Q-Tree Search: An Information-Theoretic Approach Towards Hierarchical Abstractions for Agents with Computational Limitations

Daniel T. Larsson, *Student Member, IEEE,* Dipankar Maity, *Member, IEEE,* Panagiotis Tsiotras, *Fellow, IEEE*

***Abstract*—We develop a framework to obtain graph abstractions for decision-making where the abstractions emerge as a function of the agent's available resources. We discuss the connection of the proposed approach with information-theoretic signal compression and formulate a novel optimization problem to obtain tree-based abstractions that are a function of the agent's computational resources. The structural properties of the new problem are discussed in detail and two algorithmic approaches are proposed. We discuss the quality of, and prove relationships between, the solutions obtained by the two proposed algorithms. The framework is applied to a variety of environments to obtain hierarchical abstractions.**

***Index Terms*—Hierarchical abstractions, state aggregation, graph trees, information theory, planning.**

I. INTRODUCTION

Information theory provides a principled framework for obtaining optimal compressed representations of a signal [1]. The ability to form such compressed representations, also known as abstractions, has widespread uses in many fields, ranging from signal processing and data transmission, to robotic motion planning in complex environments, and many others [1]–[18]. Particularly for autonomous systems, simplified representations of the environment which the agent operates in are preferred, as they decrease the on-board memory requirements and reduce the computational time required to find feasible or optimal solutions for planning [2], [5]–[13], [19].

Within the realm of robotics and autonomous systems, a number of studies have leveraged the power of abstractions for both exploration and path-planning purposes. Examples of such prior works include [9]–[12] in which wavelets are utilized to generate multi-resolution representations of two-dimensional environments. These compressed representations encode a simplified graph of the environment, speeding up the execution time of path-planning algorithms such as A* [5]. As the agent traverses the environment, the problem is sequentially re-solved in order to obtain a trade-off in the overall optimality of the resulting path, planning frequency, and obstacle avoidance. Related works include [5] and [6], where the authors employed a tree-based framework in order to execute path-planning tasks in two- and three-dimensional environments. In these studies, the planning problem involved the generation of a multi-resolution representation of the operating space of the robotic agent in the form of a variable-depth probabilistic quadtree or octree, based on user-provided parameters and a given initial representation of the environment. In addition, the use of probabilistic quadtrees and octrees allows for the incorporation of sensor uncertainty when creating maps, since the environment is stored in the form of an occupancy grid [20], [21]. Other works have studied the generation of quadtrees in real time, such as [13], or the creation of multi-resolution trees from a given map and pruning rules [8].

It should be noted that the use of abstractions is not unique to the robotics community. For example, researchers in reinforcement learning have previously utilized state aggregations in order to alleviate the curse of dimensionality when solving for optimal policies in Markov decision processes [4], [22]. However, there is no unifying method for how these abstractions are generated, as existing methods rely heavily on user-provided rules. Other fields, such as the formal methods community, have studied bisimulation, which considers model reduction for dynamical systems that preserve certain system characteristics [23], [24]. Interestingly, these methods provide a convergent paradigm between the controls community and researchers in computer science. We note that related work also exists in the field of statistics and estimation, where Chow-Liu trees are utilized in order to approximate probability distributions using tree structures and concepts from information theory [25].

The drawback of previous works is that they do not directly address the generation of abstractions and instead rely on them to be either provided a priori or created in a manner that is known beforehand. Critically, existing works do not take into consideration the computational limitations of the agent. That is, existing works do not consider, in their formulation, that agents with limited on-board resources may not employ the same representation, or depiction, of the environment as agents that are not resource limited. The idea that all agents do not have equal capabilities has been recently discussed in the literature pertaining to the field of bounded rationality, where the capabilities of an agent are represented by its

Support for this work has been provided by ONR awards N00014-18-1-2375 and N00014-18-1-2828 and by ARL under DCIST CRA W911NF-17-2-018.

D. Larsson is a PhD student with the Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA. Email: daniel.larsson@gatech.edu

D. Maity is a Postdoctoral fellow with the Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA. Email: dmaity@gatech.edu

P. Tsiotras is the Andrew and Lewis Chair Professor with the Guggenheim School of Aerospace Engineering and the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA. Email: tsiotras@gatech.edu

information-processing abilities [26]–[29]. Consequently, a resource-limited agent is not able to process all data collected by observing its surroundings, leading to the need for simplification of the space in which it operates. In this way, the agent must balance its resource limitations, being frugal with regard to the amount of information it processes while simultaneously achieving the task it is entrusted to complete. This leads to an intrinsic balance between optimality and decision-making complexity.

A number of existing works have modeled single-stage and sequential bounded-rational decision making in stochastic domains by employing ideas from utility and information theory to construct constrained optimization problems [18], [26]–[28]. The solution to these problems is a set of self-consistent equations that are numerically solved by alternating iterations, analogous to the Blahut-Arimoto algorithm in rate-distortion theory [18], [26], [27], [30]. Interestingly, these frameworks allow for the emergence of bounded-rational policies for a range of agents with varying capabilities [18], [26]–[28].

In this paper, we address the issue of abstraction generation for a given environment and formulate a novel optimization problem that leverages concepts from information theory to obtain representations of an environment that are a function of the agent's available resources. Specifically, we consider the case where the environment is represented as a multi-resolution quadtree, and begin by discussing connections between environment abstractions in the form of quadtrees and general signal compression, the latter of which has been extensively studied by information theorists. We then formulate an optimization problem over the space of trees that utilizes concepts from the information bottleneck method [30] and subsequently propose two algorithms to solve this problem. Theoretical guarantees of our proposed algorithmic approaches are presented and discussed. We show a number of non-trivial examples that demonstrate the utility of the approach and discuss the interpretation of the theory as applied to bounded-rational robotic agents.

The remainder of the paper is organized as follows. In Section II we introduce and review the fundamental concepts needed in this work as well as elucidate the connections between quadtrees and optimal signal compression. Then, in Section III, we formulate our problem and show how principles from information theory can be incorporated into a new optimization problem over the space of trees. In Section IV, we propose two algorithms that can be used to solve the optimization problem and present the theoretical contributions of the paper. Section V presents results of the proposed methodology which is then followed by concluding remarks in Section VI.

II. PRELIMINARIES

Throughout this paper, $\mathbb{R}$ and $\mathbb{R}_{++}$ denote the set of real and strictly positive real numbers, respectively, and $\mathbb{R}^n$ denotes the space of all real-valued vectors of dimension n.

A. Quadtree Decompositions

We consider the emergence of abstractions in the form of multi-resolution quadtree representations. Quadtrees are a common tool utilized in the robotics community to reduce the complexity of environments in order to speed path-planning or ease internal storage requirements [2], [5], [6], [13], [16]. It should be noted that, while we will restrict the discussion to quadtrees throughout the paper, the proposed approach is applicable to any tree structure. We assume that the environment $\mathcal{W} \subset \mathbb{R}^2$ (generalizable to $\mathbb{R}^d$) is given by a two-dimensional grid world where each grid element is a unit square (hypercube). In addition, it is assumed that there exists an integer $\ell > 0$ such that $\mathcal{W}$ is contained within a square (hypercube) of side length 2^ℓ. A tree representation $\mathcal{T} = (\mathcal{N}, \mathcal{E}) = (\mathcal{N}(\mathcal{T}), \mathcal{E}(\mathcal{T}))$ of $\mathcal{W}$ consists of a set of nodes $\mathcal{N}$ and edges $\mathcal{E}$ describing the interconnections between the nodes in the tree [6]. We denote the set of all possible quadtree representations of maximum depth ℓ of $\mathcal{W}$ by $\mathcal{T}^\mathcal{Q}$ and let $\mathcal{T}_\mathcal{W} \in \mathcal{T}^\mathcal{Q}$ denote the finest quadtree representation of $\mathcal{W}$; an example is shown in Figure 1. It should be noted that $\mathcal{T}_\mathcal{W}$ encodes a specific structure for $\mathcal{W}$, which we make precise in the following definition.

Definition 1. Let $t \in \mathcal{N}(\mathcal{T}_\mathcal{W})$ be any node at depth $k \in \{0, \dots, \ell\}$. Then $t' \in \mathcal{N}(\mathcal{T}_\mathcal{W})$ is a *child* of t if the following hold:

1) Node t' is at depth $k+1$ in $\mathcal{T}_\mathcal{W}$.
2) Nodes t and t' are incident to a common edge, i.e., $(t, t') \in \mathcal{E}(\mathcal{T}_\mathcal{W})$.

Conversely, we say that t is the *parent* of t' if t' is a child of t. Furthermore, we let

$$\mathcal{N}_k(\mathcal{T}_q) = \{t \in \mathcal{N}(\mathcal{T}_q) : t \text{ is at depth } k \text{ in } \mathcal{T}_\mathcal{W}\},$$

to be the set of all nodes of the tree $\mathcal{T}_q \in \mathcal{T}^\mathcal{Q}$ at depth k.

We will frequently seek to relate nodes in the tree $\mathcal{T}_q$ to those in the tree $\mathcal{T}_\mathcal{W}$, which leads us to the following definition.

Definition 2. Let $t \in \mathcal{N}(\mathcal{T}_q)$ be any node in the tree $\mathcal{T}_q \in \mathcal{T}^\mathcal{Q}$. Then the following hold:

1) The node t has children
$$\mathcal{C}(t) = \{t' \in \mathcal{N}(\mathcal{T}_\mathcal{W}) : t' \text{ is a child of } t\}.$$
2) The node t has parent
$$\mathcal{P}(t) = \left\{\hat{t} \in \mathcal{N}(\mathcal{T}_\mathcal{W}) : t \in \mathcal{C}(\hat{t})\right\}.$$
3) The node t is the root of the tree $\mathcal{T}_q$, denoted by $\mathsf{Root}(\mathcal{T}_q)$, if $\mathcal{P}(t) = \emptyset$.
4) The node t is a leaf of $\mathcal{T}_q$ if $\mathcal{C}(t) \cap \mathcal{N}(\mathcal{T}_q) = \emptyset$. Furthermore, the set of leaf nodes of $\mathcal{T}_q$ is given by
$$\mathcal{N}_{\text{leaf}}(\mathcal{T}_q) = \{t' \in \mathcal{N}(\mathcal{T}_q) : \mathcal{C}(t') \cap \mathcal{N}(\mathcal{T}_q) = \emptyset\}.$$
5) If $t \notin \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$ then $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_q) = \mathcal{N}(\mathcal{T}_q) \setminus \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$, where $\mathcal{N}_{\text{int}}(\mathcal{T}_q)$ is the set of interior nodes of $\mathcal{T}_q$.

Note that the space $\mathcal{T}^\mathcal{Q}$ encodes a specific structure on the abstractions of the environment, as shown in Figure 1. More precisely, each $\mathcal{T}_q \in \mathcal{T}^\mathcal{Q}$, $\mathcal{T}_q \neq \mathcal{T}_\mathcal{W}$, specifies a precise relation between the leaf nodes of $\mathcal{T}_\mathcal{W}$ and the leaf nodes of $\mathcal{T}_q$, an example of which is shown in Figures 1 and 2. That is, the tree $\mathcal{T}_q \in \mathcal{T}^\mathcal{Q}$ specifies an abstraction for which the leaf nodes of $\mathcal{T}_\mathcal{W}$ are mapped to leaf nodes of $\mathcal{T}_q$ in such a way that $\mathcal{T}_q$

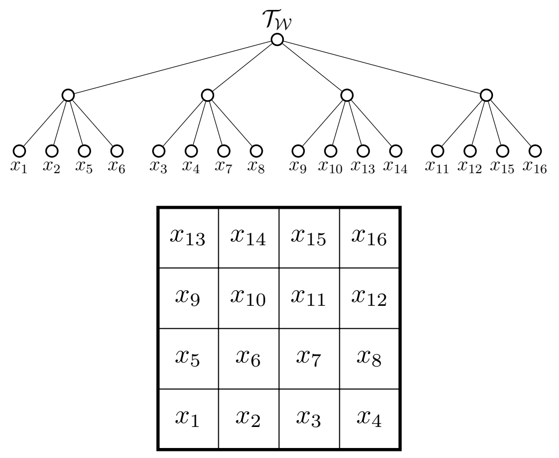

$\mathcal{T}_{\mathcal{W}}$
x_1 x_2 x_5 x_6 x_3 x_4 x_7 x_8 x_9 x_{10} x_{13} x_{14} x_{11} x_{12} x_{15} x_{16}
x_{13} x_{14} x_{15} x_{16}
x_9 x_{10} x_{11} x_{12}
x_5 x_6 x_7 x_8
x_1 x_2 x_3 x_4

Fig. 1. Representation of the tree $\mathcal{T}_{\mathcal{W}}$ and corresponding grid for a 4×4 environment.

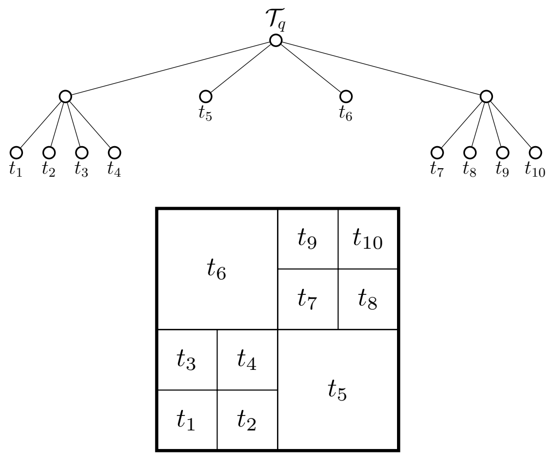


Fig. 2. Representation of some $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ and corresponding grid for a 4×4 environment.

is a pruned quadtree representation of $\mathcal{W}$. In other words, we can consider each $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ as a pruned version of $\mathcal{T}_{\mathcal{W}}$, where some nodes in the interior of $\mathcal{T}_{\mathcal{W}}$ are leaf nodes of $\mathcal{T}_q$. In this way, each $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ can be seen as encoding an abstraction, or compression, of $\mathcal{W}$ with the constraint that $\mathcal{T}_q$ be a valid quadtree depiction of $\mathcal{W}$. Varying the abstraction granularity of $\mathcal{W}$ can be toggled by selecting various trees $\mathcal{T}_q$ in the space $\mathcal{T}^{\mathcal{Q}}$. Our problem is then one of selecting a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ as a function of the agent's computational capabilities. The observation that each $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ encodes a compression of $\mathcal{W}$ connects our approach to information-theoretic frameworks that consider optimal encoder design. The optimization problem to obtain optimal encoders has been extensively studied by information theorists in the more general setting of signal compression, where no specific structure on the abstraction is enforced (i.e., the resulting encoding need not correspond to any tree representation). As such, the added constraint that our abstraction be a valid quadtree representation of $\mathcal{W}$ creates additional challenges, since direct application of information-theoretic methods is not possible. Thus, to elucidate the technical aspects of our approach, we first present a brief review of the necessary information-theoretical concepts which we will utilize later on.

