Selecting the Partial State Abstractions of MDPs: A Metareasoning Approach with Deep Reinforcement Learning

Samer B. Nashed[1*], Justin Svegliato[2*], Abhinav Bhatia[1], Stuart Russell[2], Shlomo Zilberstein[1]

***Abstract*— Markov decision processes (MDPs) are a common general-purpose model used in robotics for representing sequential decision-making problems. Given the complexity of robotics applications, a popular approach for approximately solving MDPs relies on state aggregation to reduce the size of the state space but at the expense of policy fidelity—offering a trade-off between policy quality and computation time. Naturally, this poses a challenging metareasoning problem: how can an autonomous system dynamically select different state abstractions that optimize this trade-off as it operates online? In this paper, we formalize this metareasoning problem with a notion of time-dependent utility and solve it using deep reinforcement learning. To do this, we develop several general, cheap heuristics that summarize the reward structure and transition topology of the MDP at hand to serve as effective features. Empirically, we demonstrate that our metareasoning approach outperforms several baseline approaches and a strong heuristic approach on a standard benchmark domain.**

I. Introduction

MDPs are a common general-purpose model used in robotics for representing sequential decision-making problems [23]. However, the complexity of solving MDPs scales poorly with the number of features reasoned about in the environment, limiting their applicability. To address this limitation, a range of approximate solvers for MDPs have been proposed that seek to trade a small reduction in policy quality for a large reduction in computation time.

A particularly effective approximate solver for MDPs, recently proposed by Nashed et al. [17], solves *a sequence of partially abstract MDPs* in order to solve an MDP. In a partially abstract MDP, some states are considered at maximum fidelity while other states are considered at lower fidelity using an abstract representation. This can greatly reduce the size of the state space while still resulting in a near optimal policy by using a detailed representation for states where it is most necessary. Still, for a partially abstract MDP to be effective, it requires a suitable *abstraction function* that maps a state in an MDP to an abstract state in an abstract MDP. Since there has been substantial work on generating abstraction functions for planners, ranging from symbolic planners [6], [27], [13] to stochastic planners [1], [7], [26], this paper assumes that a suitable abstraction function already exists via either learning or careful expert design.

Given a specific abstraction function, a partially abstract MDP uses *an expansion strategy* to determine which states of the MDP to reason about at maximum fidelity and which states of the MDP to reason about at lower fidelity. Concretely, this means that the expansion strategy selects the abstract states to *expand* in the partially abstract MDP. Ideally, any approach to selecting an expansion strategy (illustrated in Fig. 1) should optimize a formal notion of time-dependent utility by managing the trade-off between policy quality and computation time given the domain of operation and the available computational resources. Most importantly, such an approach should generalize to any MDP and require little knowledge in the details of that MDP.

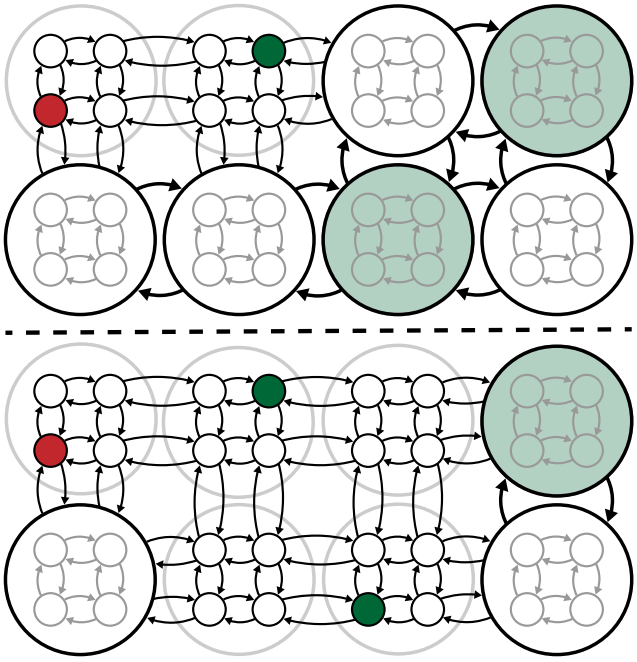

Fig. 1. Two partially abstract MDPs that were constructed using different expansion strategies: a cheap expansion strategy that often results in lower policy quality (*top*) and an expensive expansion strategy that often results in higher policy quality (*bottom*). Small circles are ground states, large circles are abstract states, and arrows are transitions between states. The red ground state is the current state, the green ground states have high reward, and the green abstract states contain a ground state that has high reward.

In this paper, we therefore (1) offer a metareasoning approach to selecting different expansion strategies that optimizes a formal notion of time-dependent utility and (2) express it as deep reinforcement learning problem. Moreover, we (3) propose several general, cheap heuristics that summarize the reward structure and transition topology of the MDP at hand to generate effective features for deep reinforcement learning. Empirically, we demonstrate that our metareasoning approach outperforms several baseline approaches and a strong heuristic approach on a standard benchmark domain.

