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 natural, general-purpose way to model stochastic decision-making problems. However, the complexity of solving MDPs forces the use of approximate solvers for many practical applications. One popular approximation method is abstraction through state aggregation, which reduces both the size of the problem and the fidelity of the solution. This also introduces a new problem: how to decide when to use different abstractions in order to optimize the trade-off between MDP policy quality and computation time. This paper formally introduces the choice of state abstraction as a metareasoning problem using time-dependent utility. We then provide several general, cheaply estimated features that serve as effective heuristics for determining abstractions. Finally, we show that one can further optimize performance by using deep reinforcement learning with these features.**

I. Introduction

Planning in stochastic domains is a common problem in robotics, and Markov decision processes (MDPs) are often an effective tool for representing such decision-making problems. However, the complexity of solving MDPs scales exponentially with respect to the number of variables considered during planning, limiting their applicability. To mitigate this limitation, many approximate methods for solving MDPs have been developed, which seek to trade a small amount of plan quality for a large amount of compute savings.

One particularly effective approximate method is the partially abstract MDP [18], where the decision-making agent dynamically selects some parts of the decision-making model to consider at maximum fidelity, while using a smaller, abstract representation for the remaining parts. This approach drastically reduces the size of the problem while mitigating suboptimal behavior by allowing the agent to use the most detailed model where it is most needed. For partially abstract MDPs to be effective, they require an *abstraction function* that maps states in the original MDP to abstract states in an abstract MDP. There has been substantial work on learning how to automatically generate abstractions for planners, including for symbolic planners [6], [14], [27] and stochastic planners [1], [7], [8], [26]. In this paper, we assume such abstractions exist via either learning or careful expert design.

Partially abstract MDPs also require an *expansion strategy* that determines which parts of the MDP to reason about at maximum fidelity during operation. Ideally, these expansion strategies (illustrated in Fig. 1) optimize the trade-off between policy quality and computation time depending on the application and resources available. Moreover, we would like the run time to be independent of both the size of the original MDP, as well as quantities such as the stochastic branching factor, which can slow down sampling methods. Finally, methods for selecting expansion strategies should be generalizable and transferable to other MDPs and require as little domain-specific information as possible.

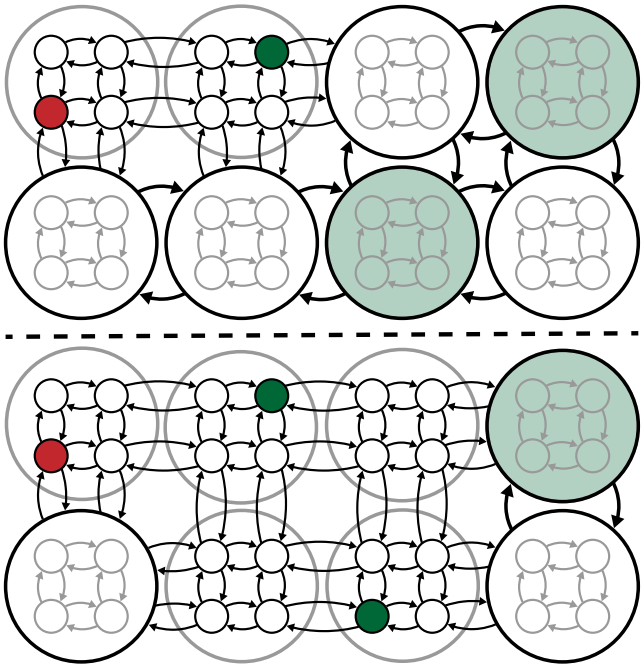

Fig. 1. A diagram of two different partially abstract MDPs, showing current agent state (*red*) and nearby high-reward states (*green*). Small circles represent ground MDP states, large circles represent abstract MDP states, and arrows represent non-zero transition probabilities between states. *Green* abstract states contain high-reward ground states. *Top*: A small expansion resulting in a cheap problem and often a lower quality policy. *Bottom*: A more expensive expansion strategy often resulting in a higher quality policy.