B. Information-Theoretical Signal Compression

The task of obtaining compressed representations of signals is addressed within the realm of information theory [1], [30]–[34]. Let $(\Omega, \mathcal{F}, \mathbb{P})$ to be a probability space with finite sample space Ω, σ-algebra $\mathcal{F}$ and probability measure $\mathbb{P} : \mathcal{F} \to [0,1]$. Let $X : \Omega \to \mathbb{R}$ denote the random variable corresponding to the original, uncompressed, signal, where X takes values in the set $\Omega_X = \{x \in \mathbb{R} : X(\omega) = x,\ \omega \in \Omega\}$ and, for any $x \in \mathbb{R}$, $p(x) = \mathbb{P}(\{\omega \in \Omega : X(\omega) = x\})$. Furthermore, let the random variable $T : \Omega \to \mathbb{R}$ denote the compressed representation of X, where T takes values in the set $\Omega_T = \{t \in \mathbb{R} : T(\omega) = t,\ \omega \in \Omega\}$. The level of compression between two random variables X and T is measured by their mutual information [1], [30], given by

$$I(T;X) \triangleq \sum_{t,x} p(t,x) \log \frac{p(t,x)}{p(t)p(x)}. \tag{1}$$

The goal is then to find a stochastic mapping (encoder), denoted by $p(t|x)$, which maps outcomes in the uncompressed space $x \in \Omega_X$, to outcomes in the compressed representation $t \in \Omega_T$ so as to minimize $I(T;X)$ (maximize compression) [30]. However, in order to obtain non-trivial solutions, a metric quantifying the quality of the resulting compression must be introduced, since maximal compression $(I(T;X) = 0)$ is always achievable. The information bottleneck (IB) method [30] defines the quality of the compression by utilizing mutual information.

More specifically, the IB method introduces an additional random variable, $Y : \Omega \to \mathbb{R}$, taking values in the set $\Omega_Y = \{y \in \mathbb{R} : Y(\omega) = y,\ \omega \in \Omega\}$. The variable Y represents information we are interested in preserving when forming the compressed representation T of X [30], [31]. The method imposes the Markov chain condition $T \leftrightarrow X \leftrightarrow Y$ which arises as a consequence of the problem formulation. To see this, note that $p(y|t,x) = p(y|x)$ since it is not possible for t to convey any additional information regarding y than what it is already in x, and thus $T \to X \to Y$. Furthermore, if $p(y|t,x) = p(y|x)$ then $p(t|y,x) = p(t|x)$ which gives $Y \to X \to T$. Therefore, $T \to X \to Y$ implies $Y \to X \to T$, which is written as $T \leftrightarrow X \leftrightarrow Y$ [1], [30].

The IB problem is formulated as

$$\min_{p(t|x)} \ I(T;X), \tag{2}$$

subject to

$$I(T;Y) \geq \hat{D}, \tag{3}$$

where the minimization is over all normalized distributions $p(t|x)$, assuming that the joint distribution $p(x,y)$ is provided and $\hat{D} \geq 0$ [30]. Through the introduction of a Lagrange multiplier, $\beta \geq 0$, we have that (2) subject to (3) has Lagrangian

$$\mathcal{K}_Y(p(t|x);\beta) \triangleq I(T;X) - \beta I(T;Y). \tag{4}$$

For given $\beta \geq 0$, the optimization problem

$$\min_{p(t|x)} \ \mathcal{K}_Y(p(t|x);\beta), \tag{5}$$

can be solved analytically, giving rise to a set of self-consistent equations [30].

The self-consistent equations obtained as a solution to (5) can be solved numerically by an algorithm that likens the Blahut-Arimoto algorithm from rate-distortion theory, albeit with no guarantee of convergence to a globally optimal solution [30]. The parameter β serves the role of adjusting the amount of relevant information regarding Y that is retained in the abstract representation T. As a result, when $\beta \to \infty$ the optimization process is concerned with the maximal preservation of information, while $\beta \to 0$ promotes maximal compression, with no regard to the information carried regarding Y. Intermediate values of β lead to a spectrum of solutions between these two extremes [30]. The mapping $p^*(t|x)$ obtained as a solution to the IB problem is generally stochastic, resulting in a deterministic mapping only when $\beta \to \infty$ [30], [33].

C. Agglomerative Information Bottleneck

The agglomerative IB (AIB) method is another framework to form compressed representations of X, which is useful when deterministic clusters that retain predictive information regarding the relevant variable Y are desired. The method uses the IB approach to solve for deterministic, or hard, encoders (i.e., $p(t|x) \in \{0,1\}$ for all t, x). Concepts from AIB will prove useful in our formulation, since each tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ encodes a hard (deterministic) abstraction of $\mathcal{W}$, where each leaf node of $\mathcal{T}_{\mathcal{W}}$ is aggregated to a specific leaf node of $\mathcal{T}_q$. That is, by viewing the uncompressed space (Ω_X) as the nodes in $\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$ and the abstracted (compressed) space (Ω_T) as the nodes in $\mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$, the abstraction operation can be specified in terms of an encoder $p(t|x)$ where $p(t|x) \in \{0,1\}$ for all t and x, where $p(t|x) = 1$ if $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$ is aggregated to $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$, and zero otherwise (see Figures 1 and 2). To better understand these connections, we briefly review the AIB before presenting the formulation of our problem.

The solution provided by AIB is an encoder $p(t|x)$ for which $p(t|x) \in \{0,1\}$ for all t, x and $\beta > 0$. AIB considers the optimization problem

$$\max_{p(t|x)} \ \mathcal{L}_Y(p(t|x);\beta), \tag{6}$$

where the Lagrangian is defined as

$$\mathcal{L}_Y(p(t|x);\beta) \triangleq I(T;Y) - \frac{1}{\beta} I(T;X), \tag{7}$$

and the maximization is performed over deterministic distributions $p(t|x)$ for given $\beta > 0$ and $p(x,y)$ [31], [32].

AIB works from bottom-up, starting with $T = X$ and with each consecutive iteration reduces the cardinality of T until $|\Omega_T| = 1$ [31]. Specifically, let T_m represent the abstracted space with m elements ($|\Omega_{T_m}| = m$) and let T_i represent the compressed space with $|\Omega_{T_i}| = i < m$ elements, where $i = m - 1$ and the number of merged elements is $n = 2$. We then merge elements $\{t_1', \ldots, t_n'\} \subseteq \Omega_{T_m}$ to a single element $t \in \Omega_{T_i}$ to obtain T_i. The set $\{t_1', \ldots, t_n'\} \subseteq \Omega_{T_m}$ selected to merge is determined by considering the difference in the IB Lagrangian induced by the merge operation, as follows. Let $p^- : \Omega_{T_m} \times \Omega_X \to \{0,1\}$ be the mapping before the merge and $p^+ : \Omega_{T_i} \times \Omega_X \to \{0,1\}$ be the resulting mapping after elements $\{t_1', \ldots, t_n'\} \subseteq \Omega_{T_m}$ are grouped to $t \in \Omega_{T_i}$. Note that, as AIB considers a sequence of merges, the mapping $p^-(t|x)$ represents an abstraction of higher cardinality as compared to $p^+(t|x)$. The merger cost is then given by $\Delta\mathcal{L}_Y : 2^{\Omega_{T_m}} \times \mathbb{R}_{++} \to \mathbb{R}$, defined as [32]

$$\Delta\mathcal{L}_Y(\{t_1', \ldots, t_n'\};\beta) \triangleq \mathcal{L}_Y(p^-(t|x);\beta) - \mathcal{L}_Y(p^+(t|x);\beta). \tag{8}$$

The above relation can be decomposed into a change in mutual information by utilizing (7) as

$$\Delta\mathcal{L}_Y(\{t_1', \ldots, t_n'\};\beta) = \\ [I(T_m;Y) - I(T_i;Y)] - \frac{1}{\beta}[I(T_m;X) - I(T_i;X)]. \tag{9}$$

This can be further simplified by noting that

$$I(T;X) = H(T) - H(T|X) = H(T), \tag{10}$$

where, $H(T)$ is the Shannon entropy of the random variable T, given by

$$H(T) \triangleq -\sum_t p(t) \log p(t), \tag{11}$$

and the conditional entropy $H(T|X) = 0$ since $p(t|x) \in \{0,1\}$ (i.e. there is no uncertainty in the random variable T when given an outcome of X). Equation (9) then becomes

$$\Delta\mathcal{L}_Y(\{t_1', \ldots, t_n'\};\beta) = \\ [I(T_m;Y) - I(T_i;Y)] - \frac{1}{\beta}[H(T_m) - H(T_i)]. \tag{12}$$

It was shown in [31], [32] that (12) can be written as

$$\Delta\mathcal{L}_Y(\{t_1', \ldots, t_n'\};\beta) = \\ p(t)\left[\mathrm{JS}_\Pi(p(y|t_1'), \ldots, p(y|t_n')) - \frac{1}{\beta}H(\Pi)\right], \tag{13}$$

where $\Pi \in \mathbb{R}^n$ is given by

$$\Pi = [\Pi_1, \ldots, \Pi_n]^\mathsf{T} \triangleq \left[\frac{p(t_1')}{p(t)}, \ldots, \frac{p(t_n')}{p(t)}\right]^\mathsf{T}, \tag{14}$$

and $\mathrm{JS}_\Pi(p_1, \ldots, p_n)$ is the Jensen-Shannon (JS) divergence [35] between the distributions $p_1, \ldots, p_n$, with weights Π defined as

$$\mathrm{JS}_\Pi(p_1, \ldots, p_n) \triangleq \sum_{s=1}^{n} \Pi_s \mathrm{D}_{\mathrm{KL}}(p_s, \bar{p}), \tag{15}$$

where, for each outcome $y \in \Omega_Y$,

$$\bar{p}(y) = \sum_{s=1}^{n} \Pi_s p_s(y). \tag{16}$$

In (15) $\mathrm{D}_{\mathrm{KL}}(\mu,\nu)$ denotes the Kullback-Leibler (KL) divergence between discrete probability distributions μ and ν given by

$$\mathrm{D}_{\mathrm{KL}}(\mu,\nu) \triangleq \sum_y \mu(y) \log \frac{\mu(y)}{\nu(y)}. \tag{17}$$

It can also be shown that

$$p(t) = \sum_{s=1}^{n} p(t'_s), \tag{18}$$

$$p(y|t) = \sum_{s=1}^{n} \Pi_s p(y|t'_s), \tag{19}$$

which can be verified by realizing that $p(t|x) \in \{0,1\}$ for all $x \in \Omega_X$ and $t \in \Omega_T$ as well as employing the condition $T \leftrightarrow X \leftrightarrow Y$ [31], [32]. Note that the merger cost (8) can be written in terms of only the distributions $p(y|t'_1), \ldots, p(y|t'_n)$ and the weight vector Π. This reduces the overall complexity of computing $\Delta\mathcal{L}_Y(\{t'_1, \ldots, t'_n\}\,; \beta)$, as opposed to utilizing equation (9), which contains sums over the sample spaces of Y, T and X [31], [32].

III. PROBLEM FORMULATION

The IB methods presented in the previous section do not impose any constraints on the mapping $p(t|x)$. That is, by solving the IB problem, one obtains a mapping $p^*(t|x)$ that is generally stochastic, and thus it is not guaranteed that it encodes a (quad)tree representation for any value of $\beta > 0$. The difficulty lies in the specific structure imposed on the abstraction by the space $\mathcal{T}^{\mathcal{Q}}$, as even AIB or deterministic IB cannot guarantee that the resulting $p^*(t|x)$ encodes a tree belonging to $\mathcal{T}^{\mathcal{Q}}$, although they do provide deterministic encoders [31]–[33]. Recall that, since each $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ represents an abstraction of $\mathcal{T}_{\mathcal{W}}$, $\mathcal{T}_q$ can be equivalently represented as $p^q(t|x)$, where $p^q(t|x) = 1$ if $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$ is abstracted to $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$ and zero otherwise. We can then define the IB Lagrangian in the space of quadtrees as the mapping $L_Y : \mathcal{T}^{\mathcal{Q}} \times \mathbb{R}_{++} \to \mathbb{R}$ given by

$$L_Y(\mathcal{T}_q; \beta) \triangleq \mathcal{L}_Y(p^q(t|x); \beta), \tag{20}$$

where $\mathcal{L}_Y(p(t|x); \beta)$ is defined in (7). Then, for a given $\beta > 0$, we can search the space of trees for the one that maximizes (20). This optimization problem is formally given by

$$\mathcal{T}_{q^*} = \operatorname*{argmax}_{\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}} L_Y(\mathcal{T}_q; \beta), \tag{21}$$

with the resulting world representation encoded by the mapping $p^{q^*}(t|x)$. That is, the leafs of $\mathcal{T}_{q^*}$ determine the optimal multi-resolution representation of $\mathcal{W}$ for the given β.

By posing the optimization problem as in (21), we have implicitly incorporated the constraints on the mapping $p(t|x)$ so that the resulting representation is a quadtree depiction of the world. While the optimization problem given by (21) allows one to form an analogous problem to that in (6) over the space of trees, the drawback of this method is the need to exhaustively enumerate all feasible quadtrees which can represent the space. In other words, (21) requires that $p^q(t|x)$ be provided for each $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$. Because of this, the problem becomes intractable for large grid sizes and thus requires reformulation to handle larger world maps.

Interestingly, we note that it is possible to arrive at a quadtree $\mathcal{T}_{q^m} \in \mathcal{T}^{\mathcal{Q}}$ by starting from $\mathcal{T}_{q^0} \in \mathcal{T}^{\mathcal{Q}}$ and considering a sequence of intermediate trees, as illustrated in Figure 3. The resulting sequence of trees can be viewed as defining a path between $\mathcal{T}_{q^0}$ and $\mathcal{T}_{q^m}$, in which each vertex of the path corresponds to a distinct intermediate tree in the sequence. Note that it is not always possible to reach any tree $\mathcal{T}_{q^m}$ starting from any tree $\mathcal{T}_{q^0}$ by considering such a sequence of trees. In order to address this, we require the following definitions.

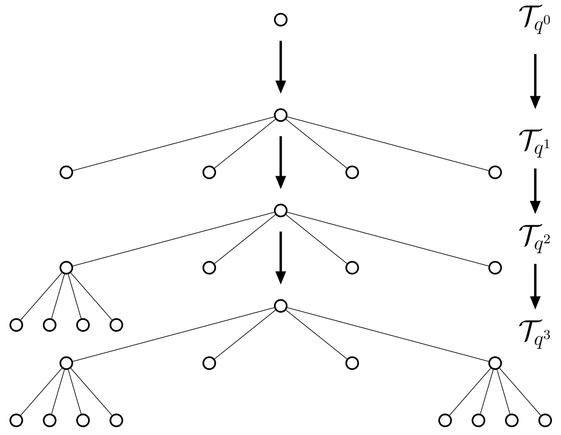


Fig. 3. Sequence of trees from $\mathcal{T}_{q^0} = \mathsf{Root}(\mathcal{T}_{\mathcal{W}}) \in \mathcal{T}^{\mathcal{Q}}$ to $\mathcal{T}_{q^3} \in \mathcal{T}^{\mathcal{Q}}$ ($m = 3$) by performing a sequence of nodal expansions. Note that $\mathcal{T}_{q^0} = \mathsf{Root}\,(\mathcal{T}_{\mathcal{W}})$ is the root node of $\mathcal{T}_{\mathcal{W}}$.

