Constrained Reinforcement Learning in Hard Exploration Problems

Pankayaraj Pathmanathan Pradeep Varakantham

Singapore Management University, Singapore
pankayarajp@smu.edu.sg, pradeepv@smu.edu.sg

Abstract

One approach to guaranteeing safety in Reinforcement Learning is through cost constraints that are imposed on trajectories. Recent works in constrained RL have developed methods that ensure constraints can be enforced even at learning time while maximizing the overall value of the policy. Unfortunately, as demonstrated in our experimental results, such approaches do not perform well on complex multi-level tasks, with longer episode lengths or sparse rewards. To that end, we propose a scalable hierarchical approach for constrained RL problems that employs backward cost value functions in the context of task hierarchy and a novel intrinsic reward function in lower levels of the hierarchy to enable cost constraint enforcement. One of our key contributions is in proving that backward value functions are theoretically viable even when there are multiple levels of decision making. We also show that our new approach, referred to as Hierarchically Limited consTraint Enforcement (HiLiTE) significantly improves on state of the art Constrained RL approaches for many benchmark problems from literature. We further demonstrate that this performance (on value and constraint enforcement) clearly outperforms existing best approaches for constrained RL and hierarchical RL.

Introduction

Reinforcement learning (RL)(Sutton and Barto 2018) is a framework to represent decision learning problem in Markov Decision Problem (MDP) environments. Recent works of (Mnih et al. 2013; Lillicrap et al. 2015; Mnih et al. 2015; Silver et al. 2016) have shown that Deep Reinforcement Learning can be used to solve large and complex decision making problems. As RL methods permeate the real world it becomes paramount for the agent to handle difficult tasks effectively while having safety measures in place.

In this paper, we are specifically interested in imposing safety constraints associated with cumulative cost or risk accrued by the RL agent. Such constraints defined on trajectories have numerous applications in robot motion planning (Ono et al. 2015; Moldovan and Abbeel 2012; Chow et al. 2015a), resource allocation (Mastronarde and van der Schaar 2010; Junges et al. 2015), and financial engineering (Abe et al. 2010; Di Castro, Tamar, and Mannor 2012). Such problems have been represented using the Constrained MDP (Dalal et al. 2018) framework and model free approaches have been proposed to solve such problems. One of the initial approaches to be developed for addressing such constraints is the Lagrangian method (Chow et al. 2015b). However, such an approach does not provide either theoretical or empirical guarantees in ensuring the constraints are enforced. To counter the issue of safety guarantees, next set of approaches focused on imposing surrogate constraints (El Chamie, Yu, and Açıkmeşe 2016; Gábor, Kalmár, and Szepesvári 1998) on individual state and action pairs. Since the surrogate constraints are typically stricter than the original constraint on the entire trajectory, they were able to provide theoretical guarantees on safety. However, the issue with such type of approaches is their conservative nature, which can potentially hamper the expected reward objective. The next set of approaches, CPO (Constrained Policy Optimization) (Achiam et al. 2017), Lyapunov (Chow et al. 2019), BVF (Satija, Amortila, and Pineau 2020) have since improved the state of art in guaranteeing safety while providing high quality solutions (with regards to expected reward). The most recent of these, referred to as BVF (Satija, Amortila, and Pineau 2020) converts the trajectory based constraint into an instantaneous state dependent constraint by using forward and backward cost value functions thus ensuring that the constraints are fulfilled at every time step. While these approaches have improved the state of art significantly, they primarily work on simple tasks (as shown in our experimental results) and do not work on real-world tasks that are typically a sequence of multiple sub-tasks (e.g., a combination of movement, object detection and discrete decision making) or of a long horizon.

To that end, a second aspect of focus in this paper is handling hard exploration problems, typically multi-level tasks or long horizon tasks. Traditionally, such hard exploration problems have been addressed via hierarchical RL approaches, where the top level of the hierarchy is responsible for taking decisions on the next sub-goal (from a not so large set of sub-goals) and the bottom level takes decisions to achieve the sub-goal. The work of (Kulkarni et al. 2016) introduced a two level Deep HRL, where the top level selects the next sub-goal to be achieved and the lower level takes primitive actions to achieve the sub-goal. But using hierarchies can result in a non stationary transition function

due to the changing policy at the lower level. In order to reduce the issue caused by the non stationarity the work of (Nachum et al. 2018) used a sampling based sub goal relabelling while the work of (Levy, Jr., and Saenko 2017) used a hindsight based sub-goal relabelling. Hierarchical RL has seen significant improvements in the last few years (Pateria et al. 2021) and our work in this paper is complementary to those new approaches.

Our objective in this paper is to ensure safety in decision making of the RL agent through trajectory level cost constraints in hard exploration problems. Unfortunately, existing hierarchical approaches cannot directly be applied to the constrained RL setting by introducing a cost associated with each sub-goal, as the number of sub-goals will then become continuous or extremely large and that makes it quite challenging to learn at the top level. To that end, we develop a scalable approach (with detailed theoretical analysis) for constrained hierarchical RL to improve safety in hard exploration problems.

Contribution summary

More specifically, our key contributions can be summarized as follows:

- A backward value function based hierarchical approach for real world tasks that are typically a sequence of multiple sub-tasks (e.g., a combination of movement, object detection and discrete decision making) or long horizon tasks. Our approach, referred to as HiLiTE (Hierarchically Limited consTraint Enforcement) limits trajectory level cost constraints to the upper level of the hierarchy and creates a new intrinsic reward for lower levels of hierarchy, thereby providing a scalable approach that guarantees enforcement of safety constraints even in hard exploration problems.
- We prove the existence of a unique steady state stationary distribution across all levels of the hierarchy with limited assumptions on exploration in episodic tasks. This uniqueness result is used to showing the equivalence of the steady state distributions induced by the forward and backward semi-Markov decision process at the upper level. These two results facilitate the practical feasibility of backward value functions in a hierarchical framework and also allows us to use the theoretical guarantees on safety that were provided in the original paper (Satija, Amortila, and Pineau 2020).
- We provide detailed experimental evaluation that demonstrates the utility of HiLiTE on multiple hard exploration benchmark problems from literature. Existing Hierarchical RL and constrained RL methods fail to either solve the tasks effectively (low expected reward) or are unable to satisfy the cost constraint or both.