This paper makes three contributions. First, we formalize the decision of which abstract states to expand in a partially abstract MDP as a metareasoning problem using time-dependent utility. Second, inspired by labeling techniques used to generate local policies, we introduce several easily estimated measures of local reward structure and transition topology. We exploit these measures to produce a heuristic approach that outperforms baseline strategies. Finally, we formulate a reinforcement learning (RL) problem for choosing expansion strategies that optimizes time-dependent utility using some of these measures to describe the state of computation. Empirically, we show how a deep RL agent further optimizes the trade-off between computation and plan quality, beyond both the baselines and the heuristic approach.

*Both authors contributed equally.
[1]University of Massachusetts Amherst, {snashed, abhinavbhati,shlomo}@cs.umass.edu
[2]University of California Berkeley, jsvegliato@berkeley.edu
Supported in part by NSF grants IIS-1813490 and IIS-1954782.

II. Related Work

There are many approaches to approximately solving MDPs, including approximate dynamic programming, computing partial policies, and using abstractions. See [18] for a more thorough discussion of how these approaches relate to partially abstract MDPs. Similarly, metareasoning using time-dependent utility, introduced by Horvitz [11], has been applied to a variety of problems [23], [24]. While the contributions in this paper use these techniques, we focus on a different problem—that of learning how and when to use a given problem abstraction dynamically online during plan execution.

Modeling and learning different abstractions online is a common problem across many areas of artificial intelligence and encompasses several related sub-problems. These include dealing with the non-Markovian nature of state-abstracted models [2], learning context-specific independences present in certain tasks [5], and learning useful temporal abstractions in the form of progressively more abstract skills controllers [15]. Though, here, we restrict our attention to abstractions in the form of state aggregation, where multiple states in the original problem are mapped to a single state in a smaller, abstract problem.

Online choice of state aggregation abstractions has been studied in the context of both RL and model-based planning. In RL, abstractions are generally used when data is scarce and the state space is large, leading to poor experiential coverage. Recent techniques include learning the best abstraction from a set of abstractions via hypothesis testing [13] and dynamically choosing abstractions of increasing granularities based on confidence intervals of the Q-values [25]. Similar techniques have been applied to sample-based tree-search. Hostetler et al. [12] introduce the PARSS algorithm, an anytime algorithm that changes abstractions during search by starting with coarse abstractions and refining them based on the variance of the Q-value over actions at particular abstract nodes.

A more extensive body of research has investigated reasoning over abstractions during planning [28]. Early work proposed hierarchies of abstractions represented as factored semi-MDPs, where there may be multiple choices for intermediate abstractions that can be swapped in or out depending on the environment [22]. Later, algorithms were proposed for dynamically removing state factors in states where they were estimated to not affect the policy [3]. Such estimates were made by comparing two partially abstract policies made with different abstractions. Some specialized domains, such as spatial, multi-agent planning, have proposed specialized partition schemes that can be adapted online [17].

In contrast, we propose a system that takes advantage of powerful deep RL methods to learn a strategy for when to use *different* abstractions. Similar to previous approaches, we mainly use cheaply estimated, general features that avoid relying on structures specific to a given MDP. Most importantly, we formally define this as a metareasoning problem that maximizes 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 $M = \langle S, A, T, R \rangle$ [4]. 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 Bellman optimality equation $V^*(s) = \max_{a \in A} \left[R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^*(s') \right]$.

b) Abstract MDPs: Specifying an abstract MDP $\bar{M}$ of a ground MDP M requires two functions [16]. First, an abstraction function $\phi : S \to \bar{S}$ maps a ground state $s \in S$ to an abstract state $\bar{s} \in \bar{S}$. Second, an inverse abstraction function $\phi^{-1} : \bar{S} \to \mathcal{P}(S)$ maps an abstract state $\bar{s} \in \bar{S}$ to a set of ground states $S \subseteq \mathcal{P}(S)$, where $\mathcal{P}(S)$ is the power set of S. The condition $\phi(s) = \bar{s} \Leftrightarrow s \in \phi^{-1}(\bar{s})$ must hold for each ground state $s \in S$ and abstract state $\bar{s} \in \bar{S}$.

