widehat{M}$ is for M. It turns out that we can characterize this relationship in a simple way.

Definition 12. *(Relaxation Error) The* **relaxation error** *is*

$$\varepsilon_R = \max \left(\left\| V^*_M - V^{\Phi^*}_M \right\|_\nu - \left\| V^*_{\widehat{M}} - V^{\widehat{\Phi}^*}_{\widehat{M}} \right\|_\nu , 0 \right) \quad ,$$

where Φ^* is the optimal option policy in M and $\widehat{\Phi}^*$ is the optimal option policy in $\widehat{M}$.

Surprisingly, the relaxation error only depends on the difference between the optimal policies over primitive actions and the optimal policies over options in M and $\widehat{M}$. If these policies have similar values then $\widehat{M}$ can be a good approximation for M, even if the dynamics of M are very noisy.

Finally, the **sampling error** ε_S is controlled by m in Algorithm 2. Increasing m corresponds to collecting more samples which consequently decreases ε_S.

Theorem 2. *(LAVI Convergence) Let $\varepsilon_S > 0, \delta \in (0, 1]$. There exists*

$$m = O\left(\frac{1}{(\varepsilon_S (1-\gamma)^2 (1-\gamma^{d_{\min}}))^2} \ln \left(\frac{LK}{\delta} \right) \right)$$

such that with probability greater than $1-\delta$, if Algorithm 2 is executed for $K \geq 1$ iterations, the greedy policy φ_K derived from V_K satisfies

$$\mathcal{L}_{1,\nu}(\varphi_K) \leq \left(\frac{2(\varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}})}{1-\gamma^{\widehat{d}_{\min}}} + \varepsilon_R \right) + \tilde{\varepsilon} + \varepsilon_S + \gamma^{(K+1)d_{\min}} \left(\frac{\left\| V^{\Phi^*}_M - V_0 \right\|_{\mathbb{L}}}{1-\gamma^{d_{\min}}} \right) \quad , \tag{27}$$

where $\tilde{\varepsilon} = \left(\frac{\gamma^{d_{\min}}}{1-\gamma^{d_{\min}}} \right) \left(1 + \frac{(1-\psi)\gamma^{d_{\min}}}{1-\gamma^{d_{\min}}} \right) (\psi V_{\text{MAX}} + (1-\psi)\varepsilon_H)$ and $\widehat{d}_{\min}$ and $d_{\min}$ are the minimum duration of any landmark-option pair in $\widehat{M}$ and M, respectively.

The proof of Theorem 2 appears in Appendix C. Surprisingly, Eq. (27) holds for the initial state distribution even though LAVI only maintains value estimates for states in $\mathbb{L}$.

Although this bound is not directly comparable to bounds derived for FVI, it has many of the same characteristics as the bounds found in the works of Farahmand et al. (2010) and Mann and Mannor (2014). For example, the first three terms on the right hand side of Eq. (27) correspond to the approximation error, ε_S is the estimation error controlled by the level of sampling, and the last term characterizes the convergence behavior of the algorithm. For FVI the approximation error is caused by choosing a function approximation architecture that is not rich enough to represent the estimates of the value function at each iteration. For LAVI the approximation error is caused by choosing a landmark set that is not sufficiently rich or using a planner that cannot reliably reach the vicinity of landmark states.

The first four terms on the right hand side of (27) describe the worst case loss of the policy derived by LAVI as $K \to \infty$. The first term corresponds to the error associated with the choice of landmarks and using a suboptimal local planner. If LAVI uses an optimal local planner, such as A^*, then $\varepsilon_{\mathcal{P}} = 0$. The second term is the relaxation error (discussed more below). $\tilde{\varepsilon}$ is controlled by the stochastic plan failure ψ and local Lipschitz error ε_H. If both, ψ and ε_H are small then $\tilde{\varepsilon}$ will be small. In addition, longer duration options (i.e., larger $d_{\min}$) decreases $\tilde{\varepsilon}$. The sample error ε_S is decreased by increasing m.

The last term corresponds to LAVI's convergence rate and is one of the keys to LAVI's speed. The convergence rate $\gamma^{d_{\min}}$ is faster than γ, the convergence rate of FVI with primitive actions (Munos & Szepesvári, 2008; Mann & Mannor, 2014). The minimum duration $d_{\min}$ is controlled by the minimum time between landmark regions. So convergence is faster when the landmarks provide greater mobility throughout the state-space. A closer look at the last term in Eq. (27) shows that the convergence error depends on $\left\|V_0 - V_M^{\Phi^*}\right\|_{\mathbb{L}}$, which is a max-norm only over the landmark states $\mathbb{L}$. This last term represents the fact that LAVI only needs to estimate the value function at the landmark states.

The relaxation error term ε_R in (27) determines how good of an approximation $\widehat{M}$ is for M given the set of landmark options. One naive bound for ε_R is in terms of a bound on the transition dynamics with respect to the primitive actions

$$\varepsilon_D = \max_{(s,a)\in S\times A} \left\| P(\cdot|s,a) - \widehat{P}(\cdot|s,a) \right\|_1$$

where $\|\cdot\|_1$ is the L_1 norm, P are the transition probabilities for M, and $\widehat{P}$ are the state transitions (degenerate probability distributions with all mass on a single next state) for $\widehat{M}$. It is not difficult to show that $\varepsilon_R \leq \frac{2\varepsilon_D}{1-\gamma}$. However, this bound is generally extremely conservative. ε_R only depends on the loss of the best option policies in M and $\widehat{M}$ whereas ε_D is influenced by primitive actions and states that may never be visited by these option policies.

The total number of samples used by LAVI is LKm, where L is the number of valid landmark-option pairs, K is the number of iterations, and m is the number of landmark-option samples. On the other hand, the number of samples used in the analysis of Fitted Value Iteration depends on the complexity of the function approximation architecture (Munos & Szepesvári, 2008). In MDPs with complex value functions, such as the Pinball domain (see Section 5.2), complex function approximation schemes are necessary to get FVI to work. On the other hand, with an appropriate landmark set, LAVI can simply "skip over" complex regions of the value function.

Notice that executing the policy derived from the output of LAVI requires sampling from the simulator. The number of samples needed depends on the discount factor γ. When γ is close to 1, more samples are needed to ensure that the policy behaves near-optimally. However, this is an acceptable cost when the simulator is relatively inexpensive compared to the overall cost of planning.

4.2 Landmark-based Options Fitted Value Iteration

It is also possible to consider using landmark-based options with OFVI. We refer to the resulting case of the algorithm as Landmark-based Options Fitted Value Iteration (LOFVI).

Theorem 3. *(LOFVI Convergence) Let $\varepsilon_S > 0, \delta \in (0, 1]$ and $\mathcal{O}$ be a set of landmark-based options. If assumption $A2(\nu, \mu)$ (Assumption 2) and $A3(\alpha = 0, d = d_{\min}, \psi = 0, \nu, j = 0)$ (Assumption 3) hold, then there exists n and m, such that with probability greater than $1-\delta$, if Algorithm 1 is executed for $K \geq 1$ iterations, the greedy policy φ_K derived from V_K satisfies*

$$\mathcal{L}_{p,\nu}(\varphi_K) \leq \left(\frac{2(\varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}})}{1-\gamma^{\widehat{d}_{\min}}} + \varepsilon_R\right) + \frac{2\gamma^{d_{\min}}}{(1-\gamma)^2}\mathbb{C}_{\nu,\mu}^{1/p} b_{p,\mu}(\mathbb{T}\mathcal{F}, \mathcal{F}) + \varepsilon_S + \left(\gamma^{d_{\min}(K+1)}\right)^{1/p}\left(\frac{2\left\|V_M^{\Phi^*} - V_0\right\|_\infty}{(1-\gamma)^2}\right) , \quad (28)$$

where $\widehat{d}_{\min}$ and $d_{\min}$ are the minimum duration of any landmark-option pair in $\widehat{M}$ and M, respectively.

Theorem 3 is comparable to Theorem 2 but provides a (p, ν)-norm bound rather than a $(1, \nu)$-norm bound. If we consider the case where $p = 1$, then we can compare Theorem 3 to Theorem 2. Both theorems share the same abstraction loss. Although their convergence rates are similar, LAVI's convergence term (Theorem 2) is significantly smaller than the convergence term for OFVI. However, LOFVI's convergence depends on an explicit representation over the state-space while LAVI's convergence depends only on values maintained at the landmark states.

The main advantage of LOFVI over LAVI is it's potential for lower sample complexity when querying the policy. Although not shown here, it is easy to extend the analysis of OFVI to produce action-value functions rather than value functions. With an explicit action-value function querying the policy does not require any additional samples. This may be an important consideration if the simulator is computationally expensive.

4.3 Additional Considerations

One barrier to applying landmark-based options is that we need access to a deterministic relaxation of the target MDP for local planning. In many domains, a deterministic model may have already been created by domain experts. However, if this relaxation is unavailable, we might wonder how one could be acquired.

One simple strategy for obtaining a deterministic MDP from the target MDP simulator would be to use the most frequently sampled next state. Algorithm 3 demonstrates one possible implementation for this strategy. The algorithm builds a deterministic model $H : X \times \mathcal{O} \to \mathbb{R} \times X$ as new state-action pairs are requested. It takes as arguments the target MDP simulator $\mathbb{S}_M$ a state-action pair (x, a), the number of samples $m \geq 1$, and the partial deterministic model H and returns a 3-tuple containing a reward r and terminal state y. Furthermore, Algorithm 3 can easily be extended to the continuous state setting by matching states that are "close" together. The cost of Algorithm 3 depends on

Algorithm 3 DREX (Deterministic RElaXation)

```
Require: $\mathbb{S}_M, x \in X, a \in A, m \in \mathbb{N}, H : X \times \mathcal{O} \to \mathbb{R} \times X$
if $H(x,a) \neq \bot$ then {Model has no entry for $(x,a)$.}
    for $i = 1, 2, \dots, m$ do
        $(r_i, \cdot, y_i) \sim \mathbb{S}_M(x,a)$
    end for
    $y \leftarrow \arg\max_{i=1,2\dots,m} \sum_{j=1}^{m} \mathbb{I}\{y_i = y_j\}$ {Assign most frequent next state.}
    $r \leftarrow \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\{y = y_i\} r_i$ {Average reward for next state $y$.}
    $H(x,o) \leftarrow (r,y)$
end if
return $H(x,o)$
```

the cost of sampling a primitive action m times, where m can be chosen to ensure that the highest probability terminal state is chosen with high probability. This approach only makes sense when there exists a most probable next state (region) for each state-action pair. Nevertheless, this may capture a wide range of real-world domains.

While we have used the example of a deterministic relaxation for local planning. Landmark options could be implemented using alternative local planning algorithms, for example, UCT (Kocsis & Szepesvári, 2006), Episodic Natural Actor Critic (Peters & Schaal, 2008), etc. These approaches have the advantage that they can be applied directly on the target MDP simulator. The theoretical guarantees provided for deterministic planners can easily be adapted to other black box planners whenever the local planning error can be bounded. However, we focus our analysis on deterministic local planners for brevity and clarity.

5. Experiments and Results

We compared PFVI and OFVI in three different tasks: (1) the optimal replacement problem (Munos & Szepesvári, 2008), (2) the pinball domain (Konidaris & Barto, 2009), and (3) an eight commodity inventory management problem (Mann & Mannor, 2014).

Our theoretical analysis from the previous sections characterizes convergence rates. However, we are also interested in the trade-off between the planning effort and performance (i.e., cumulative reward) of the resulting policy. While it is possible to compare the time-to-solution, this requires setting a potentially arbitrary performance threshold. Choosing a performance threshold can unfairly bias our judgment about which algorithm achieves the best overall performance-time trade-off. We measure this trade-off by introducing the following statistic:

$$\zeta(x,k) = \frac{V^{\varphi_k}(x)}{\sum_{i=1}^{k} t_i} \ , \tag{29}$$

where x is the start state used for evaluation and k refers to the number of iterations performed by the algorithm so far. For $i = 1, 2, \dots, k$ the value t_i is the time in seconds of the i^{th} iteration. Higher values are more desirable because they imply more performance for less time spent planning.

In all of our experiments, we simulated options by simulating individual primitive actions, until the selected option terminates or a maximum number of timesteps (100 in our

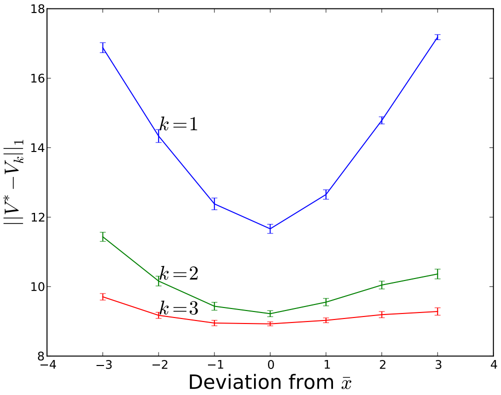


Figure 3: Optimal Replacement: Expected loss of iterates V_1, V_2, and V_3 of OFVI given the primitive actions and a single option of varying quality. Error bars represent ±1 standard deviation. Results were averaged over 20 trials.

experiments) occurs. This potentially places options-based planning methods at a disadvantage. Nevertheless, our experiments provide strong evidence that options can speed up the convergence rate of planning, which leads to a smaller time-to-solution.

All experiments were implemented in Java and executed using OpenJDK 1.7 on a desktop computer running Ubuntu 12.04 64-bit with an 8 core Intel Core i7-3370 CPU 3.40GHz and 8 gigabytes of memory.

5.1 Optimal Replacement Task

In the optimal replacement problem, we only compare PFVI and OFVI with hand crafted options. Due to the simplicity of the task, option generation is unnecessary and we include this task for comparison with previous work. In this problem, the agent selects from one of two actions K and R, whether to maintain a product (action K) at a maintenance cost $c(x)$ that depends on the product's condition x or replace (action R) the product with a new one for a fixed cost C. This problem is easy to visualize because it has only a single dimension, and the optimal value function and optimal policy can be derived in closed form (Munos & Szepesvári, 2008) so that we can compare PFVI and OFVI directly with the optimal policy. We used parameter values $\gamma = 0.6$, $\beta = 0.5$, $C = 30$ and $c(x) = 4x$ (identical to those used in the work of Munos & Szepesvári, 2008) where β is the inverse of the mean of an exponential distribution driving the transition dynamics of the task. Similar to the work of Munos and Szepesvári (2008), we used polynomials to approximate the value function. All results presented here used fourth degree polynomials. The optimal policy keeps the product up to a point $\bar{x}$ and replaces the product once the state equals or exceeds $\bar{x}$.

For the OFVI condition, we introduced a single option that keeps the product up to a point $\tilde{x} = \bar{x} + \Delta$ and terminates once the state equals or exceeds $\tilde{x}$. By modifying Δ, we controlled the optimality of the given option. As predicted by our analysis, adjusting Δ away from 0 (i.e., reducing the option quality), resulted in slower convergence when the

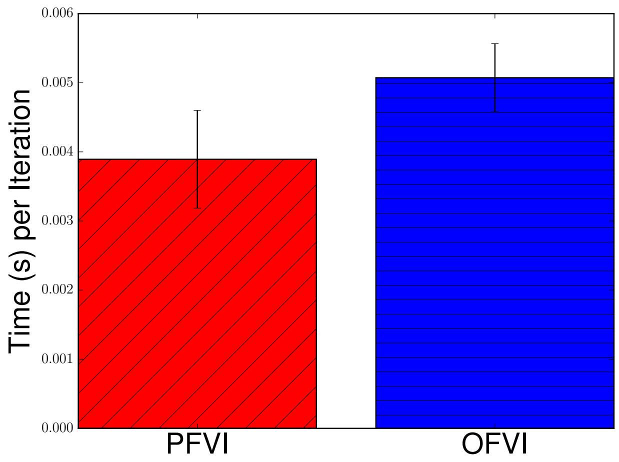
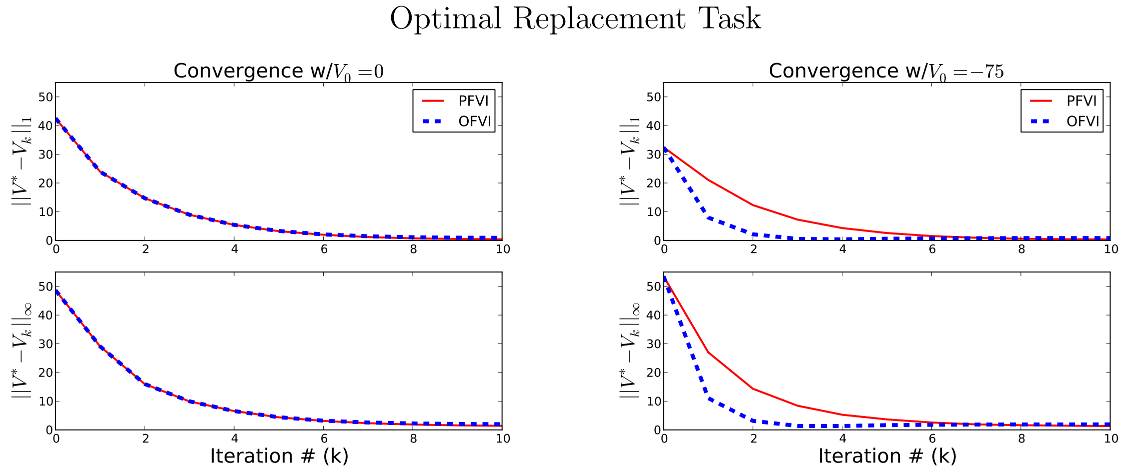


Figure 4: Optimal Replacement: Time in seconds per iteration for PFVI and OFVI on the optimal replacement task averaged over 20 trials.