Constrained Reinforcement Learning

In constrained RL, the underlying decision making problem can be represented as a Constrained Markov Decision Problem (CMDP), where the rewards, costs and transitions are not known *a priori*. A CMDP is defined as the tuple:

$$\Big\langle S, A, r, c, \gamma, p, s_0, C \Big\rangle$$

where S, A are a set of states and actions.$r : S \times A \to \mathbb{R}$ is a reward signal associated with every state action pair. Along similar lines, $c : S \to \mathbb{R}$ is a cost signal associated with every state (can also be extended to state, action pairs). $p : S \times A \to \Delta(S)$ is the probability of transitioning to a new state given the current state action pair. s_0 refers to the starting state.

The objective in a CMDP is to compute a policy, $\pi : S \times A \to [0, 1]$, which maximizes long term cumulative reward over a horizon, τ, while ensuring the cumulative cost accumulated is less than cost threshold, C. Formally,

$$\max_{\pi} \ \mathbb{E}\Big[\sum_{t=0}^{\tau} r(s_t, a_t)|s_0, \pi\Big]$$
$$\text{s.t. } \mathbb{E}\left[\sum_{t=0}^{\tau} c\left(s_t\right) \mid s_0, \pi\right] \leq C \tag{1}$$

Backward Value Function (BVF) Approach

We now describe the BVF approach by Satija *et al.* (Satija, Amortila, and Pineau 2020), which converts the trajectory based constraints to state based constraints through the use of backward cost value functions in conjunction with forward cost value functions. While the forward value function estimates the expected cost that would be accumulated starting from the current state s_t till the end of the episode (similar to that of a traditional value function on reward), the backward value function is defined as the expected cost that was accumulated by the agent until the current state s_t starting from s_0. $M(\pi)$ and $B(\pi)$ define the forward and backward Markov chains when following policy π, while $V_{\pi}^{\mathcal{C}}(s_t)$ and $\overleftarrow{V}_{\pi}^{\mathcal{C}}(s_t)$ refer to the forward and backward cost value functions defined on the respective Markov chains.

$$V_{\pi}^{\mathcal{C}}(s_t) = \mathbb{E}_{M(\pi)}\left[\sum_{k=t}^{\tau} c\left(s_t\right)\right] \tag{2}$$

$$\overleftarrow{V}_{\pi}^{\mathcal{C}}(s_t) = \mathbb{E}_{\mathcal{B}(\pi)}\left[\sum_{k=0}^{\tau_{\mathcal{B}}} c\left(s_{t-k}\right)\right] \tag{3}$$

τ_B is the finite horizon of the backward Markov chain with terminal state s_0. Further as shown in Satija *et al.* (Satija, Amortila, and Pineau 2020), the samples from the forward Markov chain can be used to estimate the backward value function as the steady state distribution of both the Markov chains are the same. Formally,

$$\overleftarrow{V}_{\pi}^{\mathcal{C}}(s_t) = \mathbb{E}_{\mathcal{M}(\pi), s_{t-K} \sim \eta^{\pi}(\cdot)}\left[\sum_{k=0}^{K} c\left(s_{t-k}\right)\right] \tag{4}$$

where η^{π} is the steady state distribution induced by policy π. Thus, the cumulative cost constraint of $\mathbb{E}\left[\sum_{t=0}^{T} c\left(s_t\right) \mid s_0, \pi\right] \leq C$ can now be estimated as constraint for each state, $_t$

$$\overleftarrow{V}_{\pi}^{\mathcal{C}}\left(s_t\right) + V_{\pi}^{\mathcal{C}}\left(s_t\right) - c\left(s_t\right) \leq C, \forall s_t \tag{5}$$

In practice, for the instantaneous constraints above to work, both the backward and forward value functions need to be estimated correctly. It has been shown that distribution induced by the backward and forward value functions are equal (Satija, Amortila, and Pineau 2020) and hence experiences of the forward markov chain can be used to estimate the backward value function. Even-though this method ensures constraint satisfaction at all time steps, the method heavily relies on the correct estimation of the cost value functions as they were used to constraint the agents exploration space at every time step. In practice, it comes to multi level, long horizon tasks this estimation can be harder thus constraining he agent's search in a non feasible space thereby affecting the overall performance.

Constrained RL for Hard Exploration Tasks

Deep reinforcement learning has been employed in robots to learn continuous control tasks such as locomotion, movement of arms in accomplishing a task etc. However, most of these tasks are atomic and rarely require complex reasoning and planning to accomplish complex multi-level tasks that are a combination of movement, object interaction and discrete decision making. When required to accomplish such hard exploration tasks with complex multi-level tasks safely in the presence of cumulative cost constraints , existing works in constrained RL fail either with regards to optimizing reward or minimizing cost. This is because the objective in optimization 1 cannot be optimized with regular RL approaches and cost estimation required in enforce the constraint of optimization 1 is significantly more challenging (spread across multiple tasks).

Hierarchical RL methods (Kulkarni et al. 2016; Nachum et al. 2018; Levy, Jr., and Saenko 2017; Dietterich 1999) have shown promise in considering such complex multi-level tasks when optimizing reward, so we adapt them to also consider constraints on cumulative costs (trajectory cost). There are multiple challenges involved in achieving this goal:

1. ***Challenge 1: Handling dependencies and proving the existence of steady state distribution at all levels***. Occupancy measure at a level of the hierarchy is dependent on the occupancy measure at the lower level and vice versa. Thus, the first challenge is in characterizing the dependencies and proving that the steady state distribution exists for all levels of the hierarchy, so that we can utilize backward value functions to covert cumulative constraints to state based constraints.
2. ***Challenge 2: Equivalence of forward and backward semi-MDP at the upper level***. In hierarchical RL, the upper levels of the hierarchy are solving a semi-MDP instead of a regular MDP. To utilize backward value functions in a similar vein to that of in constrained RL at these upper levels, the second challenge is in showing that the steady state distribution of the backward semi-MDP and the forward semi-MDP will remain the same.
3. ***Challenge 3: Implementation of hierarchy in constrained RL.*** There are multiple challenges with regards to implementing the hierarchy in constrained RL. There is only one overall cost constraint, so the first challenge is in resolving the cost constraints for the lower level. In hierarchical RL, there is typically an intrinsic reward defined based on the distance from sub-goal. When working with a cost, distance as an intrinsic reward does not work as it can incentivize the agent to not achieve a subgoal so as to meet the cost threshold. So, the second challenge is in defining a new intrinsic reward.