An abstract MDP is a tuple $\bar{M} = \langle \bar{S}, A, \bar{T}, \bar{R} \rangle$ [16]. The space of abstract states is $\bar{S} = \{\phi(s) \mid s \in S\}$ 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 $\bar{T}(\bar{s}, a, \bar{s}') = \sum_{s \in \phi^{-1}(\bar{s})} \psi(s) \sum_{s' \in \phi^{-1}(\bar{s}')} T(s, a, s')$. The abstract reward function is $\bar{R}(\bar{s}, a) = \sum_{s \in \phi^{-1}(\bar{s})} \psi(s) R(s, a)$. Note that a weighting function $\psi : S \to [0, 1]$ represents the probability of being in a ground state $s \in S$ in an abstract state $\phi(s) \in \bar{S}$.

c) Partially Abstract MDPs: A partially abstract MDP $\tilde{M}$ combines a ground MDP M and an abstract MDP $\bar{M}$ as a tuple $\tilde{M} = \langle \tilde{S}, A, \tilde{T}, \tilde{R} \rangle$ [18]. The space of partially abstract states is $\tilde{S} = \alpha \cup \beta$ with a set of ground states $\alpha = \{\phi^{-1}(\bar{s}) \mid \bar{s} \in \Gamma\}$ and a set of abstract states $\beta = \{\bar{S} \backslash \Gamma\}$ such that a set of expanded abstract states $\Gamma \subseteq \bar{S}$ is expanded by an inverse abstraction function ϕ^{-1}. The space of ground actions is A. The partially abstract transition function is $\tilde{T} : \tilde{S} \times A \times \tilde{S} \to [0, 1]$ is a piecewise composition of a ground transition function T and an abstract transition function $\bar{T}$:

$$\tilde{T}(\tilde{s}, a, \tilde{s}') = \begin{cases} T(\tilde{s}, a, \tilde{s}') & \text{if } \tilde{s} \in \alpha, \tilde{s}' \in \alpha \\ \sum_{s' \in \phi^{-1}(\tilde{s}')} T(\tilde{s}, a, s') & \text{if } \tilde{s} \in \alpha, \tilde{s}' \in \beta \\ \sum_{s \in \phi^{-1}(\tilde{s})} \psi(s) T(s, a, \tilde{s}') & \text{if } \tilde{s} \in \beta, \tilde{s}' \in \alpha \\ \bar{T}(\tilde{s}, a, \tilde{s}') & \text{if } \tilde{s} \in \beta, \tilde{s}' \in \beta \end{cases}$$

The partially abstract reward function is $\tilde{R} : \tilde{S} \times A \to \mathbb{R}$ piecewise composition of a ground reward function R and an abstract reward function $\bar{R}$:

$$\tilde{R}(\tilde{s}, a) = \begin{cases} R(\tilde{s}, a) & \text{if } \tilde{s} \in \alpha \\ \bar{R}(\tilde{s}, a) & \text{if } \tilde{s} \in \beta \end{cases}$$

IV. Selecting Partial State Abstractions

The problem of selecting abstract states to expand in a partially abstract MDP offers a trade-off between policy quality and compute time. Here, we cast this problem as a metareasoning problem, the main advantage being that it expresses this trade-off formally using time-dependent utility allowing deep RL to learn to optimize this trade-off. Many methods for similar problems use heuristics based on statistical measures to manage this trade-off. To the best of our knowledge, this is the first method for dynamically selecting expansion strategies for partially abstract MDPs.

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) [11]. We define this function below.