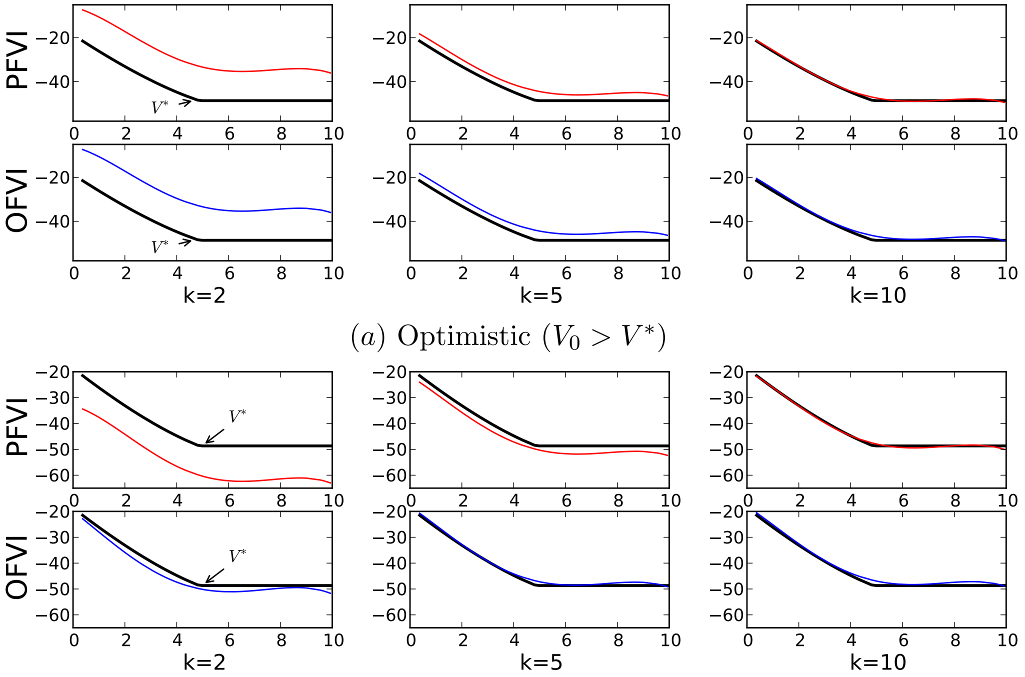


(a) Optimistic $(V_0 > V^*)$

(b) Pessimistic $(V_0 \leq V^*)$

Figure 5: Optimal Replacement: Convergence rates of PFVI and OFVI in the Optimal Replacement Task. (a) When the initial value function estimate is optimistic, there is no difference between the convergence rates of PFVI and OFVI. (b) However, when the value function estimate is pessimistic, OFVI converges faster than PFVI. Results were averaged over 20 trials.

initial value function was pessimistic (see Figure 3). For an optimistic initial value function, the behavior of PFVI and OFVI was almost identical.



(a) Optimistic $(V_0 > V^*)$

(b) Pessimistic $(V_0 \leq V^*)$

Figure 6: Optimal Replacement: Average iterates V_k ($k = 2, 5,$ and 10) for PFVI and OFVI for both (a) optimistic and (b) pessimistic initial value functions. With a pessimistic value function OFVI converges significantly faster than PFVI.

Figure 4 shows that OFVI takes slightly longer per iteration than PFVI, because OFVI considers both primitive and temporally extended actions. Figure 5a shows the average convergence rates of PFVI and OFVI (with $\Delta = 0$), when the initial value function estimate is optimistic for both max-norm and L_1-norm error. In both cases the value functions converge at almost identical rates as predicted by our analysis. Figure 5b shows the average convergence rates of PFVI and OFVI, when the initial value function estimate is pessimistic. With a pessimistic initial value function, OFVI converges significantly faster than PFVI as predicted by our analysis.

Figure 6 compares the average iterates V_k of OFVI to PFVI for $k = 2, 5,$ and 10 with optimistic (Figure 6a) and pessimistic (Figure 6b) initial value function estimates. The solid black line depicts the optimal value function V^*. With an optimistic initial value function the behavior of PFVI and OFVI is qualitatively identical. However, with a pessimistic initial value function, OFVI's second iterate is qualitatively similar to PFVI's fifth iterate.

With a pessimistic initial value function estimate, even suboptimal options were able to improve convergence rates, though to a lesser degree than when $\Delta = 0$.

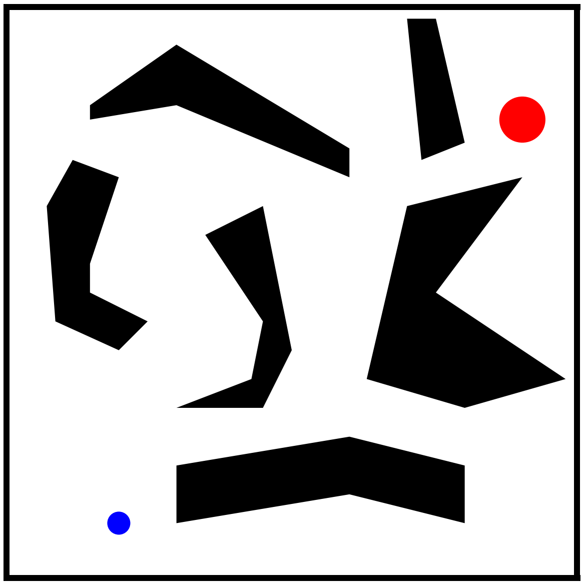


Figure 7: Instance of the pinball domain used in our experiments. Black polygons are obstacles. The large red circle is the target, and the smaller blue circle is the controlled ball.

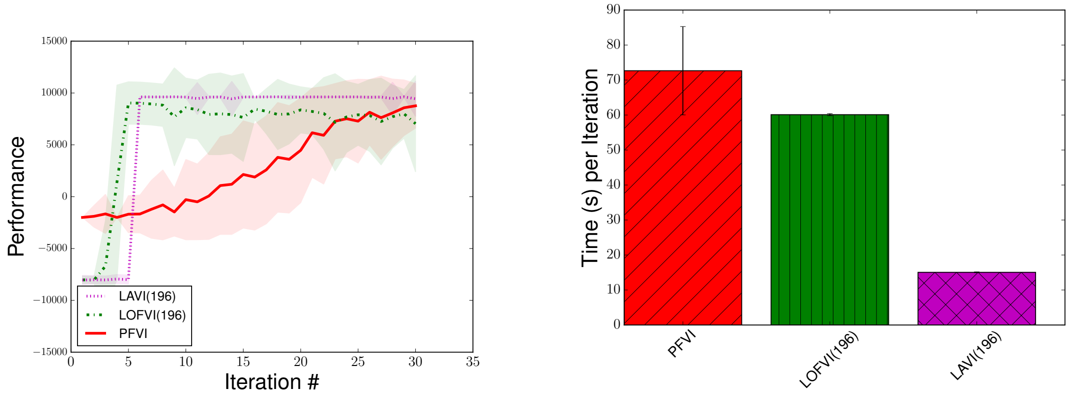
5.2 Pinball

In the Pinball domain (Konidaris & Barto, 2009) the agent applies forces to control a ball on a 2-dimensional surface containing polygonal obstacles. The agent's goal is to direct the ball to a goal region. Figure 7 depicts the instance of the Pinball domain used in our experiments. The state-space consists of four continuous dimensions $(x, y, \dot{x}, \dot{y})$ corresponding to the coordinates (x, y) of the ball and its velocity $(\dot{x}, \dot{y})$. Similar to the work of Tamar, Castro, and Mannor (2013), we added zero-mean Gaussian noise to the velocities with standard deviation 0.03. The discount factor was $\gamma = 0.95$.

The pinball domain contains five primitive actions: (1) accelerate along the X-axis, (2) decelerate along the X-axis, (3) accelerate along the Y-axis, (4) decelerate along the Y-axis, and (5) leave the velocities unchanged. Since it is unclear how to create useful hand-coded options, we decided to only compare PFVI against the LOFVI and LAVI where the options are generated. We experimented with randomly placed landmarks and landmarks placed in a grid. In both cases, one landmark was manually placed at the goal state. Randomly placed landmarks were uniformly sampled over the coordinates of the state-space. Grid landmarks were placed in a two-dimensional grid over the X and Y coordinates the state-space. All landmarks corresponded to states where the ball was at zero velocity. If a sampled landmark fell inside of an obstacle, then a new landmark was sampled so that all landmarks corresponded to valid states of the task.

The metric used to determine the distance between two states $\vec{x} = (x, y, \dot{x}, \dot{y})$ and $\vec{x}' = (x', y', \dot{x}', \dot{y}')$ is given by

$$\mu(\vec{x}, \vec{x}') = \sqrt{(x - x')^2 + (y - y')^2} + \alpha\sqrt{(\dot{x} - \dot{x}'))^2 + (\dot{y} - \dot{y}')^2} \ , \tag{30}$$



(a) (b)

Figure 8: Pinball: Comparison of planning with PFVI, LOFVI, and LAVI with 196 landmarks (+1 at the goal) arranged in a grid in the Pinball domain. (a) Performance over policies derived from each iteration of PFVI, LOFVI, and LAVI. Shaded regions represent ± 1 standard deviation. (b) Time in seconds to compute each iteration of PFVI, LOFVI, and LAVI. Results were averaged over 20 trials.

where α, which places less emphasis on the differences in velocity than the differences in coordinates. We chose $\alpha = 0.01$ through experimentation.

For PFVI and LOFVI, we tried many different function approximation architectures including Radial Basis Function Networks (RBF), Cerebellar Model Arithmetic Computer (CMAC), linear regression with various features, but we found through experimentation that nearest neighbor approximation was both fast and able to capture the complexity of the value function. For LOFVI, we used one-nearest neighbor approximation and $N = 1,000$ states were sampled at each iteration. For PFVI, we averaged the value of states within a 0.1 radius of the queried state and $N = 30,000$ states were sampled at each iteration. Without $30,000$ samples, PFVI either failed to solve the task or produced a policy that solved the task unreliably. Both PFVI and LOFVI used $L = 5$ samples for each state-option pair. We chose these settings because they resulted in the strongest performance for PFVI and LOFVI.

For the landmark options, we experimented with different numbers of landmarks. For simplicity we selected landmarks that formed a uniform grid over the pinball domain's X- and Y-coordinates. By choosing grid sizes of $10 \times 10, 12 \times 12$, and 14×14, the number of landmarks were $100, 144, 196$, respectively. With fewer than 100 landmarks performance of LOFVI and LAVI degraded significantly. The radius of the hypercube around landmarks was set to $\eta = 0.03$, and the landmark options available at a state corresponded to the landmarks that were at a distance less than 0.2 from the balls current state, which approximates a local planning horizon d^+. For brevity, we consider the results for landmark options arranged in a grid. Randomly selected landmarks gave qualitatively similar results with slightly higher variance.

Figure 8a compares the performance of PFVI, LOFVI, and LAVI in the pinball domain with 196 landmarks (+1 landmark at the goal). After about six iterations, LOFVI and

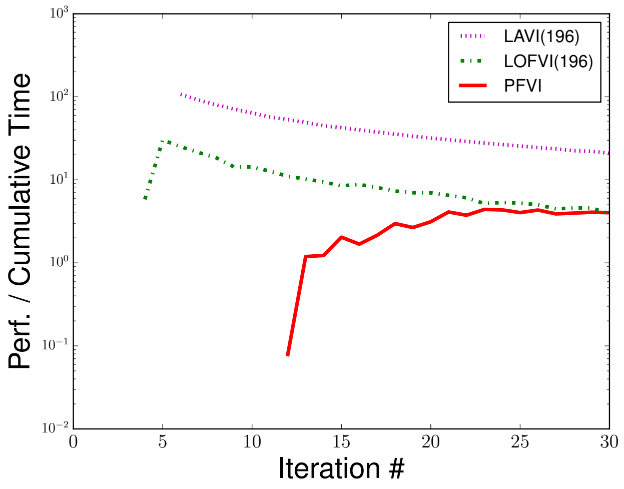


Figure 9: Pinball: Performance over cumulative time in seconds received by policies from each iteration. Higher is better. Results were averaged over 20 trials.

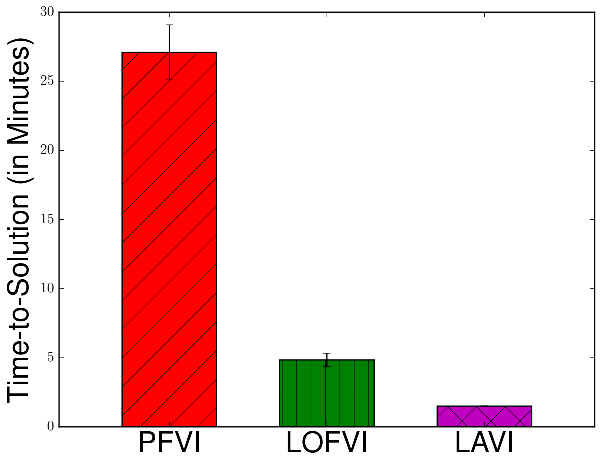


Figure 10: Pinball: Cumulative time-to-solution in minutes for PFVI, LOFVI, and LAVI averaged over 20 trials. LAVI has the smallest time-to-solution.

LAVI are able to solve the task. However, PFVI takes about 25 iterations to solve the task. Figure 8b compares the time per iteration of PFVI, LOFVI, and LAVI in the pinball domain. PFVI has the highest time per iteration cost. This is because we needed to use a lot more samples per iteration for PFVI to solve the task. Notice that LAVI is less expensive than LOFVI. This is due to the fact that LAVI only needs to sample from landmark states, whereas LOFVI samples from a larger number of states sampled at each iteration (although less than PFVI).

Figure 9 compares the performance over cumulative time spent planning. PFVI has a poor performance over time trade-off because each iteration is takes more time than LOFVI and LAVI, and it takes many iterations to achieve high performance. LOFVI and LAVI achieve similar performance, but LAVI has a higher score due to the fact that it spends less time planning per iteration. Figure 10 compares the time in minutes before PFVI, LOFVI, and LAVI produce a policy that achieves performance greater than 8,000. PFVI takes a longer than LOFVI and LAVI, because it converges slowly and uses an expensive function approximation step at each iteration. Despite the fact that LOFVI and LAVI both use the same landmark options, LAVI is faster than LOFVI, because LAVI only approximates the value function around landmark states.

5.3 Inventory Management Task

In a basic inventory management task, the objective is to maintain stock of one or more commodities to meet customer demand while at the same time minimizing ordering costs and storage costs (Scarf, 1959; Sethi & Cheng, 1997). At each time period, the agent is given the opportunity to order shipments of commodities to resupply its warehouse.

We created an inventory management problem where the agent restocks a warehouse with $n = 8$ different commodities (Mann, 2014). The warehouse has limited storage (500 units in our experiments). Demand for each commodity is stochastic and depends on the time of year. Orders can be placed twice each month for a total of 24 order periods per demand cycle.

The state $\langle \tau, x \rangle$ of the inventory management problem is a vector specifying the time of year τ and a vector x specifying the quantity of each commodity (denoted x_i for the i^{th} commodity) stored in the warehouse. During each timestep, a demand vector ξ is drawn by sampling the demand for each commodity independently from a normal distribution where the mean depended on the time of year (see Table 1 for parameters used in our experiments). The demand vector is then subtracted from the quantity of each commodity stored in the warehouse. If any of the commodities were negative after subtracting the demand vector, the agent receives an unmet demand penalty

$$p_{\text{ud}}(x - \xi) = \begin{cases} u_b + u_s \sum_{i=1}^{n} [x_i - \xi_i]_- & \text{if } \sum_{i=1}^{n} [x_i - \xi_i]_- < 0 \ , \\ 0 & \text{otherwise} \ , \end{cases} \tag{31}$$

where $u_b = 2$ represents the base unmet demand cost, $u_s = 10$ represents the per unit unmet demand cost, and $[x]_- = \begin{cases} x & \text{if } x < 0 \\ 0 & \text{otherwise} \end{cases}$.

Once the demand is subtracted, the agent is given the opportunity to either resupply its warehouse or order nothing. The set of possible primitive actions is $n^{501} = 8^{501}$. Searching over this set would be intractable. Therefore, we designed a smaller set of primitive actions.

Table 1: Commodity Properties

Commodity Index	1	2	3	4	5	6	7	8
Unit Cost ($o_{s,i}$)	1	3	1	2	0.5	1	1	1
Demand Peak (Month)	1	3	7	10	8.5	12	1	5.5
Demand Std. Deviation	2	1	2	3	2	2	1	2
Max. Expected Demand	16	10	20	4	10	9	20	16

The primitive actions available to the agent were the ability to order nothing or to order any single commodity in quantities of 25 up to the maximum size of the warehouse. This resulted in $(8 \times (500/25)) + 1 = 161$ primitive actions. An action $a = \langle i, q \rangle$ is defined by a commodity index i and a quantity q. The cost of an order is defined by

$$p_{\text{oc}}(i, q) = \begin{cases} 0 & \text{if } q = 0 \ , \\ o_b + o_{s,i} q & \text{otherwise} \ , \end{cases} \tag{32}$$

where $o_b = 8$ is the base ordering cost and $o_{s,i}$ (see Table 1) is the commodity dependent unit cost. The new state steps forward half a month into the future and the quantities in the inventory are updated to remove the purchased commodities and add the ordered commodities (if any). If the agent orders more than will fit in the inventory, then only the portion of the order that fits in the warehouse will be kept (but the agent will be charged for the complete order). At the end of each decision step, the agent receives a negative reward (i.e., cost)

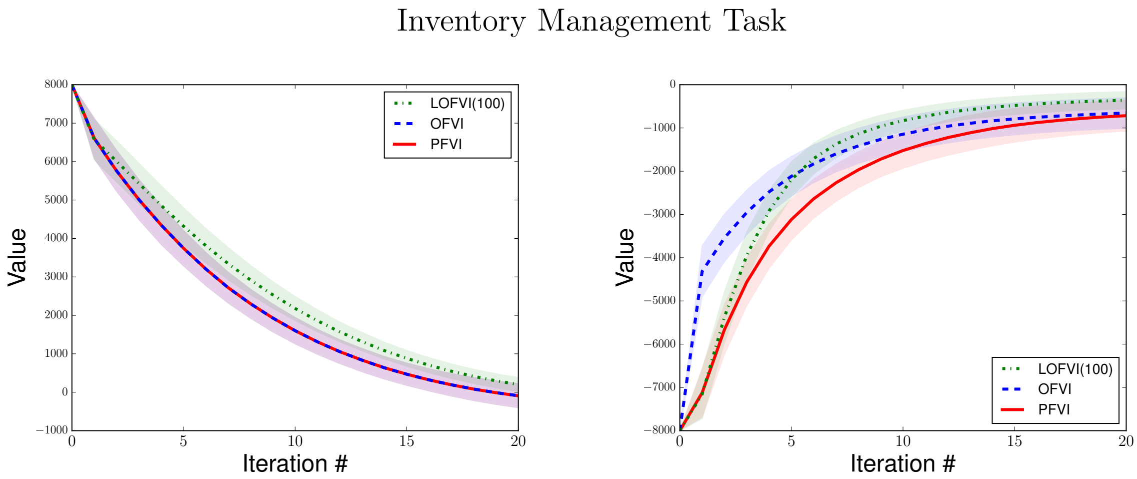
$$R(x, a) = -\left(p_{\text{ud}}(x - \xi) + p_{\text{oc}}(i, q) \right) \ , \tag{33}$$

which is the negative of the sum of the unmet demands and the order costs depending on the inventory levels x, the demand ξ, and the action $a = \langle i, q \rangle$. There is no storage cost, but the limit on the inventory forces the agent to make careful decision about which commodities to order. The discount factor was $\gamma = 0.9$.

The high dimensionality of the state space (1 dimension for each commodity and 1 dimension for time) required a function approximation architecture with good generalizability. We tried many different function approximation architectures before settling on Radial Basis Function networks (RBFs) with a grid of 1-dimensional radial bases. By limiting the dimensionality of the radial bases we were able to achieve good generalization performance with few samples. We divided the state space into 24 time periods, so that the value function approximation was implemented by 24 RBFs. The the number of bases per dimension was 25 and the basis widths were controlled by $\sigma = 0.1$. Throughout the experiments we sampled $n = 1000$ states each iteration and sampled each option $m = 20$ times.