Model: Hierarchical Constrained MDP

We extend the constrained RL model from Section to consider two levels of decision making[1]. There is an extra element in the constrained MDP tuple, which is the set of subgoals, $\mathcal{G}$[2]. The higher level (coarser) policy, $\pi^u : \mathcal{S} \rightarrow \mathcal{G}$ provides the next sub-goal, g to be considered from the set of goals and sub-goals, $\mathcal{G}$ over a temporally extended period. The lower level policy, $\pi^l : \mathcal{S} \times \mathcal{G} \rightarrow \mathcal{A}$ dictates the lower level atomic action to be taken given the current state and sub-goal being pursued.

Since higher level decisions are made every few time steps (depending on how long the lower level takes to accomplish the goal), there is temporal abstraction and because of this, upper level is no longer an MDP, but a Semi-MDP. It is given by the tuple:

$$\langle S, \mathcal{G}, r^u, \gamma, p^u, s_0, M \rangle$$

The key difference in a semi-MDP compared to an MDP is with regards to the transition probability matrix, p, which now also has to account for the duration of executing the action (a discrete set of values less than M) . Specifically, we have $p^u(s', m|s, a)$ instead of $p^u(s'|s, a)$, where $m(\leq M)$ refers to the duration of moving from s to s' on taking action a. Existing works (Limnios and Swishchuk 2020; Limnios and Opri¢an 2003) has shown that the state transition probabilities and duration probabilities in semi-MDP can be made independent given the source state and action. Hence,:

$$p^u(s', m|s, a) = p^u(s'|s, a) \cdot p^u(m|s, a)$$

where $p^u(m|s, a)$ is referred to also as the sojourn time probability (probability of duration taken is m to transition from state s on taking action a) while $p^u(s'|s, a)$ is the regular state transition probability. r^u refers to sum of all the rewards accumulated while moving towards the goal.

At the lower level, we have a regular MDP given by:

$$\langle S, A, r^l, \gamma, p^l, s_0 \rangle$$

The transition function is the same as the original transition function. However, the reward, r^l refers to the intrinsic rewards and typically represent a distance measure from the sub-goal. We now address the three challenges mentioned earlier.

Challenge 1

The steady state probability distribution induced by the policy π^u on the upper level's semi-MDP, i.e., $d^u_{\pi^u}(s'|s_0)$ is re-

[1]This can easily be extended to more than 2 levels, but for ease of exposition, we will focus on two level hierarchy.

[2]We can potentially also utilize options instead of subgoals

cursively defined as follows:

$$d^u_{\pi^u}(s'|s_0) = \beta^u(s') + \sum_{m \leq M} \sum_{s \in S} \sum_{g \in \mathcal{G}} p^u(s'|s,g) \cdot p^u(m|s,g) \cdot \pi^u(g|s) \cdot d^u_{\pi^u}(s|s_0)$$

Since the lower level is defined by a MDP the steady state probability distribution at the lower level can be defined recursively as follows

$$d^l_{\pi^l}(s'|s,g) = \beta^l(s',g) + \sum_{\tilde{s}} \sum_{a'} T^l(s'|\tilde{s},a').\pi^l(a'|\tilde{s}).d^l_{\pi^l}(\tilde{s}|s,g) \quad (6)$$

Since the lower level is triggered corresponding to a goal, g at a given state s of the upper level, the start state distribution for the lower level can be computed from the steady state distribution of the upper level:

$$\beta^l(s,g) = \pi^u(g|s) \cdot d^u_{\pi^u}(s|s_0)$$

Similarly, $p^u(s'|s,g)$ at the upper level can be replaced by the steady state probability distribution of the lower level $d^l_{\pi^l}(s'|s,g)$ since the start state of the lower level is given by the current state and goal at the lower level (s,g).

In order to show that the Hierarchical RL has a unique steady state probability distribution, we need to first show that there exists a unique stationary distribution at both levels.

Theorem 1. *(Huang 2020) Both the semi-markov process and a markov process have a unique steady state distribution if the corresponding process is ergodic. A MDP is ergodic if all the markov chains/ semi markov chains induced by all possible policies are ergodic.*

In summary, according to the Theorem 1 proving the ergodicity of the MDPs would suffice as a proof for the existence of a unique stationary distribution. Consequently, to prove ergodicity of the MDP or the Semi-MDP, we need to prove the ergodicity of the induced Markov and Semi-Markov chain (regardless of the policy).

Steady state probability distribution at lower level: To prove the existence of a unique steady state probability distribution at the **lower level**, we make the following assumptions on existence of a starting state distribution and finiteness of the lower level MDP.

Assumption 1. *The start state distribution of the lower level hierarchy is a mixture of probability distribution of the upper level's steady state probability distributions induced by all possible upper level policies. (Homogeneity)*

Homogeneity is a common assumption, as also indicated in (Huang 2020). With finite duration episodes, if there is only one starting state, there is a chance that not all states are reachable and this homogeneity assumption is needed

Assumption 2. *The reinforcement learning problem at the lower level always lasts for a finite time steps. That is there is a guarantee that the lower level problem will always terminate after some finite time steps regardless of the end state. (Finiteness)*

This is a reasonable assumption, as most (sub-)tasks of interest are of finite duration. However, in continual

By using these assumptions and closely following the work of (Huang 2020) we prove the ergodicity of the induced Markov chain regardless of the policy.

