entropy

Article

A Generalized Information-Theoretic Framework for the Emergence of Hierarchical Abstractions in Resource-Limited Systems

Daniel T. Larsson [1,*], **Dipankar Maity** [2] and **Panagiotis Tsiotras** [3]

[1] D. Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150, USA
[2] Department of Electrical and Computer Engineering, The University of North Carolina at Charlotte, Charlotte, NC 28223-0001, USA; dmaity@uncc.edu
[3] D. Guggenheim School of Aerospace Engineering, Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332-0150, USA; tsiotras@gatech.edu
* Correspondence: daniel.larsson@gatech.edu

Abstract: In this paper, a generalized information-theoretic framework for the emergence of multi-resolution hierarchical tree abstractions is developed. By leveraging ideas from information-theoretic signal encoding with side information, this paper develops a tree search problem which considers the generation of multi-resolution tree abstractions when there are multiple sources of relevant and irrelevant, or possibly confidential, information. We rigorously formulate an information-theoretic driven tree abstraction problem and discuss its connections with information-theoretic privacy and resource-limited systems. The problem structure is investigated and a novel algorithm, called G-tree search, is proposed. The proposed algorithm is analyzed and a number of theoretical results are established, including the optimally of the G-tree search algorithm. To demonstrate the utility of the proposed framework, we apply our method to a real-world example and provide a discussion of the results from the viewpoint of designing hierarchical abstractions for autonomous systems.

Keywords: hierarchical abstractions; hierarchical tree abstractions; information bottleneck method; information-theoretic privacy; multi-resolution abstractions

Citation: Larsson, D.T.; Maity, D.; Tsiotras, P. A Generalized Information-Theoretic Framework for the Emergence of Hierarchical Abstractions in Resource-Limited Systems. *Entropy* **2022**, *24*, 809. https://doi.org/10.3390/e24060809

Academic Editors: Ercan Kuruoglu and Boris Ryabko

Received: 7 March 2022
Accepted: 6 June 2022
Published: 9 June 2022

1. Introduction

Driven by the human ability to discern pertinent details from immense amounts of perceptual information, the process of identifying task-relevant structures from data has long been considered a cornerstone to the development of intelligent systems [1–4]. To this end, researchers in the autonomous systems community have spend a great deal of effort studying abstractions, which is a problem generally viewed as an information-removal procedure to discard details that are not relevant for a given task [1–5]. The central motivation for the employment of abstractions is to simplify the problem domain by removing details that can be safely ignored, thereby creating a new representation of the problem for which reasoning and decision making requires fewer computational resources [1–3]. Despite their importance, autonomous systems thus far seldom design abstractions on their own, instead relying on system designers and prior domain knowledge to provide hand-crafted rules for the emergence of abstractions as a function of task in various domains [3,5]. In spite of these shortcomings, abstractions have seen wide-spread use in a number of autonomous systems applications.

Perhaps the most notable field of research where abstractions have seen particular success is within the planning community [1]. Examples of work that employ the power of abstract representations in planning for autonomous systems include [6–14]. The idea of utilizing abstractions in planning is to form reduced graphs on which classical search algorithms, such as A* and Dijkstra, are implemented. By reducing the number of vertices in the graph, the computational burden of executing these search algorithms is reduced. However, while the cited works all leverage graph abstractions to ease the computational cost of planning, the methods by which they generate these abstractions differ. For example, in [7–11] the environment abstractions are created via the wavelet transform. In contrast, the works of [12–14] generate abstractions of the environment in the form of multi-resolution quadtree and octree data structures. Notably, the work of [12,13] develops a framework that incorporates sensor uncertainty in robotic systems by merging ideas from multi-resolution planning and probabilistic tree structures introduced in [15]. Today, the use of probabilistic trees in robotics is ubiquitous, and has led to the development of open-source software packages for their implementation [16].

Motivated by the possibly dynamic nature of the environment as well as sensing limitations inherent to autonomous systems, the abstractions employed in all the aforementioned works maintain high resolution nearest the autonomous agent (e.g., robotic ground vehicle), while aggregating other portions of the environment at various resolution levels. In this way, the region nearest to the vehicle is considered the most relevant, and thus preserved through the process of abstraction. To strike a balance between path-optimality (system performance) and the computational cost of planning, agents recursively re-plan as they traverse the world.

The design of abstractions has also been considered by information theorists in the context of optimal signal encoding for communication over capacity-limited channels [17]. In order to formulate mathematical optimization problems that yield optimal encoders it is required to identify the relevant structure of the original signal necessary to guarantee that a satisfactory system performance can be achieved. To this end, the framework of rate-distortion theory approaches the optimal encoder problem by measuring the degree of compression via the mutual information between the compressed representation and the original signal, whereas the performance of the system is quantified by a user-provided distortion function [17]. In this way, the distortion function implicitly specifies which aspects of the original signal are relevant, and should be retained, in order to guarantee low distortion. A notable drawback to the rate-distortion framework is, however, the need to specify the distortion function, which may be difficult and non-intuitive for a given task [18].

In contrast, the information bottleneck (IB) method developed in [18] approaches the optimal encoder problem to preserve relevant information more directly. That is, the IB method considers an optimal encoder problem where the degree of achieved compression is captured by the mutual information between the compressed and original signals, and the model quality is measured by the mutual information between the compressed representation and an auxiliary variable which is assumed to contain task-relevant information. The IB approach is entirely data-driven, requiring only the joint distribution of the original signal and relevant information (i.e., the data) in order to be applied.

Owing to its general statistical formulation, the IB method, or some variation of it, has been considered in a number of studies [19–26]. Among these works, reference [19] develops an approach to obtain deterministic encoders to an IB-like problem motivated by reducing the number of clusters in the compressed space as opposed to designing encoders for communication. Consequently, the deterministic IB [19] measures the degree of achieved compression not by the mutual information between the original and compressed representations, as in communication systems, but by the entropy of the reduced space. The work of [20] considers the IB problem with side-information, allowing for both relevant and irrelevant structures to be provided to aid the identification of task-relevant information during the creation of signal encoders. The authors of [21,23] consider a multivariate extension of the IB principle, employing the use of Bayesian networks to specify the compression-relevance relations between the random variables to be maintained through the abstraction process. It should be noted that, while it does not directly employ the IB principle in its formulation, the empirical coordination problem [27,28] considers an information-theoretic compression problem over a graph, where interconnections between vertices represent communication links that agents may use to correlate their sequence of outcomes. Much like the multi-IB method [21,23], the network (communication) topology specifies the statistical dependencies that are possible in the empirical coordination problem. Observe, however, that the objective of the empirical coordination problem is to characterize the set of achievable joint distributions that are possible with various network topologies and communication (code) rates between vertices, whereas the multi-IB problem is a generalization of the encoder-design problem considered by the IB method to multivariate settings where the Bayesian networks are used to specify the relationships between source, reproduction and prediction (relevance) variables.

Other variants of the IB principle include the work in [22], where the authors consider the development of a bottom-up, agglomerative, hard clustering approach that employs the IB objective in determining which clusters to myopically merge at each step of the proposed algorithm. In related work inspired by the AIB problem, the authors of [29] exploit the structure of the AIB merging rule to design algorithms that form compressed representations of images by performing a sequence of greedy merges based on minimizing the stage-wise loss of relevant information at each iteration. Crucially, however, the algorithms developed in [29] do not consider the IB problem as they aim to design a sequence of myopic merges so as to minimize the loss of only relevant information, as compared with the much more challenging IB problem of simultaneously balancing information retention *and* information-theoretic compression. Moreover, in contrast to the work presented in this paper, the algorithms developed in [29] are not accompanied by theoretical performance guarantees that certify the optimality of the abstractions, nor are the methods readily extendable to cases where information from multiple sources must be considered in the design of compressed representations. Finally, the research conducted by the authors of [24] considers the IB problem in the setting of jointly-Gaussian data. More specifically, the authors of [24] established that when the original signal and the auxiliary (relevant) variable are jointly Gaussian, the solution to the IB problem is a noisy linear projection. For completeness, we note that when the data are not jointly Gaussian, or are described by a general probability density function, a solution to the IB problem is difficult to obtain. However, a number of studies have proposed methods leveraging variational inference in order to obtain approximate solutions to the IB problem in these cases [30–33].

Employing a unified viewpoint between abstractions in autonomy and those driven by information-theoretic principles, the authors of [34–37] developed frameworks for the emergence of abstractions in autonomous systems via methods inspired by information-theoretic signal compression. For example, the work of [34] employs the use of the IB principle to generate multi-resolution quadtree abstractions for planning, developing a framework that couples environment resolution, information and path value. Moreover, the research conducted in [35] utilizes environment abstractions to reduce the computational cost of evaluating mutual-information objective functions in active sensing applications. Of the reviewed works, those most closely related to the developments in this paper is that of [36,37], where the authors develop algorithms to select multi-resolution trees that are optimal with respect to the IB objective in both the soft-constrained (Lagrangian) [36] and hard-constrained [37] settings of the IB problem.

Inspired by the recent developments in information-theoretic driven approaches for generating abstractions for autonomous agents for the purposes of planning, the contribution of this paper is the development of a generalized information-theoretic framework that allows for multi-resolution tree abstractions to be obtained when multiple sources of relevance and irrelevance are specified. The incorporation of irrelevant information allows for connections between our framework and notions of information-theoretic privacy. Moreover, our generalized approach allows for abstractions to be refined by removing aspects of the relevant variables that are correlated with the irrelevant information structure, thus allowing for more compressed representations to emerge. This is especially critical in resource-constrained systems, which must make the best use of scarce on-board memory and bandwidth-limited communication channels.

The remainder of the paper is organized as follows. We begin in Section 3 with a brief overview of information-theoretic signal compression and detail the connection between hierarchical trees and signal encoders. Section 4 contains our formal problem statement. We propose and discuss solution approaches in Section 5. In Section 6, we present a discussion and comparison between the information-bottleneck method and the information-bottleneck problem with side-information (IBSI) in the setting of hierarchical tree abstractions. Examples and results are discussed in Section 7 before concluding remarks in Section 8. Proofs for the theoretical results presented in this paper are provided in the appendices.