5.3.1 Hand-crafted Options

It is difficult to design a good policy for this problem by hand. Inventory management has received a lot of attention from the operations research community. One of the main findings is that the optimal strategy for a large class of inventory management problems belong to a simple family of policies called (s, S)-policies (Scarf, 1959). For a problem with a stationary demand distribution and a single commodity, an (s, S)-policy orders enough stock to bring the inventory level up to S whenever the inventory level falls below s and



(a) Optimistic ($V_0 > V^*$) (b) Pessimistic ($V_0 \leq V^*$)

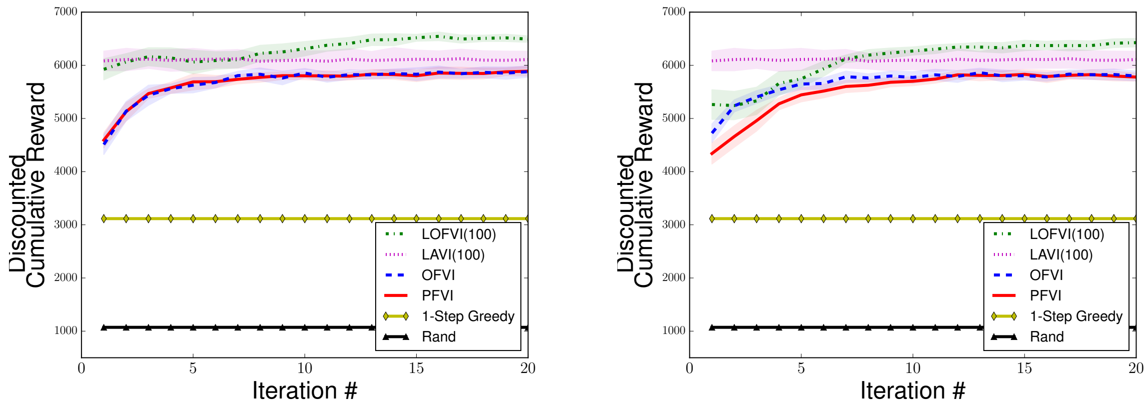
Figure 11: Inventory Management: Average value of iterates produced by LOFVI, OFVI and PFVI. Results were averaged over 20 trials. Shaded regions represent ±1 standard deviation.

orders no new stock otherwise. In an inventory management task with Markov demand, (s, S)-policies are optimal for each demand state (Sethi & Cheng, 1997).

While the problem described here does not cleanly fit into the Markov demand setting, the notion of (s, S)-policies provides a potential idea for a temporally extended action. Since there is a high base order cost (Eq. (32) and $o_b = 8$) while storage is free, a reasonable policy should prefer to make large orders whenever possible and maximize the number of timesteps where nothing is ordered. One way to encode this prior knowledge is to provide temporally extended actions that follow the policy "order nothing" until some threshold is met. In addition to the primitive actions, we provided OFVI with 20 temporally extended actions for each commodity. The policy followed by all of these temporally extended actions was to order nothing and terminate once the inventory level for a particular commodity fell below a constant level (one of 20 levels spanning from 0 to the maximum storage of the warehouse).

Since we do not know the optimal value function for this problem, we cannot compare the iterates of PFVI, OFVI, and LOFVI to a ground truth. However, we can still examine their iterates. Figure 11a shows the average iterates produced by PFVI, OFVI, and LOFVI with optimistic V_0. In this case, we can see that PFVI, OFVI, and LOFVI appear to decrease at similar rates. Figure 11b shows the average iterates produced by PFVI, OFVI, and LOFVI with pessimistic V_0. Here we see that OFVI and LOFVI increase their values (toward V^*) more quickly than PFVI. LOFVI appears to converge to a different solution than PFVI and OFVI, which is probably due to the fact that the landmark options used in this experiment may be more powerful than the actions available to PFVI and OFVI. Note that a comparison to the value function produced by LAVI is not straightforward because it does not produce an approximation for all states (only the landmark states).

Inventory Management Task



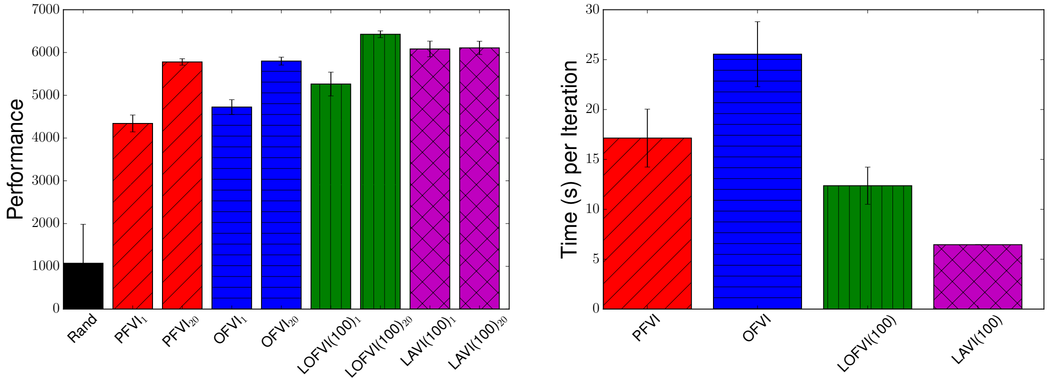
(a) Optimistic $(V_0 > V^*)$ (b) Pessimistic $(V_0 \leq V^*)$

Figure 12: Inventory Management: Performance of policies at each iteration of LAVI, LOFVI, OFVI and PFVI starting from a state with no inventory. LAVI settles on a near-optimal policy after a single iteration. Results were averaged over 20 trials. Shaded regions represent ±1 standard deviation.

We considered the performance of the policies derived from iterates of PFVI and OFVI. When V_0 is initialized pessimistically, Figure 12b shows that OFVI quickly converges to a better policy than PFVI. It takes PFVI several more iterations before an equally successful policy is found. We compared these policies with a policy that selected uniformly at random from the available primitive actions (denoted Random) and a policy that selects the action that has the immediate lowest cost (denoted 1-Step Greedy). The initial state was set to the beginning of the year with zero inventory. For the case where V_0 is optimistically initialized, Figure 12 shows that the performance of PFVI and OFVI quickly improve beyond the Random and 1-Step Greedy policies, but their performance is similar at each iterate.

5.3.2 Landmark-Based Options

For LAVI, we set $m = 20$ (where m controls the number of samples per state-option pair) after experimentation showed that this value works reasonably well. Since the state-space of the problem was too large to use a generic graph-based planner, we constructed a heuristic local planner that used a deterministic instance of the problem to transition as close as possible to landmark states. Given the definition of the inventory management task, it is easy to define a deterministic model by replacing samples from the Gaussian distributions with the expectation of those distributions. Given a landmark l, if the current state was lower than l for the i^{th} commodity, then it would order the amount needed to reach the i^{th} commodity's quantity in l plus the expected demand for the i^{th} commodity. If the current state was higher than l for the i^{th} commodity, it would place no order for that commodity. Notice that this local planner is efficient and is able to make use of the entire set of primitive actions. We used Euclidean distance and set $\eta = 0.05 \times 500$ where 500 was the maximum



(a) (b)

Figure 13: Inventory Management: (a) Comparison of performance of the first and last policies derived by PFVI, OFVI, and LAVI. (b) Comparison of time per iteration in seconds. Results were averaged over 20 trials. Error bars represent ±1 standard deviation.

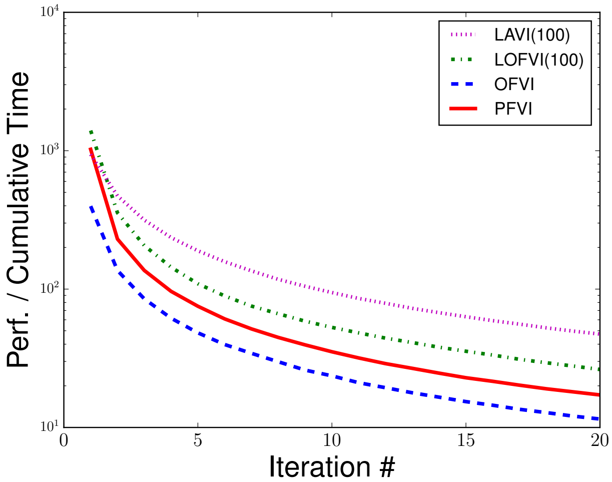


Figure 14: Inventory Management: Comparison of performance over cumulative time in seconds (higher is better). LAVI achieves higher performance for time invested even compared to LOFVI which also uses landmark options. Results were averaged over 20 trials.

inventory level and $d^+ = \infty$. The reason we set $d^+ = \infty$ was because successfully managing inventory requires making large jumps in the state-space (e.g., going from 0 inventory to maximum inventory levels) in a single timestep.

Figure 13a compares the performance of a policy that selects primitive actions uniformly at random and policies derived from the first and last iterates of PFVI, OFVI, LOFVI, and

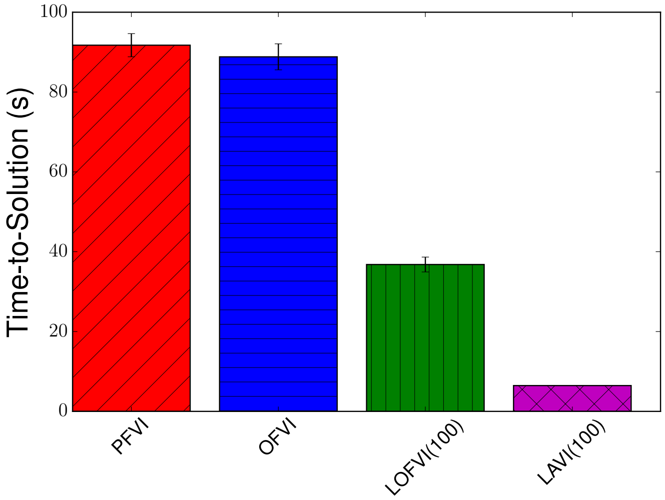


Figure 15: Cumulative time-to-solution (with performance threshold 5,500) averaged over 10 independent trials for PFVI, OFVI, LOFVI with 100 landmarks, and LAVI with 100 landmarks. LAVI has the smallest time-to-solution.

LAVI. In this task, LAVI outperforms PFVI and OFVI after on its first iteration, while LOFVI ultimately has higher performance. Figure 13b compares the time per iteration in seconds of PFVI, OFVI, LOFVI, and LAVI. In this task, LAVI is significantly faster than PFVI, OFVI, and LOFVI. Figure 14 shows that LAVI achieves a better performance-time trade-off on all iterations.

Figure 15 shows the cumulative time in seconds to derive a policy that achieves performance above 5,500 averaged over 10 independent trials. OFVI, which uses a mixture of options and primitive actions, has a slightly faster time-to-solution than PFVI, despite having to evaluate both primitive and temporally extended actions at each iteration. Thus, OFVI's faster time-to-solution is due to its faster convergence rate. LOFVI and LAVI both have a smaller time-to-solution than PFVI and OFVI. Although there are approximately the same number of primitive actions and landmark options that can be initialized at each state, LOFVI and LAVI are faster than PFVI because they converge faster. Finally, LAVI is faster than LOFVI because it uses a computationally efficient estimate of the value function based only at the landmark states, whereas LOFVI (1) uses a potentially more expensive function approximation architecture and (2) does not make explicit use of the landmark state locations.

5.4 Experimental Domains versus Theoretical Assumptions

It is a useful exercise to consider how our theoretical assumptions map onto our experimental domains.

First, we consider the concentrability coefficient (Assumptions 1 and 2). Unfortunately, we cannot estimate the concentrability coefficients for our domains because they depend on a supremum norm over sequences of policies. However, the concentrability coefficients are generally smaller in stochastic domains, where every policy has a broad future state distribution (meaning that the long-term value of a state depends a little bit on lots of states). Along this line of reasoning, we should expect that the optimal replacement and inventory management problems might have smaller concentrability coefficients than the pinball domain. Consistent with this hypothesis, we found that all of our algorithms were much more sensitive to the sampling distribution in the pinball domain than the other two domains.

In addition, Lemma 1 suggests that with options the concentrability coefficient is always less than (or equal to) the concentrability coefficient for primitive actions. This is also supported by our experiments with the pinball domain, where PFVI was much more sensitive to the sampling distribution than LOFVI or LAVI.

Second, let us consider Assumption 3 with respect to our experimental domains. Assumption 3 deals with the sparseness of options in the state-space. Informally, it says that nearly-optimal temporally extended actions are abundant in the state-space. With landmark options, this holds true for the pinball and inventory management domains by definition, and as a result LAVI and LOFVI achieve fast convergence. Our hand-crafted options in the optimal replacement and inventory management domains, on the other hand, may terminate immediately in some states and have a long duration in others. This may account for why landmark-based options resulted in faster convergence in our experiments.

Now we will consider the assumptions and definitions from the analysis of LAVI. The locally Lipschitz assumption probably holds for the inventory management domain because the high stochasticity generally smooths the value function. In the pinball domain, there are regions of the state-space that are probably not Lipschitz due to the complex obstacles. Two states on the opposite side of an obstacle can be relatively close but have extremely different values. However, LAVI only needs the locally Lipschitz assumption to hold around landmark states. Since the majority of the state-space in the pinball domain is probably smooth, the local Lipschitz assumption likely holds for most landmark configurations from our experiments.

Landmark error decreases when we add more landmark states. In the pinball and inventory management domains the task could not be solved with too few landmarks. However, we were surprised that in the pinball and inventory management domains only 100 landmarks were needed to learn reasonable solutions. Furthermore, in the pinball domain, how landmark states were chosen did not have a large effect on the performance. Thus a grid-based layout of landmarks only produced slightly better policies than uniformly sampled landmark states. This suggests that landmark error could be made small in our experimental domains with a small number of landmarks.

Local planning error is the error due to using an imperfect deterministic planner. For the inventory management task, the local planning error was 0, because we were able to use the deterministic model to specify what to order to get to a landmark state. In the pinball domain, the local planning error may be large because we used a greedy algorithm to select actions that move the agent in a straight line toward the landmark. This local planner will fail when the ball needs to be pushed around a corner, but it worked well in our experiments. This suggests that the local planning error may have been small on average.

Stochastic plan failure occurs when noise in the environment prevents a landmark option from terminating sufficiently close to the designated landmark state. In our analysis of

LAVI, even a small probability ψ of a stochastic plan failure caused a large increase in the approximation error. This is due to the possibility that failing to reach a landmark may leave the agent in a non-recoverable state. However, in the pinball (Konidaris & Barto, 2009) and inventory management (Mann & Mannor, 2014) domains, the agent can eventually recover from any mistake. So stochastic plan failure generally has a much smaller consequence than what is predicted by (27).

The relaxation error quantifies how much worse the best landmark option policy is in the target MDP than in the deterministic relaxation. Despite the fact that the pinball domain and the inventory management domain have significant stochasticity, LAVI and LOFVI were able to derive good policies. Unfortunately, it is not clear how to measure the relaxation error.

6. Discussion

We have proposed and analyzed Options Fitted Value Iteration and Landmark-based Approximate Value Iteration. For both algorithms longer temporally extended actions result in faster convergence and smaller approximation error. For OFVI, our analysis shows that when the value function estimate is pessimistic with respect to the optimal value function, the convergence rate of OFVI can take advantage of temporally extended actions that have a smaller effective discount factor than the options with minimum duration. Furthermore, options can improve convergence even when they are suboptimal and spread throughout the environment. In fact, LAVI and LOFVI both converge faster as landmark-based options are spread out further in the environment.

Approximate Modified Policy Iteration (Scherrer, Ghavamzadeh, Gabillon, & Geist, 2012) is related to planning with options in the sense that modified policy iteration performs backups from d-step rollouts (rather than 1-step rollouts) of the greedy policy. However, planning with options is more flexible because the options can have termination conditions that depend on state and time. Furthermore, the analysis from the work of Scherrer et al. (2012) does not point to any improvement in convergence rates by increasing the length of the rollouts used to perform backups.

Special representations such as factorization of a task's state and action spaces can be exploited to achieve faster planning (Hoey, St-Aubin, Hu, & Boutilier, 1999; Barry, Kaelbling, & Lozano-Prez, 2011). However, for many problems a simulator already exists or simulators are a more convenient way to represent the task. In fact, the work of Dietterich, Taleghan, and Crowley (2013) presents an example where the simulator representing the task is computationally efficient, but exact inference on the factored representation of the task is computationally intractable. Therefore, the ability to plan on black box simulators is more generally applicable than requiring a problem to be in some special representation. This is why we focus on planning with a black box simulator.

Option discovery has been investigated extensively, and many approaches explore heuristics related to finding useful subgoals (McGovern & Barto, 2001; Simsek & Barto, 2004; Stolle & Precup, 2002; Wolfe & Barto, 2005), which is similar in spirit to finding landmarks. In all of these approaches, however, the emphasis is on finding only useful subgoals. Our analysis provides instead a way to use any arbitrary set of landmarks, and quantify the quality of the obtained policy. Because of this less careful approach in selecting landmarks, and because of the use of local planning on a deterministic problem, the scalability of LAVI is significantly better, especially in high-dimensional problems.

Given a collection of policies, the works of Comanici and Precup (2010) and Mankowitz, Mann, and Mannor (2014) have investigated creating useful options by applying option interruption. Both of these methods rely on being given a collection of policies. Here we make use of a deterministic local planner instead, which gives LAVI and LOFVI more flexibility since they are not restricted to a few predefined policies.

For clarity, we have focused on learning a good approximation of the optimal value function and then showed that the resulting greedy policy has bounded loss. However, in practice we cannot directly obtain the greedy policy from a value function. It must be approximated with samples. However, our results can easily be extended to handle this by the same arguments used to prove Thm. 3 in the work of Munos & Szepesvári, 2008 or by approximating the action-value function (Farahmand, Ghavamzadeh, Szepesvári, & Mannor, 2008).

For brevity and generality, we have presented an analysis of the convergence behavior of AVI algorithms (not computational complexity). It is possible using the bounds in Theorem 1 and Theorem 2 to obtain bounds on computational complexity. However, there are two critical decisions needed to determine the computational complexity. The first is the computational complexity of sampling options. For example, the smart grid simulator Gridlab-d can efficiently simulate actions at multiple timescales (Chassin et al., 2014). On the other hand, some simulators may require sampling the outcome of a sequence of primitive actions. The second decision involves the choice of function approximation, which can vary widely.