Lemma 1. *Problem at the lower level is ergodic and has a unique stationary distribution.*

Proof Sketch : We prove ergodicity by showing that the Markov chain at the lower level is irreducible (every state is reachable from every other state) and positive recurrent:

- *Lower Level is irreducible*: Any reachable state that would be reachable in a single episode from the terminal state. Due to the homogeneity the set of reachable states from the terminal state is the same through out the training time. Thus any two states can be reached in a finite number of steps in the span of two episodes via the terminal state.
- *Lower Level HRL problem is positive recurrent*: Following in the lines of the work (Huang 2020) let's assume T^l_s as the first recurrence time of a state s. Define a set of rollouts $\epsilon \in \Phi^l_s$ such that each rollout ϵ starts at state s and terminate once it first encounters the state s. That is $\epsilon = [s.....s]$. By definition $E_{\iota \sim M^l_{\pi_l}}[T^l_s] = E_{\epsilon \in \Phi^l_s}[T^l_s]$. Let n_s be the number of lower level episodes encountered before T^l_s then $E_{\epsilon \in \Phi^l_s}[T^l_s] = \sum_{k>0} Pr(n_s = k).E_{\epsilon \in \Phi^l_s}[T^l_s | n_s = k]$. Due to the homogeneity argument there is a probability of hitting the state s given by $Pr(s \in \epsilon)$ at every $k+1$ episodes. If $Pr(s \in \epsilon) = \alpha_s$ then the $Pr(n_s = k) = (1-\alpha_s)^k.\alpha_s$. By that the probability of $Pr(n_s = \infty) = 0$. Thus k is finite. By definition T^l_s happens to fall in the $k+1th$ episode and since the episode length of each of those episodes is finite by finiteness property $E_{\epsilon \in \Phi^l_s}[T^l_s | n_s = k] \leq (k+1).T^l_{max} < +\infty$ where T^l_{max} is the maximum length of a lower level episode.

If we substitute the conclusion we get the following

$$E_{\epsilon \in \Phi^l_s}[T^l_s] = \sum_{k>0} Pr(n_s = k).E_{\epsilon \in \Phi^l_s}[T^l_s | n_s = k] \quad (7)$$

$$E_{\epsilon \in \Phi^l_s}[T^l_s] \leq \sum_{k>0} (1-\alpha_s)^k.\alpha_s.(k+1).T^l_{max} \quad (8)$$

$$E_{\epsilon \in \Phi^l_s}[T^l_s] \leq T^l_{max}.\alpha_s.\sum_{k>0} (1-\alpha_s)^k.(k+1) \quad (9)$$

$$E_{\epsilon \in \Phi^l_s}[T^l_s] \leq T^l_{max}.\alpha_s.\frac{1}{(1-\alpha_s)} \sum_{k>0} (1-\alpha_s)^{k+1}.(k+1) \quad (10)$$

$$E_{\epsilon \in \Phi^l_s}[T^l_s] \leq T^l_{max}.\alpha_s.\frac{1}{(1-\alpha_s)} \frac{(1-\alpha_s)}{\alpha_s^2} \quad (11)$$

$$E_{\epsilon \in \Phi^l_s}[T^l_s] \leq \frac{T^l_{max}}{\alpha_s} \quad (12)$$

(13)

by using the infinite sum of the Gabriel's Staircase series.

Since $\alpha_s > 0$ due to the irreducibly and T^l_{max} is finite by the finiteness property. Thus the problem at the lower level is positive recurrent.

Steady state probability distribution at upper level: Based on the Theorem 3.3 from (Limnios and Oprisan 2001), we have:

Theorem 2. *A semi-Markov chain is irreducible and positive recurrent if the corresponding embedded Markov chain is irreducible and positive recurrent.*

Proof Sketch We now show that the upper level induces a semi-Markov chain that is ergodic (irreducible and positive recurrent)

Lemma 2. *Problem at the upper level is ergodic and has a unique steady state distribution.*

Proof Sketch :

- *Upper level Semi Markov chain is irreducible*: This translates to showing that the Markov chain induced by any policy and specifically the state transition matrix is irreducible. Using Chapman Kolmogorov equation we can write the probability of going from a state s_i to s_j in n time steps as

$$p^u(s_j, n|s_i) = \sum_k p^u(s_k, m|s_i) \cdot p^u(s_j, n-m|s_k) \quad (14)$$

  where $m \leq n$ and s_k is an intermediate state. Similarly using the same Chapman Kolmogorov equation we recursively further break down $p^u(s_k, m|s_i)$ and $p^u(s_j, n-m|s_i)$ until the transition is atomic (i.e., no other state can be reached in fewer time steps). These atomic transitions are obtained from the lower level steady state probabilities (as mentioned earlier: $\forall g, p^u(s'|s, g) = d^l_{\pi^l}(s'|s, g)$) and since lower level is ergodic, all the states are sufficiently visited. Therefore, upper level Markov chain is irreducible.
- *Upper Level Semi Markov chain is positive recurrent*: Similar to the lower level's markov chain a we can formulate an infinite series for the first time recurrence T^u_s of a state s and prove that the series is convergent thus proving the positive recurrence of the embedded semi markov chain at the upper level.

Challenge 2

In this section, we show equivalence of forward and backward semi-MDP at the upper level. As mentioned in Lemma 2, the embedded Markov chain of the semi-markov chain at the upper level has a unique stationary distribution $d^u_{\pi^u}$. Thus

$$d^u_{\pi^u}(s') = \sum_s p^u(s'|s, \pi^u(s)).d^u_{\pi^u}(s) \quad (15)$$

We can define a MDP backwards in time using the state transition probability $\overleftarrow{p}^u(s, a|s')$ which gives us the probability that the previous state and action s, a had led to the current state s' where the transition probability can be written in terms of the forward MDP using bayes rule as

$$\overleftarrow{p}^u(s, a|s') = \frac{p^u(s'|s, a).d^u_{\pi_u}(s, a)}{\sum_{\tilde{s},\tilde{a}} p^u(s'|\tilde{s}, \tilde{a}).d^u_{\pi_u}(\tilde{s}, \tilde{a})} \quad (16)$$

Due to the ***existence of a unique probability distribution for the higher level MDP*** we have:

$$d^u_{\pi_u}(s, a) = d^u_{\pi_u}(s).\pi^u(a|s) \quad (17)$$

$$p^u(s', a|s, t) = p^u(s'|s, a).\pi^u(a|s) \quad (18)$$

$$d^u_{\pi_u}(s') = \sum_{\tilde{s}} \sum_{\tilde{a}} p^u(s'|\tilde{s}, \tilde{a}).d^u_{\pi_u}(\tilde{s}, \tilde{a}) \quad (19)$$

Including these equations in Equation 16, we have:

$$\overleftarrow{p}^u(s, a|s') = \frac{p^u(s', a|s).d^u_{\pi_u}(s)}{d^u_{\pi_u}(s')} \quad (20)$$

We can define a backward semi Markov process (considering the duration probability) via the backward transition probability as follows

$$\overleftarrow{p}^u(s, a|m, s') = \overleftarrow{p}^u(s, a|s') \cdot p^u(m|s', a) \quad (21)$$

We will refer to the forward and backward semi-markov chain induced by the upper level policy as $M^u_{\pi_u}$ and $B^u_{\pi_u}$ where each of those chains are governed by the forward and backward transition probabilities $p^u(s', m|s, a)$ and $\overleftarrow{P}^u(s, a|m, s')$ respectively.