Definition 1. *Given a policy of quality $q \in \Phi$ and computation time $t \in \Psi$, a **time-dependent utility** $U : \Phi \times \Psi \to \mathbb{R}$ can be expressed as the difference between two functions $U(q,t) = U_I(q) - U_C(t)$ where $U_I : \Phi \to \mathbb{R}^+$ is the **intrinsic value** and $U_C : \Psi \to \mathbb{R}^+$ 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 the time-dependent utility: we must select the abstract states to expand in the partially abstract MDP in a way that balances the quality and computation time of its resulting policy. Formally, given a set of abstract states $\Gamma_i \in \mathcal{P}(\bar{S})$ to expand in a partially abstract MDP $\tilde{M}_i$ and its resulting policy π_i of quality $q(\pi_i)$ and computation time $t(\pi_i)$, this one-step metareasoning problem is as follows.

$$\arg\max_{\Gamma_i} U(q(\pi_i), t(\pi_i))$$

Note that this problem is challenging to solve given substantial uncertainty over the policy π_i resulting from a partially abstract MDP $\tilde{M}_i$ that expands the abstract states $\Gamma_i \in \mathcal{P}(\bar{S})$.

In many real-time domains, an autonomous system lazily plans and acts online. Hence, during operation, we assume that the autonomous system is either (1) executing an old local policy π when it encounters a *visited* current state s or (2) solving for a new local policy π' when it encounters an *unvisited* current state s'. We can therefore view the union of each local policy π_i as a joint global policy π_Υ (as in recent work [18]) that grows in quality and computation time with each local policy π_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 π_i of each partially abstract MDP $\tilde{M}_i$ together compose a joint global policy π_Υ that must optimize the time-dependent utility. Formally, given the abstract states $\Upsilon = [\Gamma_1, \ldots, \Gamma_h]$ expanded in a sequence of partially abstract MDPs $[\tilde{M}_1, \ldots, \tilde{M}_h]$ over the unvisited states $\{s_1, \ldots s_h\} \in S^h$ and the joint global policy π_Υ of quality $q(\pi_\Upsilon)$ and computation time $t(\pi_\Upsilon)$, this sequential metareasoning problem is as follows.

$$\arg\max_{\Upsilon} U(q(\pi_\Upsilon), t(\pi_\Upsilon))$$

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

$$\arg\max_{\Gamma_1} U(q(\pi_1), t(\pi_1)) + \cdots + \arg\max_{\Gamma_h} U(q(\pi_h), t(\pi_h))$$

B. Reinforcement Learning for Partial State Abstractions

We 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 Γ_i to expand in the partially abstract MDP $\tilde{M}_i$. Intuitively, the *states* include the quality and computation time of the current 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.

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

- *$\Phi = \{q_0, q_1, \ldots, q_{N_\Phi}\}$ is a set of qualities.*
- *$\Psi = \{t_0, t_1, \ldots, t_{N_\Psi}\}$ is a set of computation times.*
- *$F = F_0 \times F_1 \times \cdots \times F_{N_F}$ is a set of features that summarize the reward structure and transition topology of the ground MDP M and abstract MDP $\bar{M}$.*
- *$\hat{S} = \Phi \times \Psi \times F$ is a set of states of computation: each state $s \in \hat{S}$ reflects the current global policy π_Υ of quality $q(\pi_\Upsilon) \in \Phi$ and computation time $t(\pi_\Upsilon) \in \Psi$.*
- *$\hat{A}$ is a set of actions of computation: the set of expansion strategies that each select different abstract states Γ_i to expand in a partially abstract MDP $\tilde{M}_i$.*
- *$\hat{T} : \hat{S} \times \hat{A} \times \hat{S} \to [0,1]$ is an unknown transition function that represents the probability of reaching state $s' = (q', t', f') \in \hat{S}$ after performing action $a \in \hat{A}$ in state $s = (q, t, f) \in \hat{S}$.*
- *$\hat{R} : \hat{S} \times \hat{A} \times \hat{S} \to \mathbb{R}$ is a reward function that represents the expected immediate reward, $\hat{R}(s, a, s') = U(q', t') - U(q, t)$, of reaching state $s' = (q', t', f') \in \hat{S}$ 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 global policy π_Υ of quality $q(\pi_\Upsilon) \in \Phi$ and computation time $t(\pi_\Upsilon) \in \Psi$ emits a cumulative reward equal to the time-dependent utility $U(q(\pi_\Upsilon), t(\pi_\Upsilon))$. 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 accelerates RL by guiding the agent with a reward at each time step [19].