Planning with options is an important setting because options are a more natural model for settings where decisions are made at irregular time intervals. Furthermore, algorithms that plan with options can potentially make use of the many algorithms proposed for learning options from data (Iba, 1989; Mannor et al., 2004). However, which algorithms produce good options for planning is an open question, since the majority of previous research has considered generating options for exploration. Our analysis of landmark-based options helps to address this question because landmark-options are similar in spirit to many existing techniques for option generation, such as skill chaining (Konidaris & Barto, 2009) and bottleneck discovery (McGovern & Barto, 2001; Simsek & Barto, 2004).

Options may have other benefits for planning besides improving the convergence rate (and thus the overall speed of planning). For example, options may enable a planning algorithm to “skip over” regions of the state space with highly complex dynamics without impacting the quality of the planned policy. In particular, LAVI only models the value function around the landmark states, which allows it to perform well in tasks where the value function is highly nonlinear (such as the Pinball domain in Section 5.2). In partially observable environments, options may be exploited to decrease uncertainty about the hidden state by “skipping over” regions of the state space where there is large observation variance, or “testing” hypotheses about the hidden state. Options may also play an important role in robust optimization, where the dynamics of temporally extended actions are known with greater certainty than the dynamics of primitive actions. In fact, macro-actions have already been used for planning in partially observable environments with some success (He, Brunskill, & Roy, 2011). However, these results only consider a very narrow definition of temporally extended actions that excludes closed loop policies such as options.

We have focused on generalizations of value iteration, but there are many other algorithms where planning can benefit from options. For example, approximate policy iteration (Lazaric, Ghavamzadeh, & Munos, 2010; Scherrer et al., 2012) may also exploit options to speed up convergence. Another interesting family of planning algorithms is the sparse sampling framework, which estimates the value at a single state using either breadth-first-like search (Kearns, Mansour, & Ng, 2002) or rollouts (Kocsis & Szepesvári, 2006). Options may enable sparse sampling algorithms to derive higher quality policies with a smaller dependence on the horizon.

For option generation, we assumed the existence of an efficient local planner. For many applications it may be much easier to create and/or learn an efficient local planner than a global planner. This is especially true in domains where the local dynamics tend to remain similar throughout large regions of the environment (Brunskill, Leffler, Li, Littman, & Roy, 2008).

Acknowledgments

This work was funded in part by the NSERC Discovery grant program and the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n.306638.

Appendix A. Proof of Proposition 1

Proof. (of Proposition 1) This proposition follows from Theorem 1. To see why, consider any $Z \geq 0$, there is at least one optimal policy π^* defined over primitive actions that satisfies Assumption 3 with values $\alpha = 0$, $d = 1$, $\psi = 0$, arbitrary $\nu \in M(X)$, and $j = 0$. In this case, Theorem 1 gives us the following high probability ($> 1 - \delta$) bound with $\alpha = 0$:

$$\begin{aligned} ||V^* - V^{\pi_K}||_{p,\nu} &\leq \tfrac{2\gamma}{(1-\gamma)^2}\mathbb{C}_{\nu,\mu}^{1/p}\left(b_{p,\mu}(\mathbb{T}\mathcal{F},\mathcal{F}) + \alpha\right) + \varepsilon \\ &+ \left(\left(\gamma^{(1-\psi)d+\psi}\right)^Z + \gamma^{K-Z+1}\right)^{1/p}\left(\tfrac{2\|V^*-V_0\|_\infty}{(1-\gamma)^2}\right) \\ &\leq \tfrac{2\gamma}{(1-\gamma)^2}C_{\nu,\mu}^{1/p}b_{p,\mu}(\mathbb{T}\mathcal{F},\mathcal{F}) + \varepsilon + \left(\gamma^{K+1}\right)^{1/p}\left(\tfrac{2\|V^*-V_0\|_\infty}{(1-\gamma)^2}\right) \quad , \end{aligned}$$

where we replace $\mathbb{C}_{\nu,\mu}$ with $C_{\nu,\mu}$ since Lemma 1 tells us that $\mathbb{C}_{\nu,\mu} \leq C_{\nu,\mu}$. □

Appendix B. Proof of Theorem 1 and Supporting Lemmas

In this appendix, we prove Theorem 1 and provide sufficient values for the arguments n and m, where n controls the number of states sampled at each iteration and m controls the number of samples simulated at state-action pairs.

The proof of Theorem 1 is similar in structure to Thm. 2 in the work of Munos & Szepesvári, 2008 with several changes due to the differences between options and primitive actions. The proof of Theorem 1 has the following structure:

1. In Appendix B.1, we derive Lemma 2, which bounds the number of states n and the number of samples m from each state-option pair that are necessary to achieve a high-probability bound on the error of a single iteration of AVI. Lemma 2 is used directly in the proof of Theorem 1 but not in any of the other supporting lemmas.

2. In Appendix B.2, we derive Lemma 6, which provides a pointwise bound on the loss of the policy produced by OFVI after $K \geq 1$ iterations. We start by deriving an upper bound for the policy's pointwise loss based on the value function's pointwise error (Lemma 3). To use Lemma 3, we need bounds on the value function estimate's pointwise error. So we derive upper and lower bounds on its pointwise error after K iterations (Lemma 4). Lemma 6 puts Lemmas 3 and 4 together and exploits options that follow a near-optimal policy to get a tighter bound when the estimate of the value function is pessimistic. For technical reasons, it is important in the next part of the proof (Appendix B.3) that the coefficients in the pointwise bound sum to 1. Therefore, we introduce coefficients λ_k for $k = 0, 1, 2, \ldots, K$ and show that they do indeed sum to 1 (Lemma 5).

3. In Appendix B.3, we convert the pointwise bound derived in Appendix B.2 into an L_p-norm bound, as well as, deriving the convergence behavior of OFVI (Lemma 8). Lemma 7 shows how the concentrability assumption (Assumption 2) allows us to replace the error according to the future state distribution with the error according to our sampling distribution. We use Lemma 7, as well as, Assumption 3 to prove Lemma 8.

4. Appendix B.4 proves Theorem 1. The proof uses Lemma 2 to select the number of samples needed to ensure that the error at all $K \geq 1$ iterations is low with high probability. Then we apply Lemma 8 to bound the error after K iterations.

Before moving onto the proofs, we first introduce some additional notation. In contrast to the discounted termination state probability density $\widetilde{P}$, we denote the undiscounted probability that an option o executed from a state $x \in X$ will terminate in a subset of states $Y \subseteq X$ by

$$P^o(Y|x) = \sum_{t=d_{\min}}^{\infty} P_t^o(Y|x) \ . \tag{34}$$

Notice that because (34) is undiscounted $\int \widetilde{P}^o(y|x)dy < \int P^o(y|x)dy = 1$. For an option policy $\varphi : X \to \mathcal{O}$, we will denote by $\widetilde{P}^\varphi$ discounted termination state probability distribution for executing φ once at each state (executing each option until termination) and the undiscounted termination state probability distribution P^φ analogously. Notice that for an option policy, we also have

$$P^\varphi(Y|x) = \sum_{t=d_{\min}}^{\infty} P_t^\varphi(Y|x) \tag{35}$$

for all $Y \subseteq X$ and $x \in X$.

Notice that if f is an option, an option policy, or a policy over primitive actions we can write the discounted termination state probability density by

$$\widetilde{P}^f(Y|x) = \sum_{t=d_{\min}}^{\infty} \gamma^t P_t^f(Y|x) \tag{36}$$

for all $Y \subseteq X$ and $x \in X$. When we compose options $o_1, o_2, \ldots o_m$, we write $\widetilde{P}^{o_1 o_2 \ldots o_m} = \widetilde{P}^{o_1}\widetilde{P}^{o_2}\ldots\widetilde{P}^{o_m}$, and we can write

$$\widetilde{P}^{o_1 o_2 \ldots o_m}(Y|x) = \sum_{t=m d_{\min}}^{\infty} \gamma^t \left(P^{o_1}P^{o_2}\ldots P^{o_m}\right)_t (Y|x) \tag{37}$$

for all $Y \subseteq X$ and $x \in X$.

We will assume throughout this supplementary material that when we refer to an optimal policy π^*, it is a policy over primitive actions. When $\mathcal{O}$ contains the set of primitive actions A, the fixed point of the SMDP Bellman operator $\mathbb{T}$ and the MDP Bellman operator $\mathcal{T}$ is the optimal value function V^*. Thus $\mathbb{T}^{\pi^*}$ is equivalent to $\mathcal{T}^{\pi^*}$.

B.1 Bounding the Number of Samples

The following lemma is used in Theorem 1 to select sufficient values for parameters n and m to ensure that the per iteration error is less than some $\varepsilon > 0$ with probability at least $1 - \delta$.

Lemma 2. *Let $\mathcal{M}$ be an SMDP with option set $\mathcal{O}$, $\mathcal{F} \subset B(X; V_{\text{MAX}})$ be a bounded function space with $\left(\frac{1}{8}\left(\frac{\varepsilon}{4}\right)^p, p\right)$-covering number bounded by $\mathcal{N}$, $V \in \mathcal{F}$, p be a fixed positive integer, and V' be the result of a single iteration of OFVI derived from (13) followed by (8). For any $\varepsilon, \delta > 0$,*

$$\left\|V' - \mathbb{T}V\right\|_{p,\mu} \leq b_{p,\mu}\left(\mathbb{T}V, \mathcal{F}\right) + \varepsilon$$

holds with probability at least $1 - \delta$ provided that

$$n > 128\left(\frac{8V_{\text{MAX}}}{\varepsilon}\right)^{2p}\left(\log(1/\delta) + \log(32\mathcal{N})\right) \tag{38}$$

and

$$m > \frac{8(R_{\text{MAX}} + \gamma V_{\text{MAX}})^2}{\varepsilon^2}\left(\log(1/\delta) + \log(8n|\mathcal{O}|)\right) \quad . \tag{39}$$

The proof of Lemma 2 follows from the proof of Lemma 1 from the work of Munos & Szepesvári, 2008 simply by replacing the MDP Bellman operator with the SMDP Bellman operator $\mathbb{T}$ everywhere it occurs, and noting that we must sample from $|\mathcal{O}|$ options rather than only $|A|$ primitive actions. We omit the proof here for brevity.

B.2 Bounding the Pointwise Propagation Error

We are interested in bounding the loss due to following the policy φ_K derived by OFVI rather than following the optimal policy π^*. We will use the fact that

$$\left\|V^* - V^{\varphi_K}\right\|_{p,\nu} \leq \left\|V^* - V^{\Phi^*}\right\|_{p,\nu} + \left\|V^{\Phi^*} - V^{\varphi_K}\right\|_{p,\nu} \tag{40}$$

by the triangle inequality and focus on bounding $\|V^{\Phi^*} - V^{\varphi_K}\|_{p,\nu}$, which is the loss due to following the policy φ_K produced by OFVI instead of the optimal option policy Φ^*.

Because OFVI is a value-based method, it does not directly improve the policy at each iteration. Instead performing more iterations improves the estimate of the optimal option policy's value function V^{Φ^*}. Thus, we need to relate the loss $\left\|V^{\Phi^*} - V^{\varphi_K}\right\|_{p,\nu}$ to the quality of the final value function estimate V_K produced by the OFVI algorithm. The following lemma develops a pointwise relationship between the $V^{\Phi^*} - V^{\varphi_K}$ and $V^{\Phi^*} - V_K$.

Lemma 3. *Suppose OFVI is executed for K iterations with iterates V_k for $k = 0, 1, 2, \ldots, K$. Let Φ^* be the optimal policy with respect to the given options $\mathcal{O}$ and φ_K be the greedy option policy with respect to the K^{th} and final iterate V_K, then*

$$V^{\Phi^*} - V^{\varphi_K} \leq (I - \widetilde{P}^{\varphi_K})^{-1} \left(\widetilde{P}^{\Phi^*} - \widetilde{P}^{\varphi_K}\right) \left(V^{\Phi^*} - V_K\right) \quad , \tag{41}$$

where I is the identity matrix.

Proof. Since $\mathbb{T}V^{\Phi^*} = V^{\Phi^*}$ and $\mathbb{T}^{\varphi_K} V^{\varphi_K} = V^{\varphi_K}$, we get

$$\begin{aligned}
V^{\Phi^*} - V^{\varphi_K} &= \mathbb{T}V^{\Phi^*} - \mathbb{T}^{\varphi_K} V^{\varphi_K} \\
&= \mathbb{T}V^{\Phi^*} - \mathbb{T}^{\Phi^*} V_K + \mathbb{T}^{\Phi^*} V_K - \mathbb{T}^{\varphi_K} V^{\varphi_K} \\
&= \widetilde{P}^{\Phi^*} \left(V^{\Phi^*} - V_K\right) + \mathbb{T}^{\Phi^*} V_K - \mathbb{T}^{\varphi_K} V^{\varphi_K} \\
&= \widetilde{P}^{\Phi^*} \left(V^{\Phi^*} - V_K\right) + \mathbb{T}^{\Phi^*} V_K - \mathbb{T}V_K + \mathbb{T}V_K - \mathbb{T}^{\varphi_K} V^{\varphi_K} \\
&\leq \widetilde{P}^{\Phi^*} \left(V^{\Phi^*} - V_K\right) + \mathbb{T}V_K - \mathbb{T}^{\varphi_K} V^{\varphi_K} \\
&= \widetilde{P}^{\Phi^*} \left(V^{\Phi^*} - V_K\right) + \mathbb{T}^{\varphi_K} V_K - \mathbb{T}^{\varphi_K} V^{\varphi_K} \\
&= \widetilde{P}^{\Phi^*} \left(V^{\Phi^*} - V_K\right) + \widetilde{P}^{\varphi_K} \left(V_K - V^{\varphi_K}\right) \\
&= \widetilde{P}^{\Phi^*} \left(V^{\Phi^*} - V_K\right) + \widetilde{P}^{\varphi_K} \left(V_K - V^{\Phi^*} + V^{\Phi^*} - V^{\varphi_K}\right) \quad ,
\end{aligned}$$

where the initial equality is based on the fact that V^{Φ^*} is the fixed point for $\mathbb{T}$ and V^{φ_K} is the fixed point for $\mathbb{T}^{\varphi_K}$. The first step is obtained by inserting $(-\mathbb{T}^{\Phi^*} V_K + \mathbb{T}^{\Phi^*} V_K) = 0$. The second step pulls out the discounted transition probability kernel $\widetilde{P}^{\Phi^*}$ by subtracting $\mathbb{T}^{\Phi^*} V_K$ from $\mathbb{T}V^{\Phi^*}$. Since the backups are performed by the same policy Φ^*, the immediate reward terms are canceled, leaving only $\widetilde{P}^{\Phi^*}\left(V^{\Phi^*} - V^{\varphi_K}\right)$. The third step inserts $(-\mathbb{T}V_K + \mathbb{T}V_K) = 0$. Since $\mathbb{T}^{\Phi^*} V_K \leq \mathbb{T}V_K$, we obtain the fourth step by dropping the terms $\mathbb{T}^{\Phi^*} V_K - \mathbb{T}V_K$, which is a vector whose elements are less than zero. We obtain the fifth step by noticing that since φ_K is the greedy policy with respect to V_K, $\mathbb{T}V_K = \mathbb{T}^{\varphi_K} V_K$. The sixth step pulls out $\widetilde{P}^{\varphi_K}$ by subtracting $\mathbb{T}^{\varphi_K} V^{\varphi_K}$ from $\mathbb{T}^{\varphi_K} V_K$. The seventh step inserts $(-V^{\Phi^*} + V^{\Phi^*}) = 0$.

We can manipulate the above inequality

$$\begin{aligned}
V^{\Phi^*} - V^{\varphi_K} &\leq \widetilde{P}^{\Phi^*} \left(V^{\Phi^*} - V_K\right) + \widetilde{P}^{\varphi_K} \left(V_K - V^{\Phi^*} + V^{\Phi^*} - V^{\varphi_K}\right) \\
V^{\Phi^*} - V^{\varphi_K} &\leq \left(\widetilde{P}^{\Phi^*} - \widetilde{P}^{\varphi_K}\right) \left(V^{\Phi^*} - V_K\right) + \widetilde{P}^{\varphi_K} \left(V^{\Phi^*} - V^{\varphi_K}\right) \\
\left(V^{\Phi^*} - V^{\varphi_K}\right) - \widetilde{P}^{\varphi_K} \left(V^{\Phi^*} - V^{\varphi_K}\right) &\leq \left(\widetilde{P}^{\Phi^*} - \widetilde{P}^{\varphi_K}\right) \left(V^{\Phi^*} - V_K\right) \\
\left(I - \widetilde{P}^{\varphi_K}\right) \left(V^{\Phi^*} - V^{\varphi_K}\right) &\leq \left(\widetilde{P}^{\Phi^*} - \widetilde{P}^{\varphi_K}\right) \left(V^{\Phi^*} - V_K\right) \quad ,
\end{aligned}$$

where I is the identity matrix, so that the $\left(V^{\Phi^*} - V^{\varphi_K}\right)$ terms are all on the left hand side. Since $(I - \widetilde{P}^{\varphi_K})$ is invertible and its inverse is a monotone operator, we get

$$V^{\Phi^*} - V^{\varphi_K} \leq (I - \widetilde{P}^{\varphi_K})^{-1} \left(\widetilde{P}^{\Phi^*} - \widetilde{P}^{\varphi_K}\right) \left(V^{\Phi^*} - V_K\right) \quad ,$$

which relates $(V^{\Phi^*} - V^{\varphi_K})$ to $(V^{\Phi^*} - V_K)$. □

Now that Lemma 3 provides us with a relationship between the quality of estimates of the value function and quality of the resulting policy, we need to bound the quality of value function estimates. Each iteration $k = 1, 2, \ldots, K$ of OFVI results in some error

$$\varepsilon_k = \mathbb{T} V_{k-1} - V_k \ , \tag{42}$$

which is induced by the fitting process. One of the main issues in the proof of Theorem 1 is to determine how these fitting errors propagate through the iterations.

The following lemma helps to bound the error between V^{Φ^*} and V_K by developing pointwise upper and lower bounds for $V^{\Phi^*} - V_K$ that show how error propagates recursively with each iteration.