We now define the backward and forward value functions at the upper level. Given a single lower level episode of maximum length M let us define the cumulative cost accumulated during the episode starting from a state s_t as

$$\mathbb{C}^l(s_t) = \sum_{k=t}^{M+t} c(s_k). \quad (22)$$

The forward cost value function $V^{\mathcal{C}}_{\pi_l}$ is the expectation of the cumulative cost over the steady state probabilities of the lower level. The backward and forward cost function at the upper level can thus be defined as:

$$V^{\mathcal{C}}_{\pi_u}(s_t) = \mathbb{E}_{M^u(\pi_u)} \left[\sum_{j=t}^{T} \mathbb{C}^l(s_j) \right] \quad (23)$$

$$\overleftarrow{V}^{\mathcal{C}}_{\pi_u}(s_t) = \mathbb{E}_{\mathcal{B}^u(\pi_u)} \left[\sum_{j=0}^{T_{\mathcal{B}}} \mathbb{C}^l(s_{t-j}) \right] \quad (24)$$

where the deterministic cost function c is now replaced by a stochastic cumulative cost $\mathbb{C}^l$. Index k is used to index the lower level while the index j is used to index the episode at the higher level. The stochasticity can be alleviated to some level if we replace the single cumulative return of the cost at lower level $\mathbb{C}^l$ with the expected estimation given by the lower tier's value function $V^{\mathcal{C}}_{\pi_l}$ thus resulting an alternative definition given by

$$V^{\mathcal{C}}_{\pi_u}(s_t) \approx \mathbb{E}_{M^u(\pi_u)} \left[\sum_{j=t}^{T} V^{\mathcal{C}}_{\pi_l}(s_j) \right] \quad (25)$$

$$\overleftarrow{V}^{\mathcal{C}}_{\pi_u}(s_t) \approx \mathbb{E}_{\mathcal{B}^u(\pi_u)}\left[\sum_{j=0}^{T_{\mathcal{B}}} V^{\mathcal{C}}_{\pi_l}(s_{t-j})\right] \quad (26)$$

Theorem 3. *The samples from the forward and the backward semi-MDP can interchangeably be used to estimate the backward value function.*

Proof Sketch

- The proof follows the same line as the proof in (Satija, Amortila, and Pineau 2020). Lets define a sequence of states $s_1, s_2, s_3, ...s_n$ that were obtained by the agent performing the upper level policy π_u on the embedded markov chain of the upper level $P^u(t)$. Let the actions taken by the agent those states be $a_1, a_2, a_3, ...a_n$. By markov property and eq. 20

$$\overleftarrow{P}(s_1, a_1, s_2, a_2 ... s_{n-1}, a_{n-1}|s_n, t) =$$
$$\overleftarrow{P}(s_{n-1}, a_{n-1}|s_n, t) ... \overleftarrow{P}(s_1, a_1|s_2, t)$$
$$\overleftarrow{P}(s_1, a_1, s_2, a_2 ... s_{n-1}, a_{n-1}|s_n, t) =$$
$$\frac{P(s_n, a_{n-1}|s_{n-1}, t) ... P(s_2, a_1|s_1, t).d^u_{\pi_u}(s_1)}{d^u_{\pi_u}(s_n)}$$
$$\overleftarrow{P}(s_1, a_1, s_2, a_2 ... s_{n-1}, a_{n-1}|s_n, t) \propto$$
$$P(s_n, a_{n-1}|s_{n-1}, t) ... P(s_2, a_1|s_1, t).d^u_{\pi_u}(s_1)$$

  Furthermore, $C^u(t, m) \propto P^u(t)$ and $\overleftarrow{C}^u(t, m) \propto \overleftarrow{P}^u(t)$. Thus the samples from the forward semi-MDP can be used interchangeably with the backward MDP. That completes the proof.

Thus, the backward value function can alternatively be calculated as

$$\overleftarrow{V}^{\mathcal{C}}_{\pi_u}(s_t) \approx \mathbb{E}_{\mathcal{M}^u(\pi_u)}\left[\sum_{j=0}^{T_{\mathcal{B}}} V^{\mathcal{C}}_{\pi_l}(s_{t-j})\right] \quad (27)$$

Hence, the instantaneous estimation of the of total cumulative rewards for the upper level is given by

$$\mathbb{E}_{s_t \sim d^u_{\pi_u}(\cdot)}\left[\overleftarrow{V}^{\mathcal{C}}_{\pi_u}(s_t) + V^{\mathcal{C}}_{\pi_u}(s_t) - V^{\mathcal{C}}_{\pi_l}(s_t)\right] \leq C \quad (28)$$

$V^{\pi_l}_{\mathcal{C}}$ is the forward cost function at the lower level and it is used to estimate the cumulative cost of the lower level episode.

Challenge 3

One intuitive approach to solve hierarchical constrained RL is for the upper level to generate a cost constraint for the lower level along with a goal. Then, both levels can use backward cost value functions to enforce constraints. However, such an approach has three major issues. First, the number of possible combinations of goals and costs are infinite, as cost is usually a continuous number. Second, in the lower level of a hierarchical RL, we have intrinsic reward which is typically distance from goal and this results in subgoals not being achieved to enforce the cost constraint. Finally, it is a non-trivial problem to break up the overall cost constraint into multiple cost constraints for the lower level without being overly conservative or aggressive.