This work was supported by NSF grants IIS-1813490 and IIS-1954782.

*Both authors contributed equally.

[1]University of Massachusetts, Amherst, MA, USA. Emails: {snashed, abhinavbhati, shlomo}@cs.umass.edu

[2]University of California, Berkeley, CA, USA. Emails: {jsvegliato, russell}@berkeley.edu

II. Related Work

There are many approaches to approximately solving MDPs, including performing dynamic programming, using partial policies, and employing state abstractions. See [17] for a thorough discussion of how these approaches relate to partially abstract MDPs. Similarly, reinforcement learning with time-dependent utility, introduced by Horvitz [10], has been used in a variety of metareasoning problems [24], [22], [4]. Although our approach uses reinforcement learning and time-dependent utility similar to this work, we focus on a different problem—that of learning how to select different expansion strategies dynamically online during operation.

Online planning and learning over different abstractions in general is a common problem across many areas of artificial intelligence and encompasses several related subproblems. These include handling the non-Markovian nature of state and action abstractions [2], learning context-specific independences present in certain tasks [5], and learning temporal abstractions in the form of progressively more abstract skill controllers [14]. However, here, we restrict our attention to state abstractions in the form of state aggregation where multiple states in a larger (ground) problem form a single state in a smaller (abstract) problem.

Online selection of state abstractions has been studied in the context of both reinforcement learning and planning. In reinforcement learning, abstractions are generally used if the state space is large and training data is sparse, which leads to poor experiential coverage. Methods include learning the best state abstraction from a set of state abstractions via hypothesis testing [12] and dynamically selecting state abstractions of increasing granularities based on confidence intervals of Q-values [25]. In planning, similar techniques have been to applied to sample-based tree search algorithms. For example, the PARSS algorithm is an algorithm that adjusts state abstractions during tree search by starting with coarse state abstractions and refining them given the variance of the Q-values over actions at a specific abstract state [11].

An extensive body of research has investigated reasoning over state abstractions during planning. Early work proposed a hierarchy of state abstractions, represented as factored semi-MDPs, that may have multiple intermediate state abstractions that can be swapped in and out depending on the environment [21]. Later work proposed algorithms for dynamically eliminating state factors in states that were estimated to not impact the policy by comparing two partially abstract policies made with different state abstractions [3]. Finally, there have been specific applications, such as multi-agent planning, where specialized partition schemes have been introduced and adapted to an online setting [16].

This paper proposes a metareasoning framework that takes advantage of powerful deep reinforcement learning methods to learn a policy for selecting *different* expansion strategies. Following work on SSPs, we use general, cheap heuristic features that avoid relying on the specifics of an MDP. Most importantly, we define this as a metareasoning problem that optimizes a formal notion of time-dependent utility.

III. Background

In this section, we review the formal definitions of a ground MDP, an abstract MDP, and a partially abstract MDP.

a) Ground MDPs: A ground MDP is a tuple $M = \langle S; A; T; R \rangle$. The space of states is S. The space of actions is A. The transition function $T : S \times A \times S \to [0; 1]$ represents the probability of reaching a state $s' \in S$ after performing an action $a \in A$ in a state $s \in S$. The reward function $R : S \times A \to \mathbb{R}$ represents the immediate reward of performing an action $a \in A$ in a state $s \in S$. A solution is a policy $\pi : S \to A$ indicating that an action $\pi(s) \in A$ should be performed in a state $s \in S$. A policy π induces a value function $V^\pi : S \to \mathbb{R}$ representing the expected discounted cumulative reward $V^\pi(s) \in \mathbb{R}$ for each state $s \in S$ given a discount factor $0 \leq \gamma < 1$. An optimal policy π^* maximizes the expected discounted cumulative reward for each state $s \in S$ given the equation $V^*(s) = \max_{a \in A} R(s; a) + \gamma \sum_{s' \in S} T(s; a; s') V^*(s')$.

b) Abstract MDPs: Specifying an abstract MDP M of a ground MDP M requires two functions [15]. First, an abstraction function $\phi : S \to S$ maps a ground state $s \in S$ to an abstract state $s \in S$. Second, an inverse abstraction function $\phi^{-1} : S \to P(S)$ maps an abstract state $s \in S$ to a set of ground states $S \subseteq P(S)$, where $P(S)$ is the power set of S. The condition $\phi(s) = s \iff s \in \phi^{-1}(s)$ must hold for each ground state $s \in S$ and abstract state $s \in S$.