Definition 3 ([36])**.** A tree $\mathcal{G} = (\mathcal{N}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ is a *subtree* of the tree $\mathcal{J} = (\mathcal{N}(\mathcal{J}), \mathcal{E}(\mathcal{J}))$, denoted $\mathcal{G} \subseteq \mathcal{J}$, if $\mathcal{N}(\mathcal{G}) \subseteq \mathcal{N}(\mathcal{J})$ and $\mathcal{E}(\mathcal{G}) \subseteq \mathcal{E}(\mathcal{J})$.

Definition 4. The tree $\mathcal{T}_{q'} \in \mathcal{T}^{\mathcal{Q}}$ is a *neighbor* of the tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ if $\mathcal{N}(\mathcal{T}_{q'}) \setminus \mathcal{N}(\mathcal{T}_q) = \{t'_1, \ldots, t'_n\} \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{q'})$ such that $\mathcal{P}(t'_1) = \cdots = \mathcal{P}(t'_n) = t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$.

With these definitions, we see that if $\mathcal{T}_{q'} \in \mathcal{T}^{\mathcal{Q}}$ is a neighbor of $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$, then we can obtain $\mathcal{T}_{q'}$ by adding the nodes $\{t'_1, \ldots, t'_n\}$ to $\mathcal{T}_q$, where the set $\{t'_1, \ldots, t'_n\}$ consists of the children of a leaf node t of $\mathcal{T}_q$. We call this process of adding $\mathcal{C}(t) = \{t'_1, \ldots, t'_n\}$ to $\mathcal{N}(\mathcal{T}_q)$ a *nodal expansion* of $\mathcal{T}_q$ at t. We observe that by only performing a sequence of nodal expansions, a path exists between the trees $\mathcal{T}_{q^0} \in \mathcal{T}^{\mathcal{Q}}$ and $\mathcal{T}_{q^m} \in \mathcal{T}^{\mathcal{Q}}$ if $\mathcal{T}_{q^0}$ is a subtree of $\mathcal{T}_{q^m}$ $\left(\mathcal{T}_{q^0} \subseteq \mathcal{T}_{q^m}\right)$. An illustration of nodal expansion is provided in Figure 3, where we also note that each tree $\mathcal{T}_{q^{i+1}}$ in the sequence is a neighbor to tree $\mathcal{T}_{q^i}$ with $i \in \{0,1,2\}$.

Furthermore, we may view the set of all possible quadtrees as a connected graph, where neighbors are defined according to Definition 4. An illustration of neighboring trees is provided in Figure 4. Thus, if it is possible to obtain a sequential characterization of (20), we can formulate an optimization problem requiring the generation of candidate solutions only along the path leading from $\mathcal{T}_{q^0}$ to $\mathcal{T}_{q^m}$. To this end, if we take $\mathcal{T}_{q^0} \subseteq \mathcal{T}_{q^m}$, where $\mathcal{T}_{q^0}, \mathcal{T}_{q^m} \in \mathcal{T}^{\mathcal{Q}}$, and assume that $\mathcal{T}_{q^m}$ is obtained by m expansions of $\mathcal{T}_{q^0}$, then

$$L_Y(\mathcal{T}_{q^m}; \beta) = L_Y(\mathcal{T}_{q^0}; \beta) + \sum_{i=0}^{m-1} \Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta), \tag{22}$$

where $\Delta L_Y(\cdot,\cdot;\beta)$ is defined as

$$\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) \triangleq L_Y(\mathcal{T}_{q^{i+1}}; \beta) - L_Y(\mathcal{T}_{q^i}; \beta), \tag{23}$$

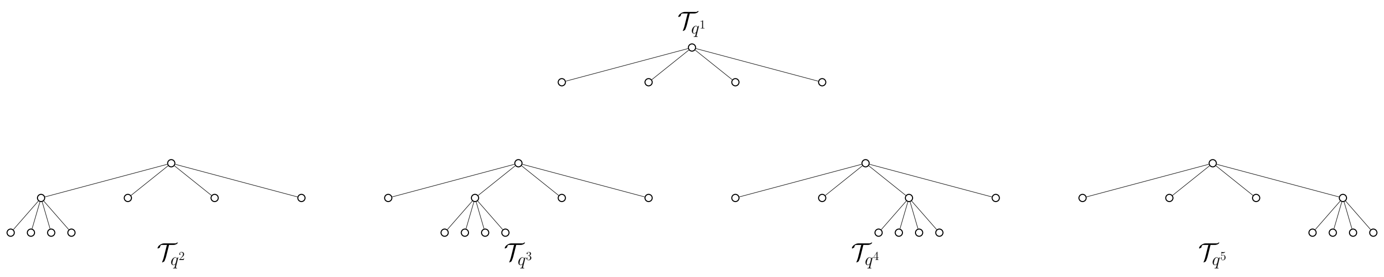


Fig. 4. Tree neighbors of $\mathcal{T}_{q^1} = \left\{\mathcal{T}_{q^2}, \mathcal{T}_{q^3}, \mathcal{T}_{q^4}, \mathcal{T}_{q^5}\right\}$.

and $\mathcal{T}_{q^{i+1}} \in \mathcal{T}^{\mathcal{Q}}$ is a neighbor of $\mathcal{T}_{q^i} \in \mathcal{T}^{\mathcal{Q}}$ with higher leaf node cardinality for $i \in \{0, \ldots, m-1\}$. Consequently, (22) gives a sequential representation of (20). Additionally, the nodal expansion operation to move from tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ to the neighbor $\mathcal{T}_{q'} \in \mathcal{T}^{\mathcal{Q}}$ has an analogous interpretation to (8) of the AIB method discussed in Section II. Consequently,

$$\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) = \Delta \mathcal{L}_Y(\{t'_1, \ldots, t'_n\}; \beta), \tag{24}$$

and thus

$$\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) = \\ p(t)\left[\mathrm{JS}_\Pi(p(y|t'_1), \ldots, p(y|t'_n)) - \frac{1}{\beta} H(\Pi)\right]. \tag{25}$$

Importantly, note that the structure of $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta)$ in (25) only depends on which leaf nodes of $\mathcal{T}_{q^i}$ are expanded, as depicted in Figure 5. This implies that $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta)$ is only a function of the nodes that are to be expanded, and not of the overall configuration of the tree, which greatly simplifies the calculation of $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta)$.

It follows that the optimization problem can be reformulated as

$$\max_m \max_{\{\mathcal{T}_{q^1}, \ldots, \mathcal{T}_{q^m}\}} L_Y(\mathcal{T}_{q^0}; \beta) + \sum_{i=0}^{m-1} \Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta). \tag{26}$$

In this formulation, the constraint encoding that the resulting representation is a quadtree is handled implicitly by $\mathcal{T}_{q^i} \in \mathcal{T}^{\mathcal{Q}}$. The additional maximization over m in (26) appears because the horizon of the problem is not known a priori and is, instead, a free parameter in the optimization problem.

Next, we propose two algorithms that can be used to solve the problem in (26). Note that, by taking $\mathcal{T}_{q^0} = \mathsf{Root}(\mathcal{T}_\mathcal{W}) \in \mathcal{T}^{\mathcal{Q}}$, we can guarantee that a path exists between $\mathcal{T}_{q^0}$ and any other $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$, since, in this case, $\mathcal{T}_{q^0} \subseteq \mathcal{T}_q$ for all $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$.

IV. ALGORITHMIC SOLUTIONS

In this section, we discuss two algorithmic approaches to solve the optimization problem (26). Specifically, we present two novel approaches: a Greedy search method, and an algorithm we call Q-tree search. Proofs of all lemmas, propositions and theorems in this section are provided in the Appendix.

A. A Greedy Approach

A greedy approach to solve (26) involves maximizing $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta)$ myopically at each step. That is, provided that $\mathcal{T}_{q^{i+1}} \in \mathcal{T}^{\mathcal{Q}}$ is a neighbor of $\mathcal{T}_{q^i} \in \mathcal{T}^{\mathcal{Q}}$, we consider the next tree $\mathcal{T}_{q^{i+1}}$ that maximizes the value of $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta)$, and we sequentially keep selecting trees ($\mathcal{T}_{q^{i+1}} \to \mathcal{T}_{q^{i+2}} \to \cdots$) until no further improvement is possible. In other words, the Greedy search algorithm continues along the current path in the space of trees until it finds a tree $\mathcal{T}_{q^i} \in \mathcal{T}^{\mathcal{Q}}$ that has no neighbor $\mathcal{T}_{q^{i+1}} \in \mathcal{T}^{\mathcal{Q}}$ for which $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) > 0$. The process is detailed in Algorithm 1. In this algorithm, the function `computeDeltaL` $\left(\mathcal{T}_{q^i}, \mathcal{T}_{q^k}, \beta\right)$ computes the value of $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^k}; \beta)$.

The Greedy search algorithm is simple to implement and requires little pre-processing. However, one can construct examples for a given $\beta > 0$ and $\mathcal{T}_{q^i} \in \mathcal{T}^{\mathcal{Q}}$ for which $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) < 0$ for all $\mathcal{T}_{q^{i+1}} \in \mathcal{T}^{\mathcal{Q}}$ that are neighbors of $\mathcal{T}_{q^i}$, and where there exists at least one neighbor $\mathcal{T}_{q^{i+2}} \in \mathcal{T}^{\mathcal{Q}}$ of $\mathcal{T}_{q^{i+1}}$ such that $\Delta L_Y(\mathcal{T}_{q^{i+1}}, \mathcal{T}_{q^{i+2}}; \beta) > 0$ and $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) + \Delta L_Y(\mathcal{T}_{q^{i+1}}, \mathcal{T}_{q^{i+2}}; \beta) > 0$. In this case, the algorithm will terminate at the condition $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) < 0$, without gaining access to $\Delta L_Y(\mathcal{T}_{q^{i+1}}, \mathcal{T}_{q^{i+2}}; \beta) > 0$. Since in this scenario $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) + \Delta L_Y(\mathcal{T}_{q^{i+1}}, \mathcal{T}_{q^{i+2}}; \beta) > 0$, further improvement of (26) is possible, but not achievable by the greedy approach. Therefore, while the Greedy search algorithm is simple to implement, it does not, in general, find globally optimal solutions. However, as $\beta \to \infty$, the Greedy search algorithm does find a global solution as $\lim_{\beta\to\infty} \Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) \geq 0$ for all $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$, as seen by the limit of (25) and non-negativity of the JS-divergence.

Algorithm 1 The Greedy Search Algorithm.

```
Input Data: $p(x, y)$, $\beta > 0$
Result: $\mathcal{T}_{q^*}$
$\mathcal{T}_{q^0} \leftarrow \mathsf{Root}(\mathcal{T}_\mathcal{W})$
while termination condition not satisfied do
    for each neighbor $\mathcal{T}_{q^k}$ of $\mathcal{T}_{q^i}$ do
        $v(k) \leftarrow$ computeDeltaL $\left(\mathcal{T}_{q^i}, \mathcal{T}_{q^k}, \beta\right)$
    end for
    if $\max v > 0$ then
        $z \leftarrow \operatorname{argmax} v$
        $\mathcal{T}_{q^{i+1}} \leftarrow$ neighbor $\mathcal{T}_{q^z}$ of $\mathcal{T}_{q^i}$
        $i \leftarrow i + 1$
    end if
end while
```

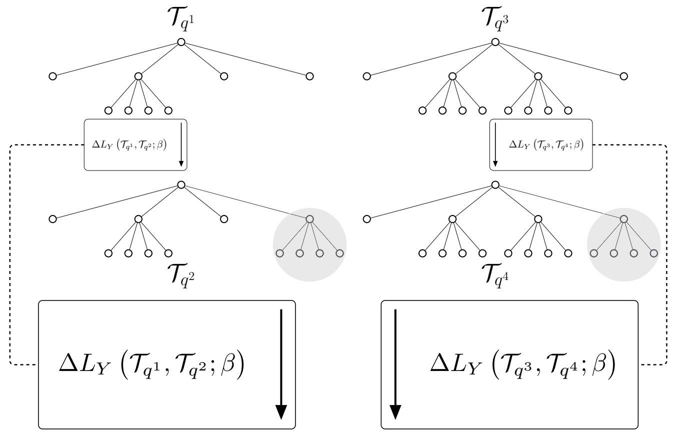

$\mathcal{T}_{q^1}$
$\mathcal{T}_{q^3}$
$\Delta L_Y(\mathcal{T}_{q^1}, \mathcal{T}_{q^2}; \beta)$
$\Delta L_Y(\mathcal{T}_{q^3}, \mathcal{T}_{q^4}; \beta)$
$\mathcal{T}_{q^2}$
$\mathcal{T}_{q^4}$
$\Delta L_Y(\mathcal{T}_{q^1}, \mathcal{T}_{q^2}; \beta)$
$\Delta L_Y(\mathcal{T}_{q^3}, \mathcal{T}_{q^4}; \beta)$

Fig. 5. Representation of the invariance of $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta)$. Changes to the previous tree are shown in grey shading. Note that, in both cases, moving from $\mathcal{T}_{q^1}$ to $\mathcal{T}_{q^2}$ or from $\mathcal{T}_{q^3}$ to $\mathcal{T}_{q^4}$ involves expanding the same node. Subsequently, $\Delta L_Y(\mathcal{T}_{q^1}, \mathcal{T}_{q^2}; \beta) = \Delta L_Y(\mathcal{T}_{q^3}, \mathcal{T}_{q^4}; \beta)$.

B. The Q-tree Search Algorithm

We now present an alternative approach, detailed in Algorithm 2, designed to overcome some of the shortfalls encountered with the Greedy search algorithm. The main drawback of utilizing the greedy approach in solving the optimization problem (26) is the short-sightedness of the algorithm and its inability to realize that poor expansions at the current step may lead to much higher-valued options in the future. This is analogous to problems in reinforcement learning and dynamic programming, where an action-value function (Q-function) is introduced to incorporate the notion of cost-to-go for selecting among feasible actions in a given state [3], [37]. The idea behind introducing such a function is to incorporate future costs, allowing agents to take actions that are not the most optimal with respect to the current one-step cost, but have lower total cost due to events that are possible in the future.

To this end, for any tree $\mathcal{T}_{q^i}$ and its neighbor $\mathcal{T}_{q^{i+1}} (\supseteq \mathcal{T}_{q^i})$ we define the function

$$Q_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) \triangleq \max\left\{ \Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) + \sum_{\tau=1}^{n} Q_Y(\mathcal{T}_{q^{i+1}}, \mathcal{T}_{q_\tau^{i+2}}; \beta),\ 0 \right\}, \tag{27}$$

where $\mathcal{T}_{q_\tau^{i+2}}$ is a neighbor of $\mathcal{T}_{q^{i+1}}$ for all $\tau = 1, \dots, n$ and

$$Q_Y(\mathcal{T}_{q'}, \mathcal{T}_{\mathcal{W}}; \beta) \triangleq \max\{\Delta L_Y(\mathcal{T}_{q'}, \mathcal{T}_{\mathcal{W}}; \beta),\ 0\}, \tag{28}$$

for all $\mathcal{T}_{q'} \in \mathcal{T}^{\mathcal{Q}}$ for which $\mathcal{T}_{\mathcal{W}} \in \mathcal{T}^{\mathcal{Q}}$ is a neighbor. If $\mathcal{T}_{q^{i+1}}$ is obtained by a nodal expansion of $\mathcal{T}_{q^i}$ at $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{q^i})$, then the quadtrees $\mathcal{T}_{q_\tau^{i+2}}$ for $\tau \in \{1, \dots, n\}$ are obtained by nodal expansions of $\mathcal{T}_{q^{i+1}}$ at nodes $\{t'_1, t'_2, \dots, t'_n\} \in \mathcal{C}(t)$, as shown in Figure 6. In Algorithm 2, the function `populateQ` $(p(x,y), \beta)$ computes the Q-values for each node in the original tree, which are then accessed as the algorithm determines which nodes are to be present in the compressed representation.