2. Notation

Let $\mathbb{R}$ denote the set of real numbers and, for any integer $n > 0$, let $\mathbb{R}^n$ denote the n-dimensional Euclidean space. The set of non-negative real numbers is denoted by $\mathbb{R}_+ = \{x \in \mathbb{R} : x \geq 0\}$. For any vector $x \in \mathbb{R}^n$, $[x]_i$ is the ith element of the vector x for $i \in \{1, \dots, n\}$. For any integer $n > 0$, the collection of all non-negative n-dimensional vectors is denoted $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n : [x]_i \geq 0, 1 \leq i \leq n\}$. Given any two vectors $x, y \in \mathbb{R}^n$, the notation $x \leq y$ is understood component-wise; that is $x \leq y$ implies $[x]_i \leq [y]_i$ for all $i \in \{1, \dots, n\}$. Unless otherwise stated, all logarithms are base e.

3. Preliminaries

The development of a framework for generating information-theoretic multi-resolution abstractions requires the introduction of concepts from both information theory and graph theory in order to rigorously define trees and encoder problems. We begin by introducing necessary topics from information theory before proceeding to introduce hierarchical trees and their connection to multi-resolution representations of the environment. In the interest of succinctness, we only introduce the relevant topics from information theory necessary for the developments of our framework, and refer the interested readers to [17,38] for a more comprehensive exposition of information theoretic principles and classical signal compression frameworks. We close this section by elucidating how multi-resolution trees can be viewed as deterministic encoders having a special structure, thereby allowing us to employ information-theoretic concepts from signal encoding theory to formulate the tree abstraction problem we consider in this paper.

Information-theoretic frameworks for compression model signals according to their statistical structure. Consequently, we require the introduction of a probability space. To this end, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with finite sample space Ω, σ-algebra $\mathcal{F}$ and probability measure $\mathbb{P}$. We define the random variables $X : \Omega \to \mathbb{R}$, $Y : \Omega \to \mathbb{R}$, and $T : \Omega \to \mathbb{R}$, where the random variable X has probability distribution (mass function) assigned according to $p(x) = \mathbb{P}(\{\omega \in \Omega : X(\omega) = x\})$, with the mass functions for Y and T defined analogously.

3.1. Mutual Information

Given two distributions $p(x)$ and $\nu(x)$ over the same set of outcomes, the Kullback-Leibler (KL) divergence between the distributions $p(x)$ and $\nu(x)$ is

$$\mathrm{D}_{\mathrm{KL}}(p(x), \nu(x)) \triangleq \sum_x p(x) \log \frac{p(x)}{\nu(x)}. \tag{1}$$

The KL-divergence is non-negative and equals zero if and only if $p(x) = q(x)$ for all outcomes x [17]. The mutual information between two random variables X and T is defined in terms of the KL-divergence as

$$I(T;X) \triangleq \mathrm{D}_{\mathrm{KL}}(p(t,x), p(t)p(x)) = \sum_{t,x} p(t,x) \log \frac{p(t,x)}{p(t)p(x)}. \tag{2}$$

The mutual information is symmetric (i.e., $I(T;X) = I(X;T)$), non-negative, and equals zero if and only if $p(t,x) = p(t)p(x)$. The mutual information plays an important role in signal compression theory where it represents the code rate, or average number of bits per source symbol. Consequently, if T is a compressed representation (or reproduction) of X, then lower values of $I(T;X)$ correspond to greater degrees of achieved compression.

In the more general setting, the mutual information is a measure of the degree of statistical correlation between the random variables X and T, where $I(T;X) = 0$ if and only if X and T are independent. The mutual information also satisfies

$$I(T;X) = H(X) - H(X|T) = H(T) - H(T|X), \tag{3}$$

where $H(X)$ is the Shannon entropy (the Shannon entropy of the random variable X is $H(X) = -\sum_x p(x) \log p(x)$) of the random variable X and $H(T|X)$ is the conditional entropy, measuring the average uncertainty in T when given knowledge of X. Lastly, the Jensen-Shannon divergence between a collection of probability distributions $\{p_1(x), \ldots, p_n(x)\}$ with weights $\Pi \in \mathbb{R}^n_+$ is given by

$$\mathrm{JS}_\Pi(p_1(x), \ldots, p_n(x)) = \sum_{i=1}^{n} [\Pi]_i \mathrm{D}_{\mathrm{KL}}(p_i(x), \bar{p}(x)), \tag{4}$$

where $0 \leq [\Pi]_i \leq 1$ for all $i \in \{1, \ldots, n\}$, $\sum_{i=1}^n [\Pi]_i = 1$ and $\bar{p}(x) = \sum_{i=1}^n [\Pi]_i p_i(x)$ [22,39,40].

3.2. Trees and Trees as Encoders

Our goal in this paper is to leverage information-theoretic signal compression principles in order to generate abstractions for autonomous systems in the form of multi-resolution tree structures. However, existing frameworks for signal encoding, such as rate-distortion theory [17] or the information bottleneck (IB) method [18], do not impose any structural constraints on the resulting encoder in order to guarantee that the solution corresponds to a tree representation. The added constraint poses a significant challenge, as existing methods do not consider such limitations on the set of feasible encoders. To tackle this problem, we will elucidate how trees can be viewed as encoders with a specific structure.

We assume that the environment $\mathcal{W} \subset \mathbb{R}^d$ is a d-dimensional grid-world and that there is an integer $\ell > 0$ such that the environment is contained within a hypercube of side length 2^ℓ. A hierarchical, multi-resolution depiction of $\mathcal{W}$ can be represented as a tree (a tree is a connected acyclic graph [41]) $\mathcal{T} = (\mathcal{N}(\mathcal{T}), \mathcal{E}(\mathcal{T}))$ where $\mathcal{N}(\mathcal{T})$ is a collection of nodes and $\mathcal{E}(\mathcal{T})$ is a collection of edges that describe the nodal interconnections [41]. We will henceforth limit the discussion to the case when the tree structure is that of a quadtree, however it should be noted that the theory developed in this paper applies straightforwardly to general tree structures. Given an environment $\mathcal{W}$, we will take $\mathcal{T}^\mathcal{Q}$ to denote the set of all feasible quadtree representations of $\mathcal{W}$, and let $\mathcal{T}_\mathcal{W} \in \mathcal{T}^\mathcal{Q}$ be the tree whose leafs define the finest resolution depiction of $\mathcal{W}$. An example is shown in Figure 1. In the sequel, we follow the notation and definitions of [36,37]. To this end, we let $\mathcal{N}_k(\mathcal{T}_q)$ denote all the nodes of the tree $\mathcal{T}_q \in \mathcal{T}^\mathcal{Q}$ at depth $k \in \{0, \ldots, \ell\}$, and, for any $t \in \mathcal{N}(\mathcal{T}_\mathcal{W})$, $\mathcal{C}(t)$ will denote the set of children of t. The set of leafs of $\mathcal{T}_q \in \mathcal{T}^\mathcal{Q}$ is given by $\mathcal{N}_{\mathrm{leaf}}(\mathcal{T}_q)$ and the interior node set is $\mathcal{N}_{\mathrm{int}}(\mathcal{T}_q) = \mathcal{N}(\mathcal{T}_q) \setminus \mathcal{N}_{\mathrm{leaf}}(\mathcal{T}_q)$.

Given a formal definition of a tree, we are now ready to discuss the connection between hierarchical trees and signal encoders. To this end, it was noted in [36,37] that hierarchical tree abstractions of $\mathcal{W}$ can be viewed as deterministic encoders having a specific structure. To this end, notice from Figure 1 that by changing the tree $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$ we alter the multi-resolution representation of the environment $\mathcal{W}$. Moreover, any tree $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$ can be created by aggregating finest resolution cells to some parent node in the tree in such a way that the resulting tree is in the space $\mathcal{T}^\mathcal{Q}$.

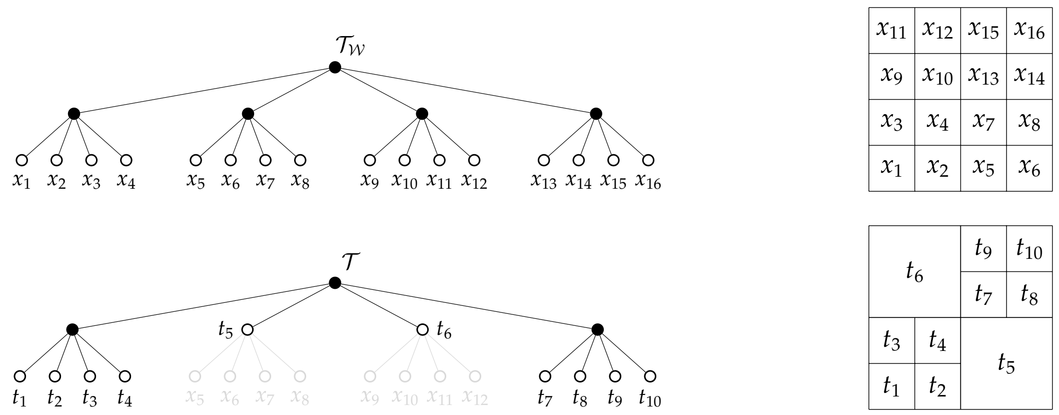

$\mathcal{T}_{\mathcal{W}}$
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14} x_{15} x_{16}
x_{11} x_{12} x_{15} x_{16}
x_9 x_{10} x_{13} x_{14}
x_3 x_4 x_7 x_8
x_1 x_2 x_5 x_6
$\mathcal{T}$
t_5 t_6
t_1 t_2 t_3 t_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12} t_7 t_8 t_9 t_{10}
t_6 t_9 t_{10} t_7 t_8
t_3 t_4 t_5 t_1 t_2

Figure 1. (**top**) The tree $\mathcal{T}_{\mathcal{W}}$ together with 4×4 grid world representation. (**bottom**) Multi-resolution abstraction of the world $\mathcal{W}$ in the form of a quadtree. Notice that the tree $\mathcal{T} \in \mathcal{T}^{\mathcal{Q}}$ is formed by aggregating finest resolution cells that are leafs of $\mathcal{T}_{\mathcal{W}}$ to their parent nodes, which are leafs of $\mathcal{T}$. Aggregated nodes are shown in grey shading. In both figures, black filled nodes are those nodes that are part of the set $\mathcal{N}_{\text{int}}(\cdot)$, whereas nodes with no fill comprise the set $\mathcal{N}_{\text{leaf}}(\cdot)$.