An abstract MDP is a tuple $M = \langle S; A; T; R \rangle$ [15]. The space of abstract states is $S = \{\phi(s) \mid s \in S\}$ such that a set of ground states S is abstracted by an abstraction function ϕ. The space of ground actions is A. The abstract transition function is $T(s; a; s') = \sum_{s \in \phi^{-1}(s)} \omega(s) \sum_{s' \in \phi^{-1}(s')} T(s; a; s')$. The abstract reward function is $R(s; a) = \sum_{s \in \phi^{-1}(s)} \omega(s) R(s; a)$. Note that a weighting function $\omega : S \to [0; 1]$ represents the probability of being in a ground state $s \in S$ in an abstract state $\phi(s) \in S$.

c) Partially Abstract MDPs: A partially abstract MDP $\bar{M}$ combines a ground MDP M and an abstract MDP M as a tuple $\bar{M} = \langle \bar{S}; A; \bar{T}; R \rangle$ [17]. The space of partially abstract states is $\bar{S} = \bigcup$ with a set of ground states $= \{\phi^{-1}(s) \mid s \in \}$ and a set of abstract states $= \{s \in \}$ such that a set of expanded abstract states S is expanded by an inverse abstraction function ϕ^{-1}. The space of ground actions is A. The partially abstract transition function $\bar{T} : \bar{S} \times A \times \bar{S} \to [0; 1]$ is composed of the ground/abstract transition functions T and T:

$$\bar{T}(\bar{s}; a; \bar{s}') = \begin{cases} T(\bar{s}; a; \bar{s}') & \text{if } \bar{s} \in \ ; \bar{s}' \in \\ \sum_{s' \in \phi^{-1}(\bar{s}')} T(\bar{s}; a; s') & \text{if } \bar{s} \in \ ; \bar{s}' \in \\ \sum_{s \in \phi^{-1}(\bar{s})} \omega(s) T(\bar{s}; a; \bar{s}') & \text{if } \bar{s} \in \ ; \bar{s}' \in \\ T(\bar{s}; a; \bar{s}') & \text{if } \bar{s} \in \ ; \bar{s}' \in \end{cases}$$

The partially abstract reward function $R : \bar{S} \times A \to \mathbb{R}$ is composed of the ground/abstract reward functions R and R:

$$R(\bar{s}; a) = \begin{cases} R(\bar{s}; a) & \text{if } \bar{s} \in \\ R(\bar{s}; a) & \text{if } \bar{s} \in \end{cases}$$

IV. Selecting Partial State Abstractions

The problem of selecting an expansion strategy to determine the abstract states to expand in a partially abstract MDP involves managing the trade-off between policy quality and computation time. We frame this as a metareasoning problem, the main advantage being that it expresses this trade-off in terms of time-dependent utility, providing deep reinforcement learning with an appropriate objective. Methods for similar metareasoning problems typically use heuristics based on statistical measures to manage this trade-off. Here, we introduce the first approach that selects expansion strategies for partially abstract MDPs decision-theoretically.

A. Metareasoning for Partial State Abstractions

We begin by introducing the metareasoning problem for partial state abstractions. This problem requires a *time-dependent utility* that represents the utility of a policy in terms of its quality and computation time. Intuitively, a policy of a specific quality computed in a second has higher utility than a policy of the same quality computed in an hour. A time-dependent utility is therefore expressed as the difference between an *intrinsic value* that reflects the utility of a policy given its quality (but not computation time) and a *time cost* that reflects the utility of a policy given its computation time (but not quality) [10]. We define this function below.

Definition 1. *Given a policy of quality $q \in$ and computation time $t \in$, a* ***time-dependent utility*** $U : \; \rightarrow \mathbb{R}$ *can be expressed as the difference between two functions $U(q,t) = U_I(q) \; U_C(t)$ where $U_I : \; \rightarrow \mathbb{R}^+$ is the* ***intrinsic value*** *and $U_C : \; \rightarrow \mathbb{R}^+$ is the* ***time cost****.*

Given this time-dependent utility, the *one-step* metareasoning problem for partial state abstractions is the problem of selecting the abstract states to expand in a given partially abstract MDP. Naturally, a solution to this problem must optimize time-dependent utility: we must select the abstract states to expand in the partially abstract MDP that balances the quality and computation time of its resulting policy. Formally, given a set of abstract states $_i \in P(S)$ to expand in a partially abstract MDP $\mathcal{M}_i$ and its resulting policy $_i$ of policy quality $q(\,_i)$ and computation time $t(\,_i)$, this one-step metareasoning problem is as follows.

$$\arg\max_{_i \in P(S)} U(q(\,_i), t(\,_i))$$

This can be challenging to solve given substantial uncertainty over the policy $_i$ resulting from a partially abstract MDP $\mathcal{M}_i$ that expands the abstract states $_i \in P(S)$.