Note that $Q_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta)$ conveys whether or not a current poor expansion (that is, one where $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) < 0$) can be overcome by future rewards by continuing expansions that are available through $\{t'_1, \dots, t'_n\}$. One may observe that this is possible due to the dependence of $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta)$ on only the nodes added by moving from $\mathcal{T}_{q^i}$ to $\mathcal{T}_{q^{i+1}}$ and not the overall configuration of the tree, as seen in (25) and the subsequent discussion. Furthermore, the sum over τ in (27) encodes the fact that it is possible for all children of $\{t'_1, \dots, t'_n\}$ to be expanded in ensuing steps if they improve the quality of the solution. In addition, we see from the definition of $Q_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta)$ that, if $\sum_{\tau=1}^{n} Q_Y(\mathcal{T}_{q^{i+1}}, \mathcal{T}_{q_\tau^{i+2}}; \beta) = 0$, then the algorithm will not ignore a one-step improvement provided $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) > 0$. In general, the solution obtained by the Greedy search algorithm will not necessarily be the same as the one obtained by the Q-tree search algorithm. Contrasting the Q-tree search algorithm to the greedy approach, we obtain the following theorem that relates the solutions obtained by these two methods.

Theorem 1. Let $\mathcal{T}_{q^0} \in \mathcal{T}^{\mathcal{Q}}$ be a tree at which both Greedy and Q-tree search algorithms are initialized. Then the solution $\mathcal{T}_{q^*_{\text{G}}}$ obtained by the Greedy search algorithm is a subtree of the solution $\mathcal{T}_{q^*_{\text{Q}}}$ obtained by the Q-search method.

Proof. The proof is presented in Appendix A. □

As a direct consequence of Theorem 1, solutions obtained by the Q-tree search algorithm will contain at least as many leaf-nodes as the solution of the greedy approach, and, at the same time, produce a better solution (if one exists) with respect to (26) for a given $\beta > 0$.

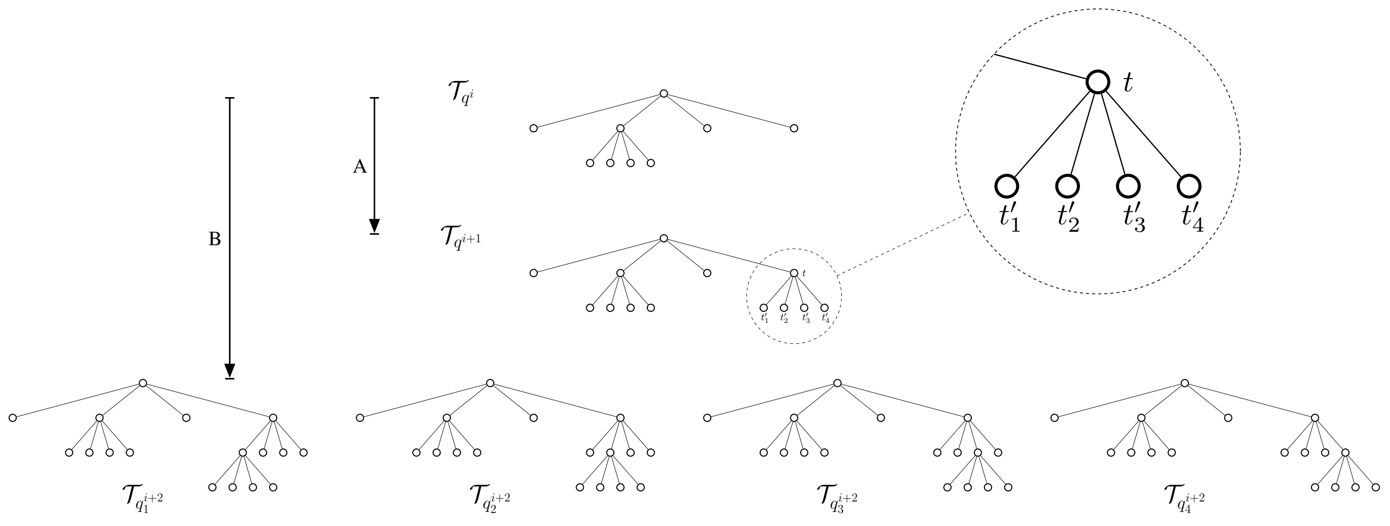
Before we discuss the properties of the solution obtained by the Q-tree search algorithm, we provide the following definition.

Definition 5. A tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ is *minimal* with respect to the cost $L_Y(\cdot; \beta)$ if, for all $\mathcal{T}_{q'} \in \mathcal{T}^{\mathcal{Q}}$ such that $\mathcal{T}_{q'} \subset \mathcal{T}_q$, $L_Y(\mathcal{T}_{q'}; \beta) < L_Y(\mathcal{T}_q; \beta)$.

From Definition 5 we see that, if a tree is minimal, then it is not possible to reduce the number of leaf nodes of the tree without reducing the value of the objective function $L_Y(\cdot; \beta)$. In what follows, we will show that the tree obtained by the

Algorithm 2 The Q-tree search Algorithm.

```
Input Data: $p(x,y)$, $\beta > 0$
Result: $\mathcal{T}_{q^*}$
$\mathcal{T}_{q^0} \leftarrow \mathsf{Root}(\mathcal{T}_{\mathcal{W}})$
$Q_Y(\cdot,\cdot;\beta) \leftarrow$ populateQ $(p(x,y), \beta)$ using (27)
while termination condition not satisfied do
    for each neighbor $\mathcal{T}_{q^k}$ of $\mathcal{T}_{q^i}$ do
        $v(k) \leftarrow Q_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^k}; \beta)$
    end for
    if $\max v > 0$ then
        $z \leftarrow \operatorname{argmax} v$
        $\mathcal{T}_{q^{i+1}} \leftarrow$ neighbor $\mathcal{T}_{q^z}$ of $\mathcal{T}_{q^i}$
        $i \leftarrow i + 1$
    end if
end while
```



$\mathcal{T}_{q^i}$
A
B
$\mathcal{T}_{q^{i+1}}$
t
t'_1
t'_2
t'_3
t'_4
$\mathcal{T}_{q_1^{i+2}}$
$\mathcal{T}_{q_2^{i+2}}$
$\mathcal{T}_{q_3^{i+2}}$
$\mathcal{T}_{q_4^{i+2}}$

Fig. 6. Illustration of the dependency of $Q_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) = \max\left\{\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) + \sum_{\tau=1}^{n} Q_Y(\mathcal{T}_{q^{i+1}}, \mathcal{T}_{q_\tau^{i+2}}; \beta),\ 0\right\}$ on the trees $\mathcal{T}_{q^{i+1}}$ and $\mathcal{T}_{q_\tau^{i+2}}$ for $\tau \in \{1, \ldots, n\}$. Moving from tree $\mathcal{T}_{q^i}$ to tree $\mathcal{T}_{q^{i+1}}$ involves expanding the node $t \in \mathcal{N}_{\text{leaf}}\left(\mathcal{T}_{q^i}\right)$ (line A) and incurs the one-step cost $\Delta L_Y\left(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta\right)$. The trees $\mathcal{T}_{q_\tau^{i+2}}$ for $\tau \in \{1, \ldots, n\}$ are then the trees created by expanding each of the childeren nodes of t, given by $\mathcal{C}(t) = \{t'_1, t'_2, t'_3, t'_4\}$, one at a time. Thus, the trees $\mathcal{T}_{q_\tau^{i+2}}$ differ from the tree $\mathcal{T}_{q^i}$ by two expansions (line B), allowing $Q_Y(\cdot, \cdot; \beta)$ to capture the notion of cost-to-come. Note that $n = 4$ for the special case of quadtrees.

Q-tree search algorithm is minimal and optimal with respect to (26). In order to present these theoretical results, some additional definitions are required, which are provided next.

Definition 6. Given any node $t \in \mathcal{N}(\mathcal{T}_q)$, the *subtree* of $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ *rooted at node* t is denoted by $\mathcal{T}_{q(t)}$ and has node set

$$\mathcal{N}\left(\mathcal{T}_{q(t)}\right) = \left\{t' \in \mathcal{N}(\mathcal{T}_q) : t' \in \bigcup_i \mathcal{D}_i\right\},$$

where $\mathcal{D}_1 = \{t\}$, $\mathcal{D}_{i+1} = \mathcal{A}(\mathcal{D}_i)$ and where

$$\mathcal{A}(\mathcal{D}_i) = \left\{t' \in \mathcal{N}(\mathcal{T}_{\mathcal{W}}) : t' \in \bigcup_{m \in \mathcal{D}_i} \mathcal{C}(m)\right\}.$$

A visualization of $\mathcal{T}_{q(t)}$ for some $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ is provided in Figure 7. Furthermore, recall that $\Delta L_Y(\mathcal{T}_q, \mathcal{T}_{q'}; \beta)$ is only a function of the nodes that are added to tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ to obtain $\mathcal{T}_{q'} \in \mathcal{T}^{\mathcal{Q}}$, as shown by (25) and depicted in Figure 5. Thus, it is convenient to describe $\Delta L_Y(\mathcal{T}_q, \mathcal{T}_{q'}; \beta)$ explicitly as a function of the nodes of the trees $\mathcal{T}_q$ and $\mathcal{T}_{q'}$ as given in the following definition.

Definition 7. The *node-wise* $\Delta \hat{L}_Y$*-function* for any node $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ is given by

$$\Delta \hat{L}_Y(t; \beta) = \Delta \mathcal{L}_Y(\{t'_1, \ldots, t'_n\}; \beta),$$

where $\{t'_1, \ldots, t'_n\} = \mathcal{C}(t) \subset \mathcal{N}(\mathcal{T}_{\mathcal{W}})$. Furthermore, $\Delta \hat{L}(t; \beta) = 0$ for all $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$.

As a consequence of Definition 7, note that if we let $\mathcal{T}_{q'}$ be a neighbor of $\mathcal{T}_q$ such that $\{t'_1, \ldots, t'_n\} = \mathcal{C}(t) = \mathcal{N}(\mathcal{T}_{q'}) \setminus \mathcal{N}(\mathcal{T}_q)$ where $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$ then,

$$\Delta L_Y(\mathcal{T}_q, \mathcal{T}_{q'}; \beta) = \Delta \hat{L}_Y(t; \beta). \tag{29}$$

Moreover, since $Q_Y(\cdot, \cdot; \beta)$ in (27) is recursively defined in terms of $\Delta L_Y(\cdot, \cdot; \beta)$, we have the following definition.

Definition 8. The *node-wise* $\hat{Q}_Y$*-function* for any node $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ is given by

$$\hat{Q}_Y(t; \beta) = \max\left\{\Delta \hat{L}_Y(t; \beta) + \sum_{t' \in \mathcal{C}(t)} \hat{Q}_Y(t'; \beta),\ 0\right\},$$

and where $\hat{Q}_Y(t; \beta) = 0$ for all $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$.

From Definition 8, if $\mathcal{T}_{q'} \in \mathcal{T}^{\mathcal{Q}}$ is a neighbor of $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ where nodes $\{t'_1, \ldots, t'_n\} = \mathcal{C}(t) \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{q'})$ are merged to a node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$ to obtain tree $\mathcal{T}_q$, then we have

$$Q_Y(\mathcal{T}_q, \mathcal{T}_{q'}; \beta) = \hat{Q}_Y(t; \beta). \tag{30}$$

As a result of Definitions 7 and 8, if $\left\{\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}, \ldots, \mathcal{T}_{q^{i+j}}\right\} \subset \mathcal{T}^{\mathcal{Q}}$ is a sequence of trees such that $\mathcal{T}_{q^{i+k+1}}$ is a neighbor of $\mathcal{T}_{q^{i+k}}$ for all $k \in \{0, \ldots, j-1\}$, then

$$\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+j}}; \beta) = \sum_{z \in \mathcal{B}_{ij}} \Delta \hat{L}_Y(z; \beta), \tag{31}$$

where $\mathcal{B}_{ij} = \mathcal{N}_{\text{int}}(\mathcal{T}_{q^{i+j}}) \setminus \mathcal{N}_{\text{int}}(\mathcal{T}_{q^i})$. Additionally, we should note the connection between (31) and (22). Namely, it can be shown that

$$L_Y(\mathsf{Root}\left(\mathcal{T}_{\mathcal{W}}\right); \beta) = 0, \tag{32}$$

which follows from the non-negativity of the mutual information and properties of entropy. Taking $\mathcal{T}_{q^0} = \mathsf{Root}\left(\mathcal{T}_{\mathcal{W}}\right)$ in (22) and utilizing (32), we see that for any $\mathcal{T}_{q^m} \in \mathcal{T}^{\mathcal{Q}}$,

$$L_Y(\mathcal{T}_{q^m}; \beta) = \sum_{i=0}^{m-1} \Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta). \tag{33}$$

Then, since (31) provides a relation for the right-hand side of (33) we have, for any $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$,

$$L_Y(\mathcal{T}_q; \beta) = \sum_{z \in \mathcal{N}_{\text{int}}(\mathcal{T}_q)} \Delta \hat{L}_Y(z; \beta), \tag{34}$$

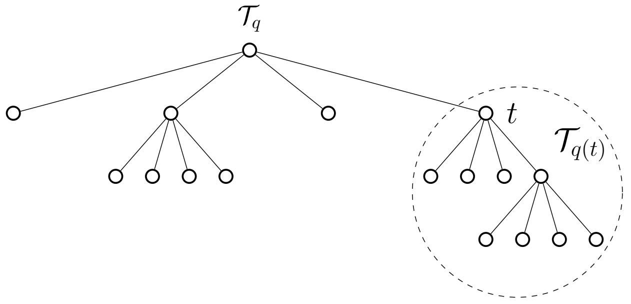


Fig. 7. Visual representation of $\mathcal{T}_{q(t)}$, where $\mathcal{T}_{q(t)} \subseteq \mathcal{T}_q$ for some $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ and node $t \in \mathcal{N}(\mathcal{T}_q)$.

since $\mathcal{N}_{\text{int}}(\mathsf{Root}(\mathcal{T}_{\mathcal{W}})) = \emptyset$, which follows from Definition 2. We see from (34) that the value of $L_Y(\mathcal{T}_q;\beta)$ for any tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ and $\beta > 0$ is the sum of the node-wise $\Delta\hat{L}_Y(\cdot;\beta)$ function over the interior nodes of the tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$. With this in place, we now have the following two lemmas, which will be useful for proving the optimality of the Q-tree search algorithm.