We use deep RL to learn an optimal policy that maps states of computation to actions of computation by performing a series of simulations that each use an expansion strategy to select the abstract states to expand in a partially abstract MDP. A deep RL agent learns a policy as a neural network by performing actions and observing rewards in the world, 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 transition topology and reward structure of the ground MDP and abstract MDP. Next, by ignoring large unreachable regions of the state space, it can reduce the overhead of learning when to select an expansion strategy. 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.

C. Representing 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, $q = \frac{\bar{c}^*}{c}$, where $\bar{c}^*$ 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 $q(\pi)$ of a policy π is defined as the approximation ratio,

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

where V^π is the value function of the policy π and V^* is the value function of the optimal policy π^*, given a probability $d(s)$ of starting in 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 $\bar{V}^*$.

Given the quality $q(\pi_\Upsilon)$ and computation time $t(\pi_\Upsilon)$ of the current global policy π_Υ, we can define the time-dependent utility $U(q(\pi_\Upsilon), t(\pi_\Upsilon))$ using an intrinsic value $U_I(q(\pi_\Upsilon))$ and a time cost $U_C(t(\pi_\Upsilon))$. First, given a tunable parameter α, we model the intrinsic value as $U_I(q(\pi_\Upsilon)) = \alpha q(\pi_\Upsilon)$. Second, given a tunable parameter β, we model the time cost as $U_C(t(\pi_\Upsilon)) = \sum_{i \in h} [e^{\beta t(\pi_i)} - 1]$ such that π_i is the local policy solved for the unvisited states $\{s_1, \dots s_h\} \in S^h$. The rates α and β are typically given in the problem and based on the value/urgency for a policy [9].

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

$$\begin{aligned}\hat{R}(s, a, s') &= U(q(\pi'_\Upsilon), t(\pi'_\Upsilon)) - U(q(\pi_\Upsilon), t(\pi_\Upsilon)) \\ &= \alpha[q(\pi'_\Upsilon) - q(\pi_\Upsilon)] - e^{\beta t(\pi)}\end{aligned}$$

V. Representing the State of Computation

In this section, we introduce 6 features that define the states of the sequential metareasoning problem for partial state abstractions, all of which can be computed or estimated for a given ground MDP M and abstract MDP $\bar{M}$. These features reflect either the *reward structure* or *transition topology* of the ground MDP M or abstract MDP $\bar{M}$.

a) Reward Structure: Two features are rather simple: f_1 is the number of positive reward ground states reachable within h actions normalized by the total number of such states. f_2 is the minimum number of actions required to reach the nearest positive reward state, normalized by diam(M).

f_3 is more complex. In general, a main weakness of state aggregation schemes is that they induce artificial information boundaries within the state space. For example, when an abstract state is expanded, ground states that transition to other ground states beyond the expanded abstract state lose information about successor ground states, since legitimate successor ground states are aggregated with other ground states that are not reachable in one action. This can be detrimental if this loss in information leads to an inability to distinguish between actions that reach a high reward ground state versus actions that prevent visiting such states. Therefore, we define f_3 as

$$f_3 = 1/(1 + |\text{diam}(\bar{s}) - d|),$$

where diam($\bar{s}$) is the diameter of the subgraph of the ground states in abstract state $\bar{s}$ and d is the distance to a high reward ground state. f_3 approaches 1 as such states near this boundary and 0 as they move away from this boundary.

b) Transition Topology: In addition to features that describe the availability of immediate reward and therefore the potential for suboptimal decision making, we also use information about the local topology of the MDP. An expensive prerequisite for using partially abstract MDPs is constructing the abstract MDP $\bar{M}$, particularly its transition function $\bar{T}$. We reuse this computation to compute two features.

First, we analyze the abstract transition function $\bar{T}$. We let f_4 be the entropy of the abstract successor state distribution, assuming actions are selected uniformly at random from the current abstract state. Essentially, this is a rough measure of the likelihood that actions taken in this abstract state will have significantly different outcomes that may be worth reasoning over more closely. The higher the entropy, the more likely different actions will produce different values.