In real-time settings, an autonomous system often lazily plans and acts online. Hence, during operation, we assume that the autonomous system is either (1) executing an old local policy when it encounters a *visited* current state s or (2) solving for a new local policy ^{0} when it encounters an *unvisited* current state s^0. We can therefore view the union of each local policy $_i$ as a joint global policy as in our recent work [17] that grows in quality and computation time with each local policy $_i$. Intuitively, this presents a *sequential* metareasoning problem for selecting the abstract states to expand in a sequence of partially abstract MDPs where the resulting local policies $_i$ of each partially abstract MDP $\mathcal{M}_i$ together compose a joint global policy that must optimize time-dependent utility. Formally, given the abstract states $= [\,_1, \ldots, \,_h]$ expanded in a sequence of partially abstract MDPs $[\mathcal{M}_1, \ldots, \mathcal{M}_h]$ over the unvisited states $\{s_1, \ldots s_h\} \in S^h$ and the joint global policy of quality $q(\,)$ and computation time $t(\,)$, this sequential metareasoning problem is as follows.

$$\arg\max U(q(\,), t(\,))$$

In practice, it is often beneficial to approximate this *sequential* metareasoning problem as a sequence of independent *one-step* metareasoning problems as follows.

$$\arg\max_{_1 \in P(S)} U(q(\,_1), t(\,_1)) + \quad + \arg\max_{_h \in P(S)} U(q(\,_h), t(\,_h))$$

B. Reinforcement Learning for Partial State Abstractions

We now cast the sequential metareasoning problem for partial state abstractions as an MDP. Each time an unvisited state $s_i \in S$ is encountered, the MDP must select the abstract states $_i$ to expand in the partially abstract MDP $\mathcal{M}_i$. Intuitively, the *states* include the quality and computation time of the current joint global policy along with the reward structure and transition topology of the ground MDP and abstract MDP while the *actions* include expansion strategies that select the abstract states to expand in the partially abstract MDP. We define this metareasoning problem below.

Definition 2. *The* ***sequential metareasoning problem for partial state abstractions*** *is a tuple $\langle \; ; \; ; F; \hat{S}; \hat{A}; \hat{T}; \hat{R} \rangle$ given a ground MDP M and an abstract MDP $\bar{M}$:*

- $= \{q_0, q_1, \ldots, q_N \; \}$ *is a set of qualities.*
- $= \{t_0, t_1, \ldots, t_N \; \}$ *is a set of computation times.*
- $F = F_0 \quad F_1 \quad \quad F_{N_F}$ *is a set of features that summarize the reward structure and transition topology of the ground MDP M and abstract MDP $\bar{M}$.*
- $\hat{S} =$ F *is a set of states of computation: each state $s \in \hat{S}$ reflects the current joint global policy of quality $q(\,) \in$ and computation time $t(\,) \in$.*
- $\hat{A}$ *is a set of actions of computation: the set of expansion strategies that each select different abstract states $_i$ to expand in a partially abstract MDP $\mathcal{M}_i$.*
- $\hat{T} : \hat{S} \; \hat{A} \; \hat{S} \; \rightarrow [0, 1]$ *is an unknown transition function that represents the probability of reaching state $s^0 = (q^0, t^0, f^0) \in \hat{S}$ after performing action $a \in \hat{A}$ in state $s = (q, t, f) \in \hat{S}$.*
- $\hat{R} : \hat{S} \; \hat{A} \; \hat{S} \; \rightarrow \mathbb{R}$ *is a reward function that represents the expected immediate reward, $\hat{R}(s, a, s^0) = U(q^0, t^0) \; U(q, t)$, of reaching state $s^0 = (q^0, t^0, f^0) \in \hat{S}$ after performing action $a \in \hat{A}$ in state $s = (q, t, f) \in \hat{S}$.*

Note that the reward function is consistent with the objective of optimizing the time-dependent utility: executing a sequence of expansion strategies until a joint global policy of quality $q(\,) \in$ and computation time

$t(\ ) \in$ emits a cumulative reward equal to the time-dependent utility $U(q(\ ); t(\ ))$. This is a form of *reward shaping*—equivalent to emitting a reward of $U(q; t)$ once at the end of an episode in terms of the objective—that guides reinforcement learning with a reward at each time step [18].

We use deep reinforcement learning to learn an optimal metareasoning policy by performing a series of simulations that each use an expansion strategy to select the abstract states to expand in a sequence of partially abstract MDPs. Here, an agent learns a policy as a neural network by performing actions and observing rewards in the environment, making it a good fit for metareasoning for three reasons. First, by balancing exploitation and exploration, it can learn how to select an expansion strategy given the reward structure and transition topology of the ground MDP and abstract MDP. Next, by ignoring large unreachable regions of the state space, it can reduce the overhead of learning which expansion strategy to select. Finally, by using a neural network that extracts the relationship between large input and output spaces, it can encode the effects of an expansion strategy on the resulting policy of a partially abstract MDP in a way that generalizes to novel states of computation.