Lemma 1. Let $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$. Then $\hat{Q}_Y(t;\beta) > 0$ if and only if there exists a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta) > 0$. Furthermore, if $\hat{Q}_Y(t;\beta) > 0$, then there exists a tree $\mathcal{T}_{q^*} \in \mathcal{T}^{\mathcal{Q}}$ such that $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*(t)})} \Delta\hat{L}_Y(z;\beta) = \hat{Q}_Y(t;\beta)$, and for all other trees $\mathcal{T}_{q'} \in \mathcal{T}^{\mathcal{Q}}$ with $t \in \mathcal{N}(\mathcal{T}_{q'})$ and $\mathcal{T}_{q'(t)} \neq \mathcal{T}_{q^*(t)}$ it holds that $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q'(t)})} \Delta\hat{L}_Y(z;\beta) \leq \hat{Q}_Y(t;\beta)$.

Proof. The proof is presented in Appendix B. □

The ensuing result implies that if a node with positive $\hat{Q}_Y(\cdot;\beta)$ is not expanded, then the resulting tree is sub-optimal with respect to (26).

Lemma 2. Let $\mathcal{T}_{q^*} \in \mathcal{T}^{\mathcal{Q}}$ be the solution returned by the Q-tree search algorithm and let $\mathcal{T}_{q'} \in \mathcal{T}^{\mathcal{Q}}$ be such that $\mathcal{T}_{q'} \subset \mathcal{T}_{q^*}$. Then

$$L_Y(\mathcal{T}_{q'};\beta) < L_Y(\mathcal{T}_{q^*};\beta).$$

Proof. The proof is presented in Appendix C. □

Importantly, Lemma 1 establishes that a node with $\hat{Q}_Y(\cdot;\beta) > 0$ should be expanded, whereas Lemma 2 states that if the nodes with $\hat{Q}(\cdot;\beta) > 0$ are not expanded, then the resulting tree is sub-optimal with respect to $L_Y(\cdot;\beta)$. The next theorem formally establishes the optimality of solutions found by the Q-tree search algorithm.

Theorem 2. Let $\mathcal{T}_{\tilde{q}} \in \mathcal{T}^{\mathcal{Q}}$ to be a minimal tree that is also optimal with respect to the cost $L_Y(\cdot;\beta)$. Assume, without loss of generality[1], that the Q-tree search algorithm is initialized at the tree $\mathcal{T}_{q^0} \in \mathcal{T}^{\mathcal{Q}}$, where $\mathcal{T}_{q^0} \subseteq \mathcal{T}_{\tilde{q}}$ and let $\mathcal{T}_{q^*} \in \mathcal{T}^{\mathcal{Q}}$ be the solution returned by the Q-tree search algorithm. Then $\mathcal{T}_{q^*} = \mathcal{T}_{\tilde{q}}$.

Proof. The proof is presented in Appendix D. □

[1]The fully abstracted tree with single node $\mathsf{Root}(\mathcal{T}_{\mathcal{W}})$ is a subtree of any quadtree

Theorem 2 establishes that the Q-tree search will find the globally optimal tree with respect to the cost $L_Y(\cdot;\beta)$, provided the algorithm is initiated at a tree $\mathcal{T}_{q^0} \in \mathcal{T}^{\mathcal{Q}}$ such that $\mathcal{T}_{q^0} \subseteq \mathcal{T}_{\tilde{q}}$. Therefore, by selecting $\mathcal{T}_{q^0} = \mathsf{Root}(\mathcal{T}_{\mathcal{W}})$ we can guarantee that the Q-tree search algorithm will find the globally optimal solution. Having established these results, we now discuss some details of our framework before demonstrating the utility of the approach.

C. Influence of $p(x,y)$

A tacit assumption regarding the probability distribution $p(x,y)$ has been made in the development of this framework. Namely, provided that $p(x) > 0$, we can write the distribution $p(x,y)$ as $p(x,y) = p(y|x)p(x)$. This poses no technical concern in the case when $p(x) > 0$ for all $x \in \Omega_X$. In contrast, when $p(x) \not> 0$ for all $x \in \Omega_X$, it may happen that an aggregate node and all of its children have no probability mass. This situation arises if $p(x) = 0$ for all $x \in \Omega_X$ which belong to the aggregate node $t \in \Omega_T$. In this case, we have from (18) that $p(t) = 0$, but it is not clear that (25) is well-defined. Additionally, the need to investigate this scenario is clear from Definition 7 and the subsequent discussion, as it illustrates the connection between the change in the objective function value when moving from tree $\mathcal{T}_{q^i} \in \mathcal{T}^{\mathcal{Q}}$ to tree $\mathcal{T}_{q^{i+1}} \in \mathcal{T}^{\mathcal{Q}}$ to the node-specific quantities. Thus, in order to apply the Greedy or Q-tree search algorithms for general $p(x)$, we must establish that (25) is well defined in these cases. This leads us to the following proposition.

Proposition 1. Let $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ and assume $p(x) = \varepsilon/N$ for all $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})$ with $N = |\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})|$ for some $\varepsilon > 0$. Then $\lim_{\varepsilon\to 0^+} \Delta\hat{L}(t;\beta) = 0$ for all $\beta > 0$.

Proof. The proof is presented in Appendix E. □

The utility of Proposition 1 is that it allows for the direct application of both the Greedy and Q-tree search algorithms for any $p(x)$ without modification to the respective algorithms. This allows us not only to form abstractions as a function of $\beta > 0$, but lets us also dictate *where* information is important by changing $p(x)$. To see why $p(x)$ allows us to dictate where information is important, let the joint distribution $p(x,y)$ be defined by $p(y|x)$ and $p(x)$ as $p(x,y) = p(y|x)p(x)$, and consider

$$p(y|t) = \frac{1}{p(t)} \sum_{x\in\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})} p(y|x)p(x). \tag{35}$$

From (35) we see that nodes $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})$ that are aggregated to $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ and have $p(x) = 0$ do not contribute to the conditional distribution $p(y|t)$, and thus have lower importance to the optimization problem, since these nodes convey no information regarding Y. Therefore, abstract nodes $t \in \Omega_T$ for which the underlying $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})$ have high $p(y|x)$ and $p(x)$ will have the greatest information content regarding Y, as these conditions will increase the value of $p(y|t)$. Furthermore, we see from (35) that, when $p(x)$ is uniform, the algorithm does not discriminate as to where the information in the environment is located, as each value of $p(y|x)$ for $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})$ is given equal weight when computing $p(y|t)$. Consequently, as $\beta \to \infty$ the algorithms become concerned with retaining all the relevant information in the environment, regardless of where this information is located. This is illustrated in the numerical examples we discuss next.

V. Numerical Examples

In this section, we present a numerical example to demonstrate the emergence of abstractions in a grid-world setting. To this end, consider the environment shown in Figure 8d having dimension 256×256. We view this map as representing an environment where the intensity of the color indicates the probability that a given cell is occupied. In this view, the map in Figure 8d can be thought of as an occupancy grid (OG) where the original space, X, is considered to be the elementary cells shown in the figure.

We wish to compress X to an abstract representation T (a quadtree), while preserving as much information regarding cell occupancy as possible. Thus, we take Y to be the relevant random variable corresponding to the occupancy and study this problem while varying $\beta > 0$. Therefore, $\Omega_Y = \{0, 1\}$ where $y = 0$ corresponds to free space and $y = 1$ to occupied space. It is assumed that $p(x)$ is provided and $p(y|x)$ is given by the occupancy grid, where $p(x, y) = p(y|x)p(x)$.

A. Region-Agnostic Abstraction

In this section, we assume that $p(x)$ is uniform. By changing β we obtain a family of solutions, with the leaf node cardinality of the resulting tree returned by the respective algorithm shown in Table I. As seen in Table I, the number of leaf nodes of the trees found by both algorithms is increasing with β. Furthermore, the Q-tree search and Greedy search leaf node cardinalities converge as β tends toward infinity, as expected. Additionally, as seen in Table I, the information contained in the compressed representation T regarding the relevant variable Y, given by $I(T;Y)$, approaches the information that the original space X contains about Y, quantified by $I(X;Y)$. Note also that $I(T;Y) \leq I(X;Y)$, which follows from the Markov chain $Y \to X \to T$ and the data processing inequality. This encodes the fact that the information contained about the relevant variable Y retained by the abstraction T cannot exceed that given by the original space X. Furthermore, from Table I, we notice that the Q-tree search algorithm finds solutions that are more informative regarding the relevant variable Y than the Greedy search algorithm, indicating that the Greedy search algorithm terminates prematurely, and that further improvement is possible for the given $\beta > 0$. We also see that the solutions of the Greedy search algorithm and of the Q-tree search converge as β approaches infinity.

Shown in Figure 9 is the information plane, where the normalized $I(T;Y)$ is plotted versus the normalized $I(T;X)$. In this way, the information plane displays the amount of relevant information retained in a solution vs. the level of compression of X. In viewing this figure, recall that Theorem 2 establishes the global optimality of solutions obtained by Q-tree search, and hence no solution above the Q-tree search line is achievable in the space $\mathcal{T}^{\mathcal{Q}}$, since this would imply that there exist solutions (quadtrees) in $\mathcal{T}^{\mathcal{Q}}$ encoding more information about Y for the same level of compression.

With this in mind, Figure 9 also corroborates that the Greedy search algorithm generally finds solutions that are sub-optimal with respect to $L_Y(\cdot;\beta)$, since trees found by the Greedy search algorithm retain less information about Y for the same level of compression as the information-plane curve of the Greedy search algorithm lies below that of the Q-tree search algorithm. Moving along the curve is done by varying β, with increasing β moving the solution to the right in this plane, towards more informative, higher cardinality solutions. In addition, Figure 9 includes a comparison of Q-tree and Greedy search approaches with the performance of k-class

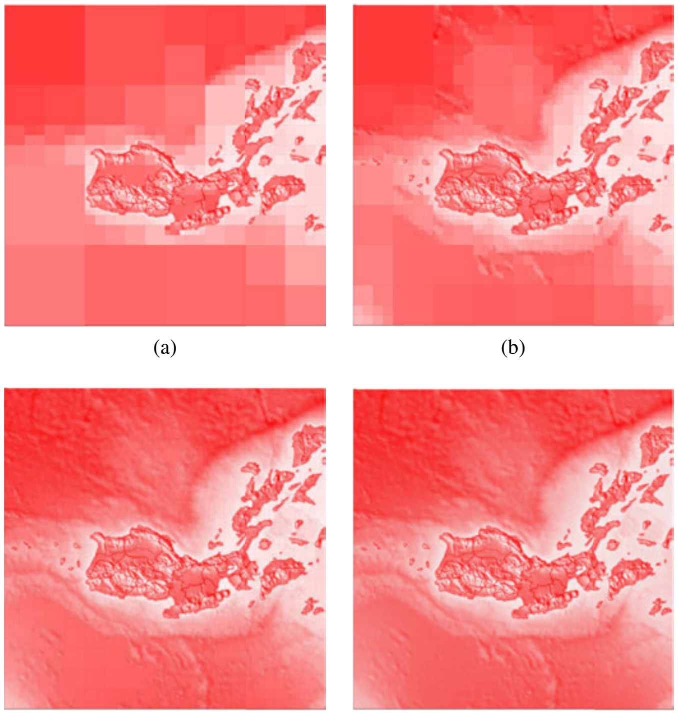


Fig. 8. Visualizations of Q-tree search solutions returned at various β together with original (uncompressed) map for uniform $p(x)$. Shading of red scales with the probability of occupancy. (a)-(c) are the resulting representations for $\beta = 400$, $\beta = 2 \times 10^3$ and $\beta = 30 \times 10^3$, respectively. (d) is the original representation.

TABLE I
NORMALIZED LEAF-NODE CARDINALITY AND RELEVANT INFORMATION FOR SOLUTIONS RETURNED BY Q-TREE SEARCH (Q) AND GREEDY SEARCH ALGORITHMS (G) AS A FUNCTION OF β FOR UNIFORM $p(x)$.

	$\%\lvert\Omega_X\rvert$		$I(T;Y)/I(X;Y)$	
$\beta/10^3$	Q	G	Q	G
0.25	5.11	0.15	0.72	0.0
0.4	10.80	0.42	0.8871	0.5839
0.55	12.35	0.67	0.9163	0.6410
0.7	13.39	2.93	0.9315	0.7202
2	19.76	12.73	0.9714	0.9104
20	62.50	52.06	0.9990	0.9942
70	82.67	76.33	0.9998	0.9987

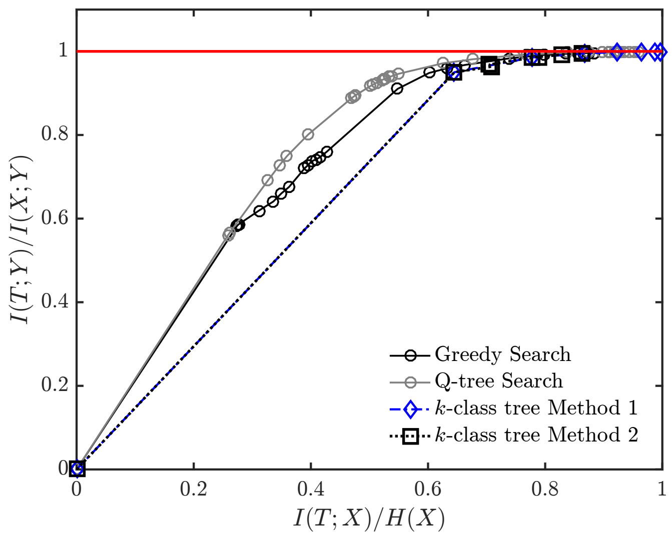


Fig. 9. Information plane ($I(T;Y)/I(X;Y)$ vs. $I(T;X)/H(X)$) for uniform $p(x)$ showing the Q-tree, Greedy and k-class tree solutions. Method 1 corresponds to creating new class intervals by halving the previous class intervals, whereas Method 2 divides the interval $[0,1]$ into k evenly spaced intervals for any integer value of k. The k-class tree solution moves along the curve in the information plane by changing the number of classes.

trees, as introduced in [8]. From this comparison we see that the k-class tree solutions lie below that of both Greedy and Q-tree search, implying that this ad-hoc tree pruning approach finds solutions that are suboptimal as compared to both Greedy and Q-tree search since the k-class trees retain less relevant information for a given level, or amount, of compression.

A sample of environment depictions for various values of β obtained from the Q-tree search algorithm are shown in Figures 8a-8c. As seen in these figures, the solution returned by the Q-tree search algorithm approaches that of the original space as $\beta \to \infty$, with a spectrum of solutions obtained as β is varied. These figures show that areas containing high information content, as specified by Y, are refined first while leaving the regions with less information content to be refined at a higher value of β.