Instead of trying to find an approach that addresses all three issues and learns well in hard exploration settings, we pursued a slightly different approach that is significantly more scalable and avoids some of these issues entirely. The key idea is to enforce the constraints only at the upper level and ensure the lower level reaches the goal with a minimum cost. This approach completely avoids the first and third issues (of infinite goal/cost combinations and splitting of overall cost constraint) and has to only deal with the second issue of designing a new intrinsic reward. In case of multiple level hierarchies we can use BVF based instantaneous cost estimation on the upper level and treat the cost as auxiliary reward (negative reward/regret) in the lower levels. Ergodicity property can be achieved the same way as for a single level, through the use of Assumption 1.

Since the objective is to reach the sub-goal at the minimum cost, we considered rewards that are positive only when the agent reaches the goal and elsewhere it is zero. The overall objective for the lower level is to maximize $Q_L(s, a) - \lambda.Q^C_L(s, a)$, where Q_L is the Q value function at the lower level corresponding to the new intrinsic reward and $Q^C_L(s, a)$ is the cost Q value functions which is defined as an estimation of the accumulated future cost given a current state action pair s, a.

Meanwhile, at the upper level the constraints are enforced as follows

$$\mathbb{E}_{s_t \sim \eta^{\pi}_h(\cdot)}\left[\overleftarrow{V}^{\pi_h}_{\mathcal{C}}(s_t) + V^{\pi_h}_{\mathcal{C}}(s_t) - V^{\pi_l}_{\mathcal{C}}(s_t)\right] \leq C_0 \quad (29)$$

where the forward cost value function at the lower level $V^{\pi_l}_{\mathcal{C}}(s_t)$ as the instantaneous cost for the upper level. A combination of all these ideas together is our new approach, referred to as HiLite (Hierarchical Limited consTraint Enforcement).

Experiments

In this work we designed the experiments to explore the drawbacks of existing constrained RL methods with regards to hard exploration problems. We benchmark the algorithms on modified (to consider costs) versions of two hard exploration environments –namely Grid (Satija, Amortila, and Pineau 2020) and Four rooms (Jain, Khetarpal, and Precup 2018) environments.In the modified version of the Grid environment we adopt the grid world structure from the works of (Leike et al. 2017), (Satija, Amortila, and Pineau 2020) where the objective of an agent is to move from a given start state (blue) reach the key state (aqua) and then move to the goal state (green). The agent is rewarded with a +5000 reward when it reaches the goal state after visiting the key state but is only rewarded a +1000 for visiting the goal state only. The pits (red) carry the cost of +10 every time the agent steps on one. Thus the overall objective of the agent is to reach the goal state after visiting the key state while keeping the number of pits visited within a constraint. In case of the Four Rooms, the environment from (Jain, Khetarpal, and Precup 2018) was modified to facilitate costs. The agent starting from the start state (blue) gets a reward of +1000 for

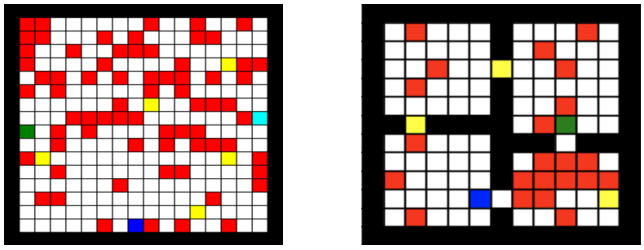

Figure 1: (a) Grid; (b) Four Rooms. In case of the Grid, the black blocks represent the walls while the red blocks denote the pits that would result in the agent inducing a cost upon visitation. The agent starts at the blue block and the goal is to reach the green block while avoiding the red blocks as much as possible. Here the aqua colored block denotes the key. If the agent picks up the key before reaching the goal the agent is rewarded +5000 points as opposed to only +1000 points if it is to reach the goal without the key. the yellow blocks indicate the sub goals that where used by the policy at upper level along with the goal state(green) and key state (aqua). In case of the Four Rooms, the black blocks represent the walls while the red blocks denote the pits that would result in the agent inducing a cost upon visitation. The agent starts at the blue block and the goal is to reach the green block while avoiding the red blocks as much as possible. Here the yellow blocks indicate the sub goals that where used by the policy at upper level along with the goal state(green).

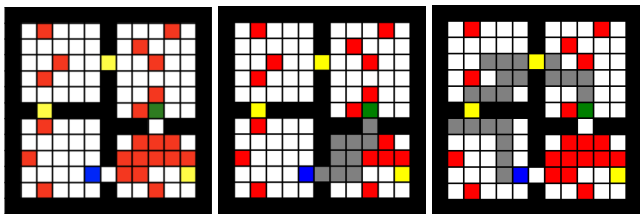

Figure 2: (a) Four Rooms Environment with Costs; (b) Unsafe SARSA in Four Rooms; (c) HiLiTE in Four Rooms. Here the black blocks represent the walls while the red blocks denote the pits that would result in the agent inducing a cost upon visitation. The agent starts at the blue block and the goal is to reach the green block while avoiding the red blocks as much as possible. Here the yellow blocks indicate the sub goals that where used by the policy at upper level along with the goal state(green). The gray boxes denote the path the agent took to reach the goal.

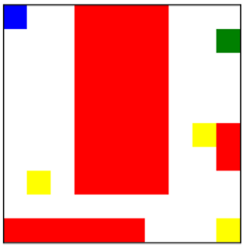

Figure 3: Puddle Environment: In here the red blocks denote the puddles that would result in the agent inducing a cost upon visitation. The agent starts at the blue block and the goal is to reach the green block while avoiding the red blocks as much as possible. Here the yellow blocks indicate the sub goals that where used by the policy at upper level along with the goal.

reaching the goal (green). Every time the agent steps on a pit it accumulated a cost of +10 and the goal of the agent is to reach the goal state while keeping the amount of pit states visited within the minimum bound.

In case of the modified version of the puddle environment (Jain, Khetarpal, and Precup 2018) as denoted in Figure 3, the agent needs to move in a continuous state space and reach the goal (green) starting from the start state (blue) while avoiding the puddles (red). The agent would receive a reward of +1000 if it reaches the goal and it incurs a cost of +10 every time it steps on a puddle. Here agent has to take a longer route in order to reach the goal while avoiding the puddles (cost states).