C. Calculating Time-Dependent Utility

Typically, in metareasoning, a solution quality q is defined as the approximation ratio, $q = \frac{c^*}{c}$, where c^* is the cost of the optimal solution and c is the cost of the given solution. However, since computing the cost of an optimal solution to a complex problem is often infeasible, a solution quality can be estimated as the approximation ratio, $q = \frac{c_*}{c}$, where c_* is a lower bound on the cost of the optimal solution and c is the cost of the given solution. Generally, a solution quality $q = 0$ means no solution was computed while a solution quality $q = 1$ means an optimal solution was computed.

We need a specific definition of solution quality in the context of MDPs. Here, the quality $q(\ )$ of a policy $\ $ is defined as the approximation ratio,

$$q(\ ) = \frac{V\ }{V^*} = \frac{\sum_{s \in S} d(s) V\ (s)}{\sum_{s \in S} d(s) V^*(s)};$$

where $V\ $ is the value function of the policy $\ $ and V^* is the value function of the optimal policy *, given a probability $d(s)$ of starting in a state $s \in S$. However, since computing the value of an optimal policy of a complex MDP is often infeasible, the optimal value function V^* must be replaced with an upper bound on the value function V^*.

Given the quality $q(\ )$ and computation time $t(\ )$ of the current joint global policy $\ $, we can define the time-dependent utility $U(q(\ ); t(\ ))$ using an intrinsic value $U_I(q(\ ))$ and a time cost $U_C(t(\ ))$. First, given a tunable parameter $\ $, we model the intrinsic value as $U_I(q(\ )) = q(\ )$. Second, given a tunable parameter $\ $, we model the time cost as $U_C(t(\ )) = \sum_{i \in h} [e^{t(\ _i)} - 1]$ such that $_i$ is the local policy solved for the unvisited states $\{s_1, \ldots, s_h\} \in S^h$. The rates $\ $ and $\ $ are typically given in the problem depending on the value and urgency for a policy [8].

Given this time-dependent utility, it is possible to express the reward function of the metareasoning problem. Formally, given the current state of computation $s = (q(\ ); t(\ ); \ ) \in \hat{S}$ and the successor state of computation $s' = (q(\ '); t(\ '); \ ) \in \hat{S}$ that reflect the current joint global policy $\ $ and successor joint global policy $\ '$ along with an expansion strategy $a \in \hat{A}$ used to solve for a new local policy $\ $ that improves the joint global policy $\ $, we can express the reward function in the following way.

$$\begin{aligned} \hat{R}(s; a; s') &= U(q(\ '); t(\ ')) - U(q(\ ); t(\ )) \\ &= [q(\ ') - q(\ )] - e^{t(\ )} \end{aligned}$$

V. Representing the State of Computation

In this section, we introduce 6 features that compose the state of computation in the sequential metareasoning problem for partial state abstractions. These features can easily be computed for a ground MDP M and abstract MDP M and reflect their *reward structure* or *transition topology*.

A. Reward Structure

We define 3 features below describing the availability of immediate reward around the current ground/abstract state.

1) ***Reward Frequency****:* The feature f_1 is the number of positive reward ground states reachable within h actions of the current ground state normalized by the total number of reachable ground states.

2) ***Reward Proximity****:* The feature f_2 is the minimum number of actions required to reach the nearest positive reward ground state from the current ground state normalized by the diameter diam(M) of the ground MDP M.

3) ***Reward Information****:* A main weakness of state abstractions is that they induce artificial information boundaries within the state space. For example, when a set of ground states is compressed into an abstract state, the abstract MDP loses information about any ground state that has successor ground states in other abstract states. This is because successor ground states may be aggregated with other ground states that are not reachable in a single action, which is detrimental when a ground state with high reward successor ground states can no longer be distinguished from a ground state without high reward successor ground states. Therefore, the feature f_3 is $1/(1 + |\text{diam}(s) - |)$, where diam($s$) is the diameter of the graph of the ground states in the current abstract state $s \in S$ and $\ $ is the distance to the nearest high reward ground state such that this value approaches 1 or 0 as these ground states move toward or away from this boundary.

B. Transition Topology

We define 3 features below describing the local transition topology surrounding the current ground/abstract state.

1) ***Transition Entropy****:* The feature f_4 is the entropy of the abstract successor state distribution at the current abstract state assuming that actions are selected randomly. This is a rough measure of the probability that actions performed at the current abstract state will transition to different abstract states that may be worth reasoning over more closely. A higher entropy at the current abstract state indicates a higher probability of transitioning to different abstract states.