Lemma 4. *Suppose Φ^* is the optimal policy with respect to the options $\mathcal{O}$, OFVI is executed for K iterations with iterates V_k for $k = 0, 1, 2, \ldots, K$ and iteration errors ε_k for $k = 1, 2, \ldots, K$ as defined by (42), then we have the following upper bound*

$$V^{\Phi^*} - V_K \leq \sum_{k=1}^{K} \left(\widetilde{P}^{\Phi^*} \right)^{K-k} \varepsilon_k + \left(\widetilde{P}^{\Phi^*} \right)^{K} (V^* - V_0) \ , \tag{43}$$

and the following lower bound

$$V^{\Phi^*} - V_K \geq \varepsilon_K + \sum_{k=1}^{K-1} \left(\widetilde{P}^{\varphi_{K-1}} \widetilde{P}^{\varphi_{K-2}} \ldots \widetilde{P}^{\varphi_k} \right) \varepsilon_k + \left(\widetilde{P}^{\varphi_{K-1}} \widetilde{P}^{\pi_{K-2}} \ldots \widetilde{P}^{\varphi_0} \right) \left(V^{\Phi^*} - V_0 \right) \ . \tag{44}$$

Proof. First we derive an upper bound for $V^{\Phi^*} - V_K$. By equation (42), we have

$$\begin{array}{rcl}
V^{\Phi^*} - V_k & = & \mathbb{T} V^{\Phi^*} - \mathbb{T} V_{k-1} + \varepsilon_k \\
& = & \mathbb{T}^{\Phi^*} V^{\Phi^*} - \mathbb{T}^{\Phi^*} V_{k-1} + \mathbb{T}^{\Phi^*} V_{k-1} - \mathbb{T} V_{k-1} + \varepsilon_k \\
& \leq & \mathbb{T} V^{\Phi^*} - \mathbb{T}^{\Phi^*} V_{k-1} + \varepsilon_k \\
& = & \widetilde{P}^{\Phi^*} \left(V^{\Phi^*} - V_{k-1} \right) + \varepsilon_k \ .
\end{array}$$

By recursing on this inequality, we obtain an upper bound

$$V^{\Phi^*} - V_K \leq \sum_{k=1}^{K} \left(\widetilde{P}^{\Phi^*} \right)^{K-k} \varepsilon_k + \left(\widetilde{P}^{\Phi^*} \right)^{K} (V^* - V_0) \ .$$

Now we will derive a lower bound for $V^{\Phi^*} - V_K$. Let φ_k denote the greedy policy with respect to V_k. By (42), we have

$$\begin{array}{rcl}
V^{\Phi^*} - V_k & = & \mathbb{T} V^{\Phi^*} - \mathbb{T} V_{k-1} + \varepsilon_k \\
& = & \mathbb{T} V^{\Phi^*} - \mathbb{T}^{\varphi_{k-1}} V^{\Phi^*} + \mathbb{T}^{\varphi_{k-1}} V^{\Phi^*} - \mathbb{T} V_{k-1} + \varepsilon_k \\
& \geq & \mathbb{T}^{\varphi_{k-1}} V^{\Phi^*} - \mathbb{T} V_{k-1} + \varepsilon_k \\
& = & \widetilde{P}^{\varphi_{k-1}} \left(V^{\Phi^*} - V_{k-1} \right) + \varepsilon_k \ .
\end{array}$$

By recursing on this inequality, we obtain a lower bound

$$V^{\Phi^*} - V_K \geq \varepsilon_K + \sum_{k=1}^{K-1} \left(\widetilde{P}^{\varphi_{K-1}}\widetilde{P}^{\varphi_{K-2}} \dots \widetilde{P}^{\varphi_k}\right) \varepsilon_k + \left(\widetilde{P}^{\varphi_{K-1}}\widetilde{P}^{\pi_{K-2}} \dots \widetilde{P}^{\varphi_0}\right) \left(V^{\Phi^*} - V_0\right) \ .$$

□

Lemma 3 gives a relationship between the quality of value function estimates and the quality of the resulting greedy policy, while Lemma 4 gives upper and lower bounds on value function estimates. The next step is to combine the results from these lemmas to derive a pointwise error bound for $V^{\Phi^*} - V^{\varphi_K}$.

We will make use of the following definition in deriving the point-wise error bound. The lambda values are used to simplify the notation, but we also use the fact that they are carefully designed so that they sum to 1.

Definition 13. *For $t = 1, 2, \dots, \infty$, let*

$$\lambda_k = \frac{1-\gamma}{1-\gamma^{K+1}}\gamma^{(K-k)} \tag{45}$$

for $k = 0, 1, \dots, K$.

The following lemma shows that the λ_k values sum to 1.

Lemma 5. *The $\lambda.$ values defined by (45) satisfy $\sum_{k=0}^{K} \lambda_k = 1$.*

Proof. We have

$$\begin{array}{rcl} \sum_{k=0}^{K} \lambda_k & = & \sum_{k=0}^{K} \frac{1-\gamma}{1-\gamma^{K+1}}\gamma^{(K-k)} \\ & = & \frac{1-\bar{\gamma}}{1-\gamma^{K+1}} \sum_{k=0}^{K} \gamma^k \\ & = & \frac{1}{1-\gamma^{K+1}} \sum_{k=0}^{K}(1-\gamma)\gamma^k \\ & = & \frac{1}{1-\gamma^{K+1}}(1-\gamma^{K+1}) \\ & = & 1 \ . \end{array}$$

□

Now we are ready to derive the point-wise error bound for $V^{\Phi^*} - V^{\varphi_K}$.

Lemma 6. *Let $Z \in \{0, 1, 2, \dots, K\}$, φ_k be the greedy policy with respect to the k^{th} iterate V_k derived by OFVI, Φ be an option policy such that $Q^*(x, \Phi(x)) \geq V^*(x) - \alpha$ for all $x \in X$, and $\bar{\gamma} = \gamma^{d_{\min}}$. If $A3(\alpha, d, \psi, \nu, j)$ (Assumption 3) is true and the first Z iterates of OFVI are pessimistic (i.e., for all $x \in X$ and $k \in \{0, 1, 2, \dots, Z\}$, $V^{\Phi^*}(x) \geq V_k(x)$), then the difference between V^{Φ^*} and the value of the option policy φ_K returned by OFVI is bounded by*

$$V^{\Phi^*} - V^{\varphi_K} \leq \Delta \sum_{t=1}^{\infty}\sum_{k=0}^{K} \lambda_k P_{k,t} |\xi_k| \ ,$$

where the λ_k's are defined by (45),

$$\Delta = \left(\frac{2\bar{\gamma}(1-\gamma^{K+1})}{(1-\gamma)^3}\right) \ ,$$

$$P_{k,t} = \begin{cases} \left[\left(P^{\Phi^*}\right)^{K-Z}\left(P^{\Phi}\right)^{Z-k}\right]_t & 0 \leq k \leq Z \\ \frac{1}{2}\left[\left(P^{\Phi^*}\right)_t^{K-k} + \left(P^{\varphi_{K-1}} P^{\varphi_{K-2}} \dots P^{\varphi_k}\right)_t\right] & Z < k < K \\ \mathbb{1} & k = K \end{cases}$$

for $t \geq 1$*, and*

$$\xi_k = \begin{cases} V^{\Phi^*} - V_0 & k = 0 \\ \varepsilon_k + \alpha & 1 \leq k \leq Z \\ \varepsilon_k & Z < k \leq K \end{cases} .$$

Proof. We can place an upper bound (43) and a lower bound (44) on the relationship between V_K and V^{Φ^*}. Then we can use this information to bound the difference between V^{φ_K} and V^{Φ^*}. However, in this lemma, we will exploit the pessimism of the first Z iterates and the option policy Φ to achieve a more informative bound.

When an iterate V_k is pessimistic $V^{\Phi^*} - V_k$ is lower bounded by 0. For an upper bound, we have

$$\begin{aligned} V^{\Phi^*} - V_k &= V^{\Phi^*} - \mathbb{T}^{\Phi} V^{\Phi^*} + \mathbb{T}^{\Phi} V^{\Phi^*} - V_k \\ &\leq \alpha + \mathbb{T}^{\Phi} V^{\Phi^*} - V_k \\ &= \alpha + \mathbb{T}^{\Phi} V^{\Phi^*} - \mathbb{T} V_{k-1} + \varepsilon_k \\ &= \alpha + \mathbb{T}^{\Phi} V^{\Phi^*} - \mathbb{T}^{\Phi} V_{k-1} + \mathbb{T}^{\Phi} V_{k-1} - \mathbb{T} V_{k-1} + \varepsilon_k \\ &\leq \alpha + \mathbb{T}^{\Phi} V^{\Phi^*} - \mathbb{T}^{\Phi} V_{k-1} + \varepsilon_k \\ &\leq \widetilde{P}^{\Phi}\left(V^* - V_{k-1}\right) + \left(\varepsilon_k + \alpha\right) \ , \end{aligned}$$

where the initial inequality inserts the term $(-\mathbb{T}^{\Phi} V^{\Phi^*} + \mathbb{T}^{\Phi} V^{\Phi^*}) = 0$. The first step follows from the fact that following Φ for a single decision and then following Φ^* produces an α-optimal policy, so $V^{\Phi^*} - \mathbb{T}^{\Phi} V^{\Phi^*} \leq \alpha$. The second step is due to the definition of ε_k from (42). The third step inserts $(-\mathbb{T}^{\Phi} V_{k-1} + \mathbb{T}^{\Phi} V_{k-1}) = 0$. The fourth step removes $\mathbb{T}^{\Phi} V_{k-1} - \mathbb{T} V_{k-1}$ because the sum of those two terms is less than or equal to zero (since $\mathbb{T}$ updates using the max operator, while $\mathbb{T}^{\Phi}$ updates using the policy Φ). The fifth and final step pulls out the discounted transition probability kernel $\widetilde{P}^{\Phi}$.

By recursing on this inequality $Z \geq 0$ times we obtain

$$V^{\Phi^*} - V_Z \leq \begin{cases} V^{\Phi^*} - V_0 & Z = 0 \\ \left(\sum_{j=1}^{Z}\left(\widetilde{P}^{\Phi}\right)^{Z-j}\left(\varepsilon_j + \alpha\right)\right) + \left(\widetilde{P}^{\Phi}\right)^{Z}\left(V^{\Phi^*} - V_0\right) & 1 \leq Z \leq K \end{cases} . \tag{46}$$

By combining our upper bound recursion from (43) with (46), we obtain terms

$$u_k \xi_k = \begin{cases} \left[\left(\widetilde{P}^{\Phi^*}\right)^{K-Z}\left(\widetilde{P}^{\Phi}\right)^{Z}\right]\left(V^{\Phi^*} - V_0\right) & k = 0 \\ \left[\left(\widetilde{P}^{\Phi^*}\right)^{K-Z}\left(\widetilde{P}^{\Phi}\right)^{Z-k}\right]\left(\varepsilon_k + \alpha\right) & k = 1, 2, \dots, Z \\ \left[\left(\widetilde{P}^{\Phi^*}\right)^{K-k}\right]\varepsilon_k & k = Z+1, Z+2, \dots, K \end{cases}$$

such that

$$V^{\Phi^*} - V_K \leq \sum_{k=0}^{K} u_k \xi_k$$

upper bounds the difference between V^{Φ^*} and the final iterate derived by OFVI, V_K.

Now, since 0 lower bounds the difference between V^{Φ^*} and the first Z iterates of OFVI, we can use 0 as our lower bound for the first Z iterations and fill in the rest of the iterates with (44). This gives us the terms

$$l_k \xi_k = \begin{cases} 0 & 0 \leq k \leq Z \\ \left[\widetilde{P}^{\varphi_{K-1}} \widetilde{P}^{\varphi_{K-2}} \ldots \widetilde{P}^{\varphi_k} \right] \varepsilon_k & Z < k < K-1 \\ \varepsilon_K & K \end{cases} ,$$

such that

$$V^{\Phi^*} - V_K \geq \sum_{k=0}^{K} l_k \xi_k$$

lower bounds the difference between V^{Φ^*} and the final iterate V_K. This implies that $|V^{\Phi^*} - V_K| \leq \sum_{k=0}^{K} (u_k - l_k) \xi_k$.

By Lemma 3, we have

$$\begin{aligned} V^{\Phi^*} - V^{\varphi_K} &\leq (I - \widetilde{P}^{\varphi_K})^{-1} \left(\widetilde{P}^{\Phi^*} - \widetilde{P}^{\varphi_K} \right) \left(\textstyle\sum_{k=0}^{K} (u_k - l_k) \xi_k \right) \\ &\leq (I - \widetilde{P}^{\varphi_K})^{-1} \left| \widetilde{P}^{\Phi^*} - \widetilde{P}^{\varphi_K} \right| \left(\textstyle\sum_{k=0}^{K} (u_k + l_k) |\xi_k| \right) \\ &\leq \bar{\gamma} (I - \widetilde{P}^{\varphi_K})^{-1} \left(\textstyle\sum_{k=0}^{K} (u_k + l_k) |\xi_k| \right) \\ &= \bar{\gamma} \left(\textstyle\sum_{i=0}^{\infty} (\widetilde{P}^{\varphi_K})^i \right) \left(\textstyle\sum_{k=0}^{K} (u_k + l_k) |\xi_k| \right) \\ &\leq \bar{\gamma} \left(\textstyle\sum_{i=0}^{\infty} \bar{\gamma}^i \right) \left(\textstyle\sum_{k=0}^{K} (u_k + l_k) |\xi_k| \right) \\ &\leq \bar{\gamma} \left(\textstyle\sum_{i=0}^{\infty} \gamma^i \right) \left(\textstyle\sum_{k=0}^{K} (u_k + l_k) |\xi_k| \right) \\ &\leq \tfrac{\bar{\gamma}}{1-\gamma} \textstyle\sum_{k=0}^{K} (u_k + l_k) |\xi_k| \ , \end{aligned}$$

where for the first step we have taken the absolute value of both sides of the inequality, and for the second step we used the fact that $\gamma^{d_{\min}} \geq \left| \widetilde{P}^{\Phi^*} - \widetilde{P}^{\varphi_K} \right|$. In the remainder of the proof we will denote $\gamma^{d_{\min}}$ by $\bar{\gamma}$.

For $k = 0$, we have

$$\begin{aligned} &\frac{\bar{\gamma}}{1-\gamma} (u_0 + l_0) |\xi_0| \\ &= \left(\tfrac{2}{2}\right) \tfrac{\bar{\gamma}}{1-\gamma} \left(u_0 |\xi_0| \right) \\ &= \left(\tfrac{2\bar{\gamma}}{1-\gamma} \right) \left(\tfrac{1}{2} \left[\left(\widetilde{P}^{\Phi^*} \right)^{K-Z} \left(\widetilde{P}^{\Phi} \right)^{Z} \right] \right) |\xi_0| \\ &\leq \left(\tfrac{2\bar{\gamma}}{1-\gamma} \right) \left[\left(\widetilde{P}^{\Phi^*} \right)^{K-Z} \left(\widetilde{P}^{\Phi} \right)^{Z} \right] |\xi_0| \\ &\leq \left(\tfrac{2\bar{\gamma}}{(1-\gamma)} \right) \textstyle\sum_{t=1}^{\infty} \gamma^K P_{0,t} |\xi_0| \\ &= \left(\tfrac{2\bar{\gamma}(1-\gamma^{K+1})}{(1-\gamma)^2} \right) \textstyle\sum_{t=1}^{\infty} \left(\tfrac{1-\gamma}{1-\gamma^{K+1}} \gamma^K \right) P_{0,t} |\xi_0| \\ &\leq \Delta \sum_{t=1}^{\infty} \lambda_0 P_{0,t} |\xi_0| \ . \end{aligned}$$

For $k = 1, 2, \ldots, Z$, we have

$$\frac{\bar{\gamma}}{1-\gamma} (u_k + l_k) |\xi_k|$$

$$\begin{aligned}
&= \left(\tfrac{2}{2}\right)\tfrac{\bar{\gamma}}{1-\gamma}\left(u_k|\xi_k|\right)\\
&= \left(\tfrac{2\bar{\gamma}}{1-\gamma}\right)\left(\tfrac{1}{2}\left[\left(\widetilde{P}^{\Phi^*}\right)^{K-Z}\left(\widetilde{P}^{\Phi}\right)^{Z-k}\right]\right)|\xi_k|\\
&\le \left(\tfrac{2\bar{\gamma}}{1-\gamma}\right)\left[\left(\widetilde{P}^{\Phi^*}\right)^{K-Z}\left(\widetilde{P}^{\Phi}\right)^{Z-k}\right]|\xi_k|\\
&\le \left(\tfrac{2\bar{\gamma}}{(1-\gamma)}\right)\sum_{t=1}^{\infty}\gamma^{K-k}P_{k,t}|\xi_k|\\
&= \left(\tfrac{2\bar{\gamma}(1-\gamma^{K+1})}{(1-\gamma)^2}\right)\sum_{t=1}^{\infty}\left(\tfrac{1-\gamma}{1-\gamma^{K+1}}\gamma^{K-k}\right)P_{k,t}|\xi_k|\\
&\le \Delta\sum_{t=1}^{\infty}\lambda_k P_{k,t}|\xi_k|\ .
\end{aligned}$$

For $k = Z+1, Z+2, \ldots, K$, we have

$$\frac{\bar{\gamma}}{1-\gamma}(u_k + l_k)|\xi_k|$$

$$\begin{aligned}
&= \left(\tfrac{2}{2}\right)\tfrac{\bar{\gamma}}{1-\gamma}\left(u_k + l_k\right)|\xi_k|\\
&= \left(\tfrac{2\bar{\gamma}}{1-\gamma}\right)\tfrac{1}{2}\left(\left(\widetilde{P}^{\Phi^*}\right)^{K-k} + \left[\widetilde{P}^{\varphi_{K-1}}\widetilde{P}^{\varphi_{K-2}}\ldots\widetilde{P}^{\varphi_k}\right]\right)|\xi_k|\\
&\le \left(\tfrac{2\bar{\gamma}}{(1-\gamma)}\right)\sum_{t=1}^{\infty}\gamma^{K-k}P_{k,t}|\xi_k|\\
&= \left(\tfrac{2\bar{\gamma}(1-\gamma^{K+1})}{(1-\gamma)^2}\right)\sum_{t=1}^{\infty}\left(\tfrac{1-\gamma}{1-\gamma^{K+1}}\gamma^{K-k}\right)P_{k,t}|\xi_k|\\
&= \Delta\sum_{t=1}^{\infty}\lambda_k P_{k,t}|\xi_k|\ .
\end{aligned}$$

By plugging in the results from these three inequalities, we obtain $V^{\Phi^*} - V^{\varphi_K} \le \Delta\sum_{t=1}^{\infty}\sum_{k=0}^{K}\lambda_k P_{k,t}|\xi_k|$. □

B.3 From Pointwise to L_p-norm Propagation Error

Lemma 6 gives us a pointwise bound on the loss of the policy φ_K derived by OFVI compared to following the optimal option policy Φ^*, but the most common function approximation architectures minimize an L_p-norm (not pointwise loss). In this subsection, we derive Lemma 8 that transforms our pointwise bound to an L_p-norm bound weighted by an arbitrary distribution $\nu \in \mathcal{M}(X)$. The key to this transformation is based on $A2(\nu,\mu)$ (Assumption 2), which allows us to bound the pointwise transition probability kernels $P_{k,t}$ from Lemma 6 by $\hat{c}(\cdot)\mu$ from Assumption 2 at each iteration $k \in \{1,2,\ldots,K\}$. The following lemma provides the first step in this transformation.

Lemma 7. *Suppose that $A2(\nu,\mu)$ (Assumption 2) holds, then*

$$\nu P_{k,t} \le \max_{m\in\{1,2,\ldots,i+t\}}\hat{c}_t(m)\mu\ , \tag{47}$$

where $\nu,\mu \in M(X)$.