To make the connection to an information-theoretic framework for compression more precise, we let $X : \Omega \to \mathbb{R}$ be the random variable corresponding to the uncompressed signal. In our setting, the uncompressed signal can be the original map of the environment, and therefore the outcomes of X are the finest resolution grid cells of $\mathcal{W}$. For example, in the full-resolution (4×4) environment $\mathcal{T}_{\mathcal{W}}$ depicted in Figure 1, we have $X : \Omega \to \{x_1, \ldots, x_{16}\}$. Notice that each tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ defines a compressed random variable $T_q : \Omega \to \mathbb{R}$ whose outcomes are the elements of the set $\mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$. The relationship between X and T_q can be characterized by a deterministic encoder $p_q(t|x)$ where $p_q(t|x) = 1$ if and only if the finest resolution cell $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$ is aggregated to the node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$ in the tree $\mathcal{T}_q$. However, it is important to note that not all deterministic encoders correspond to valid tree representations of $\mathcal{W}$, which is a challenge we will discuss in the development of our proposed solution approach.

The observation that a tree can be represented as a deterministic encoder, allows us to express information-theoretic quantities as a function of the tree, described next. Consider, for example, the case when a given joint distribution $p(x, y)$ is provided, describing how the finest resolution cells are correlated with a specified random variable Y, which we assume contains task-relevant information. Imagine now that we wish to compress the signal X in the form of a hierarchical tree so that the resulting tree is maximally retentive regarding the relevant variable Y. The resulting joint distribution $p_q(t, x, y)$ can be computed according to $p_q(t, x, y) = p_q(t|x)p(x, y)$, which is a function of the tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$, where we have employed the fact that T_q is conditionally independent of Y when given X. From this, we note that the distributions $p_q(t, y)$, $p_q(t, x)$, $p_q(t)$, $p(x)$ and $p(y)$ can be obtained via the appropriate marginalization of the joint distribution $p_q(t, x, y)$. Therefore, we can write the mutual information as a function of the tree as

$$I_X(\mathcal{T}_q) \triangleq I(T_q; X) = \sum_{t,x} p_q(t, x) \log \frac{p_q(t, x)}{p_q(t)p(x)}, \tag{5}$$

and

$$I_Y(\mathcal{T}_q) \triangleq I(T_q; Y) = \sum_{t,y} p_q(t, y) \log \frac{p_q(t, y)}{p_q(t)p(y)}, \tag{6}$$

where $I_X(\mathcal{T}_q)$ quantifies the degree of compression and $I_Y(\mathcal{T}_q)$ quantifies the amount of relevant information contained in the tree $\mathcal{T}_q$. In this setting, we note that the distributions $p(x)$ and $p(y)$ do not depend on the tree $\mathcal{T}_q$, as they can be obtained directly from the input distribution $p(x, y)$. In an analogous manner, one may also define the amount of irrelevant information, represented by a random variable Z, contained in the tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ as

$$I_Z(\mathcal{T}_q) \triangleq I(T_q; Z) = \sum_{t,z} p_q(t,z) \log \frac{p_q(t,z)}{p_q(t)p(z)}, \tag{7}$$

where we assume that $p(x,y,z)$ is provided and $p_q(t,x,y,z) = p_q(t|x)p(x,y,z)$.

The expressions (5)–(7) may be generalized to the case where we have a collection $\{Y_1,\ldots,Y_n\}$ of relevant and $\{Z_1,\ldots,Z_m\}$ of irrelevant variables, respectively, as follows. Given the joint distribution $p(x,y_1,\ldots,y_n,z_1,\ldots,z_m)$ specifying the correlations between relevant and irrelevant variables, the information of each variable contained in the tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ is given by

$$I_{Y_i}(\mathcal{T}_q) = I(T_q; Y_i) = \sum_{t,y_i} p_q(t,y_i) \log \frac{p_q(t,y_i)}{p_q(t)p(y_i)}, \quad i \in \{1,\ldots,n\}, \tag{8}$$

and

$$I_{Z_j}(\mathcal{T}_q) = I(T_q; Z_j) = \sum_{t,z_j} p_q(t,z_j) \log \frac{p_q(t,z_j)}{p_q(t)p(z_j)}, \quad j \in \{1,\ldots,m\}, \tag{9}$$

where $p_q(t,x,y_1,\ldots,y_n,z_1,\ldots,z_m) = p_q(t|x)p(x,y_1,\ldots,y_n,z_1,\ldots,z_m)$. Having related trees to signal encoders and showing how mutual information terms can be written as a function of the tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$, we now turn to formally state the problem we consider for the remainder of the paper.

4. Problem Formulation

In Section 3, we discussed the relation between trees and signal encoders and showed how the observation that a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ can be represented as a deterministic encoder $p_q(t|x)$ allows us to quantify the information contained in the tree. With these observations, we can now formally state the problem we consider in this paper.

Problem 1. *Given the environment $\mathcal{W}$, vectors $\beta \in \mathbb{R}^n_+$ and $\gamma \in \mathbb{R}^m_+$, a scalar $\alpha \geq 0$ and the joint distribution $p(x,y_1,\ldots,y_n,z_1,\ldots,z_m)$, consider the problem of maximizing*

$$\max_{\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}} \sum_{i=1}^{n} [\beta]_i I_{Y_i}(\mathcal{T}_q) - \sum_{j=1}^{m} [\gamma]_j I_{Z_j}(\mathcal{T}_q) - \alpha I_X(\mathcal{T}_q).$$

It should be noted that Problem 1 cannot be solved by applying existing algorithms (e.g., the Blahut-Arimoto algorithm [17,18] or the iterative IB method [18,39]) from signal encoding theory as the set of feasible solutions is discrete, in addition to the presence of the constraint that $p_q(t|x)$ must correspond to a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$. The added constraint poses significant technical challenges, as it is not obvious how this constraint can be represented mathematically so as to render Problem 1 solvable via numerical methods. Moreover, as a result of the discrete nature of $\mathcal{T}^{\mathcal{Q}}$, it follows that Problem 1 cannot be solved via standard (sub-)gradient approaches from optimization theory, as it belongs to a class of combinatorial optimization problems. Despite these challenges, in the next section we propose a novel and tractable numerical algorithm to find a solution to Problem 1 with theoretical guarantees.

Before proceeding, we provide a few comments regarding the relation of Problem 1 to other areas of research. Namely, Problem 1 is similar to problems considered in the information-theoretic security community [42–44] where $\{Z_1,\ldots,Z_m\}$ are viewed as private variables whose information content we wish not to disclose to an un-trusted party. In this setting, the value of $I_{Z_j}(\mathcal{T})$ represents the amount of private information disclosed by the tree $\mathcal{T} \in \mathcal{T}^{\mathcal{Q}}$ and the vector of weights $\gamma \in \mathbb{R}^m_+$ encodes the relative cost of private information disclosure, allowing for the privacy variables to be distinguished in their importance of revelation. Alternatively, one may interpret the privacy aspects of Problem 1 via conditional entropy. Using (3) and (7), we can write $I_{Z_j}(\mathcal{T}_q) = H(Z_j) - H(Z_j|T_q)$ and note that $H(Z_j)$ is constant, given the data $p(x,y_1,\ldots,y_n,z_1,\ldots,z_m)$. Consequently, performing the maximization in Problem 1 encourages solutions $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ for which $H(Z_j|T_q)$ is as large as possible, amounting to trees that attempt to make Z_j and T_q independent since $H(Z_j|T_q) \leq H(Z_j)$. Then, Fano's inequality ([17], pp. 37–41) implies that the lower bound of the error probability of any estimator designed to infer Z_j from T_q increases as a function of $H(Z_j|T_q)$. It follows that when $H(Z_j|T_q)$ large, the probability of error when estimating the value of Z_j from T_q increases [17,42]. Consequently, information regarding Z_j remains protected.

It should be noted that the incorporation of additional irrelevant variables when designing abstractions has been considered in other works. Previous approaches that introduce irrelevance variables when forming abstractions, such as the IBSI method [20], employ the viewpoint that the information provided via $\{Z_1, \ldots, Z_m\}$ is general task-irrelevant information, with no motivation from an information-theoretic security standpoint. In the IBSI approach, the incorporation of irrelevant information helps improve the quality of abstractions with respect to the task-relevant variable, as aspects of the task-relevant variable that are correlated with the irrelevant information can be discarded when forming the compressed representations. In summary, we note that, while our formulation given by Problem 1 can be interpreted from an information-theoretic security standpoint, the main motivation for our approach is not one of security. Rather, it is the development of a general information-theoretic framework that allows for both relevant and irrelevant information to be specified and balanced versus compression in the design of multi-resolution tree abstractions for autonomous systems. However, as the discussion above shows, the proposed framework could also be useful in obscuring private information contained in quadtree abstractions.

5. Solution Approach

In this section, we discuss an approach to find a solution to Problem 1 and introduce a tractable numerical algorithm that searches for an optimal tree as a function of the weight parameters $\beta \in \mathbb{R}^n_+$, $\gamma \in \mathbb{R}^m_+$, and $\alpha \geq 0$. In what follows, it will be convenient to write the objective of Problem 1 in terms of the function $J : \mathcal{T}^{\mathcal{Q}} \times \mathbb{R}^n_+ \times \mathbb{R}^m_+ \times \mathbb{R}_+ \to \mathbb{R}$, defined by

$$J(\mathcal{T}_q; \beta, \gamma, \alpha) = \sum_{i=i}^{n} [\beta]_i I_{Y_i}(\mathcal{T}_q) - \sum_{j=1}^{m} [\gamma]_j I_{Z_j}(\mathcal{T}_q) - \alpha I_X(\mathcal{T}_q). \tag{10}$$

Then our problem is one of selecting a tree $\mathcal{T}_{\tilde{q}} \in \mathcal{T}^{\mathcal{Q}}$ such that

$$\mathcal{T}_{\tilde{q}} \in \operatorname*{argmax}_{\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}} J(\mathcal{T}_q; \beta, \gamma, \alpha). \tag{11}$$

The evaluation of the objective (10) for a given tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ may be computationally expensive, as it requires the computation of each joint distribution $p(t, y_i)$, $i \in \{1, \ldots, n\}$, and $p(t, z_j)$, $j \in \{1, \ldots, m\}$, as well as the evaluation of the mutual information terms (8) and (9), each of which requires summation over the sample spaces of Ω_{T_q}, Ω_{Y_i} and Ω_{Z_j} for each $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$. Such a computation is especially burdensome if the sample spaces have a large number of elements. Instead, we seek an easier, less computationally costly incremental approach toward evaluating the objective (10) for any $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$.