To demonstrate the utility of HiLiTe, we compare it with multiple baseline approaches: (1) Unconstrained Hierarchical RL (Unsafe-HRL); (2) Backward Value Function approach (BVF) (Satija, Amortila, and Pineau 2020); (3) Lyapunov approach (Dalal et al. 2018). The cost limit provided to these approaches is mentioned in brackets next to the acronym. For instance, BVF with cost constraint of 30, is shown as BVF(30).

Results

We now provide results obtained using our HiLiTE approach in comparison to the baseline approaches on two hierarchical problems with cost constraints. Results on continuous state space problems in the appendix.

Example Policy: Before we delve into the performance results, we would like to provide an example problem and the policies provided by existing methods and our approach to illustrate the effectiveness. Figure 2(a) provides the 4 room environment. As can be seen in this case, there exists a short path from start state to the goal state. However, given the presence of many red cells in between and due to a cost constraint, shortest path or anything close to that is not a viable option. Due to this challenge, this is a hard exploration problem, where a very long path has to be explored to get to the goal. Figure 2(c) shows the path taken by our approach, which is able to find a more round about path, while

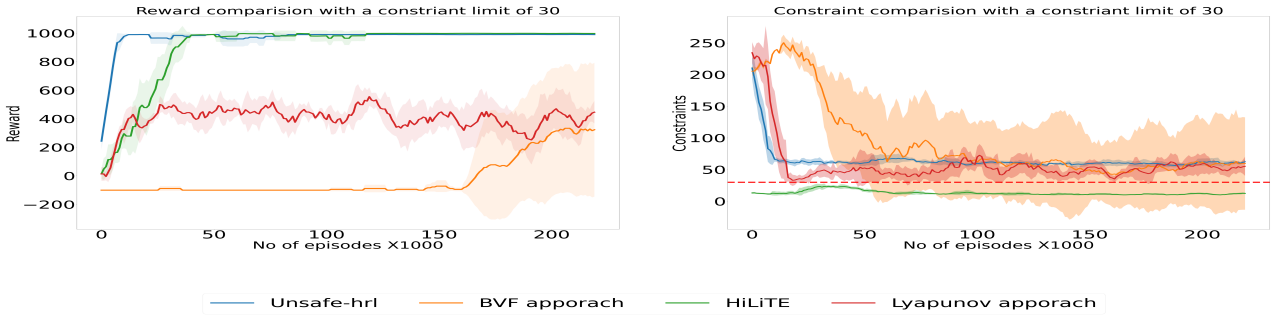

Figure 4: Expected reward and cost comparison of all the approaches with maximum allowed cumulative cost of 30 on 4 room problem.

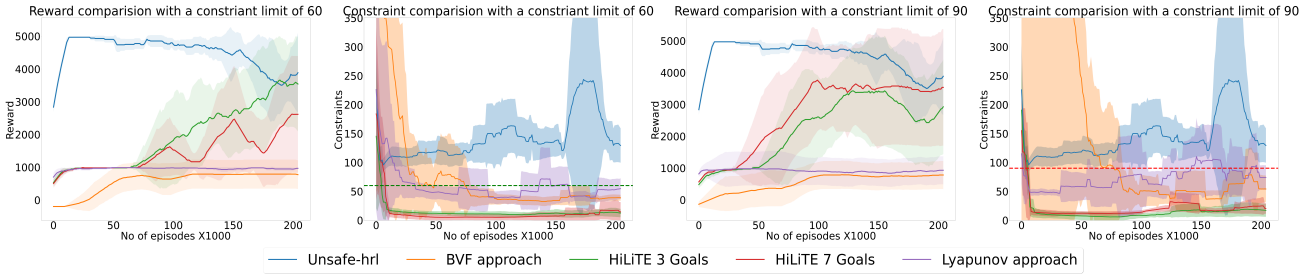

Figure 5: Expected reward, cost comparison of all the approaches with maximum allowed cumulative cost of 60 and 90 respectively on the Grid problem

avoiding the red cells as much as it can to enforce the cost constraint.

Results on Four rooms: The first set of results are on the 4 room problem, where agent has to explore round about paths due to hard cost constraints. Figure 4 provides the results with regards to expected reward and expected cost. The key observation is that none of the existing approaches are able to satisfy the cost constraint of 30 including BVF and Lyapunov approaches, two of the leading works for Constrained RL. More importantly, HiLiTE was able to obtain the same expected reward as Unsafe-hrl and it was able to learn that policy very quickly even in the presence of the cost constraint.

Results on Grid: Here, we varied the complexity of the grid environment by having 3 and 7 sub-goals, while also considering multiple cost limits of 60 and 90. On this set of examples, while both BVF and Lyapunov approaches were able to satisfy the cost constraint, the reward obtained was significantly lower than HiLiTE. Irrespective of the number of sub-goals, we were able to observe similar behavior in terms of high rewards (almost reaching the expected reward of Unsafe-hrl) while satisfying the cost constraints in a very comfortable manner.

Results of Puddle: Figure. 6 provides the results with regards to expected reward and expected cost in the puddle environment. In this case the BVF fails to achieve the goal when the constraints were enforced. In case of the Lyapunov approach, even though initally it manages to achieve a better reward it does so while violating the constraint. As it learns to satisfy the constraints it eventually fails to reach the goal. HiLiTE was able to learn to get a better reward while keeping the incurred cost below the cost constraint of 20.

We have more results in the appendix on continuous state problems, but we were able to clearly demonstrate that our scalable way of introducing hierarchy into constraint reinforcement learning framework works exceedingly well both with regards to reward and cost enforcement.

Acknowledgments

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-017).

References

Abe, N.; Melville, P.; Pendus, C.; Reddy, C. K.; Jensen, D. L.; Thomas, V. P.; Bennett, J. J.; Anderson, G. F.; Cooley, B. R.; Kowalczyk, M.; Domick, M.; and Gardinier, T. 2010. Optimizing Debt Collections Using Constrained Reinforcement Learning. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, 75–84. New York, NY, USA: Association for Computing Machinery. ISBN 9781450300551.