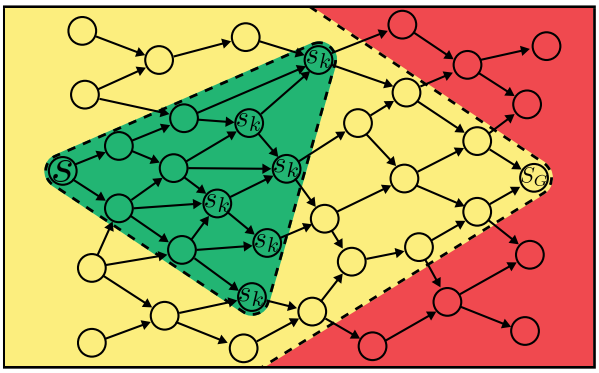

Fig. 2. An example of (k, h)-reachability where $k = 3$ and $h \leq 4$. The *green* region is *all* states reachable from a given ground state s within k actions. The *yellow*/*red* regions are the states from which the set of important ground states S_G is still reachable/unreachable within h actions.

*2) **State Visitation**:* The feature f_5 is the expected discounted number of times that the current abstract state will be visited given an abstract start state distribution and an optimal abstract policy. We compute this for all abstract states $s^0 \in S$ by performing dynamic programming with the equation

$$(s^0) = d(s^0) + \sum_{s \in S} T(s, \quad (s), s^0) \quad (s),$$

where (s) is the optimal abstract policy and $d(s^0)$ is the abstract start state distribution.

*3) **Important Ground State Reachability**:* The feature f_6 is a novel measure of reachability from a given ground state to a set of nearby important ground states.

Definition 3. *A set of important ground states S_G is **(k,h)-reachable** from a given ground state s if, after performing any arbitrary sequence of actions $a_1, \ldots, a_k$, there is at least one important ground state $s_g \in S_G$ that is still reachable within h actions given a probability > 0.*

Similar to recent work on SSPs [20], [19], this measure establishes an envelope (illustrated in Fig. 2) of ground states in which the probability of reaching a set of important ground states S_G from a given ground state s is always greater than zero. In general, while certain MDPs may have transition topologies that permit calculating (k, h)-reachability exactly, it usually must be estimated. To do this, we provide a constant time estimation procedure in Algorithm 1. Here, the accuracy of the estimate improves with the number of samples parameterized by n and m. We choose k to be proportional to the diameter of the current abstract state as this is the maximum number of actions that can performed by the local policy solved for the current abstract state.

Algorithm 1: ESTIMATE (k, h)-REACHABILITY

Input: An MDP M, a ground state s, a set of important ground states S_G, and the parameters k, h, n, and m
Output: The probability that the set of important ground states S_G are (k, h)-reachable from the ground state s

$S_k \leftarrow \emptyset$
for $i \in \{1, \ldots, n\}$ **do**
  $s' \leftarrow s$
  for $j \in \{1, \ldots, k\}$ **do**
    $s' \leftarrow$ SIMULATERANDOMACTION(M, s')
  $S_k \leftarrow S_k \cup \{s'\}$
$\sigma \leftarrow 0$, $\rho \leftarrow \emptyset$
for $s_k \in S_k$ **do**
  for $i \in \{1, \ldots, m\}$ **do**
    $s' \leftarrow s_k$
    for $j \in \{1, \ldots, h\}$ **do**
      $s' \leftarrow$ SIMULATERANDOMACTION(M, s')
      if $s' \in S_G$ **or** $\exists p \in \rho$ such that $s' \in p$ **then**
        $\sigma \leftarrow \sigma + 1$
        $\rho \leftarrow \rho \cup$ PATH(s_k, s')
        break
return σ / n

VI. EXPERIMENTS

We now evaluate the proposed approach (DQN) against a set of baseline approaches on a standard benchmark domain.

a) Hypothesis: Any approach to the metareasoning problem for partial state abstractions should try to optimize time-dependent utility by selecting the abstract states to expand in a sequence of partially abstract MDPs. Ideally, the approach should identify two cases. First, there are cases in which *cheap* and *expensive* expansion strategies result in *roughly equal policy quality*, reducing computation time at negligible sacrifice to policy quality. Second, there are cases in which *expensive* expansion strategies result in *much higher policy quality* than a *cheap* expansion strategy, boosting policy quality at marginal amortized computation time. **Our hypothesis is that the proposed approach will learn to exploit these two cases and hence optimize time-dependent utility beyond the baseline approaches.**

b) Experimental Setup: All approaches were *evaluated* on 100 random simulations. For each simulation, we record three metrics: the values for the *policy quality*, *computation time*, and *time-dependent utility* of the final policy. The proposed approach was *trained* on 1000 random simulations using deep Q-learning with standard settings. The neural network has two hidden layers of 64 and 32 nodes with ReLU activation and a linear output layer of 3 nodes. The step size is 0.0001. The exploration strategy is -greedy action selection with an exploration probability that is annealed from 1 to 0.1 over 1000 episodes. The experience buffer capacity is 1 . The number of steps is 20000. The buffer initialization period is 200. The target network update interval is 1000. The minibatch size is 64. All simulations for training and evaluation were generated using different randomization seeds to measure generalizability to unfamiliar simulations.