Proof. We have two cases to consider (case 1) $1 \le k \le Z$ and (case 2) $Z < k \le K$.

For case 1, we have

$$\begin{array}{rcl} \nu P_{k,t} & = & \nu \left[\left(P^{\Phi^*}\right)^{K-Z} \left(P^{\varphi}\right)^{Z-k} \right]_t \\ & \leq & \hat{c}_t(K-k)\mu \ . \end{array}$$

For case 2, we have

$$\begin{array}{rcl} \nu P_{k,t} & = & \nu \frac{1}{2} \left[\left(P^{\Phi^*}\right)_t^{K-k} + \left(P^{\varphi_{K-1}} P^{\varphi_{K-2}} \ldots P^{\varphi_k}\right)_t \right] \\ & = & \frac{1}{2} \left[\nu \left(P^{\Phi^*}\right)_t^{K-k} + \nu \left(P^{\varphi_{K-1}} P^{\varphi_{K-2}} \ldots P^{\varphi_k}\right)_t \right] \\ & \leq & \frac{1}{2} \left[\hat{c}_t(K-k)\mu + \hat{c}_t(K-k)\mu \right] \\ & = & \hat{c}_t(K-k)\mu \ . \end{array}$$

□

To derive the L_p-norm bound, we need the following additional notation to represent the set of options that can be initialized from state $x \in X$ and have duration longer than some $d \geq 1$.

Definition 14. *Let $d \geq 1$, $x \in X$ be a state, and $\mathcal{O}$ be a set of options. The set $\mathcal{O}_{x,d}$ denotes the subset of options $o \in \mathcal{O}$ that can be initialized from the state x, such that $\inf_{Y \subseteq X} \mathbb{E}\left[D^o_{x,Y}\right] \geq d$.*

Notice that by assumption $\mathcal{O}_{x,d_{\min}} \equiv \mathcal{O}_x$.

Lemma 8. *Let $K \geq 1$, $\varepsilon > 0$, and $Z \in \{0, 1, 2, \ldots, K\}$. Suppose that Assumption 2 and Assumption 3 are true and that the first Z iterates of OFVI are pessimistic, then*

$$\|V^* - V^{\pi_K}\|_{p,\nu} \leq \frac{2\bar{\gamma}}{(1-\gamma)^2} \mathbb{C}^{1/p}_{\nu,\mu}(\varepsilon+\alpha) + \left(\gamma^{d_{\min}(K+1)+(1-\psi)(d-d_{\min})\lfloor Z/\hat{j} \rfloor}\right)^{1/p} \left(\frac{2\left\|V^{\Phi^*} - V_0\right\|_\infty}{(1-\gamma)^2} \right) \tag{48}$$

holds, provided that the approximation errors ε_k satisfy $\|\varepsilon_k\|_{p,\mu} \leq \varepsilon$ for all $k = 1, 2, \ldots, K$.

Proof. First note that

$$\Phi(x) = \begin{cases} \arg\max_{o \in \mathcal{O}_{x,d}} Q^*(x,o) & \text{if } x \in \omega_{\alpha,d} \\ \Phi^*(x) & \text{otherwise} \end{cases} \ .$$

is a policy such that $Q^{\Phi^*}(x, \Phi(x)) \geq V^{\Phi^*}(x) - \alpha$ for all $x \in X$. Therefore, by Lemma 6, we have

$$V^{\Phi^*} - V^{\varphi_K} \leq \Delta \sum_{t=1}^{\infty} \sum_{k=0}^{K} \lambda_k P_{k,t} |\xi_k| \ .$$

Now, we have

$$\begin{array}{rcl} \left\|V^{\Phi^*} - V^{\varphi_K}\right\|^p_{p,\nu} & = & \int \nu(x) \left|V^{\Phi^*}(x) - V^{\pi_K}(x)\right|^p dx \\ & \leq & \int \nu(x) \left(\Delta \sum\limits_{t=1}^{\infty} \sum\limits_{k=0}^{K} \lambda_k P_{k,t} |\xi_k|(x) \right)^p dx \\ & \leq & \Delta^p \int \nu(x) \left(\left[\sum\limits_{t=1}^{\infty} \sum\limits_{k=1}^{K} \lambda_k P_{k,t} |\varepsilon_k + \alpha| + \lambda_0 P_{0,t} |V^{\Phi^*} - V_0| \right] (x) \right)^p dx \ , \end{array}$$

where the initial equality is due to the definition of $\|\cdot\|_{p,\nu}$. The first step replaces $V^{\Phi^*} - V^{\varphi_K}$ with $\Delta \sum_{t=1}^{\infty} \sum_{k=0}^{K} \lambda_k P_{k,t} |\xi_k|$. The last step pulls $k = 0$ out of the sum and moves Δ outside of the integral.

Recall by Lemma 5 that $\sum\limits_{k=0}^{K} \lambda_k = 1$. We apply Jensen's inequality twice; once with convex function $|\cdot|^p$ and parameters λ_k for $k = 0, 1, \ldots, K$, and once with parameters determined by the stochastic operators $\sum_{t=1}^{\infty} P_{k,t}$, to obtain

$$\left\|V^{\Phi^*} - V^{\varphi_K}\right\|_{p,\nu}^{p} \leq \Delta^p \int \nu \left[\sum_{t=1}^{\infty} \sum_{k=1}^{K} \lambda_k P_{k,t} |\varepsilon_k + \alpha|^p + \lambda_0 P_{0,t} |V^{\Phi^*} - V_0|^p\right](x) dx \ .$$

Noticing that $|V^{\Phi^*} - V_0|$ is bounded by $\|V^{\Phi^*} - V_0\|_{\infty}$, we obtain

$$\begin{aligned} \left\|V^{\Phi^*} - V^{\varphi_K}\right\|_{p,\nu}^{p} \ \leq \ & \Delta^p \Big[\textstyle\sum_{t=1}^{\infty} \sum_{k=1}^{K} \lambda_k P_{k,t} |\varepsilon_k + \alpha|^p + \\ & \textstyle\int \nu(x) \lambda_0 P_{0,t} \|V^{\Phi^*} - V_0\|_{\infty}^{p} dx \Big] \ . \end{aligned}$$

By Assumption 2 and Lemma 7, we have that

$$\nu P_{k,t} \leq \max_{m \in \{1,2,\ldots,K-k+t'-1\}} \hat{c}_{K-k+t'-1}(m) \mu \ ,$$

where $t' = t - (K - k) + 1$. Thus we have

$$\begin{aligned} \sum_{k=1}^{K} \sum_{t=1}^{\infty} \lambda_k \nu P_{k,t} |\varepsilon_k + \alpha|^p \ \leq \ & \sum_{k=1}^{K} \sum_{t'=1}^{\infty} \tfrac{1-\gamma}{1-\gamma^{K+1}} \gamma^{K-k} \cdot \\ & \max_{m \in \{1,2,\ldots,K-k+t'-1\}} \hat{c}_{K-k+t'-1}(m) \|\varepsilon_k + \alpha\|_{p,\mu}^{p} \\ \leq \ & \sum_{k=1}^{K} \sum_{t'=1}^{\infty} \tfrac{1-\gamma}{\gamma^{t'-1}(1-\gamma^{K+1})} \gamma^{K-k} \gamma^{t'-1} \cdot \\ & \max_{m \in \{1,2,\ldots,K-k+t'-1\}} \hat{c}_{K-k+t'-1}(m) \|\varepsilon_k + \alpha\|_{p,\mu}^{p} \\ \leq \ & \tfrac{(1-\gamma)^2}{1-\gamma^{K+1}} \sum_{k=1}^{K} \sum_{t'=0}^{\infty} \gamma^{k+t'} \cdot \\ & \max_{m \in \{1,2,\ldots,k+t'\}} \hat{c}_{k+t'}(m) \|\varepsilon_k + \alpha\|_{p,\mu}^{p} \\ \leq \ & \tfrac{1}{1-\gamma^{K+1}} (1-\gamma)^2 \sum_{t=1}^{\infty} t \gamma^{t-1} \max\nolimits_{m \in \{1,2,\ldots,t\}} \hat{c}_t(m) \|\varepsilon_k + \alpha\|_{p,\mu}^{p} \\ \leq \ & \tfrac{1}{1-\gamma^{K+1}} \mathbb{C}_{\nu,\mu} \left(\varepsilon + \alpha\right)^p \ , \end{aligned}$$

where $\mathbb{C}_{\nu,\mu}$ is the SMDP discounted average concentrability coefficient from Assumption 2. By replacing $\sum_{t=1}^{\infty} \sum_{k=1}^{K} \lambda_k P_{k,t} |\varepsilon_k + \alpha|^p$, we get

$$\begin{aligned} \left\|V^{\Phi^*} - V^{\varphi_K}\right\|_{p,\nu}^{p} \ \leq \ & \left(\tfrac{2\bar{\gamma}(1-\gamma^{K+1})}{(1-\gamma)^2}\right)^p \Big[\tfrac{1}{1-\gamma^{K+1}} \mathbb{C}_{\nu,\mu} \left(\varepsilon + \alpha\right)^p + \\ & \textstyle\sum_{t=1}^{\infty} \int \nu(x) \lambda_0 P_{0,t} \|V^{\Phi^*} - V_0\|_{\infty}^{p} dx \Big] \ . \end{aligned} \tag{49}$$

Consider the second term in the last step of (49). By replacing $P_{0,t}$ with its definition, we get

$$\begin{aligned}
\int \nu(x) \sum_{t=1}^{\infty} \lambda_0 P_{0,t} dx &= \int \nu(x) \sum_{t=1}^{\infty} \tfrac{(1-\gamma)}{1-\gamma^{K+1}} \gamma^K \left[\left((P^{\Phi^*})^{K-Z} \left(P^{\Phi} \right)^Z \right)_{K+t-1} \right] dx \\
&= \int \nu(x) \sum_{t=1}^{\infty} \tfrac{(1-\gamma)}{\gamma^{t-1}(1-\gamma^{K+1})} \gamma^K \gamma^{t-1} \left[\left((P^{\Phi^*})^{K-Z} \left(P^{\Phi} \right)^Z \right)_{K+t-1} \right] dx \\
&\leq \tfrac{(1-\gamma)^2}{1-\gamma^{K+1}} \int \nu(x) \left[(\widetilde{P}^{\Phi^*})^{K-Z} \left(\widetilde{P}^{\Phi} \right)^Z \right] dx \\
&\leq \tfrac{(1-\gamma)^2}{1-\gamma^{K+1}} \gamma^{d_{\min} K + (1-\psi)(d-d_{\min}) \lfloor Z/\hat{j} \rfloor} \quad ,
\end{aligned} \tag{50}$$

where the initial equality is due to expanding $P_{0,t}$ by its definition. The first step simplifies and drops the dependence on $(1-\bar{\gamma})(I - \widetilde{P}^{\varphi_K})^{-1} \leq 1$. The final step replaces $\left(\widetilde{P}^{\varphi_K} \right)^{K-Z}$ with $\gamma^{d_{\min}(K-Z)}$ and $\left(\widetilde{P}^{\Phi} \right)^Z$ with $\gamma^{(1-\psi)d\lfloor Z/\hat{j} \rfloor + \psi d_{\min} Z}$. Under any case, $\left(\widetilde{P}^{\Phi} \right)^Z \leq \gamma^{d_{\min} K}$. Under Assumption 3 with probability $(1-\psi)$ either (a) the state transitioned to is in $\omega_{\alpha,d}$, in which case the effective discount factor is γ^d, or (b) following the bridge policy Φ from the current state reaches a state in $\omega_{\alpha,d}$ in no more than $\hat{j}$ timesteps. On the timesteps that the agent is not in $\omega_{\alpha,d}$ the effective discount factor is $\gamma^{d_{\min}}$, but with probability $(1-\psi)$ this can only happen $Z - \lfloor Z/\hat{j} \rfloor$ times during. Thus $\left(\widetilde{P}^{\Phi} \right)^Z \leq \gamma^{(1+\psi)d_{\min} Z + (1-\psi)(d-d_{\min})\lfloor Z/\hat{j} \rfloor} \leq \gamma^{d_{\min} K + (1-\psi)(d-d_{\min})\lfloor Z/\hat{j} \rfloor}$.

By replacing the second term from (49) with (50), we get

$$\begin{aligned}
\left\| V^{\Phi^*} - V^{\varphi_K} \right\|_{p,\nu}^p \leq \left(\tfrac{2\bar{\gamma}(1-\gamma^{K+1})}{(1-\gamma)^2} \right)^p \Bigg[& \tfrac{1}{1-\gamma^{K+1}} \mathbb{C}_{\nu,\mu} (\varepsilon + \alpha)^p + \\
& \tfrac{(1-\gamma)^2}{1-\gamma^{K+1}} \gamma^{d_{\min} K + (1-\psi)(d-d_{\min})\lfloor Z/\hat{j} \rfloor} \| V^{\Phi^*} - V_0 \|_\infty^p \Bigg] \quad .
\end{aligned}$$

Since $\left(1-\gamma^{K+1}\right)^p \left(\frac{1}{1-\gamma^{K+1}} \right) \leq 1$, then

$$\left\| V^{\Phi^*} - V^{\varphi_K} \right\|_{p,\nu}^p \leq \left(\tfrac{2\bar{\gamma}}{(1-\gamma)^2} \right)^p \left[\mathbb{C}_{\nu,\mu} (\varepsilon + \alpha)^p + (1-\gamma)^2 \gamma^{d_{\min} K + (1-\psi)(d-d_{\min})\lfloor Z/\hat{j} \rfloor} \| V^{\Phi^*} - V_0 \|_\infty^p \right] \quad .$$

Thus, we have

$$\left\| V^{\Phi^*} - V^{\varphi_K} \right\|_{p,\nu} \leq \tfrac{2\bar{\gamma}}{(1-\gamma)^2} \mathbb{C}_{\nu,\mu}^{1/p} (\varepsilon + \alpha) + \left(\gamma^{d_{\min}(K+1) + (1-\psi)(d-d_{\min})\lfloor Z/\hat{j} \rfloor} \right)^{1/p} \left(\tfrac{2\|V^{\Phi^*} - V_0\|_\infty}{(1-\gamma)^2} \right) \quad .$$

□

B.4 Proof of Theorem 1

Proof. (of Theorem 1) We use Lemma 2 to select appropriate values for n and m, when $\varepsilon' = \epsilon(1-\gamma)^2/(2\bar{\gamma}\mathbb{C}_{\nu,\mu}^{1/p})$ and $\delta' \leftarrow \frac{\delta}{K}$.

Since the iterates $V_1, V_2, \ldots, V_K$ are random objects, we cannot directly apply Lemma 2 to bound the error at each iteration. However, this problem was resolved in the proof of

Thm. 2 from the work of Munos & Szepesvári, 2008 by using the fact that the algorithm collects independent samples at each iteration.

The iterate V_{k+1} depends on the random variable V_k and the random samples S_k containing the $n \times m \times |\mathcal{O}|$ next states, rewards, and trajectory lengths. Let the function

$$f(S_k, V_k) = \mathbb{I}\left\{\|V_{k+1}(V_k, S_k) - \mathbb{T}V_k\|_{p,\mu} \leq d_{p,\mu}(\mathbb{T}V_k, \mathcal{F}) + \varepsilon'\right\} - (1 - \delta') \ ,$$

where we have written $V_{k+1}(V_k, S_k)$ to emphasize V_{k+1}'s dependence on both random variables V_k and S_k. Notice that V_k and S_k are independent because S_k was not used to generate V_k and the simulator $\mathbb{S}$ generates independent samples. Because V_k and S_k are independent random variables, we can apply Lemma 5 from the work of Munos & Szepesvári, 2008. This lemma tells us that $\mathbb{E}\left[f(S_k, V_k) \mid V_k\right] \geq 0$ provided that $\mathbb{E}\left[f(S_k, v)\right] \geq 0$ for all $v \in \mathcal{F}$. For any $v \in \mathcal{F}$, by Lemma 2, and by our choice of n and m, we have that $P\left(\|V_{k+1}(v, S_k) - \mathbb{T}v\|_{p,\mu} \leq d_{p,\mu}(\mathbb{T}v, \mathcal{F}) + \varepsilon'\right) \geq 1 - \delta'$. This implies that $\mathbb{E}\left[f(S_k, v)\right] \geq 0$. By Lemma 5 from the work of Munos & Szepesvári, 2008, we have that $\mathbb{E}\left[f(S_k, V_k) \mid V_k\right] \geq 0$. Thus we have $P\left(\|V_{k+1}(V_k, S_k) - \mathbb{T}V_k\|_{p,\mu} \leq d_{p,\mu}(\mathbb{T}v, \mathcal{F}) + \varepsilon'\right) \geq 1 - \delta'$. By the union bound, this ensures that $\|\varepsilon\|_{p,\mu} \leq \varepsilon$ for all K iterations with probability at least $1 - K\delta' = 1 - K(\delta/K) = 1 - \delta$.

The result follows by applying Lemma 8 with $\|\epsilon_k\|_{p,\mu} \leq d_{p,\mu}(\mathbb{T}V_k, \mathcal{F}) + \varepsilon'$.

$$\begin{array}{rcl}
\|V^* - V^{\varphi_K}\|_{p,\nu} & \leq & \left\|V^* - V^{\Phi^*}\right\|_{p,\nu} + \left\|V^{\Phi^*} - V^{\varphi_K}\right\|_{p,\nu} \\
& \leq & \mathcal{L}_{p,\nu}(\Phi^*) + \frac{2\bar{\gamma}}{(1-\gamma)^2}\mathbb{C}_{\nu,\mu}^{1/p}\left(b_{p,\mu}(\mathbb{T}\mathcal{F}, \mathcal{F}) + \alpha + \varepsilon'\right) \\
& & + \left(\gamma^{d_{\min}(K+1)+(1-\psi)(d-d_{\min})\lfloor Z/\hat{j}\rfloor}\right)^{1/p}\left(\frac{2\|V^{\Phi^*}-V_0\|_\infty}{(1-\gamma)^2}\right) \\
& = & \mathcal{L}_{p,\nu}(\Phi^*) + \frac{2\bar{\gamma}}{(1-\gamma)^2}\mathbb{C}_{\nu,\mu}^{1/p}\left(b_{p,\mu}(\mathbb{T}\mathcal{F}, \mathcal{F}) + \alpha + \epsilon(1-\gamma)^2/(2\bar{\gamma}\mathbb{C}_{\nu,\mu}^{1/p})\right) \\
& & + \left(\gamma^{d_{\min}(K+1)+(1-\psi)(d-d_{\min})\lfloor Z/\hat{j}\rfloor}\right)^{1/p}\left(\frac{2\|V^{\Phi^*}-V_0\|_\infty}{(1-\gamma)^2}\right) \\
& = & \mathcal{L}_{p,\nu}(\Phi^*) + \frac{2\bar{\gamma}}{(1-\gamma)^2}\mathbb{C}_{\nu,\mu}^{1/p}\left(b_{p,\mu}(\mathbb{T}\mathcal{F}, \mathcal{F}) + \alpha\right) + \epsilon \\
& & + \left(\gamma^{d_{\min}(K+1)+(1-\psi)(d-d_{\min})\lfloor Z/\hat{j}\rfloor}\right)^{1/p}\left(\frac{2\|V^{\Phi^*}-V_0\|_\infty}{(1-\gamma)^2}\right) \ .
\end{array}$$

□

Appendix C. Proof of Theorems 2, and 3

In this appendix, we prove Theorems 2, and 3. First we will analyze LAVI by proving Theorem 2. Then we use one of the lemmas developed for analyzing LAVI to analyze OFVI with landmark-based options and prove Theorem 3. Throughout this appendix we assume that rewards are bound to $[-R_{\text{MAX}}, 0]$, so that stochastic shortest path problems are well defined.