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained Policy Optimization. *CoRR*, abs/1705.10528.

Chow, Y.; Nachum, O.; Faust, A.; Ghavamzadeh, M.; and Duéñez-Guzmán, E. A. 2019. Lyapunov-based Safe

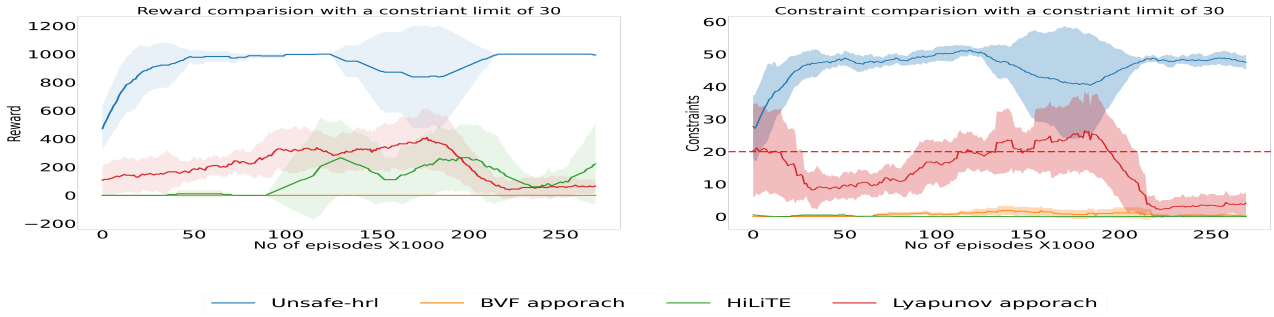

Figure 6: Expected reward and cost comparison of all the approaches with maximum allowed cumulative cost of 20 on Puddle environment.

Policy Optimization for Continuous Control. *CoRR*, abs/1901.10031.

Chow, Y.; Pavone, M.; Sadler, B. M.; and Carpin, S. 2015a. Trading Safety Versus Performance: Rapid Deployment of Robotic Swarms with Robust Performance Constraints. *CoRR*, abs/1511.06982.

Chow, Y.; Tamar, A.; Mannor, S.; and Pavone, M. 2015b. Risk-Sensitive and Robust Decision-Making: a CVaR Optimization Approach. *CoRR*, abs/1506.02188.

Dalal, G.; Dvijotham, K.; Vecerik, M.; Hester, T.; Paduraru, C.; and Tassa, Y. 2018. Safe Exploration in Continuous Action Spaces. arXiv:1801.08757.

Di Castro, D.; Tamar, A.; and Mannor, S. 2012. Policy Gradients with Variance Related Risk Criteria.

Dietterich, T. G. 1999. Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. *CoRR*, cs.LG/9905014.

El Chamie, M.; Yu, Y.; and Açıkmeşe, B. 2016. Convex synthesis of randomized policies for controlled Markov chains with density safety upper bound constraints. In *2016 American Control Conference (ACC)*, 6290–6295.

Gábor, Z.; Kalmár, Z.; and Szepesvári, C. 1998. Multi-criteria Reinforcement Learning. 197–205.

Huang, B. 2020. Steady State Analysis of Episodic Reinforcement Learning. *CoRR*, abs/2011.06631.

Jain, A.; Khetarpal, K.; and Precup, D. 2018. Safe Option-Critic: Learning Safety in the Option-Critic Architecture. *CoRR*, abs/1807.08060.

Junges, S.; Jansen, N.; Dehnert, C.; Topcu, U.; and Katoen, J. 2015. Safety-Constrained Reinforcement Learning for MDPs. *CoRR*, abs/1510.05880.

Kulkarni, T. D.; Narasimhan, K.; Saeedi, A.; and Tenenbaum, J. B. 2016. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. *CoRR*, abs/1604.06057.

Leike, J.; Martic, M.; Krakovna, V.; Ortega, P. A.; Everitt, T.; Lefrancq, A.; Orseau, L.; and Legg, S. 2017. AI Safety Gridworlds. *CoRR*, abs/1711.09883.

Levy, A.; Jr., R. P.; and Saenko, K. 2017. Hierarchical Actor-Critic. *CoRR*, abs/1712.00948.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning.

Limnios, N.; and Oprisan, G. 2001. *Semi-Markov Processes and Reliability*. ISBN 978-1-4612-6640-2.

Limnios, N.; and Opri¢an, G. 2003. Ch. 14. An introduction to semi-markov processes with application to reliability. In *Stochastic Processes: Modelling and Simulation*, volume 21 of *Handbook of Statistics*, 515–556. Elsevier.

Limnios, N.; and Swishchuk, A. 2020. Discrete-Time Semi-Markov Random Evolutions in Asymptotic Reduced Random Media with Applications. *Mathematics*, 8(6).

Mastronarde, N.; and van der Schaar, M. 2010. Fast Reinforcement Learning for Energy-Efficient Wireless Communications. *CoRR*, abs/1009.5773.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing Atari with Deep Reinforcement Learning.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M. A.; Fidjeland, A.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature*, 518: 529–533.

Moldovan, T. M.; and Abbeel, P. 2012. Safe Exploration in Markov Decision Processes. *CoRR*, abs/1205.4810.

Nachum, O.; Gu, S. S.; Lee, H.; and Levine, S. 2018. Data-Efficient Hierarchical Reinforcement Learning. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Ono, M.; Pavone, M.; Kuwata, Y.; and Balaram, J. 2015. Chance-Constrained Dynamic Programming with Application to Risk-Aware Robotic Space Exploration. *Auton. Robots*, 39(4): 555–571.

Pateria, S.; Subagdja, B.; Tan, A.-h.; and Quek, C. 2021. Hierarchical Reinforcement Learning: A Comprehensive Survey. *ACM Comput. Surv.*, 54(5).

Satija, H.; Amortila, P.; and Pineau, J. 2020. Constrained Markov Decision Processes via Backward Value Functions. In *ICML*.

Silver, D.; Huang, A.; Maddison, C.; Guez, A.; Sifre, L.; Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T.; Leach, M.; Kavukcuoglu, K.; Graepel, T.; and Hassabis, D. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529: 484–489.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. The MIT Press, second edition.