c) Standard Benchmark Domain: We consider the Earth observation domain proposed in early work on ground MDPs [9] and recently modified in recent work on partially abstract MDPs [17]. In this domain, a satellite orbiting Earth indefinitely must take photos of points of interest P with weather levels W that change stochastically. The satellite starts at longitude $x \in X$ with its camera focused at latitude $y \in Y$. Given the rates $_Y$ and $_X$, the satellite can then either do NOOPERATION, shift its camera NORTH to latitude $(y + _Y) \in Y$, shift its camera SOUTH to latitude $(y \quad _Y) \in Y$, or take an IMAGE of a point of interest at latitude $y \in Y$ and longitude $x \in X$. Concurrent to each action, the satellite orbits from east to west described by

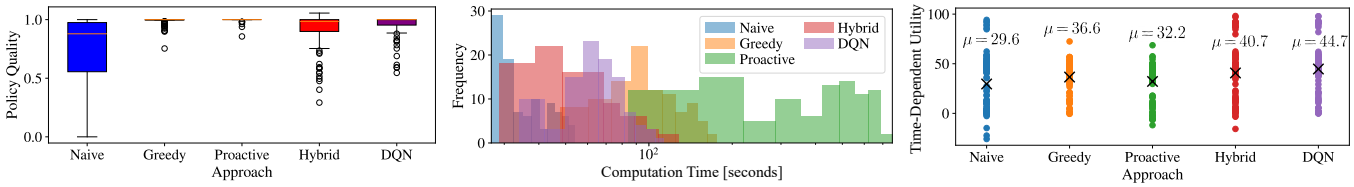

Fig. 3. *Left:* The policy quality of the final policy relative to the optimal policy over all evaluation simulations for each approach. *Center:* The frequency of computation times of the final policy over all evaluation simulations for each approach. *Right:* The distribution of time-dependent utilities of the final policy over all evaluation simulations for each approach. Putting these figures together, we observe that the proposed approach optimizes time-dependent utility more effectively than the baseline approaches by learning how to manage the trade-off between policy quality and computation time.

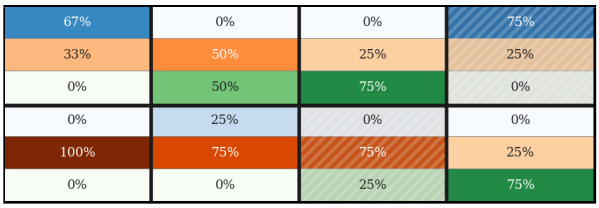

Fig. 4. An example policy for the proposed approach that selects expansion strategies. There are eight abstract states that each contain 2×4 ground states where hatch marks denote a point of interest. Each band within an abstract state represents a specific expansion strategy: *blue* for NAIVE, *orange* for GREEDY, and *green* for PROACTIVE such that darker shading denotes higher probability. This policy shows how the proposed approach exploits reward structure and transition topology of the earth observation MDP to dynamically select the expansion strategy optimizing time-dependent utility.

longitude $((x + \ _x) \bmod jXj)\ 2\ X$ where the modulo operator creates periodic boundary conditions to represent continuous orbits around earth. Most importantly, given the IMAGE action, the satellite earns a reward proportional to image quality such that image quality is a function of the weather $w\ 2\ W$. The formal definitions of the ground, abstract, and partially abstract MDPs are in recent work [17].

d) Baseline Approaches: We consider pure and hybrid approaches that expand *the current abstract state* and *a set of informative abstract states*. The NAIVE approach expands no informative abstract states. The GREEDY approach expands informative abstract states that contain a point of interest within 1 abstract state of the current abstract state. The PROACTIVE approach expands informative abstract states that are reachable from the current abstract state to any abstract state that contains a point of interest within 2 abstract states of the current abstract state. The HYBRID approach uses either the NAIVE, GREEDY, or PROACTIVE approach depending on the $(k; h)$-reachability of the current ground state and the occupancy frequencies of the abstract MDP.

e) Experimental Results: Fig. 3 shows that the proposed approach *optimizes* time-dependent utility beyond the baseline approaches. First, NAIVE, GREEDY, and PROACTIVE exhibit poor time-dependent utility (29:6, 36:6, 32:2). This is because they lead to either high policy quality in too much computation time or low policy quality. Next, HYBRID exhibits better time-dependent utility (40:7) by heuristically reasoning over expansion strategies via careful expert design. Finally, DQN exhibits the best time-dependent utility (44:7) by performing explicit optimization using deep reinforcement learning. Overall, the proposed approach decision-theoretically selects expansion strategies (like in Fig. 4) based on whether investing computation time would result in a worthwhile improvement in policy quality.