Each iteration Algorithm 2 performs the operator $\widehat{T}_m$ defined by

$$\left(\widehat{T}_m V\right)(x) = \max_{o \in \mathcal{O}_x} \frac{1}{m} \sum_{j=0}^{m} \left(\widetilde{R}_{x,o}^{(j)} + \gamma^{\tau^{(j)}} \Delta(V, y^{(j)})\right) \qquad (51)$$

Table 2: Errors Impacting Landmark-based VI

Error Name	**Symbol**	**Due to . . .**
Landmark Error	$\varepsilon_{\mathbb{L}}$	selected landmarks
Loc. Planning Error	$\varepsilon_{\mathcal{P}}$	$\mathcal{P}$'s sub-optimality in $\widehat{M}$
Loc. Lipschitz Error	ε_H	terminate in y where $\sigma(y,l) > 0$
Stoc. Plan Failure	ψ	prob. terminating far from $\mathbb{L}$
Relaxation Error	ε_R	increased cost in M
Sampling Error	ε_S	finite # samples

at each state $x \in X$ where

$$\Delta(V, y) = \begin{cases} \max_{l \in \mathbb{L}_\eta(y)} V(l) & \text{if } \mathbb{L}_\eta(y) \neq \emptyset \\ 0 & \text{otherwise} \end{cases} . \tag{52}$$

If we consider the limit of $\widehat{T}_m$ as $m \to \infty$, then we obtain T defined by

$$(TV)(x) = \max_{o \in \mathcal{O}_x} \left(\widetilde{R}_{x,o} + \sum_{t=1}^{\infty} \int_{y \in X} \gamma^t P_t^o(y|x) \Delta(V, y) dy \right) \tag{53}$$

for each state $x \in X$.

However, we would like to compare T to the Bellman optimality operator $\mathbb{T}$ defined by

$$(\mathbb{T}V)(x) = \max_{o \in \mathcal{O}_x} \left(\widetilde{R}_{x,o} + \sum_{t=1}^{\infty} \int_{y \in X} \gamma^t P_t^o(y|x) V(y) dy \right) , \tag{54}$$

for which it is well known to converge to the optimal value function with respect to option set $\mathcal{O}$.

Throughout this analysis we will work with vectors with dimension $|X|$ but we mostly focus on $|\mathbb{L}|$ elements. For vectors V and V' in $[-V_{\text{MAX}}, 0]^{|X|}$, we define the max-norm with respect to the subset of states in $\mathbb{L} \subset X$ by

$$\left\| V - V' \right\|_{\mathbb{L}} = \max_{l \in \mathbb{L}} \left| V(l) - V'(l) \right| , \tag{55}$$

which measures the difference between V and V' only at the states in $\mathbb{L}$.

Table 2 provides an overview of the errors that contribute to the sub-optimality of policies derived from LAVI.

C.1 Proof of Theorem 2

We proceed by bounding the value estimation error, which we use to bound the loss of the derived policy. Next we bound the error due to using a deterministic local planner. Finally, we use these results to prove Theorem 2.

C.1.1 Bounding the Value Estimation Error

Lemma 9. *(Bound Value Estimation Error with $\widehat{T}_m$) Let $\epsilon_1 > 0$, $\delta \in (0,1]$, $K \geq 1$, $V \in [-V_{\mathrm{MAX}}, 0]^{|S|}$, $\mathbb{L}$ be the set of landmark states, $\mathcal{O}$ the set of landmark options, and Φ^* be the optimal policy on M with respect to $\mathcal{O}$. If*

$$m > \frac{1}{2(\epsilon_1(1-\gamma))^2} \ln\left(\frac{2LK}{\delta}\right) \ , \tag{56}$$

all landmark-option pairs have a duration of at least $d_{\min}$, and Assumption 5 holds, then with probability at least $1 - \delta/K$

$$\left\|\widehat{T}_m V - V_M^{\Phi^*}\right\|_{\mathbb{L}} \leq \gamma^{d_{\min}}\left(\psi V_{\mathrm{MAX}} + (1-\psi)\kappa\eta + (1-\psi)\left\|V - V_M^{\Phi^*}\right\|_{\mathbb{L}}\right) + \epsilon_1$$

where ψ is the stochastic plan failure, η is a distance threshold, and κ is the Lipschitz coefficient from Assumption 5. By recursing on this inequality K times, we obtain

$$\left\|V_K - V_M^{\Phi^*}\right\|_{\mathbb{L}} \leq \frac{\gamma^{d_{\min}}\left(\psi V_{\mathrm{MAX}} + (1-\psi)\kappa\eta\right) + \epsilon_1}{1-\gamma^{d_{\min}}} + \left((1-\psi)\gamma^{d_{\min}}\right)^K \left\|V^{\Phi^*} - V_0\right\|_{\mathbb{L}} \tag{57}$$

with probability at least $1-\delta$.

Proof. Notice that for all $l \in \mathbb{L}$

$$\begin{aligned}
\left|(\widehat{T}_m V)(l) - V_M^{\Phi^*}(l)\right| &= \left|(\widehat{T}_m V)(l) - (\mathbb{T}V_M^{\Phi^*})(l)\right| \\
&= \left|(\widehat{T}_m V)(l) - (TV_M^{\Phi^*})(l) + (TV_M^{\Phi^*})(l) - (\mathbb{T}V_M^{\Phi^*})(l)\right| \\
&\leq \left|(\widehat{T}_m V)(l) - (TV_M^{\Phi^*})(l)\right| + \left|(TV_M^{\Phi^*})(l) - (\mathbb{T}V_M^{\Phi^*})(l)\right|
\end{aligned} \tag{58}$$

where the initial equality is due to the fact that $V_M^{\Phi^*}(l) = (\mathbb{T}V_M^{\Phi^*})(l)$, the first step inserts $\left(-(TV_M^{\Phi^*})(l) + (TV_M^{\Phi^*})(l)\right) = 0$, and the last step uses the triangle inequality.

Now let $\mathcal{X}$ denote the event that $\left|\left(\widehat{T}_m V\right)(l) - (TV)(l)\right| \leq \epsilon_1$. If event X occurs, then the first term in the last step of inequality (58) is

$$\begin{aligned}
\left|(\widehat{T}_m V)(l) - (TV_M^{\Phi^*})(l)\right| &= \left|\max_{o \in \mathcal{O}_l}\left(\widetilde{R}_{l,o} + \sum_{t=1}^{\infty}\int_{y \in X} \gamma^t P_t^o(y|x)\Delta(V,y)dy\right) - \right. \\
&\qquad \left. \max_{o \in \mathcal{O}_l}\left(\widetilde{R}_{l,o} + \sum_{t=1}^{\infty}\int_{y \in X} \gamma^t P_t^o(y|x)\Delta(V_M^{\Phi^*},y)dy\right)\right| + \epsilon_1 \\
&\leq \max_{o \in \mathcal{O}_l}\sum_{t=1}^{\infty}\int_{y \in X} \gamma^t P_t^o(y|x)\left|\Delta(V,y) - \Delta(V_M^{\Phi^*},y)\right| dy + \epsilon_1 \\
&\leq \gamma^{d_{\min}} \max_{o \in \mathcal{O}_l}\sum_{t=d_{\min}}^{\infty}\int_{y \in X} P_t^o(y|x)\left|\Delta(V,y) - \Delta(V_M^{\Phi^*},y)\right| dy + \epsilon_1 \ ,
\end{aligned}$$

where the last step in the previous inequality is due to the fact that all landmark-option pairs execute for at least $d_{\min}$ timesteps (meaning that they have effective discount factor less than or equal to $\gamma^{d_{\min}}$). By our choice of m and using Hoeffding's inequality it can easily be shown that event $\mathcal{X}$ occurs for a landmark-option pair with probability at least

$1 - \delta/LK$. Since there are $L = \sum\limits_{l \in \mathbb{L}} |\mathcal{O}_l|$ total landmark-option pairs, then using the union bound we have that event $\mathcal{X}$ holds for all of these landmark-options pairs with probability at least $1 - \sum_{i=1}^{L} \delta/LK = 1 - L\,(\delta/LK) = 1 - \delta/K$.

Now, if $\mathbb{L}_\eta(y) = \emptyset$, then $|\Delta(V, y) - \Delta(V_M^{\Phi^*}, y)| = |0 - 0| = 0$. On the other hand, if $\mathbb{L}_\eta(y) \neq \emptyset$, then we have

$$\begin{aligned}
\left|\Delta(V, y) - \Delta(V_M^{\Phi^*}, y)\right| &= \left| \max_{l' \in \mathbb{L}_\eta(y)} V(l') - \max_{l' \in \mathbb{L}_\eta(y)} V_M^{\Phi^*}(l') \right| \\
&\leq \max_{l' \in \mathbb{L}_\eta(y)} \left| V(l') - V_M^{\Phi^*}(l') \right| \\
&\leq \left\| V - V_M^{\Phi^*} \right\|_{\mathbb{L}} \ ,
\end{aligned}$$

which implies that $\left|(\widehat{T}_m V)(l) - (T V_M^{\Phi^*})(l)\right| \leq \gamma^{d_{\min}}(1 - \psi) \left\| V - V_M^{\Phi^*} \right\|_{\mathbb{L}} + \epsilon_1$ holds for all landmarks with probability at least $1 - \delta/K$.

The second term in the last step of inequality (58) is

$$\begin{aligned}
\left|(T V_M^{\Phi^*})(l) - (\mathbb{T} V_M^{\Phi^*})(l)\right| &= \Bigg| \max_{o \in \mathcal{O}_l} \left(\widetilde{R}_{l,o} + \sum_{t=1}^{\infty} \int\limits_{y \in X} \gamma^t P_t^o(y|x) \Delta(V_M^{\Phi^*}, y) dy \right) - \\
&\qquad \max_{o \in \mathcal{O}_l} \left(\widetilde{R}_{l,o} + \sum_{t=1}^{\infty} \int\limits_{y \in X} \gamma^t P_t^o(y|x) V_M^{\Phi^*}(y) dy \right) \Bigg| \\
&\leq \max_{o \in \mathcal{O}_l} \sum_{t=1}^{\infty} \int\limits_{y \in X} \gamma^t P_t^o(y|x) \left| \Delta(V_M^{\Phi^*}, y) - V_M^{\Phi^*}(y) \right| dy \\
&\leq \gamma^{d_{\min}} \max_{o \in \mathcal{O}_l} \sum_{t=d_{\min}}^{\infty} \int\limits_{y \in X} P_t^o(y|x) \left| \Delta(V_M^{\Phi^*}, y) - V_M^{\Phi^*}(y) \right| dy
\end{aligned}$$

where the last step in the previous inequality is due to the fact that all landmark-option pairs execute for at least $d_{\min}$ timesteps.

With probability at most ψ, we have $|\Delta(V_M^{\Phi^*}, y) - V_M^{\Phi^*}(y)| \leq |0 - V_M^{\Phi^*}(y)| \leq V_{\text{MAX}}$ and with probability $1 - \psi$ we have $|\Delta(V_M^{\Phi^*}, y) - V_M^{\Phi^*}(y)| \leq |\max_{l' \in \mathbb{L}_\eta(y)} V^{\Phi^*}(l') - V^{\Phi^*}(y)| \leq \kappa\eta$. Thus, $\left|(T V^{\Phi^*})(l) - (\mathbb{T} V^{\Phi^*})(l)\right| \leq \gamma^{d_{\min}} \left(\psi V_{\text{MAX}} + (1 - \psi)\kappa\eta\right)$.

By replacing the two terms in the last step of inequality (58) we obtain our result $\left\| TV - V_M^{\Phi^*} \right\|_{\mathbb{L}} \leq \gamma^{d_{\min}} \left(\psi V_{\text{MAX}} + (1 - \psi) \left(\kappa\eta + \left\| V - V^{\Phi^*} \right\|_{\mathbb{L}} \right) \right) + \epsilon_1$ with probability at least $1 - \delta/K$. □

C.1.2 Bounding the Policy Error

Lemma 10. *(Bound Policy Error) Let $\epsilon_2 > 0$, $\mathcal{O}$ be a set of landmark options and $d_{\min} \geq 1$ be the minimum duration of all state-options pairs, $V \in [-V_{\text{MAX}}, 0]^{|\mathbb{L}|}$, $\xi \geq 0$, and Φ^* be the optimal policy with respect to $\mathcal{O}$. Suppose that $\left\| V - V_M^{\Phi^*} \right\|_{\mathbb{L}} \leq \xi$ and*

$$\varphi(x) = \arg\max_{o \in \mathcal{O}_x} \left(\widetilde{R}_{x,o} + \sum_{t=1}^{\infty} \int\limits_{y \in X} \gamma P_t^o(y|x) \max_{l \in \mathbb{L}} \Delta(V, y) dy \right)$$

is the greedy policy with respect to V. If Assumption 5 holds, then

$$\left\| V_M^{\Phi^*} - V_M^{\varphi} \right\|_\infty \leq \frac{\gamma^{d_{\min}} \left((1 - \psi)\left(\xi + \kappa\eta\right) + \psi V_{\text{MAX}} \right)}{1 - \gamma^{d_{\min}}} \tag{59}$$

holds.

Proof. Let $x \in X$, $o = \varphi(x)$, and $o^* = \Phi^*(x)$. Since $\varphi(x) = o$, then

$$\widetilde{R}_{x,o^*} + \sum_{t=1}^{\infty} \int_{y \in X} \gamma^t P_t^{o^*}(y|x)\Delta(V,y)dy \leq \widetilde{R}_{x,o} + \sum_{t=1}^{\infty} \int_{y \in X} \gamma^t P_t^{o}(y|x)\Delta(V,y)dy \ . \tag{60}$$

Let $G = \{y \in X \mid \mathbb{L}_\eta(y) \neq \emptyset\}$ and $\bar{G} = X \backslash G$. The set G contains all states that are closer than η to at least one landmark state. By rearranging we obtain

$$\begin{aligned}
\widetilde{R}_{x,o^*} - \widetilde{R}_{x,o} &\leq \sum_{t=1}^{\infty} \int_{y \in X} \gamma^t \left[P_t^o(y|x) - P_t^{o^*}(y|x)\right] \Delta(V,y)dy \\
\widetilde{R}_{x,o^*} - \widetilde{R}_{x,o} &\leq \sum_{t=1}^{\infty} \int_{y \in G} \gamma^t \left[P_t^o(y|x) - P_t^{o^*}(y|x)\right] \Delta(V,y)dy \\
&\quad + \int_{y \in \bar{G}} \gamma^t \left[P_t^o(y|x) - P_t^{o^*}(y|x)\right] \Delta(V,y)dy \\
\widetilde{R}_{x,o^*} - \widetilde{R}_{x,o} &\leq \sum_{t=1}^{\infty} \int_{y \in G} \gamma^t \left[P_t^o(y|x) - P_t^{o^*}(y|x)\right] \max_{l' \in \mathbb{L}_\eta(y)} V(l')dy \\
\widetilde{R}_{x,o^*} - \widetilde{R}_{x,o} &\leq \sum_{t=1}^{\infty} \left(\int_{y \in G} \gamma^t \left[P_t^o(y|x) - P_t^{o^*}(y|x)\right] \left(\max_{l' \in \mathbb{L}_\eta(y)} V_M^{\Phi^*}(l') + \xi \right) dy \right) \\
\widetilde{R}_{x,o^*} - \widetilde{R}_{x,o} &\leq \sum_{t=1}^{\infty} \left(\int_{y \in G} \gamma^t \left[P_t^o(y|x) - P_t^{o^*}(y|x)\right] \left(V_M^{\Phi^*}(y) + \kappa\eta + \xi \right) dy \right) \ ,
\end{aligned}$$

where the initial inequality rearranges (60) to isolate the reward terms. The first step is obtained by dividing the sum over states into states where $\mathbb{L}_\eta(y) \neq \emptyset$ and states where $\mathbb{L}_\eta(y) = \emptyset$. The third step replaces $\Delta(V,y)$ by its definition. Since $\Delta(V,y) = 0$ for all y where $\mathbb{L}_\eta(y) = \emptyset$, the second term on the right hand side disappears. The fourth step replaces $V(l') \leq V_M^{\Phi^*}(l') + \xi$, and the final step uses Assumption 5 to replace $V_M^{\Phi^*}(l')$ with $V_M^{\Phi^*}(y) + \kappa\eta$.