We see that β resembles a sort of a "gain" that can be increased, resulting in progressively more informative solutions of higher cardinality. Thus, once the map is given, changing *only* the value of β gives rise to a variety of solutions of varying resolution. That is, our framework finds the optimal tree $\mathcal{T}_{q^*}$ with respect to $L_Y(\cdot;\beta)$ without the need to specify pre-defined pruning rules or a host of parameters that define the granularity of the abstraction a priori. Interestingly, β plays a similar role in this work as in [18], [26], [27]. Namely, as $\beta \to 0$, highly compressed representations of the space are obtained whereas for large values of β, we asymptotically approach the original map. Thus, we can view β as a "rationality parameter," analogous to [18], [26], [27], where agents with low β are considered to be more resource limited, thus utilizing simpler, lower cardinality representations of the environment.

B. Region-Specific Abstractions

We have discussed how the Greedy and Q-tree search algorithms can be used to obtain abstractions as a function of $\beta > 0$ under the assumption that the distribution $p(x)$ is uniform. We now relax this assumption and discuss the ability to obtain region-specific abstractions in the environment through a non-uniform $p(x)$, without modification to the underlying framework or algorithms, as discussed in Section IV-C.

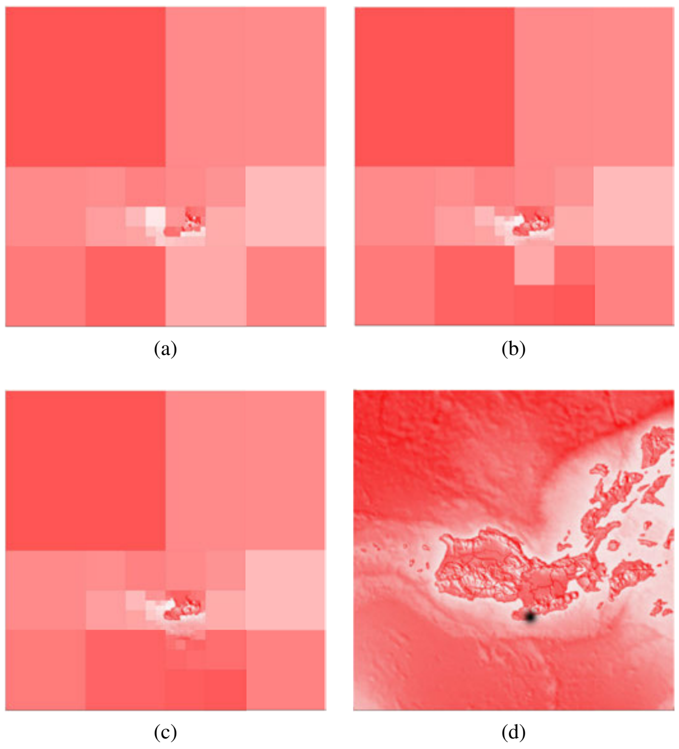


Fig. 10. Visualizations of Q-tree search solutions returned at various β together with original (uncompressed) map for non-uniform $p(x)$. Shading of red scales with the probability of occupancy. (a)-(c) are the resulting representations for $\beta = 160$, $\beta = 3 \times 10^3$ and $\beta = 30 \times 10^3$, respectively. (d) is the original representation together with $p(x)$.

We utilize the same environment as in Figure 8d, but with a non-uniform distribution $p(x)$, as shown in Figure 10d. In this example, we take $p(x)$ to be a two-dimensional Gaussian distribution with mean $\mu = [142, 75]^{\intercal}$ and covariance matrix $\Sigma = 10 I_{2\times 2}$.

Table II summarizes the Q-tree search results for various values of β. Normalized leaf-node cardinalties shown in Tables I and II differ due to the difference in $p(x)$ in the sense that regions with $p(x) = 0$ do not contain any information regarding Y, as seen by (35) and the subsequent discussion. Finally, visualizations of the resulting solutions obtained from the Q-tree search algorithm are provided in Figures 10a-10c.

TABLE II
NORMALIZED LEAF-NODE CARDINALITY AND RELEVANT INFORMATION FOR SOLUTIONS RETURNED BY Q-TREE SEARCH AS A FUNCTION OF β FOR NON-UNIFORM $p(x)$.

$\beta/10^2$	$\%\lvert\Omega_X\rvert$	$I(T;Y)/I(X;Y)$
1.5	0.19	0.7733
1.6	0.22	0.8513
2	0.28	0.9255
20	0.63	0.9912
100	0.86	0.9998

These figures corroborate the previous observations, where we can clearly see that the algorithm refines only regions for which $p(x) > 0$. Furthermore, the refinement is progressive and of increasing resolution as $\beta \to \infty$.

C. Generalization to Other Environments and Algorithmic Complexity

In the previous section, we presented an example which demonstrates the utility of the Q-tree search algorithm, as applied to a 256×256 example. In this section, we present results which show that the algorithm can be used for larger maps, as well as establish results for a number of additional environments.

In order to better understand the behavior of the Q-tree search algorithm, we consider results obtained by averaging over randomly generated 128×128 environments. Table III summarizes these results, obtained by averaging the performances of the Q-tree search algorithm over 100 randomly generated environments for various β. The table shows how both the relevant information and the number of leaf nodes of the solution returned by Q-tree search increases with β. These trends are a direct result of the properties of the problem (21) in that, as β is increased, the IB method will look for solutions that are progressively more informative regarding the relevant variable, at the cost of increased leaf node cardinality.

What is less obvious, however, is the reduction in the variance with increased β. This observation is a result of the interplay between the input distribution $p(x, y)$ and the value(s) of β. Namely, while the amount of relevant information contained in the compressed representation increases with increased β, the actual amount of retained information, and subsequently the cardinality of the set of leaf nodes, for fixed β, is dependent on the specific $p(x, y)$ considered. That is, by fixing β and varying the distribution $p(x, y)$, one will obtain solutions that, in general, differ in the values of $|\Omega_T|$ and $I(T; Y)$. As β is progressively increased and the solution gradually retains more relevant information, the particular values of β which create changes in $I(T; Y)$ are unknown and are a function of $p(x, y)$ (see, for example, [27], [32] and references therein). Recall, though, that all relevant information is retained as $\beta \to \infty$, regardless of the distribution $p(x, y)$. Thus, as β increases, the dependence of the solution on the distribution $p(x, y)$ is effectively reduced, since all solutions at high enough β will retain *all* relevant information. In contrast, the influence of $p(x, y)$ is more predominant at lower values of β, since changes in $p(x, y)$ alters the values of β at which the transitions in information retention occur, leading to higher variability in the solutions obtained.

TABLE III
AVERAGE Q-TREE SEARCH RESULTS OBTAINED BY AVERAGING OVER 100 RANDOMLY GENERATED 128×128 ENVIRONMENTS ASSUMING UNIFORM $p(x)$. TABULAR ENTRIES SHOW MEAN ± STANDARD DEVIATION.

$\beta/10^2$	$\lvert\Omega_T\rvert/\lvert\Omega_X\rvert$	$I(T;Y)/I(X;Y)$
0.5	0.3439 ± 0.4608	0.3582 ± 0.48
0.505	0.7152 ± 0.4157	0.7462 ± 0.433
0.51	0.9460 ± 0.0956	0.9851 ± 0.0995
0.515	0.9561 ± 0.0027	0.9951 ± 0.0004
0.601	0.9649 ± 0.0025	0.9967 ± 0.0003
0.878	0.98 ± 0.0018	0.9987 ± 0.0001
100	1.0 ± 0	1.0 ± 0

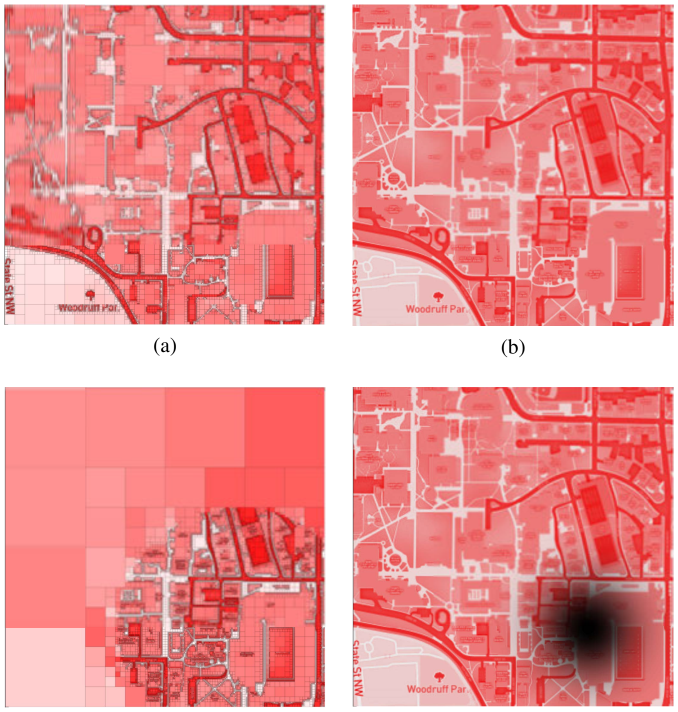


Fig. 11. Visualizations of Q-tree search solutions returned at various β together with original (uncompressed) map. Shading of red scales with the probability of occupancy. Map size is 512×512. (a) $\beta = 300$ representation and (b) original map for uniform $p(x)$. (c) $\beta = 1 \times 10^3$ representation and (d) original map with $p(x)$ when $p(x)$ is non-uniform.

In addition to the above discussion pertaining to averaged results, we also present a region-agnostic and region-specific abstraction case applied to a 512×512 map shown in Figures 11a-11d, illustrating the applicability of the Q-tree search algorithm to larger grid sizes. However, in order to appropriately address the scalability of the presented algorithms, we provide some comments on algorithmic complexity. To this end, note that the complexity of Q-tree search algorithm is given by $\mathcal{O}(\hat{N})$, where $\hat{N} = |\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})|$. To arrive at this result, we note that the function `populateQ(·,·)` in Algorithm 2 performs $\tilde{N} = |\mathcal{N}(\mathcal{T}_{\mathcal{W}})|$ operations, where $\tilde{N}$ and $\hat{N}$ are related by a constant factor. In addition to this, Q-tree search may, in the worst case, expand $\tilde{N} - \hat{N}$ nodes. This process requires $\mathcal{O}(\hat{N})$ computations, resulting in the overall complexity of the algorithm on the order of $\mathcal{O}(\hat{N})$.

To conclude this section, we present an additional set of results to demonstrate our approach on robotic applications. Particularly, we consider the use of abstract representations for the purpose of indoor occupancy grid compression and path-planning for autonomous agents. These results are shown in Figures 12a-12d. It is noted that a high degree of compression is achievable for the indoor occupancy grid in Figure 12a-12b without the loss of any relevant information. Specifically, we note that approximately 10.5% of the original space nodes were needed in order to retain all information regarding occupancy. This represents a rather drastic reduction in the number of nodes required to represent the space, which can then help simplify other tasks such as path-planning – an idea we exploit in Figures 12c-12d. The path-planning

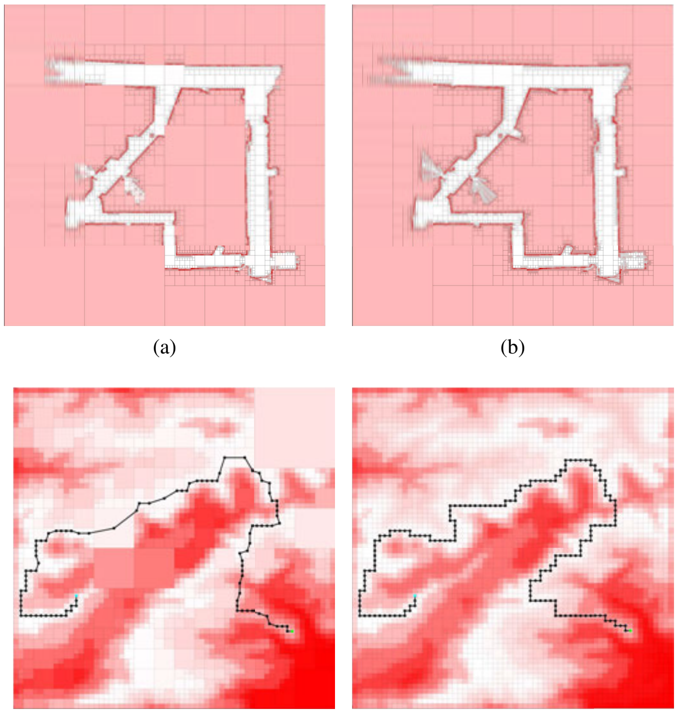

(a) (b) (c) (d)

Fig. 12. Application to robotics examples. (a)-(b) Indoor occupancy grid compression for $\beta = 490$ and $\beta = 30 \times 10^3$ with uniform $p(x)$. (c)-(d) Sample paths on abstract representations for $\beta = 300$ and $\beta = 14.9 \times 10^3$ with uniform $p(x)$.

example shown in Figures 12c-12d shows the applicability of our framework to planning problems. In this example, an autonomous agent is required to navigate from a given start location to a goal position while finding a path that minimizes the sum of occupancy values of cells along the path. We utilize Dijkstra's graph-search algorithm to plan the paths in the reduced graphs. In the example, the abstractions are generated by employing our framework, where then a corresponding reduced graph can be constructed and used for path planning. Thus, by means of abstraction, we decreases the computational time required as compared to solving the problem on the original representation/resolution, since the reduced graph contains fewer nodes/vertices.

VI. CONCLUSIONS

In this paper, we have developed a novel framework for the emergence of abstractions that are not provided to the agent a priori but, instead, arise as a result of the available agent computational resources. We utilize concepts from information theory, such as the information bottleneck and agglomerative information bottleneck methods, to formulate a new optimization problem over the space of trees. The structural properties of the framework were discussed with applications to bounded rationality and information-limited agents. Finally, we propose and analyze two algorithms, which were implemented to obtain solutions for a number of two-dimensional environments with applications to robotics.

The importance of this work lies in the development of a framework that allows for the emergence of abstractions in a principled manner. The proposed algorithms demonstrate the utility of the approach, requiring only the specification of a relevant variable that contains the information we wish to retain in the resulting compressed representation. The framework then searches for trees that not only compress the original space, but maximally preserve the information regarding the relevant variable. The results can be utilized in decision-making problems to systematically compress the given state representation or in path-planning algorithms to develop reduced complexity representations of the original planning space.

APPENDIX A
PROOF OF THEOREM 1

Note that

$$\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) \leq \Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) + \sum_{\tau=1}^{n} Q_Y(\mathcal{T}_{q^{i+1}}, \mathcal{T}_{q_\tau^{i+2}}; \beta), \quad (36)$$

since $Q_Y(\mathcal{T}_{q^{i+1}}, \mathcal{T}_{q_\tau^{i+2}}; \beta) \geq 0$. In the Greedy search algorithm, a node is expanded, adding $\{t'_1, \ldots, t'_n\}$ to $\mathcal{N}(\mathcal{T}_{q^i})$ to obtain $\mathcal{N}(\mathcal{T}_{q^{i+1}})$, if $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) > 0$. If $\Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) > 0$ then by (36) and (27) it follows that

$$0 < \Delta L_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) + \sum_{\tau=1}^{n} Q_Y(\mathcal{T}_{q^{i+1}}, \mathcal{T}_{q_\tau^{i+2}}; \beta) = Q_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta),$$

and therefore $Q_Y(\mathcal{T}_{q^i}, \mathcal{T}_{q^{i+1}}; \beta) > 0$. Hence nodes expanded by the Greedy search algorithm will also be expanded by Q-tree search. Since the two algorithms are initialized at a common $\mathcal{T}_{q^0} \in \mathcal{T}^{\mathcal{Q}}$, it follows that $\mathcal{T}_{q_G^*} \subseteq \mathcal{T}_{q_Q^*}$.