To this end, we write the objective (10) for any $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ as

$$J(\mathcal{T}_q; \beta, \gamma, \alpha) = J(\mathcal{T}_0; \beta, \gamma, \alpha) + \sum_{u=0}^{q-1} [J(\mathcal{T}_{u+1}; \beta, \gamma, \alpha) - J(\mathcal{T}_u; \beta, \gamma, \alpha)], \tag{12}$$

where $\{\mathcal{T}_0, \ldots, \mathcal{T}_{q-1}\} \subseteq \mathcal{T}^{\mathcal{Q}}$ is a collection of trees in the space $\mathcal{T}^{\mathcal{Q}}$. While the relation (12) is valid for any tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ and any collection $\{\mathcal{T}_0, \ldots, \mathcal{T}_{q-1}\} \subseteq \mathcal{T}^{\mathcal{Q}}$, it was noted in [36,37] that when the tree $\mathcal{T}_0 \in \mathcal{T}^{\mathcal{Q}}$ and the sequence $\{\mathcal{T}_0, \ldots, \mathcal{T}_{q-1}\} \subseteq \mathcal{T}^{\mathcal{Q}}$ is selected in a specific way, the objective (12) reduces to a special form. Specifically, if we select $\mathcal{T}_0 \in \mathcal{T}^{\mathcal{Q}}$ as the tree consisting of a single node where all finest resolution cells are aggregated, and the sequence $\{\mathcal{T}_0, \ldots, \mathcal{T}_{q-1}\} \subseteq \mathcal{T}^{\mathcal{Q}}$ is constructed by expanding a leaf node of $\mathcal{T}_u$ to create $\mathcal{T}_{u+1}$ for $u \in \{0, \ldots, q-1\}$, then (12) can be expressed in terms of the local changes made in moving from the tree $\mathcal{T}_i$ to $\mathcal{T}_{i+1}$. Formally, when the tree $\mathcal{T}_0 \in \mathcal{T}^{\mathcal{Q}}$ is selected to be the root tree (the root tree $\mathcal{R}_{\mathcal{W}}$ is the tree $\mathcal{R}_{\mathcal{W}} \in \mathcal{T}^{\mathcal{Q}}$ such that $\mathcal{N}_{\text{int}}(\mathcal{R}_{\mathcal{W}}) = \varnothing$), and the sequence $\{\mathcal{T}_u\}_{u=0}^{q}$ is constructed so that $\mathcal{N}(\mathcal{T}_{u+1}) \setminus \mathcal{N}(\mathcal{T}_u) = \mathcal{C}(t) = \{t_1', \ldots, t_4'\}$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$ for all $u \in \{0, \ldots, q-1\}$, the objective (12) takes the form

$$J(\mathcal{T}_q; \beta, \gamma, \alpha) = \sum_{s \in \mathcal{N}_{\text{int}}(\mathcal{T}_q)} \Delta J(s; \beta, \gamma, \alpha), \tag{13}$$

where

$$\Delta J(t; \beta, \gamma, \alpha) = \sum_{i=1}^{n} [\beta]_i \Delta I_{Y_i}(t) - \sum_{j=1}^{m} [\gamma]_j \Delta I_{Z_j}(t) - \alpha \Delta I_X(t), \tag{14}$$

and $\Delta I_{Y_i}(t) = I_{Y_i}(\mathcal{T}_{u+1}) - I_{Y_i}(\mathcal{T}_u)$, $\Delta I_{Z_j}(t) = I_{Z_j}(\mathcal{T}_{u+1}) - I_{Z_j}(\mathcal{T}_u)$, $\Delta I_X(t) = I_X(\mathcal{T}_{u+1}) - I_X(\mathcal{T}_u)$ are given by

$$\Delta I_{Y_i}(t) = p(t)\mathrm{JS}_{\Pi}(p(y_i|t_1'), \ldots, p(y_i|t_4')), \quad i \in \{1, \ldots, n\}, \tag{15}$$

$$\Delta I_{Z_j}(t) = p(t)\mathrm{JS}_{\Pi}(p(z_j|t_1'), \ldots, p(z_j|t_4')), \quad j \in \{1, \ldots, m\}, \tag{16}$$

$$\Delta I_X(t) = p(t)H(\Pi), \tag{17}$$

$$p(y_i|t) = \sum_{u=1}^{4} [\Pi]_u p(y_i|t_u'), \tag{18}$$

$$p(z_j|t) = \sum_{u=1}^{4} [\Pi]_u p(z_j|t_u'), \tag{19}$$

$$p(t) = \sum_{u=1}^{4} p(t_u'), \tag{20}$$

$$\Pi = \left[\frac{p(t_1')}{p(t)}, \ \frac{p(t_2')}{p(t)}, \ \frac{p(t_3')}{p(t)}, \ \frac{p(t_4')}{p(t)} \right]. \tag{21}$$

The relations (15)–(21) are computed via direct calculation in terms of the difference in mutual information between two encoders corresponding to the trees $\mathcal{T}_{u+1}, \mathcal{T}_u \in \mathcal{T}^{\mathcal{Q}}$ that satisfy $\mathcal{N}(\mathcal{T}_{u+1}) \setminus \mathcal{N}(\mathcal{T}_u) = \mathcal{C}(t)$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$. Observe that the condition $\mathcal{N}(\mathcal{T}_{u+1}) \setminus \mathcal{N}(\mathcal{T}_u) = \mathcal{C}(t)$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$ implies that the trees $\mathcal{T}_{u+1}$ and $\mathcal{T}_u$ differ only by a single nodal expansion. An example is shown in Figure 2. To show that the term $J(\mathcal{T}_0; \beta, \gamma, \alpha) = 0$ in (12) when the tree $\mathcal{T}_0$ is taken to be the root tree, we note from (3) that $0 \leq I(T_0; Y_i) \leq H(T_0)$, $0 \leq I(T_0; Z_j) \leq H(T_0)$ and $0 \leq I(T_0; X) \leq H(T_0)$. Then, since the root tree has only a single leaf node, it follows that the distribution $p_0(t)$ is deterministic. As a result, $H(T_0) = 0$ and thus $J(\mathcal{T}_0; \beta, \gamma, \alpha) = 0$.

It is important to note that the incremental relations (15)–(21) depend only on the node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$ expanded in moving from the tree $\mathcal{T}_u$ to $\mathcal{T}_{u+1}$, and not on any other nodes in the tree. As a result, the evaluation of the incremental changes in information are dependent only on the changes induced by expanding the node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$, thereby alleviating the need to sum over all the outcomes of the random variable T_u as otherwise required in order to evaluate the mutual information. Furthermore, the observation that the objective and information terms can be decomposed into an incremental form according to (13)–(21) allows for tractable algorithms to be designed in order to obtain a solution to Problem 1. Lastly, it is important to note that there is no loss of generality in using the expression (13). To see why this is the case, we present the following definition.

Definition 2 ([41]). *A tree $\mathcal{G} = (\mathcal{N}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ is a* subtree *of the tree $\mathcal{H} = (\mathcal{N}(\mathcal{H}), \mathcal{E}(\mathcal{H}))$, denoted by $\mathcal{G} \subseteq \mathcal{H}$, if $\mathcal{N}(\mathcal{G}) \subseteq \mathcal{N}(\mathcal{H})$ and $\mathcal{E}(\mathcal{G}) \subseteq \mathcal{E}(\mathcal{H})$.*

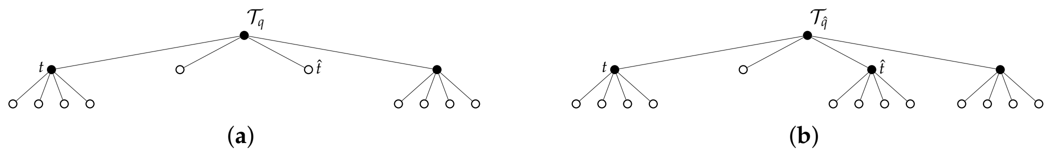


Figure 2. Two trees that differ by only a single leaf node expansion. In moving from tree (**a**) to (**b**), the node $\hat{t}$ is expanded, adding its children as leafs to create the tree shown in (**b**). Interior nodes are shown in black, whereas leaf nodes are white. (**a**) Some tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ for which $\hat{t} \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$. (**b**) The tree $\mathcal{T}_{\hat{q}} \in \mathcal{T}^{\mathcal{Q}}$ which is created by expanding the node $\hat{t} \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$.

Note that the root tree is a subtree of every tree in the space $\mathcal{T}^{\mathcal{Q}}$. As a result, one can always express the cost (10) as (13), since each tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ can be obtained by starting at the root tree $\mathcal{T}_0 \in \mathcal{T}^{\mathcal{Q}}$ and creating a sequence $\{\mathcal{T}_u\}_{u=0}^{q}$ such that $\mathcal{N}(\mathcal{T}_{u+1}) \setminus \mathcal{N}(\mathcal{T}_u) = \mathcal{C}(t)$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$ and all $u \in \{0, \ldots, q-1\}$. Next, we leverage the structure of our problem to design a tractable algorithm in order to find the solution to Problem 1.

5.1. The Generalized Tree Search Algorithm (G-Tree Search)

In this section, we show how the structural properties of Problem 1 discussed in the previous section can be exploited in order to yield a tractable algorithm to find a multi-resolution tree that is a solution to (11). Specifically, among all trees $\mathcal{T} \in \mathcal{T}^{\mathcal{Q}}$, we seek those trees that ensure no improvement of (10) is possible, as these trees provide the best trade-off between relevant information retention, irrelevant information removal, and compression. The following definition establishes the notion of optimality we employ throughout this paper.

Definition 3. *A tree $\mathcal{T} \in \mathcal{T}^{\mathcal{Q}}$ is* optimal with respect to J *if $J(\tilde{\mathcal{T}}; \beta, \gamma, \alpha) \leq J(\mathcal{T}; \beta, \gamma, \alpha)$ for all trees $\tilde{\mathcal{T}} \in \mathcal{T}^{\mathcal{Q}}$.*

To differentiate between candidate solutions, we specify additional properties considered favorable for an optimal multi-resolution tree. One such property is that the tree be minimal, which is defined as follows.