Second, we let f_5 be the relative expected discounted state occupancy of the current abstract state. This is the expected, discounted number of times that a given abstract state will be visited, assuming a particular start state distribution (often

Algorithm 1: ESTIMATE (k, h)-REACHABILITY

Input: MDP M, goals S_G, start s, constants k, h, n, m
Output: Probability that S_G is (k, h)-reachable from s
$S_k \leftarrow \emptyset$
for all $i \in \{1, \ldots, n\}$ **do**
  $s' \leftarrow s$
  for all $j \in \{1, \ldots, k\}$ **do**
    $s' \leftarrow$ SIMULATERANDOMACTION(M, s')
  $S_k \leftarrow S_k \cup s'$
$\sigma \leftarrow 0$, $\rho \leftarrow \emptyset$
for all $s_k \in S_k$ **do**
  for all $i \in \{1, \ldots, m\}$ **do**
    $s' \leftarrow s_k$
    for all $j \in \{1, \ldots, h\}$ **do**
      $s' \leftarrow$ SIMULATERANDOMACTION(M, s')
      if $s' \in S_G$ **or** $\exists p \in \rho$ s.t. $s' \in p$ **then**
        $\sigma \leftarrow \sigma + 1$
        $\rho \leftarrow \rho \cup$ PATH(s_k, s')
        break
return σ / n

uniform) and a particular policy. This can be calculated with respect to the abstract policy $\bar{\pi}$ for all abstract states by dynamic programming using the following update equation:

$$\lambda(\bar{s}') = \bar{d}(\bar{s}') + \gamma \sum_{\bar{s}} \bar{T}(\bar{s}, \bar{\pi}(\bar{s}), \bar{s}')\lambda(\bar{s}),$$

where $\bar{d}(\bar{s})$ is the probability of starting in abstract state $\bar{s}$. Since the abstract policy is fixed, this is computed just once. This equation yields the expected discounted state occupancy. To find the relative value, we simply divide all values by the maximum value over all $\bar{s}$.

Finally, f_6 is a novel measure estimating a local policy's impact on the reachability of nearby high-reward states.

Definition 3. *A subset of states $S_G \subset S$ is* **(k,h)-*reachable*** *with respect to state s if, after executing any arbitrary sequence of actions $a_1, \ldots, a_k$ beginning at state s, at least one state $s' \in S_G$ is still reachable in h or fewer additional actions with probability $\epsilon > 0$.*

Similar to other labeling schemes [21], [20], this measure establishes an envelope of states where the reachability of the set of states S_G from state s is always non-zero. Fig. 2 shows such an envelope. Some MDPs may have transition structures that permit calculating (k, h)-reachability exactly, but in general it must be estimated. Here, we provide a constant time algorithm (Algorithm 1) for estimating (k, h)-reachability, where the quality of the estimate increases as the number of samples, parameterized by n and m, increases. We choose k proportional to the diameter of the current abstract state, since this is the maximum number of actions determined by the current partially abstract MDP solution.

VI. EXPERIMENTS

We now evaluate the proposed approach against a set of baseline approaches of increasing sophistication on a standard benchmark domain for partial state abstractions.

a) Hypothesis: Any approach to our metareasoning problem must manage the trade-off between quality and computation time by selecting the abstract states to expand in a given partially abstract MDP. Intuitively, we describe

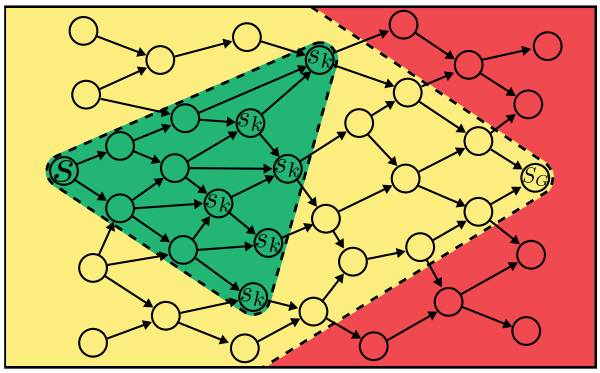

Fig. 2. An example of (k, h)-reachability where $k = 3$ and $h \geq 4$. The *red* and *yellow* regions represent states from which the set of goals S_G is unreachable and reachable, respectively. The *green* region represents *all* states reachable from state s within k actions.