APPENDIX B
PROOF OF LEMMA 1

The proof is given by induction. We first establish necessity and sufficiency for some $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_{\mathcal{W}})$, where $\ell > 0$ is the maximum depth of $\mathcal{T}_{\mathcal{W}}$.
($\Rightarrow$) Assume $\hat{Q}_Y(t; \beta) > 0$ for some $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_{\mathcal{W}})$. We thus have

$$0 < \hat{Q}_Y(t; \beta) = \max\left\{\Delta \hat{L}_Y(t; \beta) + \sum_{t' \in \mathcal{C}(t)} \hat{Q}_Y(t'; \beta);\ 0\right\}.$$

Hence,

$$\Delta \hat{L}_Y(t; \beta) + \sum_{t' \in \mathcal{C}(t)} \hat{Q}_Y(t'; \beta) > 0.$$

Since $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_{\mathcal{W}})$ it follows that $t' \in \mathcal{C}(t) \subset \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$ and thus $\hat{Q}_Y(t'; \beta) = 0$, which implies that $\Delta \hat{L}_Y(t; \beta) =$

$\hat{Q}_Y(t;\beta) > 0$. Now consider the tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $\mathcal{N}_{\text{leaf}}(\mathcal{T}_{q(t)}) = \mathcal{C}(t)$. Then, for the subtree $\mathcal{T}_{q(t)} \subseteq \mathcal{T}_{\mathcal{W}}$

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta) = \Delta\hat{L}_Y(t;\beta) > 0.$$

($\Leftarrow$) Assume there exists a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta) > 0.$$

Note that, since $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_{\mathcal{W}})$ then $\mathcal{N}(\mathcal{T}_{q(t)}) = \{t\} \cup \mathcal{C}(t)$, with $\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) = \{t\}$ and $\mathcal{N}_{\text{leaf}}(\mathcal{T}_{q(t)}) = \mathcal{C}(t) \subset \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$. Therefore,

$$0 < \sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta) = \Delta\hat{L}_Y(t;\beta),$$

and

$$\begin{aligned}\hat{Q}_Y(t;\beta) &= \max\big\{\Delta\hat{L}_Y(t;\beta) + \sum_{t'\in\mathcal{C}(t)} \hat{Q}_Y(t';\beta),\ 0\big\},\\ &= \max\big\{\Delta\hat{L}_Y(t;\beta),\ 0\big\},\\ &= \Delta\hat{L}_Y(t;\beta) > 0.\end{aligned}$$

Furthermore, for the tree $\mathcal{T}_q$ we have $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta) = \hat{Q}_Y(t;\beta)$ and since $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_{\mathcal{W}})$, for any other tree $\mathcal{T}_{\tilde{q}}$ such that $\mathcal{T}_{\tilde{q}(t)} \neq \mathcal{T}_{q(t)}$, it holds that $\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)}) = \emptyset$, which implies that $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)})} \Delta\hat{L}_Y(z;\beta) = 0 \le \hat{Q}_Y(t;\beta)$. Thus, the lemma is true for all nodes $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_{\mathcal{W}})$.

We now establish necessity and sufficiency for all $k \in \{1,\dots,\ell-1\}$. To this end, assume that for some $k \in \{1,\dots,\ell-1\}$ and any $t' \in \mathcal{N}_k(\mathcal{T}_{\mathcal{W}})$, $\hat{Q}_Y(t';\beta) > 0$ if and only if there exists a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t')})} \Delta\hat{L}_Y(z;\beta) > 0$. Furthermore, if $\hat{Q}_Y(t';\beta) > 0$ then there exists a tree $\mathcal{T}_{q^*} \in \mathcal{T}^{\mathcal{Q}}$ such that $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*(t')})} \Delta\hat{L}_Y(z;\beta) = \hat{Q}_Y(t';\beta)$, and for all other trees $\mathcal{T}_{\tilde{q}} \in \mathcal{T}^{\mathcal{Q}}$ with $t' \in \mathcal{N}(\mathcal{T}_{\tilde{q}})$ and $\mathcal{T}_{\tilde{q}(t')} \neq \mathcal{T}_{q^*(t')}$, $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t')})} \Delta\hat{L}_Y(z;\beta) \le \hat{Q}_Y(t';\beta)$. Using this hypothesis, we prove that the lemma also holds for all $t \in \mathcal{N}_{k-1}(\mathcal{T}_{\mathcal{W}})$.

($\Rightarrow$) Consider $t \in \mathcal{N}_{k-1}(\mathcal{T}_{\mathcal{W}})$ and assume that $\hat{Q}_Y(t;\beta) > 0$. Define the set

$$\mathcal{S} = \Big\{t' \in \mathcal{C}(t) : \hat{Q}_Y(t';\beta) > 0\Big\} \subset \mathcal{N}_k(\mathcal{T}_{\mathcal{W}}).$$

If $\mathcal{S} = \emptyset$ then from Definition 8, $0 < \hat{Q}_Y(t;\beta) = \max\{\Delta\hat{L}_Y(t;\beta), 0\}$, and therefore $\hat{Q}_Y(t;\beta) = \Delta\hat{L}_Y(t;\beta) > 0$. Now, consider any tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $\mathcal{T}_{q(t)}$ has node set $\mathcal{N}(\mathcal{T}_{q(t)}) = \{t\} \cup \mathcal{C}(t)$. Note that $\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) = \{t\}$ and $\mathcal{N}_{\text{leaf}}(\mathcal{T}_{q(t)}) = \mathcal{C}(t)$. Thus, for the subtree $\mathcal{T}_{q(t)}$,

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta) = \Delta\hat{L}_Y(t;\beta) = \hat{Q}_Y(t;\beta).$$

Therefore $\hat{Q}_Y(t;\beta) > 0$ implies that there exists a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta) > 0$.

Now consider $\mathcal{S} \neq \emptyset$. By hypothesis, there exists a tree $\mathcal{T}_{q^*} \in \mathcal{T}^{\mathcal{Q}}$ such that

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*(t')})} \Delta\hat{L}_Y(z;\beta) = \hat{Q}_Y(t';\beta), \quad \forall t' \in \mathcal{S}.$$

Consider a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $\mathcal{T}_{q(t)}$ has the properties

$$\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) = \{t\} \bigcup_{t'\in\mathcal{S}} \mathcal{N}_{\text{int}}(\mathcal{T}_{q^*(t')}),$$

and

$$\mathcal{N}_{\text{leaf}}(\mathcal{T}_{q(t)}) = (\mathcal{C}(t) \setminus \mathcal{S}) \bigcup_{t'\in\mathcal{S}} \mathcal{N}_{\text{leaf}}(\mathcal{T}_{q^*(t')}).$$

Therefore, using the fact that $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*(t')})} \Delta\hat{L}_Y(z;\beta) = \hat{Q}_Y(t';\beta)$, for all $t' \in \mathcal{S}$, we have

$$\begin{aligned}&\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta)\\ &\qquad = \Delta\hat{L}_Y(t;\beta) + \sum_{t'\in\mathcal{S}} \sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*(t')})} \Delta\hat{L}_Y(z;\beta),\\ &\qquad = \Delta\hat{L}_Y(t;\beta) + \sum_{t'\in\mathcal{S}} \hat{Q}_Y(t';\beta).\end{aligned}$$

Also note that $\hat{Q}_Y(t';\beta) = 0$ for all $t' \in \mathcal{C}(t) \setminus \mathcal{S}$ and hence,

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta) = \Delta\hat{L}_Y(t;\beta) + \sum_{t'\in\mathcal{S}} \hat{Q}_Y(t';\beta) + \sum_{t'\in\mathcal{C}(t)\setminus\mathcal{S}} \hat{Q}_Y(t';\beta).$$

Furthermore, note that from Definition 8, if $\hat{Q}_Y(t;\beta) > 0$ then

$$\hat{Q}_Y(t;\beta) = \Delta\hat{L}_Y(t;\beta) + \sum_{t'\in\mathcal{C}(t)} \hat{Q}_Y(t';\beta),$$

and thus,

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta) = \hat{Q}_Y(t;\beta) > 0.$$

Therefore, it follows that if $\hat{Q}_Y(t;\beta) > 0$, there exists a tree such that $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta) > 0$ and $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta\hat{L}_Y(z;\beta) = \hat{Q}_Y(t;\beta)$. Furthermore, consider any $\mathcal{T}_{\tilde{q}} \in \mathcal{T}^{\mathcal{Q}}$ such that $\mathcal{T}_{\tilde{q}(t)} \neq \mathcal{T}_{q(t)}$. Then

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)})} \Delta\hat{L}_Y(z;\beta) = \Delta\hat{L}_Y(t;\beta) + \sum_{t'\in\mathcal{C}(t)\cap\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)})} \left(\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t')})} \Delta\hat{L}_Y(z;\beta)\right).$$

Note that $t' \in \mathcal{N}_k(\mathcal{T}_{\mathcal{W}})$ and that

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t')})} \Delta\hat{L}_Y(z;\beta) \le \hat{Q}_Y(t';\beta).$$

Consequently,

$$\begin{aligned}\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)})}&\Delta\hat{L}_Y(z;\beta)\\&\leq\Delta\hat{L}_Y(t;\beta)+\sum_{t'\in\mathcal{C}(t)\cap\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)})}\hat{Q}_Y(t';\beta),\\&\leq\Delta\hat{L}_Y(t;\beta)+\sum_{t'\in\mathcal{C}(t)}\hat{Q}_Y(t';\beta),\\&=\hat{Q}_Y(t;\beta).\end{aligned}$$

($\Leftarrow$) Let $t\in\mathcal{N}_{k-1}(\mathcal{T}_\mathcal{W})$ and assume that there exists a tree $\mathcal{T}_q\in\mathcal{T}^\mathcal{Q}$ such that

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})}\Delta\hat{L}_Y(z;\beta)>0,$$

and consider any $t'\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})\cap\mathcal{C}(t)\subset\mathcal{N}_k(\mathcal{T}_\mathcal{W})$. From the hypothesis we have that

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t')})}\Delta\hat{L}_Y(z;\beta)\leq\hat{Q}_Y(t';\beta).$$

Therefore,

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})}\Delta\hat{L}_Y(z;\beta)=$$
$$\Delta\hat{L}_Y(t;\beta)+\sum_{t'\in\mathcal{C}(t)\cap\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})}\left(\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t')})}\Delta\hat{L}_Y(z;\beta)\right),$$

which yields

$$\begin{aligned}0<\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})}&\Delta\hat{L}_Y(z;\beta)\leq\\ \Delta\hat{L}_Y(t;\beta)+&\sum_{t'\in\mathcal{C}(t)\cap\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})}\hat{Q}_Y(t';\beta)+\\&\underbrace{\sum_{t'\in\mathcal{C}(t)\setminus\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})}\hat{Q}_Y(t';\beta)}_{\geq 0}.\end{aligned}$$

Hence,

$$0<\Delta\hat{L}_Y(t;\beta)+\sum_{t'\in\mathcal{C}(t)}\hat{Q}_Y(t';\beta)\leq\hat{Q}_Y(t;\beta).$$

Therefore, the existence of a tree $\mathcal{T}_q\in\mathcal{T}^\mathcal{Q}$ with $\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})}\Delta\hat{L}_Y(z;\beta)>0$ where $t\in\mathcal{N}_{k-1}(\mathcal{T}_\mathcal{W})$ implies $\hat{Q}_Y(t;\beta)>0$.

Thus, we have shown that the lemma holds for $k-1$ and for all $t\in\mathcal{N}_{k-1}(\mathcal{T}_\mathcal{W})$.

Appendix C
Proof of Lemma 2

Let $t\in\mathcal{N}(\mathcal{T}_\mathcal{W})$ be any node such that $t\in\mathcal{N}_{\text{leaf}}(\mathcal{T}_{q'})\cap\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*})$, where $\mathcal{T}_{q'}\subset\mathcal{T}_{q^*}$. Note that $\hat{Q}_Y(n;\beta)>0$ for all $n\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*(t)})$ and $\hat{Q}_Y(n;\beta)=0$ for all $n\in\mathcal{N}_{\text{leaf}}(\mathcal{T}_{q^*(t)})$, which follows from the design of the Q-tree search algorithm. Thus, we have that

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*(t)})}\Delta\hat{L}_Y(z;\beta)=\hat{Q}_Y(t;\beta)>0,$$

which holds for all $t\in\mathcal{N}_{\text{leaf}}(\mathcal{T}_{q'})\cap\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*})$. Furthermore, using (34),

$$L_Y(\mathcal{T}_{q'};\beta)+\sum_{t\in\mathcal{N}_{\text{leaf}}(\mathcal{T}_{q'})\cap\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*})}\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*(t)})}\Delta\hat{L}_Y(z;\beta)=L_Y(\mathcal{T}_{q^*};\beta).$$

The above is equivalent to

$$L_Y(\mathcal{T}_{q'};\beta)+\sum_{t\in\mathcal{N}_{\text{leaf}}(\mathcal{T}_{q'})\cap\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*})}\hat{Q}_Y(t;\beta)=L_Y(\mathcal{T}_{q^*};\beta).$$

Lastly, it is known that Q-tree search did not terminate at $\mathcal{T}_{q'}$. Thus, $\sum_{t\in\mathcal{N}_{\text{leaf}}(\mathcal{T}_{q'})\cap\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*})}\hat{Q}_Y(t;\beta)>0$, where $N_{\text{leaf}}(\mathcal{T}_{q'})\cap\mathcal{N}_{\text{int}}(\mathcal{T}_{q^*})\neq\emptyset$ if $\mathcal{T}_{q'}\neq\mathcal{T}_{q^*}$, and therefore

$$L_Y(\mathcal{T}_{q'};\beta)<L_Y(\mathcal{T}_{q^*};\beta).$$

Appendix D
Proof of Theorem 2

Let $t\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}})$ and consider the tree $\mathcal{T}_{\bar{q}}\in\mathcal{T}^\mathcal{Q}$ with node set $\mathcal{N}(\mathcal{T}_{\bar{q}})=\{t\}\cup\mathcal{N}(\mathcal{T}_{\tilde{q}})\setminus\mathcal{N}(\mathcal{T}_{\tilde{q}(t)})$. We have from (34) that

$$L_Y(\mathcal{T}_{\bar{q}};\beta)=\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}})\setminus\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)})}\Delta\hat{L}_Y(z;\beta).$$

From the above expression and (22) and (31), we have

$$L_Y(\mathcal{T}_{\tilde{q}};\beta)=L_Y(\mathcal{T}_{\bar{q}};\beta)+\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)})}\Delta\hat{L}_Y(z;\beta).$$

Since $\mathcal{T}_{\tilde{q}}$ is minimal, for any subtree $\mathcal{T}_{\bar{q}}$ we have $L_Y(\mathcal{T}_{\bar{q}};\beta)<L_Y(\mathcal{T}_{\tilde{q}};\beta)$, and therefore

$$\sum_{z\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)})}\Delta\hat{L}_Y(z;\beta)>0,\quad\forall t\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}}).$$

Hence, from Lemma 1, it follows that $\hat{Q}_Y(t;\beta)>0$ for all $t\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}})$. Thus, all nodes in $\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}})$ are expanded in $\mathcal{T}_{q^*}$, which implies that $\mathcal{T}_{q^*}\supseteq\mathcal{T}_{\tilde{q}}$. Then, either $\mathcal{T}_{q^*}=\mathcal{T}_{\tilde{q}}$, which implies $L_Y(\mathcal{T}_{\tilde{q}};\beta)=L_Y(\mathcal{T}_{q^*};\beta)$, or $\mathcal{T}_{q^*}\supset\mathcal{T}_{\tilde{q}}$, which, from Lemma 2, implies that $L_Y(\mathcal{T}_{\tilde{q}};\beta)<L_Y(\mathcal{T}_{q^*};\beta)$. However, since $\mathcal{T}_{\tilde{q}}$ is optimal, we have $L_Y(\mathcal{T}_{\tilde{q}};\beta)\geq L_Y(\mathcal{T}_{q^*};\beta)$, leading to a contradiction. Thus, $\mathcal{T}_{q^*}=\mathcal{T}_{\tilde{q}}$ and consequently $L_Y(\mathcal{T}_{\tilde{q}};\beta)=L_Y(\mathcal{T}_{q^*};\beta)$.