VII. CONCLUSION

This paper introduces the metareasoning problem of selecting different expansion strategies and solves it using deep reinforcement learning with several general, cheap heuristics that reflect the MDP at hand. Empirically, we show that our metareasoning approach outperforms several baseline approaches and a strong heuristic approach on a standard benchmark domain. In future work, we will explore the generalizability of this work to MDPs of varied topologies.

REFERENCES

[1] D. Abel, D. Arumugam, L. Lehnert, and M. Littman. State abstractions for lifelong reinforcement learning. In *ICML*, 2018.
[2] A. Bai, S. Srivastava, and S. J. Russell. Markovian state and action abstractions for MDPs via hierarchical MCTS. In *IJCAI*, 2016.
[3] J. Baum, A. E. Nicholson, and T. I. Dix. Proximity-based non-uniform abstractions for approximate planning. *JAIR*, 43, 2012.
[4] A. Bhatia, J. Svegliato, S. B. Nashed, and S. Zilberstein. Tuning the hyperparameters of anytime planning: A metareasoning approach with deep reinforcement learning. In *ICAPS*, 2022.
[5] R. Chitnis, T. Silver, B. Kim, et al. CAMPS: Learning context-specific abstractions for efficient planning. *arXiv:2007.13202*, 2020.
[6] N. S. Flann. Learning appropriate abstractions for planning in formation problems. In *6th ICML*, 1989.
[7] X. Fu, G. Yang, P. Agrawal, and T. Jaakkola. Learning task informed abstractions. In *38th ICML*, 2021.
[8] E. A. Hansen and S. Zilberstein. Monitoring and control of anytime algorithms. *AIJ*, 126(1-2):139–157, 2001.
[9] A. Hertle, C. Dornhege, T. Keller, R. Mattmüller, et al. An experimental comparison of classical, fond and probabilistic planning. In *KI*, 2014.
[10] E. Horvitz and G. Rutledge. Time-dependent utility and action under uncertainty. In *7th UAI*, 1991.
[11] J. Hostetler, A. Fern, and T. Dietterich. Sample-based tree search with fixed and adaptive state abstractions. *JAIR*, 60, 2017.
[12] N. Jiang, A. Kulesza, and S. Singh. Abstraction selection in model-based reinforcement learning. In *ICML*, 2015.
[13] C. A. Knoblock. Generating abstractions for planning. *AIJ*, 68(2), 1994.
[14] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto. Robot learning from demonstration by constructing skill trees. *IJR*, 31(3), 2012.
[15] L. Li, T. J. Walsh, and M. L. Littman. Towards a unified theory of state abstraction for MDPs. In *ISAIM*, 2006.
[16] A. Ma, M. Ouimet, and J. Cortés. Dynamic domain reduction for multi-agent planning. In *MRS*, 2017.
[17] S. B. Nashed, J. Svegliato, M. Brucato, C. Basich, R. Grupen, and S. Zilberstein. Solving Markov decision processes with partial state abstractions. In *ICRA*, 2021.
[18] A. Y. Ng, D. Harada, and S. J. Russell. Policy invariance under reward transformations. In *ICML*, 1999.
[19] L. Pineda and S. Zilberstein. Soft labeling in stochastic shortest path problems. In *18th AAMAS*, 2019.
[20] L. E. Pineda, K. H. Wray, and S. Zilberstein. Fast SSP solvers using short-sighted labeling. In *31st AAAI*, 2017.
[21] K. Steinkraus and L. P. Kaelbling. Combining dynamic abstractions in large MDPs. Technical report, MIT, 2004.
[22] J. Svegliato, P. Sharma, and S. Zilberstein. A model-free approach to meta-level control of anytime algorithms. In *ICRA*, 2020.
[23] J. Svegliato, K. H. Wray, S. J. Witwicki, J. Biswas, and S. Zilberstein. Belief space metareasoning for exception recovery. In *IROS*, 2019.
[24] J. Svegliato, K. H. Wray, and S. Zilberstein. Meta-level control of anytime algorithms with online performance prediction. In *27th IJCAI*, 2018.
[25] M. Tamassia, F. Zambetta, W. L. Raffe, F. Mueller, and X. Li. Dynamic choice of state abstraction in Q-learning. In *ECAI*, 2016.
[26] M. Tomar, A. Zhang, R. Calandra, M. E. Taylor, and J. Pineau. Model-invariant state abstractions for model-based RL. *arXiv:2102.09850*, 2021.
[27] A. Unruh and P. S. Rosenbloom. Abstraction in problem solving and learning. In *11th IJCAI*, 1989.