Now we have

$$\begin{aligned}
V_M^{\Phi^*}(x) - V_M^{\varphi}(x) &= \widetilde{R}_{x,o^*} - \widetilde{R}_{x,o} + \sum_{t=1}^{\infty} \int_{y\in X} \gamma^t \left[P_t^{o^*}(y|x)V_M^{\Phi^*}(y) - P_t^{o}(y|x)V_M^{\varphi}(y)\right] dy \\
&= \widetilde{R}_{x,o^*} - \widetilde{R}_{x,o} \\
&+ \sum_{t=1}^{\infty} \int_{y\in \bar{G}} \gamma^t \left[P_t^{o^*}(y|x)V_M^{\Phi^*}(y) - P_t^{o}(y|x)V_M^{\varphi}(y)\right] dy \\
&+ \sum_{t=1}^{\infty} \int_{y\in G} \gamma^t \left[P_t^{o^*}(y|x)V_M^{\Phi^*}(y) - P_t^{o}(y|x)V_M^{\varphi}(y)\right] dy \\
&\leq \widetilde{R}_{x,o^*} - \widetilde{R}_{x,o} \\
&+ \sum_{t=1}^{\infty} \int_{y\in G} \gamma^t \left[P_t^{o^*}(y|x)V_M^{\Phi^*}(y) - P_t^{o}(y|x)V_M^{\varphi}(y)\right] dy \\
&+ \gamma^{d_{\min}}\psi V_{\mathrm{MAX}} \\
&\leq \left(\sum_{t=1}^{\infty} \int_{y\in G} \gamma^t \left[P_t^{o}(y|x) - P_t^{o^*}(y|x)\right]\left(V_M^{\Phi^*}(y) + \xi + \eta\kappa\right) dy\right) \\
&+ \left(\sum_{t=1}^{\infty} \int_{y\in G} \gamma^t \left[P_t^{o^*}(y|x)V_M^{\Phi^*}(y) - P_t^{o}(y|x)V_M^{\varphi}(y)\right] dy\right) \\
&+ \gamma^{d_{\min}}\psi V_{\mathrm{MAX}} \\
&= \left(\sum_{t=1}^{\infty} \int_{y\in G} \gamma^t \Big[P_t^{o^*}(y|x)V_M^{\Phi^*}(y) - P_t^{o^*}(y|x)\left(V_M^{\Phi^*}(y) + \xi + \eta\kappa\right)\right. \\
&+ \left. P_t^{o}(y|x)\left(V_M^{\Phi^*}(y) + \xi + \eta\kappa\right) - P_t^{o}(y|x)V_M^{\varphi}(y)\Big] dy\right) + \gamma^{d_{\min}}\psi V_{\mathrm{MAX}} \\
&\leq \gamma^{d_{\min}}\left(\sum_{t=1}^{\infty} \int_{y\in G} \Big[P_t^{o}(y|x)\left(V_M^{\Phi^*}(y) - V_M^{\varphi}(y) + \xi + \eta\kappa\right)\Big] dy\right) + \gamma^{d_{\min}}\psi V_{\mathrm{MAX}} \\
&\leq \gamma^{d_{\min}}(1-\psi)\left(\left\|V_M^{\Phi^*} - V_M^{\varphi}\right\|_\infty + \xi + \eta\kappa\right) + \gamma^{d_{\min}}\psi V_{\mathrm{MAX}} \ .
\end{aligned}$$

By recursing on this inequality, we obtain

$$\left\|V_M^{\Phi^*} - V_M^{\varphi}\right\|_\infty \leq \frac{\gamma^{d_{\min}}\left((1-\psi)(\xi+\eta\kappa) + \psi V_{\mathrm{MAX}}\right)}{1-\gamma^{d}} \ .$$

□

C.1.3 Bounding Error in the Deterministic Relaxation

Lemma 11. *Let $\widehat{\Phi}^*$ be the optimal policy over options in $\widehat{M}$, then*

$$\left\|V_{\widehat{M}}^{*} - V_{\widehat{M}}^{\widehat{\Phi}^*}\right\|_\infty \leq \frac{2(\varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}})}{1-\gamma^{d^-}}$$

holds.

Proof. For any $l \in \mathbb{L}$, we have

$$V_{\widehat{M}}^{*}(l) - V_{\widehat{M}}^{\widehat{\Phi}^*}(l) \leq V_{\widehat{M}}^{*}(l) - \max_{l'\in\mathbb{L}_l}\left(\widetilde{R}_{\mathcal{P}(l,l')} + \gamma^{\mathcal{P}(l,l')}V_{\widehat{M}}^{\widehat{\Phi}^*}(l')\right) \ .$$

By the definition of landmark error, we have

$$V^*_{\widehat{M}}(l) - V^{\widehat{\Phi}^*}_{\widehat{M}}(l) \leq \max_{l' \in \mathbb{L}_s} \left(\widetilde{R}_{p^*_G(l,l')} + \gamma^{\left|p^*_G(l,l')\right|} V^*_{\widehat{M}}(l') \right) - \max_{l' \in \mathbb{L}_l} \left(\widetilde{R}_{\mathcal{P}(l,l')} + \gamma^{|\mathcal{P}(l,l')|} V^{\widehat{\Phi}^*}_{\widehat{M}}(l') \right) + \varepsilon_{\mathbb{L}} \ ,$$

and by the definition of planning error, we have

$$V^*_{\widehat{M}}(l) - V^{\widehat{\Phi}^*}_{\widehat{M}}(l) \leq \max_{l' \in \mathbb{L}_l} \left(\widetilde{R}_{\mathcal{P}(l,l')} + \gamma^{|\mathcal{P}(l,l')|} V^{\widehat{\Phi}^*}_{\widehat{M}}(l') \right) - \max_{l' \in \mathbb{L}_l} \left(\widetilde{R}_{\mathcal{P}(l,l')} + \gamma^{|\mathcal{P}(l,l')|} V^*_{\widehat{M}}(l') \right) + \varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}} \ .$$

Now if we take the max over the set of valid landmark destinations from l, we get

$$\begin{aligned} V^*_{\widehat{M}}(l) - V^{\widehat{\Phi}^*}_{\widehat{M}}(l) \quad &\leq \quad \max_{l' \in \mathbb{L}_l} \left(\widetilde{R}_{\mathcal{P}(l,l')} + \gamma^{|\mathcal{P}(l,l')|} V^{\widehat{\Phi}^*}_{\widehat{M}}(l') - \widetilde{R}_{\mathcal{P}(l,l')} - \gamma^{|\mathcal{P}(l,l')|} V^*_{\widehat{M}}(l') \right) + \varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}} \\ &\leq \quad \max_{l' \in \mathbb{L}_l} \gamma^{|\mathcal{P}(l,l')|} \left(V^*_{\widehat{M}}(l') - V^{\widehat{\Phi}^*}_{\widehat{M}}(l') \right) + \varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}} \ . \end{aligned}$$

Since all landmarks are separated by paths of length at least d^- while planning with $\mathcal{P}$, we obtain $\left\| V^*_{\widehat{M}} - V^{\widehat{\Phi}^*}_{\widehat{M}} \right\|_{\mathbb{L}} \leq \frac{\varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}}}{1 - \gamma^{d^-}}$.

Therefore by similar reasoning as above, we can see that for any state $x \in X$

$$V^*_{\widehat{M}}(x) - V^{\widehat{\Phi}^*}_{\widehat{M}}(x) \leq \varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}} + \frac{\gamma\left(\varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}}\right)}{1 - \gamma^{d^-}} \leq \frac{2\left(\varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}}\right)}{1 - \gamma^{d^-}} \ .$$

□

C.1.4 Proof of Theorem 2

Proof. (of Theorem 2)

We apply Lemma 9 with $\epsilon_1 = \frac{\varepsilon_S(1-\gamma^{d_{\min}})^2}{(1-\psi)\gamma^{d_{\min}}}$ and $\delta \in (0, 1]$ and Lemma 10 to obtain

$$\begin{aligned} &\left\| V^{\Phi^*}_M - V^{\varphi_K}_M \right\|_{1,\nu} \leq \frac{\gamma^{d_{\min}}}{1 - \gamma^{d_{\min}}} \Bigg(\psi V_{\mathrm{MAX}} + (1 - \psi) \Bigg(\kappa\eta + \\ &\left(\left(\frac{\gamma^{d_{\min}}}{1 - \gamma^{d_{\min}}} \right) \left[\psi V_{\mathrm{MAX}} + (1 - \psi)\kappa\eta \right] + \frac{\epsilon_1}{1 - \gamma^{d_{\min}}} + (1 - \psi)\gamma^{d_{\min} K} \left\| V^{\Phi^*}_M - V_0 \right\|_{\mathbb{L}} \right) \Bigg) \Bigg) \ . \end{aligned} \tag{61}$$

By replacing $\kappa\eta$ with ε_H and ϵ_1, we get

$$\begin{aligned} \left\| V^{\Phi^*}_M - V^{\varphi_K}_M \right\|_{1,\nu} \leq \frac{\gamma^{d_{\min}}}{1 - \gamma^{d_{\min}}} \left(1 + \frac{(1 - \psi)\gamma^{d_{\min}}}{1 - \gamma^{d_{\min}}} \right) \left(\psi V_{\mathrm{MAX}} + (1 - \psi)\varepsilon_H \right) + \\ \varepsilon_S + (1 - \psi)^2 \gamma^{(K+1)d_{\min}} \left(\frac{\left\| V^{\Phi^*}_M - V_0 \right\|_{\mathbb{L}}}{1 - \gamma^{d_{\min}}} \right) \end{aligned} \tag{62}$$

after rearranging terms.

Due to the definition of relaxation error (Definition 12),

$$\begin{aligned} \left\| V^*_M - V^{\Phi^*}_M \right\|_{1,\nu} \quad &\leq \quad \left\| V^*_{\widehat{M}} - V^{\widehat{\Phi}^*}_{\widehat{M}} \right\|_{1,\nu} + \varepsilon_R \\ &\leq \quad \tfrac{2(\varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}})}{1 - \gamma^{\widehat{d}_{\min}}} + \varepsilon_R \end{aligned} \tag{63}$$

where the last step is due to Lemma 11.

By combining (62) and (63), we obtain

$$\begin{aligned}\|V^* - V^{\varphi_K}\|_{1,\nu} &\leq \tfrac{2(\varepsilon_{\mathbb{L}}+\varepsilon_{\mathcal{P}})}{1-\gamma^{\hat{d}_{\min}}} + \varepsilon_R + \tilde{\varepsilon} + \varepsilon_S + (1-\psi)^2\gamma^{(K+1)d_{\min}}\left(\tfrac{\left\|V_M^{\Phi^*} - V_0\right\|_{\mathbb{L}}}{1-\gamma^{d_{\min}}}\right) \\ &\leq \tfrac{2(\varepsilon_{\mathbb{L}}+\varepsilon_{\mathcal{P}})}{1-\gamma^{\hat{d}_{\min}}} + \varepsilon_R + \tilde{\varepsilon} + \varepsilon_S + \gamma^{(K+1)d_{\min}}\left(\tfrac{\left\|V_M^{\Phi^*} - V_0\right\|_{\mathbb{L}}}{1-\gamma^{d_{\min}}}\right)\end{aligned}$$

where $\tilde{\varepsilon} = \left(\frac{\gamma^{d_{\min}}}{1-\gamma^{d_{\min}}}\right)\left(1 + \frac{(1-\psi)\gamma^{d_{\min}}}{1-\gamma^{d_{\min}}}\right)(\psi V_{\text{MAX}} + (1-\psi)\varepsilon_H)$. □

C.2 Proof of Theorem 3

Proof. (of Theorem 3)

By Corollary 1, we have that

$$\mathcal{L}_{p,\nu}(\varphi_K) \leq \mathcal{L}_{p,\nu}(\Phi^*) + \frac{2\gamma^{d_{\min}}}{(1-\gamma)^2}\mathbb{C}_{\nu,\mu}^{1/p} b_{p,\mu}(\mathbb{T}\mathcal{F}, \mathcal{F}) + \varepsilon_S + \left(\gamma^{d_{\min}(K+1)}\right)^{1/p}\left(\frac{2\left\|V^* - V_0\right\|_\infty}{(1-\gamma)^2}\right) \ . \tag{64}$$

By (63), we have that

$$\mathcal{L}_{p,\nu}(\Phi^*) \leq \frac{2(\varepsilon_{\mathbb{L}} + \varepsilon_{\mathcal{P}})}{1-\gamma^{\hat{d}_{\min}}} + \varepsilon_R \ .$$

The result follows by replacing $\mathcal{L}_{p,\nu}(\Phi^*)$ in (64) with the right hand side from the previous inequality. □

References

Barry, J. L., Kaelbling, L. P., & Lozano-Prez, T. (2011). DetH*: Approximate Hierarchical Solution of Large Markov Decision Processes. In *International Joint Conference on Artificial Intelligence*.

Bertsekas, D. P., & Tsitsiklis, J. (1996). *Neuro-dynamic programming*. Athena Scientific.

Brunskill, E., Leffler, B. R., Li, L., Littman, M. L., & Roy, N. (2008). CORL: A Continuous-State Offset-Dynamics Reinforcement Learner. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI-08)*.

Chassin, D. P., Fuller, J. C., & Djilali, N. (2014). GridLAB-D: An agent-based simulation framework for smart grids. *Journal of Applied Mathematics*, *2014*.

Comanici, G., & Precup, D. (2010). Optimal policy switching algorithms for reinforcement learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pp. 709–714.

Dietterich, T. G., Taleghan, M. A., & Crowley, M. (2013). PAC optimal planning for invasive species management: Improved exploration for reinforcement learning from simulator-defined MDPs. In *Proceedings of the National Conference on Artificial Intelligence*.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271.

Farahmand, A., Ghavamzadeh, M., Szepesvári, C., & Mannor, S. (2008). Regularized fitted Q-iteration: Application to planning. In *Recent Advances in Reinforcement Learning*, pp. 55–68. Springer.

Farahmand, A., Munos, R., & Szepesvári, C. (2010). Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems*.

Fernández, F., & Veloso, M. (2006). Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the* 5th *International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 720–727.

Hart, P., Nilsson, N., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, *4*(2), 100–107.

He, R., Brunskill, E., & Roy, N. (2011). Efficient planning under uncertainty with macro-actions. *Journal of Artificial Intelligence Research*, *40*, 523–570.

Hoey, J., St-Aubin, R., Hu, A. J., & Boutilier, C. (1999). SPUDD: Stochastic Planning Using Decision Diagrams. In *Proceedings of Uncertainty in Artificial Intelligence*, Stockholm, Sweden.

Iba, G. A. (1989). A heuristic approach to the discovery of macro-operators. *Machine Learning*, *3*, 285–317.

Jong, N. K., & Stone, P. (2008). Hierarchical model-based reinforcement learning: Rmax + MAXQ. In *Proceedings of the* 25th *International Conference on Machine Learning*.

Kearns, M., Mansour, Y., & Ng, A. Y. (2002). A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning*, *49*(2-3), 193–208.

Kocsis, L., & Szepesvári, C. (2006). Bandit based Monte-Carlo Planning. In *Machine Learning: ECML–2006*, pp. 282–293. Springer.

Konidaris, G., & Barto, A. (2009). Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems 22*, pp. 1015–1023.

Konidaris, G., & Barto, A. G. (2007). Building portable options: Skill transfer in reinforcement learning.. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Vol. 7, pp. 895–900.

Konidaris, G., Kuindersma, S., Barto, A., & Grupen, R. (2010). Constructing skill trees for reinforcement learning agents from demonstration trajectories. In *Advances in Neural Information Processing Systems*, pp. 1162–1170.

Lazanas, A., & Latombe, J.-C. (1992). Landmark-based robot navigation. Tech. rep., Stanford University.

Lazaric, A., Ghavamzadeh, M., & Munos, R. (2010). Analysis of a classification-based policy iteration algorithm. In *Proceedings of the 27th International Conference on Machine Learning*.

Littman, M. L., Dean, T. L., & Kaelbling, L. P. (1995). On the complexity of solving Markov decision problems. In *Proceedings of the* 11th *conference on Uncertainty in artificial intelligence*, pp. 394–402.

Mankowitz, D. J., Mann, T. A., & Mannor, S. (2014). Time-regularized interrupting options. In *Proceedings of the* 31*st* *International Conference on Machine Learning*.

Mann, T. A. (2014). Cyclic Inventory Management (CIM). https://code.google.com/p/rddlsim/source/browse/trunk/files/rddl2/examples/cim.rddl2. Accessed: 2015-06-29.

Mann, T. A., & Mannor, S. (2014). Scaling up approximate value iteration with options: Better policies with fewer iterations. In *Proceedings of the 31st International Conference on Machine Learning*.

Mannor, S., Menache, I., Hoze, A., & Klein, U. (2004). Dynamic abstraction in reinforcement learning via clustering. In *Proceedings of the* 21st *International Conference on Machine learning*, ICML '04, pp. 71–, New York, NY, USA. ACM.

McGovern, A., & Barto, A. G. (2001). Automatic discovery of subgoals in reinforcement learning using diverse density. In *Proceedings of the* 18th *International Conference on Machine Learning*, pp. 361 – 368, San Fransisco, USA.

Minner, S. (2003). Multiple-supplier inventory models in supply chain management: A review. *International Journal of Production Economics*, *81–82*, 265–279. Proceedings of the 11th International Symposium on Inventories.

Munos, R. (2005). Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*.

Munos, R., & Szepesvári, C. (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, *9*, 815–857.

Peleg, D., & Schäffer, A. A. (1989). Graph spanners. *Journal of Graph Theory*, *13*(1), 99–116.

Peters, J., & Schaal, S. (2008). Reinforcement learning of motor skills with policy gradients. *Neural Networks*, *21*, 682–691.

Precup, D., & Sutton, R. S. (1997). Multi-time models for temporally abstract planning. In *Advances in Neural Information Processing Systems 10*.

Precup, D., Sutton, R. S., & Singh, S. (1998). Theoretical results on reinforcement learning with temporally abstract options. In *Machine Learning: ECML–1998*, pp. 382–393. Springer.

Puterman, M. L. (1994). *Markov Decision Processes - Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc.

Riedmiller, M. (2005). Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *Machine Learning: ECML–2005*, pp. 317–328. Springer.

Sanders, P., & Schultes, D. (2005). Highway hierarchies hasten exact shortest path queries. In Brodal, G., & Leonardi, S. (Eds.), *Algorithms: ESA–2005*, Vol. 3669 of *Lecture Notes in Computer Science*, pp. 568–579. Springer Berlin Heidelberg.

Scarf, H. (1959). The optimality of (s,S) policies in the dynamic inventory problem. Tech. rep. NR-047-019, Office of Naval Research.

Scherrer, B., Ghavamzadeh, M., Gabillon, V., & Geist, M. (2012). Approximate Modified Policy Iteration. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, United Kingdom.

Sethi, S. P., & Cheng, F. (1997). Optimality of (s,S) policies in inventory models with markovian demand. *Operations Research*, *45*(6), 931–939.

Shantia, A., Begue, E., & Wiering, M. (2011). Connectionist reinforcement learning for intelligent unit micro management in starcraft. In *Proceedings of the International Joint Conference on Neural Networks*, pp. 1794–1801. IEEE.

Silver, D., & Ciosek, K. (2012). Compositional planning using optimal option models. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh.

Simsek, O., & Barto, A. G. (2004). Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *Proceedings of the* 21$^{\text{st}}$ *International Conference on Machine Learning*, pp. 95–102, New York, NY, USA. ACM.

Sorg, J., & Singh, S. (2010). Linear options. In *Proceedings of the* 9th *International Conference on Autonomous Agents and Multiagent Systems*, pp. 31–38.

Stolle, M., & Precup, D. (2002). Learning options in reinforcement learning. In *Abstraction, Reformulation, and Approximation*, pp. 212–223. Springer.

Stone, P., Sutton, R. S., & Kuhlmann, G. (2005). Reinforcement learning for robocup soccer keepaway. *Adaptive Behavior*, *13*(3), 165–188.

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

Tamar, A., Castro, D. D., & Mannor, S. (2013). TD methods for the variance of the reward-to-go. In *Proceedings of the 30th International Conference on Machine Learning.*

Wolfe, A. P., & Barto, A. G. (2005). Identifying useful subgoals in reinforcement learning by local graph partitioning. In *Proceedings of the* 22$^{\text{nd}}$ *International Conference on Machine Learning*, pp. 816–823.

Yoon, S. W., Fern, A., & Givan, R. (2007). FF-Replan: A Baseline for Probabilistic Planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 7, pp. 352–359.