Definition 4. *A tree $\mathcal{T} \in \mathcal{T}^{\mathcal{Q}}$ is* minimal with respect to J *if $J(\tilde{\mathcal{T}}; \beta, \gamma, \alpha) < J(\mathcal{T}; \beta, \gamma, \alpha)$ for all trees $\tilde{\mathcal{T}} \in \mathcal{T}^{\mathcal{Q}}$ such that $\tilde{\mathcal{T}} \subset \mathcal{T}$.*

A tree that is both optimal and minimal will be called an *optimal minimal tree*. Importantly, an optimal minimal tree is guaranteed to not contain any redundant nodal expansions. In other words, removing any portion of an optimal minimal tree is guaranteed to result in a pruned tree that is strictly worse with respect to the objective function. In contrast, if an optimal tree is not minimal, then some portion(s) of the tree can be pruned with no loss in the objective value, indicating that the non-minimal tree contains redundant nodal expansions. Thus, of all optimal trees, the minimal solution is preferred as it contains the fewest number of leaf nodes among solution candidates and also requires the least amount of resources to store in memory. Our goal is then to design an algorithm that returns, as a function of $\beta \in \mathbb{R}_+^n$, $\gamma \in \mathbb{R}_+^m$ and $\alpha \geq 0$, an optimal minimal tree.

In theory, one may take a number of approaches to find a solution (not necessarily optimal) to Problem 1. One approach is the brute-force method of generating each tree in the space $\mathcal{T}^{\mathcal{Q}}$ and picking one that satisfies (11); a process which is akin to grid-search methods in optimization theory. However, such an exhaustive approach does not scale well to large environments. Alternatively, one may notice that the node-wise structure of the cost (14) renders the implementation of a greedy approach straightforward. Specifically, given any tree $\mathcal{T}_u \in \mathcal{T}^{\mathcal{Q}}$ one may expand the leaf node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$ that results in the greatest change in the cost $\Delta J(t; \beta, \gamma, \alpha)$. By expanding a node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$, we remove $\{t\}$ and add its children $\mathcal{C}(t)$ to the leaf set to generate the tree $\mathcal{T}_{u+1}$, leaving other nodes unchanged. One may continue this process until a tree is reached for which no further improvement is possible, as quantified by the one-step incremental objective value $\Delta J(t;\beta,\gamma,\alpha)$. This myopic steepest-ascent-like approach is not guaranteed to find an optimal solution, however, as the process may fail to identify expansions that are suboptimal with respect to the one-step objective $\Delta J(t;\beta,\gamma,\alpha)$, but lead to higher-valued expansions in future iterations. Consequently, we seek to incorporate the value of expansions-to-come when deciding whether or not to expand a leaf node of the current tree.

To this end, we introduce a generalized tree search algorithm we call G-tree search. The G-tree search algorithm works from top-down, starting at the root tree $\mathcal{R}_{\mathcal{W}} \in \mathcal{T}^{\mathcal{Q}}$ and utilizing the function $G : \mathcal{N}(\mathcal{T}_{\mathcal{W}}) \times \mathbb{R}^n_+ \times \mathbb{R}^m_+ \times \mathbb{R}_+ \to \mathbb{R}_+$, defined as

$$G(t;\beta,\gamma,\alpha) = \begin{cases} \max\{\Delta J(t;\beta,\gamma,\alpha) + \sum_{t' \in \mathcal{C}(t)} G(t';\beta,\gamma,\alpha), 0\}, & \text{if } t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}}), \\ 0, & \text{otherwise}, \end{cases} \tag{22}$$

in order to decide which nodes to expand. Specifically, given any tree $\mathcal{T}_u \in \mathcal{T}^{\mathcal{Q}}$, G-tree search will inspect the G-values, computed according to (22), for each node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$ and expand a node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$ for which $G(t;\beta,\gamma,\alpha) > 0$. Once a node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$ is selected for expansion, a new tree $\mathcal{T}_{u+1} \in \mathcal{T}^{\mathcal{Q}}$ is defined by removing the node t and adding its children, $\mathcal{C}(t)$, to the set of leafs, leaving the other nodes in the tree $\mathcal{T}_u$ unchanged. In this way, the tree $\mathcal{T}_{u+1}$ is related to $\mathcal{T}_u$ via $\mathcal{N}_{\text{leaf}}(\mathcal{T}_{u+1}) = (\mathcal{N}_{\text{leaf}}(\mathcal{T}_u) \setminus \{t\}) \cup \mathcal{C}(t)$. The process then repeats until we find a tree $\mathcal{T}_{\tilde{q}} \in \mathcal{T}^{\mathcal{Q}}$ for which there does not exist $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\tilde{q}})$ such that $G(t;\beta,\gamma,\alpha) > 0$. Note that by designing the algorithm in this way, the constraint $\mathcal{T} \in \mathcal{T}^{\mathcal{Q}}$ is naturally enforced. The G-tree search method is detailed in Algorithm 1. Note that the pseudo-code for a greedy tree search is identical to that of G-tree search in Algorithm 1 with each $G(t;\beta,\gamma,\alpha)$ replaced by $\Delta J(t,\beta,\alpha,\gamma)$. We will discuss the shortcomings of the greedy approach in more detail in Section 6.1.

Algorithm 1 The G-tree Search Algorithm.

```
input  : $p(x, y_1, \ldots, y_n, z_1, \ldots, z_m)$, $\beta \in \mathbb{R}^n_+$, $\gamma \in \mathbb{R}^m_+$, $\alpha \geq 0$
output: a tree $\mathcal{T}_{\tilde{q}} \in \mathcal{T}^{\mathcal{Q}}$
$G(t;\beta,\gamma,\alpha) \leftarrow$ ComputeGvalues($p(x, y_1, \ldots, y_n, z_1, \ldots, z_m), \beta, \gamma, \alpha$);
initialize: $\mathcal{N}_{\text{leaf}}(\mathcal{T}_0) = \mathcal{N}_{\text{leaf}}(\mathcal{R}_{\mathcal{W}})$, $u = 0$;
while there exists $t \in \mathcal{N}_{leaf}(\mathcal{T}_u)$ such that $G(t;\beta,\gamma,\alpha) > 0$ do
    Select any node $t \in \{\hat{t} \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u) : G(t;\beta,\gamma,\alpha) > 0\}$;
    Create the tree $\mathcal{T}_{u+1}$: $\mathcal{N}_{\text{leaf}}(\mathcal{T}_{u+1}) \leftarrow (\mathcal{N}_{\text{leaf}}(\mathcal{T}_u) \cup \mathcal{C}(t)) \setminus \{t\}$;
    $(I_{Y_i}(\mathcal{T}_{u+1}), I_{Z_j}(\mathcal{T}_{u+1}), I_X(\mathcal{T}_{u+1})) \leftarrow$ UpdateInformation($t$);
    $u \leftarrow u + 1$;
end
Set $\mathcal{T}_{\tilde{q}} = \mathcal{T}_u$ and return solution;
```

A few comments are in order regarding the G-tree search method. First, the routine `ComputeGvalues`$(\cdot)$ populates the G-values, as follows. The routine utilizes the joint distribution $p(x, y_1, \ldots, y_n, z_1, \ldots, z_m)$ in order to compute the values of $\Delta I_{Y_i}(t)$, $\Delta I_{Z_j}(t)$ and $\Delta I_X(t)$ for all $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$ and $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$. Given the weights $(\beta,\gamma,\alpha) \in \mathbb{R}^n_+ \times \mathbb{R}^m_+ \times \mathbb{R}_+$ one may compute $\Delta J(t;\beta,\gamma,\alpha)$ and apply the rule (22) to obtain the G-values via a recursion that begins at the leafs of $\mathcal{T}_{\mathcal{W}}$. The pseudo-code for the `ComputeGvalues` procedure is shown in Algorithm 2. Lastly, the function `UpdateInformation`(t) updates the information contained in the tree at the current time-step of the solution. It does so by utilizing the values of $\Delta I_{Y_i}(t)$, $\Delta I_{Z_j}(t)$ and $\Delta I_X(t)$ for each $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$, which were computed in the process of evaluating the nodal G-values described above. The information contained in the tree $\mathcal{T}_{u+1}$ is then given by $I_{Y_i}(\mathcal{T}_{u+1}) = I_{Y_i}(\mathcal{T}_u) + \Delta I_{Y_i}(t)$, $I_{Z_j}(\mathcal{T}_{u+1}) = I_{Z_j}(\mathcal{T}_u) + \Delta I_{Z_j}(t)$ and $I_X(\mathcal{T}_{u+1}) = I_X(\mathcal{T}_u) + \Delta I_X(t)$ where $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$. Recall that starting the algorithm at the root tree $\mathcal{T}_0 \in \mathcal{T}^{\mathcal{Q}}$ implies, for all i and j, we have $I_{Y_i}(\mathcal{T}_0) = 0$, $I_{Z_j}(\mathcal{T}_0) = 0$ and $I_X(\mathcal{T}_0) = 0$.