two general cases. First, there are cases where *cheap* and *expensive* expansion strategies result in *roughly equal quality*, thereby reducing computational overhead at no sacrifice to quality. Second, there are cases where *expensive* expansion strategies result in *much higher quality* than a *cheaper* expansion strategy, thereby boosting quality at marginal amortized computation. Our hypothesis is that the proposed approach will identify these cases and balance quality and computation time more effectively than the baseline approaches.

b) Experimental Setup: To test this hypothesis, our approach and the baseline approaches were run on 100 random simulations. For each simulation, we record three metrics: the final *policy quality*, *computation time*, and *time-dependent utility*. We expect the proposed approach to produce higher time-dependent utilities than the baseline approaches.

The proposed approach was trained on 1000 random instances using deep Q-learning with standard settings. The action-value neural network (DQN) 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 ϵ-greedy action selection with an exploration probability ϵ that is annealed from 1 to 0.1 over 1000 episodes. The experience buffer capacity is ∞. The number of steps is 20000. The buffer initialization period is 200. The target network update interval is 1000. The minibatch size is 64. We use different seeds to initialize training simulations and evaluation simulations to ensure that our approach generalizes to unseen or unfamiliar simulations.

c) Standard Benchmark Domain: We consider the Earth observation domain proposed in early work on ground MDPs [10] and recently modified in state-of-the-art work on partially abstract MDPs [18]. 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 Δ_Y and Δ_X, the satellite can then either do NOOPERATION, shift its camera NORTH to latitude $(y + \Delta_Y) \in Y$, shift its camera SOUTH to latitude $(y - \Delta_Y) \in Y$, 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

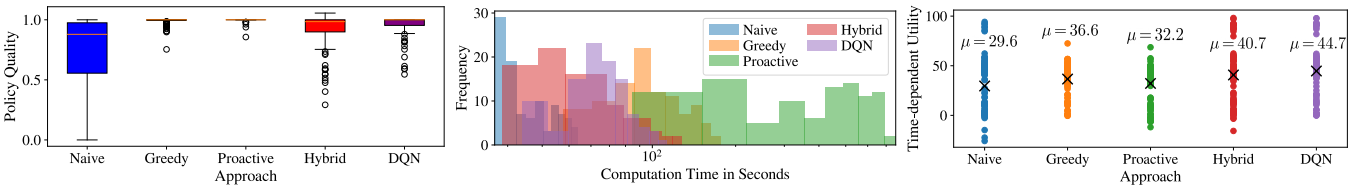

Fig. 3. *Left:* Policy quality relative to the optimal solution. *Center:* Distribution of computation times for each approach over all trials. Together, the left and center plots indicate that both the hybrid and DQN approaches frequently exploit low-cost abstraction strategies, gaining compute savings for small decreases in policy quality. The advantage of the DQN approach is that, if the relative value of computation changes, it may be retrained to optimize the new objective. *Right:* Distribution of time-dependent utilities earned by each approach. The hybrid approach, heuristically informed by the (k, h)-reachability and abstract occupancy frequency measures, improves utility by roughly 10% on average over GREEDY. The DQN approach optimizes the trade-off even further by an additional 10%. Note that multiple modes in the utility are likely due to different point of interest configurations.

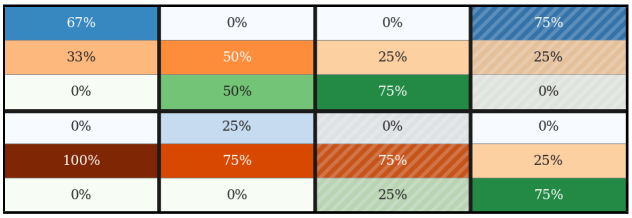

Fig. 4. An example DQN policy for selecting expansion strategies. Eight abstract states (2×4) are represented with hatched abstract states containing points of interest. Each band within an abstract state represents the probability of choosing a particular expansion strategy: *blue* for NAIVE, *orange* for GREEDY, and *green* for PROACTIVE with darker shading corresponding to higher probability. This policy in particular highlights how the DQN approach exploits different structures within an MDP to dynamically optimize the expansion strategy.

by longitude $((x + \Delta_X) \mod |X|) \in X$ 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 $w \in W$. The formal definitions of the ground, abstract, and partially abstract MDPs are in recent work [18].