Appendix E
Proof of Proposition 1

Assume $\beta>0$, $t\in\mathcal{N}_{\text{int}}(\mathcal{T}_\mathcal{W})$ and $p(x)=\varepsilon/N$ for all $x\in\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})$ with $N=|\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})|$. By (25) and Definition 7, we have

$$\Delta\hat{L}(t;\beta)=p(t)\left[\text{JS}_\Pi(p(y|t'_1),\ldots,p(y|t'_{|\mathcal{C}(t)|}))-\frac{1}{\beta}H(\Pi)\right],$$

where, without loss of generality, $\left\{t'_1, \ldots, t'_{|\mathcal{C}(t)|}\right\} = \mathcal{C}(t)$. Moreover, since $p(t|x)$ is deterministic,

$$p(t) = \sum_{x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})} p(t|x)p(x) = \sum_{x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})} p(x) = \varepsilon,$$

and since $p(x) = \varepsilon/N$ for all $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})$, it follows that

$$p(t') = \frac{\varepsilon}{|\mathcal{C}(t)|}, \quad t' \in \mathcal{C}(t).$$

Consequently,

$$\Pi = \left\{\frac{p(t'_1)}{p(t)}, \ldots, \frac{p(t'_{|\mathcal{C}(t)|})}{p(t)}\right\} = \left\{\frac{1}{|\mathcal{C}(t)|}, \ldots, \frac{1}{|\mathcal{C}(t)|}\right\},$$

and therefore,

$$H(\Pi) = \log|\mathcal{C}(t)|. \tag{37}$$

Now define

$$a_s(y) \triangleq \sum_{x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(s)})} p(x, y),$$

where $y \in \Omega_Y$ and $s \in \mathcal{N}(\mathcal{T}_{\mathcal{W}})$. Thus,

$$\sum_y a_{t'}(y) = \frac{\varepsilon}{|\mathcal{C}(t)|}, \tag{38}$$

and

$$\sum_y a_t(y) = \varepsilon,$$

for all $t' \in \mathcal{C}(t)$. Since $\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t')}) \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})$, it follows that $0 \leq a_{t'}(y) \leq a_t(y) \leq \varepsilon$. Thus, for $t' \in \mathcal{C}(t)$ we have, from the definition of the KL-divergence,

$$\mathrm{D}_{\mathrm{KL}}(p(y|t'), p(y|t)) = \sum_y p(y|t') \log \frac{p(y|t')}{p(y|t)},$$

where

$$p(y|t) = \frac{1}{p(t)} \sum_{x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t)})} p(x, y) = \frac{1}{\varepsilon} a_t(y),$$

and similarly,

$$p(y|t') = \frac{1}{p(t')} \sum_{x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(t')})} p(x, y) = \frac{|\mathcal{C}(t)|}{\varepsilon} a_{t'}(y).$$

Hence,

$$\begin{aligned}\mathrm{D}_{\mathrm{KL}}(p(y|t'), p(y|t)) &= \sum_y p(y|t') \log \frac{|\mathcal{C}(t)| a_{t'}(y)}{a_t(y)}, \\ &= \log|\mathcal{C}(t)| + \sum_y p(y|t') \log \frac{a_{t'}(y)}{a_t(y)}, \\ &= \log|\mathcal{C}(t)| + \frac{|\mathcal{C}(t)|}{\varepsilon} \sum_y a_{t'}(y) \log \frac{a_{t'}(y)}{a_t(y)}. \end{aligned} \tag{39}$$

Since $0 \leq a_{t'}(y) \leq a_t(y)$ for all $y \in \Omega_Y$ we have from (38) and (39) that

$$\begin{aligned}\frac{|\mathcal{C}(t)|}{\varepsilon} \sum_y a_{t'}(y) \log \frac{a_{t'}(y)}{a_t(y)} &\leq \frac{|\mathcal{C}(t)|}{\varepsilon} \sum_y a_{t'}(y) \log \frac{a_t(y)}{a_t(y)}, \\ &= \frac{|\mathcal{C}(t)|}{\varepsilon} \log(1) \sum_y a_{t'}(y), \\ &= 0.\end{aligned}$$

From the previous expression, along with (39), it follows that

$$0 \leq \mathrm{D}_{\mathrm{KL}}(p(y|t'), p(y|t)) \leq \log|\mathcal{C}(t)|, \quad \forall t' \in \mathcal{C}(t). \tag{40}$$

Using (40) and the definition of JS-divergence, we see that

$$\begin{aligned}\mathrm{JS}_\Pi(p(y|t'_1), \ldots, p(y|t'_{|\mathcal{C}(t)|}) &= \sum_{i=1}^{|\mathcal{C}(t)|} \Pi(i) \mathrm{D}_{\mathrm{KL}}(p(y|t'_i), p(y|t)), \\ &\leq \log|\mathcal{C}(t)|.\end{aligned}$$

Therefore, from the non-negativity of the JS-divergence as well as (37) and (40) we have,

$$\begin{aligned}&-\frac{1}{\beta}\varepsilon \log|\mathcal{C}(t)| \leq \\ &\quad p(t)\left[\mathrm{JS}_\Pi(p(y|t'_1), \ldots, p(y|t'_{|\mathcal{C}(t)|})) - \frac{1}{\beta} H(\Pi)\right] \\ &\qquad \leq \frac{\beta - 1}{\beta} \varepsilon \log|\mathcal{C}(t)|.\end{aligned}$$

Now taking the limit as $\varepsilon \to 0^+$ yields $\lim_{\varepsilon \to 0^+} \Delta \hat{L}(t; \beta) = 0$ for all $\beta > 0$.

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, 2006.

[2] S. Behnke, "Local multiresolution path planning," in *RoboCup 2003*, ser. Lecture Notes in Artificial Intelligence, vol. 3020, 2004, pp. 332–343.

[3] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 4th ed. Athena Scientific, 2012, vol. 2.

[4] M. M. Botvinick, "Hierarchical reinforcement learning and decision making," *Current Opinion in Neurobiology*, vol. 22, no. 6, pp. 956–962, Dec. 2012.

[5] F. Hauer, A. Kundu, J. M. Rehg, and P. Tsiotras, "Multi-scale perception and path planning on probabilistic obstacle maps," in *IEEE International Conference on Robotics and Automation*, Seattle, WA, May 26-30, 2015, pp. 4210–4215.

[6] F. Hauer and P. Tsiotras, "Reduced complexity multi-scale path-planning on probabilistic maps," in *IEEE Conference on Robotics and Automation*, Stockholm, Sweden, May 16-21, 2016, pp. 83–88.

[7] S. Kambhampati and L. S. Davis, "Multiresolution path planning for mobile robots," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 3, pp. 135–145, September 1986.

[8] G. K. Kraetzschmar, G. P. Gassull, and K. Uhl, "Probabilistic quadtrees for variable-resolution mapping of large environments," *IFAC Proceedings Volumes*, vol. 37, no. 8, pp. 675–680, July 2004.

[9] R. V. Cowlagi and P. Tsiotras, "Multiresolution path planning with wavelets: A local replanning approach," in *American Control Conference*, Seattle, WA, June 11-13, 2008, pp. 1220–1225.

[10] ——, "Multi-resolution path planning: Theoretical analysis, efficient implementation, and extensions to dynamic environments," in *IEEE Conference on Decision and Control*, Atlanta, GA, December 15-17, 2010, pp. 1384–1390.

[11] ——, "Multiresolution motion planning for autonomous agents via wavelet-based cell decompositions," *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 42, no. 5, pp. 1455–1469, October 2012.

[12] ——, "Hierarchical motion planning with kinodynamic feasibility guarantees," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 379–395, April 2012.

[13] E. Einhorn, C. Schröter, and H.-M. Gross, "Finding the adequate resolution for grid mapping - cell sizes locally adapting on-the-fly," in *IEEE Conference on Robotics and Automation*, Shanghai, China, May 9-13, 2011, pp. 1843–1848.

[14] P. Tsiotras, D. Jung, and E. Bakolas, "Multiresolution hierarchical path-planning for small UAVs using wavelet decompositions," *Journal of Intelligent and Robotic Systems*, vol. 66, pp. 505–522, June 2012.

[15] B. M. Kurkoski and H. Yagi, "Quantization of binary-input discrete memoryless channels," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 4544–4552, August 2014.

[16] E. Nelson, M. Corah, and N. Michael, "Environment model adaptation for mobile robot exploration," *Autonomous Robots*, vol. 42, pp. 257–272, Feb. 2018.

[17] H. Vangala, E. Viterbo, and Y. Hong, "Quantization of binary input DMC at optimal mutual information using constrained shortest path problem," in *International Conference on Telecommunications*, Sydney, Australia, April 27-29, 2015, pp. 151–155.

[18] D. T. Larsson, D. Braun, and P. Tsiotras, "Hierarchical state abstractions for decision-making problems with computational constraints," in *IEEE Conference on Decision and Control*, Melbourne, Australia, December 12-15, 2017, pp. 1138–1143.

[19] R. C. Holte and B. Y. Choueiry, "Abstraction and reformulation in artificial intelligence," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 358, no. 1435, pp. 1197–1204, July 2003.

[20] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Ocotomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, pp. 189–206, Apr. 2013.

[21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT press, 2005.

[22] L. Li, T. J. Walsh, and M. L. Littman, "Towards a unified theory of state abstraction for MDPs," in *International Symposium on Artificial Intelligence and Mathematics*, Fort Lauderdale, FL, January 4-6, 2006, pp. 1–10.

[23] G. J. Pappas, "Bisimilar linear systems," *Automatica*, vol. 39, no. 12, pp. 2035–2047, December 2003.

[24] A. Girard and G. J. Pappas, "Approximate bisimulation: A bridge between computer science and control theory," *European Journal of Control*, vol. 17, no. 5-6, pp. 568–578, 2011.

[25] C. K. Chow and C. N. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, May 1968.

[26] N. Tishby and D. Polani, "Information theory of decisions and actions," in *Preception-Action Cycle*, ser. Springer Series in Cognitive and Neural Systems, V. Cutsuridis, A. Hussain, and J. G. Taylor, Eds. Springer, New York, NY, December 2010, ch. 19, pp. 601–636.

[27] T. Genewein, F. Leibfried, J. Grau-Moya, and D. A. Braun, "Bounded rationality, abstraction, and hierarchical decision-making: An information-theoretic optimality principle," *Frontiers in Robotics and AI*, vol. 27, no. 2, pp. 1–24, November 2015.

[28] P. A. Ortega and D. A. Braun, "Information, utility and bounded rationality," in *International Conference on Artifical General Intelligence*, Mountain View, CA, August 3-6, 2011, pp. 269–274.

[29] B. L. Lipman, "Information processing and bounded rationality: A survey," *The Canadian Journal of Economics*, vol. 28, no. 1, pp. 42–67, February 1995.

[30] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *The Allerton Conference on Communication, Control and Computing*, Monticello, IL, September 22-24, 1999, pp. 368–377.

[31] N. Slonim and N. Tishby, "Agglomerative information bottleneck," in *Advances in Neural Information Processing Systems*, Denver, CO, November 29 - December 4, 1999, pp. 617–623.

[32] N. Slonim, "The information bottleneck: Theory and applications," Ph.D. dissertation, The Hebrew University, 2002.

[33] D. Strouse and D. J. Schwab, "The deterministic information bottleneck," *Neural Computation*, vol. 29, no. 6, pp. 1611–1630, June 2017.

[34] S. Hassanpour, D. Wübben, and A. Dekorsy, "Overview and investigation of algorithms for the information bottleneck method," in *International ITG Conference on Systems, Communications and Coding*, Hamburg, Germany, February 6-9, 2017, pp. 1–6.

[35] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, January 1991.

[36] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*. Elsevier Science Publishing Co., 1976.

[37] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

Daniel T. Larsson received the B.S.E degree in aerospace engineering in 2014 and the M.S degree in aerospace engineering in 2016, both from Arizona State University, Tempe, AZ, USA. He is currently working towards the Ph.D degree in aerospace engineering with the Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include information-constrained control, information-limited decision making, as well as planning and decision making in uncertain environments.

Dipankar Maity received the B.E. degree in Electronics and Telecommunication Engineering from Jadavpur University, India in 2013, and the Ph.D degree in Electrical and Computer Engineering from University of Maryland College Park, USA in 2018. During his Ph.D, he was a visiting scholar at the Technische Universität München (TUM) and at the Royal Institute of Technology (KTH) Sweden. Currently, he is a Postdoctoral Fellow at Georgia Institute of Technology. His research interests include temporal logic based controller synthesis, Control with logical constraints, control with communication constraints, intermittent-feedback control, event-triggered control, stochastic games, and integration of these ideas in the context of cyber-physical-systems.

Panagiotis Tsiotras is the David and Andrew Lewis Chair Professor in the D. Guggenheim School of Aerospace Engineering at the Georgia Institute of Technology (Georgia Tech), and the Director of the Dynamics and Controls Systems Laboratory (DCSL) in the same school, as well as Associate Director of the Institute for Robotics and Intelligent Machines at Georgia Tech. He holds degrees in Aerospace Engineering (PhD, MS), Mechanical Engineering (Dipl. Eng.), and Mathematics (MS). He has held visiting research appointments at MIT, JPL, INRIA Rocquencourt, and Mines ParisTech. His research interests include optimal control of nonlinear systems and ground, aerial and space vehicle autonomy. He is the recipient of the NSF CAREER award, the Outstanding Aerospace Engineer award from Purdue, and the Technical Excellence Award in Aerospace Control from IEEE. He is a Fellow of AIAA, IEEE, and AAS.