Algorithm 2 The ComputeGvalues routine.

```
input  : $p(x, y_1, \ldots, y_n, z_1, \ldots, z_m)$, $\beta \in \mathbb{R}^n_+$, $\gamma \in \mathbb{R}^m_+$, $\alpha \geq 0$, depth $\ell$.
output: $G(t; \beta, \gamma, \alpha)$ for all $t \in \mathcal{N}(\mathcal{T}_\mathcal{W})$
initialize: $G(t; \beta, \gamma, \alpha) = 0$ for all $t \in \mathcal{N}(\mathcal{T}_\mathcal{W})$;
for $k = \ell - 1$ to 0 do
    for $t \in \mathcal{N}_k(\mathcal{T}_\mathcal{W})$ do
        $\{t'_1, \ldots, t'_4\} = \mathcal{C}(t)$;
        $p(t) = p(t'_1) + \ldots + p(t'_4)$;
        if $p(t) > 0$ then
            $\Pi = \left[\frac{p(t'_1)}{p(t)}, \ldots, \frac{p(t'_4)}{p(t)}\right]$;
            $p(z_j|t) = \sum_{u=1}^4 [\Pi]_u p(z_j|t'_u)$, $p(y_i|t) = \sum_{u=1}^4 [\Pi]_u p(y_i|t'_u)$;
            $\Delta I_X(t) = p(t) H(\Pi)$;
            $\Delta I_{Y_i}(t) = p(t) \mathrm{JS}_\Pi(p(y_i|t'_1), \ldots, p(y_i|t'_n))$;
            $\Delta I_{Z_j}(t) = p(t) \mathrm{JS}_\Pi(p(z_j|t'_1), \ldots, p(z_j|t'_n))$;
        else
            $\Delta I_{Y_i}(t) = 0, \Delta I_{Z_j}(t) = 0, \Delta I_X(t) = 0$;
        end
        $\Delta J(t; \beta, \gamma, \alpha) = \sum_{i=1}^n [\beta]_i \Delta I_{Y_i}(t) - \sum_{j=1}^m [\gamma]_j \Delta I_{Z_j}(t) - \alpha \Delta I_X(t)$;
        $G(t; \beta, \gamma, \alpha) = \max\{\Delta J(t; \beta, \gamma, \alpha) + \sum_{u=1}^4 G(t'_u; \beta, \gamma, \alpha), 0\}$;
    end
end
return $G(t; \beta, \gamma, \alpha)$;
```

5.2. Theoretical Analysis of the G-Tree Search Algorithm

In this section, we discuss the theoretical properties of the G-tree search algorithm introduced in Section 5.1. Our main result is that the G-tree search algorithm returns an optimal minimal tree. In our analysis, we will oftentimes refer to the part of a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ that is descendant (or rooted) at some node $t \in \mathcal{N}(\mathcal{T}_q)$. To make this notion precise, we have the following definition.

Definition 5 ([36]). *Let $t \in \mathcal{N}(\mathcal{T}_q)$ be a node in the tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$.* 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

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

An example of a subtree is shown in Figure 3. Each time the G-tree search visits and expands a node $t \in \mathcal{N}_{\mathrm{int}}(\mathcal{T}_\mathcal{W})$, the algorithm can be viewed as determining the part of the subtree rooted at t for which a net increase in the objective can be achieved. For example, consider the case when the algorithm is provided with a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$. In order to determine whether or not expanding some $t \in \mathcal{N}_{\mathrm{leaf}}(\mathcal{T}_q)$ will lead to a tree of greater objective value than $\mathcal{T}_q$, the algorithm must determine if expanding the node t leads to future expansions that result with a tree $\mathcal{T}_{\tilde{q}} \in \mathcal{T}^{\mathcal{Q}}$ for which $J(\mathcal{T}_{\tilde{q}}; \beta, \gamma, \alpha) > J(\mathcal{T}_q; \beta, \gamma, \alpha)$. Of course, if $\Delta J(t; \beta, \gamma, \alpha) > 0$ for some $t \in \mathcal{N}_{\mathrm{leaf}}(\mathcal{T}_q)$, then it is clear that expanding the node t leads to a tree that improves the value of the objective. However, when $\Delta J(t;\beta,\gamma,\alpha) \leq 0$, the decision of whether or not to expand $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$ is not so clear, as the algorithm must then consider if, by continuing the expansion process along the children of t, can result in a tree that improves of the overall objective value. In essence, we are interested in investigating how the G-function in (22) relates to the incremental objective value of a subtree rooted at any $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ and to show that, if there exists a subtree rooted at t that results in a overall improvement of the objective, then $G(t;\beta,\gamma,\alpha) > 0$. To answer this question, we have the following results.

Lemma 6. *Let $t \in \mathcal{N}_{int}(\mathcal{T}_{\mathcal{W}})$, $\beta \in \mathbb{R}^n_+, \gamma \in \mathbb{R}^m_+$ and $\alpha \geq 0$. Then, $G(t;\beta,\gamma,\alpha) \geq \sum_{s \in \mathcal{N}_{int}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha)$ for all $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$.*

Proof. The proof is presented in Appendix A. □

Corollary 7. *Let $t \in \mathcal{N}_{int}(\mathcal{T}_{\mathcal{W}})$, $\beta \in \mathbb{R}^n_+$, $\gamma \in \mathbb{R}^m_+$ and $\alpha \geq 0$. If there exists a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $\sum_{s \in \mathcal{N}_{int}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha) > 0$ then $G(t;\beta,\gamma,\alpha) > 0$.*

Proof. The result is immediate from Lemma 6. □

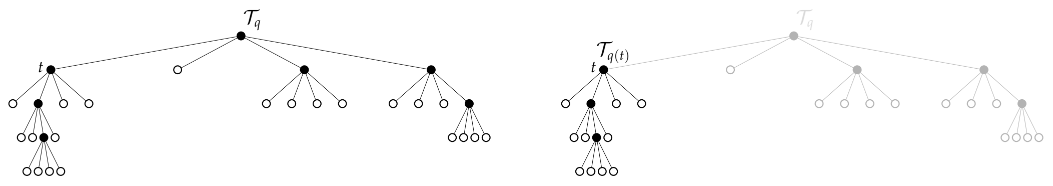


Figure 3. Example of a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ and the subtree $\mathcal{T}_{q(t)}$ rooted at t. (**a**) The tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ together with a node $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_q)$ shown. (**b**) the subtree $\mathcal{T}_{q(t)}$ rooted at t with original tree $\mathcal{T}_q$ shown in background.

As a result of Lemma 6 and Corollary 7, we are guaranteed that, if there is a subtree rooted at t for which an increase in the overall objective is possible, then $G(t;\beta,\gamma,\alpha) > 0$. Furthermore, we are guaranteed that the value of the G-function (22) is bounded below by the incremental value of the objective contributed by any subtree rooted at t. The converse to Lemma 6 and Corollary 7 is also of important; namely if we know $G(t;\beta,\gamma,\alpha) > 0$ for some $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$, then it is of interest in establishing whether or not this implies that there is a subtree rooted at t for which a net increase in the objective is possible. This leads us to the following results.

Lemma 8. *Let $t \in \mathcal{N}_{int}(\mathcal{T}_{\mathcal{W}})$, $\beta \in \mathbb{R}^n_+$, $\gamma \in \mathbb{R}^m_+$ and $\alpha \geq 0$. If $G(t;\beta,\gamma,\alpha) > 0$, then there exists a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $G(t;\beta,\gamma,\alpha) = \sum_{s \in \mathcal{N}_{int}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha)$.*

Proof. The proof is presented in Appendix B. □

Corollary 9. *Let $t \in \mathcal{N}_{int}(\mathcal{T}_{\mathcal{W}})$, $\beta \in \mathbb{R}^n_+$, $\gamma \in \mathbb{R}^m_+$ and $\alpha \geq 0$. If $G(t;\beta,\gamma,\alpha) > 0$, then there exists a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $\sum_{s \in \mathcal{N}_{int}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha) > 0$.*

Proof. The proof follows from Lemma 8. □

Importantly, Lemma 8 establishes the connection between the G-function and the incremental objective value, as well as the existence of a subtree rooted at $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ for which a net positive objective increment is possible, in the case when $G(t;\beta,\gamma,\alpha) > 0$. Moreover, Lemma 8 and Corollary 9 together guarantee that if the G-value of a node is strictly positive, then there exists a subtree rooted at the node $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ such that expanding t (and possibly continuing the expansions process along the children of t) will result in a tree that has strictly greater objective value. Also, observe that by combining the results of Corollaries 7 and 9 we obtain the following lemma.

Lemma 10. *Let $t \in \mathcal{N}_{int}(\mathcal{T}_{\mathcal{W}})$, $\beta \in \mathbb{R}_+^n$, $\gamma \in \mathbb{R}_+^m$ and $\alpha \geq 0$. Then $G(t;\beta,\gamma,\alpha) > 0$ if and only if there exists a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $\sum_{s \in \mathcal{N}_{int}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha) > 0$.*

Proof. The result is a consequence of Corollaries 7 and 9. □

Lemma 10 is important, as it provides necessary and sufficient conditions linking the existence of a subtree rooted at any node $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ to the value of the nodal G-function value. We are now in a position to prove the optimality of the solution returned by G-tree search, as stated by the following theorem.

Theorem 11. *Assume $\beta \in \mathbb{R}_+^n$, $\gamma \in \mathbb{R}_+^m$ and $\alpha \geq 0$. Then, the G-tree search algorithm returns an optimal minimal tree with respect to J.*

Proof. The proof is given in Appendix C. □

As a consequence of Theorem 11, we can guarantee that for any set of parameters $(\beta,\gamma,\alpha) \in \mathbb{R}_+^n \times \mathbb{R}_+^m \times \mathbb{R}_+$, the G-tree search algorithm will return a tree that is the optimal and minimal solution to Problem 1.

5.3. Complexity Analysis

While Theorem 11 establishes that the G-tree search algorithm introduced in Section 5.1 returns an optimal minimal tree that satisfies (11), it is also important to characterize the number of operations required to execute the algorithm, in the worst case. To this end, we note that a tree $\mathcal{T}_{\mathcal{W}} \in \mathcal{T}^{\mathcal{Q}}$ corresponding to some grid in the d-dimensional space with side-length 2^ℓ has $N = \sum_{k=0}^{\ell} 2^{dk}$ total nodes, where $N = 2^{d\ell} \sum_{k=0}^{\ell} 2^{d(k-\ell)} = |\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})|(\sum_{k=0}^{\ell} 2^{d(k-\ell)}) \leq |\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})|(2^d/(2^d-1)) \leq 2|\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})|$ for any integer $d > 0$. As a result, the number of nodes in the tree is on the order of the number of leaf nodes of $\mathcal{T}_{\mathcal{W}}$. Thus, executing the G-tree search algorithm in Algorithm 1 once the G-function is known, requires order $|\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})|$ operations, as the search may visit, in the worst case, every node in the tree. Now, note that for a given number of relevant variables $n > 0$ and irrelevant variables $m \geq 0$, the computation of the G-function requires on the order of $n + m + 2$ operations per node in the tree, corresponding to the calculation of $\Delta I_{Y_i}(t), \Delta I_{Z_j}(t)$ and $\Delta I_X(t)$ and G-function values. Thus, visiting each node in the tree requires on the order of $(n + m + 2)|\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})|$ operations. Consequently, updating the G-values and running the G-tree search requires on the order of $(n + m + 3)|\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})|$ operations in the worst case for a given setting of the problem.