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: Summarized in Fig. 3 and exemplified in Fig. 4, our results show that combining deep RL with our metareasoning formalism is an effective approach to optimizing abstract state expansion within partially abstract MDPs. Specifically, our approach learns to select expansion strategies that *optimize* the trade-off between computation time and quality. Moreover, together, our contributions reduce the problem of abstract state expansion to that of finding the correct rates α and β to parameterize the time-dependent utility. This process is straightforward in many robotics domains where the expected reward and cost of computation are measured in easily comparable units.

VII. CONCLUSION

We formulate the metareasoning problem of choosing abstract state expansion strategies online to create partially abstract MDPs. We solve this problem using deep RL and show that the learned state expansion strategies are more performant than several heuristic baselines. Future work will explore generalizability in MDPs with different 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] R. Bellman. Dynamic programming. *Science*, 1966.
[5] R. Chitnis, T. Silver, B. Kim, L. P. Kaelbling, and T. Lozano-Perez. CAMPS: Learning context-specific abstractions for efficient planning in factored MDPs. *arXiv preprint 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] D. Han, M. Wooldridge, and S. Tschiatschek. MDP abstraction with successor features. *arXiv preprint arXiv:2110.09196*, 2021.
[9] E. A. Hansen and S. Zilberstein. Monitoring and control of anytime algorithms. *AIJ*, 126(1-2):139–157, 2001.
[10] A. Hertle, C. Dornhege, T. Keller, R. Mattmüller, et al. An experimental comparison of classical, fond and probabilistic planning. In *KI*, 2014.
[11] E. Horvitz and G. Rutledge. Time-dependent utility and action under uncertainty. In *7th UAI*, 1991.
[12] J. Hostetler, A. Fern, and T. Dietterich. Sample-based tree search with fixed and adaptive state abstractions. *JAIR*, 60, 2017.
[13] N. Jiang, A. Kulesza, and S. Singh. Abstraction selection in model-based reinforcement learning. In *ICML*, 2015.
[14] C. A. Knoblock. Generating abstractions for planning. *AIJ*, 68(2), 1994.
[15] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto. Robot learning from demonstration by constructing skill trees. *IJR*, 31(3), 2012.
[16] L. Li, T. J. Walsh, and M. L. Littman. Towards a unified theory of state abstraction for MDPs. In *ISAIM*, 2006.
[17] A. Ma, M. Ouimet, and J. Cortés. Dynamic domain reduction for multi-agent planning. In *MRS*, 2017.
[18] 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.
[19] A. Y. Ng, D. Harada, and S. J. Russell. Policy invariance under reward transformations. In *ICML*, 1999.
[20] L. Pineda and S. Zilberstein. Soft labeling in stochastic shortest path problems. In *18th AAMAS*, 2019.
[21] L. E. Pineda, K. Wray, and S. Zilberstein. Fast SSP solvers using short-sighted labeling. In *31st AAAI*, 2017.
[22] K. Steinkraus and L. P. Kaelbling. Combining dynamic abstractions in large MDPs. Technical report, MIT, 2004.
[23] J. Svegliato, P. Sharma, and S. Zilberstein. A model-free approach to meta-level control of anytime algorithms. In *ICRA*, 2020.
[24] J. Svegliato, K. 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*, 2016.
[26] M. Tomar, A. Zhang, R. Calandra, M. E. Taylor, and J. Pineau. Model-invariant state abstractions for model-based rl. *arXiv preprint arXiv:2102.09850*, 2021.
[27] A. Unruh and P. S. Rosenbloom. Abstraction in problem solving and learning. In *11th IJCAI*, 1989.
[28] S. Zilberstein. Resource-bounded sensing and planning in autonomous systems. *Autonomous Robots*, 3:31–48, 1996.