6. The IB and IBSI Principles as Special Cases

In this section, we show how obtaining multi-resolution trees via the information bottleneck (IB) [18] and the information bottleneck with side-information (IBSI) [20] are special cases of Problem 1. Moreover, we establish a relation between these two approaches, showing how a tree solution to the IBSI problem can be obtained from the solution of the IB problem, the latter of which does not consider the removal of irrelevant information from the abstractions. Facilitating a theoretical connection between these two approaches helps us understand the impact of irrelevant information on the resulting tree solutions, and furthermore allows us to incorporate irrelevant information after-the-fact, which is useful in applications when only relevant details are known ahead of time. We begin with a brief overview of the IB and IBSI problems in the context of our problem.

6.1. Multi-Resolution Trees via the IB Principle and Drawbacks of Myopic Tree Search

Recall from Section 3 that the IB problem furnishes an approach to design compressed representations of the original signal that are maximally informative regarding task-relevant information. The IB method was first introduced in [18] and considers the optimization problem

$$\min_{p(t|x)} I(T;X) - \beta I(T;Y), \tag{23}$$

where the minimization is over all conditional distributions $p(t|x)$, and $\beta \geq 0$ trades the importance of compression and relevant information retention. It is important to note that the original formulation of the IB problem does not impose any constraints on the encoder $p(t|x)$ beyond those required to ensure that $p(t|x)$ is a valid probability distribution. However, as discussed in Section 3, multi-resolution trees can be viewed as deterministic encoders that have special structure. Consequently, one may employ the IB principle in order to generate multi-resolution abstractions that compress the environment and retain task-relevant information, which is given by the problem

$$\max_{\mathcal{T}\in\mathcal{T}^{\mathcal{Q}}} I_Y(\mathcal{T}) - \frac{1}{\beta} I_X(\mathcal{T}), \tag{24}$$

where (24) is obtained by multiplying (23) by the constant $-1/\beta$ for $\beta > 0$ and restricting the search to the space $\mathcal{T}^{\mathcal{Q}}$. Recently, the problem (24) was considered by the authors of [36], who introduce an algorithm called Q-tree search that returns, as a function of $\beta > 0$, an optimal solution to (24). Interestingly, the Q-tree search algorithm emerges as a special case of the more general G-tree search method developed in this paper by defining

$$\Delta L(t;\beta) = \Delta J(t;1,0,1/\beta), \tag{25}$$

and by taking

$$Q(t;\beta) = G(t;1,0,1/\beta). \tag{26}$$

As a result, we see that the Q-tree search method in [36] is a special case of G-tree search, where there is only a single relevant variable with unit weight, there is no irrelevant information, and $\alpha = 1/\beta$.

The single variable case also provides some intuition into the differences between G-tree search and an approach that relies on a one-step steepest-ascent (greedy) method to find a solution to (24). The pseudo-code for a greedy tree search is the same as that of G-tree search in Algorithm 1 with each $G(t;\beta,\gamma,\alpha)$ replaced with $\Delta J(t;\beta,\gamma,\alpha)$. Implementing a greedy approach, a visited node is expanded only if the one-step cost $\Delta J(t;\beta,\gamma,\alpha) > 0$. In the single variable case, this means from (25) that

$$\Delta L(t;\beta) = p(t)\left[\mathrm{JS}_\Pi(p(y|t_1'),\ldots,p(y|t_4')) - \frac{1}{\beta}H(\Pi)\right] > 0, \tag{27}$$

or, equivalently (it can be shown that $\Delta L(t;\beta) \to 0$ as $p(t) \to 0$. See ([36], Proposition 1) for more details), $\mathrm{JS}_\Pi(p(y|t_1'),\ldots,p(y|t_4')) - \frac{1}{\beta}H(\Pi) > 0$, where $t_i' \in \mathcal{C}(t)$ for all $i \in \{1,\ldots,4\}$. By changing $\beta > 0$, we alter the preference between trees that represent highly compressed versions of X (low β) and trees that are more informative regarding Y (large β). From (27), we see that the critical value of $\beta > 0$ for the node $t \in \mathcal{N}_{\mathrm{int}}(\mathcal{T}_{\mathcal{W}})$, denoted $\beta_{\mathrm{cr}} : \mathcal{N}_{\mathrm{int}}(\mathcal{T}_{\mathcal{W}}) \to [0,\infty]$, is given by

$$\beta_{\mathrm{cr}}(t) = \begin{cases} \frac{H(\Pi)}{\mathrm{JS}_\Pi(p(y|t_1'),\ldots,p(y|t_4'))}, & \text{if } \mathrm{JS}_\Pi(p(y|t_1'),\ldots,p(y|t_4')) > 0, \\ \infty, & \text{otherwise,} \end{cases} \tag{28}$$

where $\mathcal{C}(t) = \{t_1',\ldots,t_4'\}$. The node-wise critical β-values, $\beta_{\mathrm{cr}}(t)$, determine at what point the node will be expanded, should it be visited by the greedy search algorithm. Namely, if $\beta \leq \beta_{\mathrm{cr}}(t)$ then the node $t \in \mathcal{N}(\mathcal{T}_{\mathcal{W}})$ will not be expanded, even if the algorithm were to visit the node during the search. Moreover, we see from (28) that among those nodes for which $H(\Pi)$ is constant, those that have a greater diversity among the distributions $\{p(y|t'_1),\ldots,p(y|t'_4)\}$ will have lower critical β-values than those nodes with less diversity in the conditional distributions $\{p(y|t'_1),\ldots,p(y|t'_4)\}$. Intuitively, what this means is that nodes that provide more Y-information for a fixed amount of X-information will be expanded at lower values of β compared with those that provide less Y-information (e.g., more homogeneous cells), should these nodes be reached.

On the other hand, (28) also shows why a greedy approach may fail to find an optimal solution, even in the single variable setting. To see why this is the case, consider the 4×4 environment in Figure 4. In this example, $Y:\Omega\to\{0,1\}$ is the relevant variable, where the grid shading shows the distribution $p(y|x)$. The environment is symmetric in the sense that each of the quadrants contains one cell for which $p(y|x)=c$ for some $0<c\leq1$. If we assume $p(x)$ to be a uniform distribution, then when t is taken to be the root node of $\mathcal{T}_{\mathcal{W}}$, one will find that $\mathrm{JS}_{\Pi}(p(y|t'_1),\ldots,p(y|t'_4))=0$, $H(\Pi)>0$, where $\{t'_1,\ldots,t'_4\}=\mathcal{C}(t)$. Consequently, from (28), we have $\beta_{\mathrm{cr}}(t)=\infty$. As a result, a greedy implementation will never expand the root node and one will always recover the trivial abstraction, even though there is relevant information in the environment. In contrast, the G-tree search method does not suffer from this drawback, as it incorporates the reward-to-come of future expansions, thereby allowing it to find trees that recover all the relevant information in the environment for finite values of β.

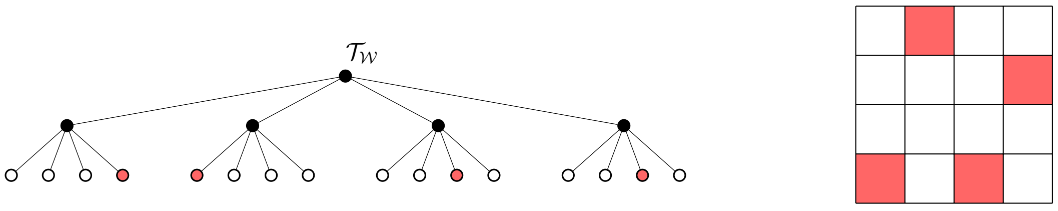


Figure 4. Tree (**left**) and grid (**right**) of a 4×4 example where greedy (myopic) tree search fails. Shading of red scales with $p(y|x)$ (all red shades are equal). Note that, in this environment, parents of leaf nodes contain relevant information, whereas the root node does not, as each of the quadrants is equal in their prediction of Y due to environment symmetry.

6.2. Multi-Resolution Trees via the IB and IBSI Principles

In a similar manner to the IB method considered in Section 6.1, the IBSI approach [20] considers the problem

$$\min_{p(t|x)} I(T;X)-\beta I(T;Y)+\gamma I(T;Z), \tag{29}$$

where the minimization is over all conditional distributions $p(t|x)$, and $\gamma\geq0$ weights the relative importance of relevant information retention and irrelevant information removal. Constrained to the space of multi-resolution tree abstractions, we obtain the IBSI problem over the space of trees given by

$$\max_{\mathcal{T}\in\mathcal{T}^{\mathcal{Q}}} I_Y(\mathcal{T})-\frac{1}{\beta}[\gamma I_Z(\mathcal{T})-I_X(\mathcal{T})]. \tag{30}$$

The IBSI problem over the space of trees is therefore a special case of Problem 1 where the relevant information has unit weight, the irrelevant information has weight γ/β and $\alpha={}^1/_\beta$ for $\beta>0$. In (30), the scalar $\gamma\geq0$ specifies the relative importance of relevant information retention and irrelevant information removal, whereas $\beta>0$ balances the importance of compression. We can solve (30) via G-tree search by defining

$$\Delta M(t;\beta,\gamma)=\Delta J(t;1,\gamma/\beta,1/\beta), \tag{31}$$

and taking the function $S:\mathcal{N}(\mathcal{T}_{\mathcal{W}})\times(0,\infty)\times\mathbb{R}_+\to\mathbb{R}_+$ to be given by the rule

$$S(t;\beta,\gamma) = G(t;1,\gamma/\beta,1/\beta). \tag{32}$$

We will call the special case of the G-tree search algorithm implemented with the G-function in (32) S-tree search. As a result of (26) and (32), we see that multi-resolution trees via the IB or IBSI principles are obtained by employing the G-tree search. Moreover, the resulting abstractions are guaranteed to be minimal and optimal with respect to their objectives, as specified by Theorem 11.

While (26) and (32) show how trees via the IB and IBSI principles can be obtained as special cases of G-tree search, these relations do not provide us with an understanding of how the introduction of irrelevant information changes the solution of the problem. Thus, to better understand the impact of the presence of irrelevant information on the resulting tree abstractions, we present the following results.

Lemma 12. *Let $t \in \mathcal{N}(\mathcal{T}_{\mathcal{W}})$. Then $S(t;\beta,\gamma) \leq Q(t;\beta)$ for all $\gamma \geq 0$ and $\beta > 0$.*

Proof. The proof is presented in Appendix D. □

Corollary 13. *Let $\gamma \geq 0$, $\beta > 0$ and assume $\mathcal{T}_{q_{\mathrm{Q}}^*}, \mathcal{T}_{q_{\mathrm{S}}^*} \in \mathcal{T}^{\mathcal{Q}}$ are the trees returned by Q-tree search and S-tree search respectively. Then $\mathcal{T}_{q_{\mathrm{S}}^*} \subseteq \mathcal{T}_{q_{\mathrm{Q}}^*}$.*

Proof. The proof is presented in Appendix E. □

Corollary 13 essentially states that trees that emerge as a solution to (30) contain no more leaf nodes than those that solve (24), since $\mathcal{T}_{q_{\mathrm{S}}^*} \subseteq \mathcal{T}_{q_{\mathrm{Q}}^*}$. Consequently, for fixed $\beta > 0$, the presence of irrelevant information works to reduce the number of leaf nodes of the resulting abstraction. This is consistent with the original motivation for the inclusion of irrelevant information discussed in [20]. Namely, the purpose of introducing irrelevant information is so as the improve the quality of the abstractions (or reduce the rate of the code in communication systems) by removing the aspects of the relevant information that are correlated with the irrelevance variable Z [20]. In this way, a higher degree of compression can be achieved since we may remove the irrelevant components of the relevant information, a process which is not considered in the IB framework. When applied to multi-resolution tree abstractions, as in our case, this is manifested as a reduction in the number of leaf nodes, as established by Corollary 13.

Having established the influence of irrelevant information on the resulting abstractions, it is also practical to derive an explicit relation between the Q- and S-functions, so that a multi-resolution abstraction that considers irrelevant information can be obtained from the Q-tree search solution. For this result, we define the function $P : \mathcal{N}(\mathcal{T}_{\mathcal{W}}) \times (0,\infty) \times [0,\infty) \to \mathbb{R}$ according to the rule

$$P(t;\beta,\gamma) = Q(t;\beta) - \gamma \Delta I_Z(t) - \sum_{t' \in \mathcal{B}_t^c} Q(t';\beta) + \sum_{t' \in \mathcal{B}_t} \left[P(t';\beta,\gamma) - Q(t';\beta)\right], \tag{33}$$

when $t \in \mathcal{N}_{\mathrm{int}}(\mathcal{T}_{\mathcal{W}})$, where $P(t;\beta,\gamma) = 0$ for $t \in \mathcal{N}_{\mathrm{leaf}}(\mathcal{T}_{\mathcal{W}})$, and $\mathcal{B}_t = \{t' \in \mathcal{C}(t) : P(t';\beta,\gamma) > 0\}$, $\mathcal{B}_t^c = \mathcal{C}(t) \setminus \mathcal{B}_t$.

Lemma 14. *Let $\gamma \geq 0$ and $\beta > 0$. Then $P(t;\beta,\gamma) \leq Q(t;\beta)$ for all $t \in \mathcal{N}(\mathcal{T}_{\mathcal{W}})$.*

Proof. The proof is presented in Appendix F. □

We then have the following result.

Proposition 15. *Let $\gamma \geq 0$, $\beta > 0$. If the function $Q(t;\beta)$ is known for every $t \in \mathcal{N}(\mathcal{T}_{\mathcal{W}})$ then*

$$S(t;\beta,\gamma) = \max\{P(t;\beta,\gamma), 0\}.$$

Proof. The proof is presented in Appendix G. □

Proposition 15 allows us to obtain a multi-resolution tree that incorporates irrelevant information from knowledge of only the Q-function employed by Q-tree search, as well as the values of $\Delta I_Z(t)$ for each $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$, but does not require $\Delta I_X(t)$ or $\Delta I_Y(t)$ for $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$. Furthermore, Proposition 15 allows us to incorporate irrelevant information after initially designing a tree that is maximally task-relevant via the IB principle (24). We now turn our attention to demonstrating the utility of the G-tree search method on a non-trivial numerical example.

7. Numerical Example and Discussion

In this section, we demonstrate the utility of our approach on a real-world example. We consider the image shown in Figure 5a, which is of size 256×256. The image in Figure 5a is then segmented, so that each pixel in the original image is classified into one of six distinct categories; the segmented image together with the original image is shown in Figure 5b. Segmented images such as the one shown in Figure 5b arise frequently in autonomous driving scenarios, where it is of interest to remove irrelevant details from the representation so as to focus available resources on only those aspects of the image that are considered important (e.g., the location of the obstacles or the shape of the road). In the segmented image shown in Figure 5c, we see that the task of maintaining relevant information regarding the road corresponds to retaining the red color while the remaining colors, such light green and yellow, are not relevant to the task of identifying the road and should be removed from the representation.

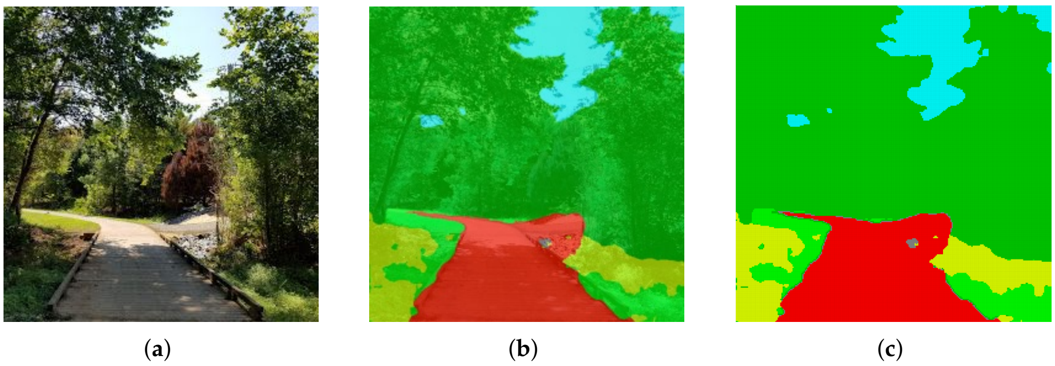


Figure 5. (a) Original 256×256 image. (b) Segmented 256×256 image with original image in background. (c) Segmented 256×256 image passed to G-tree search.

The input data are provided to the G-tree search algorithm as follows. We consider each finest-resolution pixel as an outcome of the uncompressed random variable X. Since the agent may not, in general, have the resources (time, computational, etc.) in order to determine the color (or category) information of each pixel with certainty, we model each color in Figure 5c as a random variable. To this end, for each color in Figure 5c we introduce a random variable, where colors that are assumed to be relevant are denoted as Y_i and those considered irrelevant as Z_j. For example, if we would like to generate abstractions where red and blue are relevant (and therefore should be retained) and yellow as irrelevant (and should be removed), we may define Y_1 to correspond to the category (or color) red and Y_2 to blue, whereas Z_1 may represent yellow.

Strictly speaking, knowledge of the distributions $p(y_i|x)$, $p(z_j|x)$ and $p(x)$ is sufficient to apply our method, as from relations (15)–(21) we see that these distributions allow the determination of $\Delta I_{Y_i}(t)$, $\Delta I_{Z_j}(t)$ and $\Delta I_X(t)$ for all i, j and t. The conditional distributions $p(y_i|x)$ and $p(z_j|x)$ are obtained from the image segmentation step, where $p(y_i|x)$ is the probability that the cell x has the color corresponding to Y_i, with an analogous interpretation for $p(z_j|x)$ and Z_j.In this example, $p(x)$ is assumed to be uniform, although any valid distribution is permissible in our framework (the G-tree search approach can handle any valid distribution $p(x)$ without modification. The use of a non-uniform $p(x)$ will lead to region-specific abstraction, where the G-tree search algorithm refines in regions only where $p(x) > 0$. For more information, the interested reader is referred to [36]). The joint distributions $p(x, y_i)$ and $p(x, z_j)$ are then assigned according to $p(x, y_i) = p(y_i|x)p(x)$ and $p(x, z_j) = p(z_j|x)p(x)$, respectively. In the more general setting where the input is the joint probability mass function $p(x, y_1, \ldots, y_n, z_1, \ldots, z_m)$, the distributions $p(x, y_i)$, $p(x, z_j)$ and $p(x)$ can be obtained via marginalization, and the conditional distributions $p(y_i|x)$ and $p(z_j|x)$ required to compute (15)–(17) are acquired by applying standard rules for conditional probability.

In order to provide a basis for the discussion that follows, we show in Figure 6 a selection of abstractions obtained by trading relevant information and compression (i.e., the IB problem setting) in the case where red is the relevant variable. A number of observations can be deduced from the abstractions in Figure 6. Firstly, it should be noted that G-tree search finds a tree that retains all the available red information and contains only about 1.42% of the nodes of the finest-resolution space. Next, notice that by changing the parameter α, we change the relative importance of compression and information retention. Consequently, at larger values of α, we obtain abstractions that contain less red information but contain fewer leaf nodes (achieve a greater degree of compression) as compared to the abstractions that arise as α is decreased.

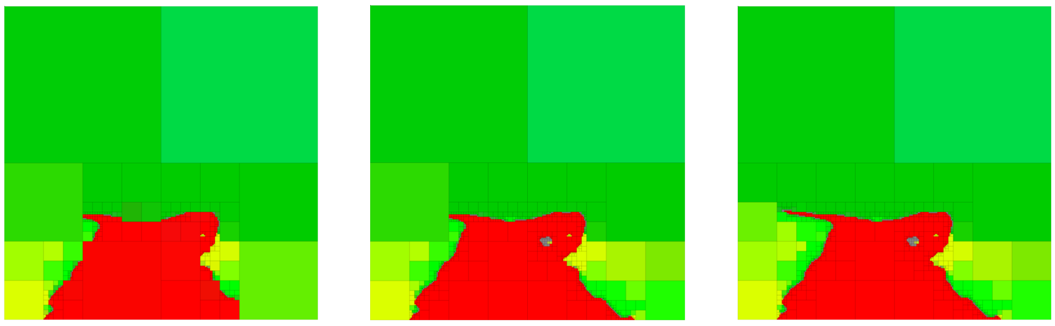

(**a**) (**b**) (**c**)

Figure 6. Sample abstractions obtained via the generalized tree search algorithm by defining the color red as relevant and ignoring other colors. In this scenario, the G-tree search method reduces to the IB problem (see Section 6). We assume the (scalar) weighing parameter $\beta = 1$ and change only α in (10) to generate the abstractions shown. (**a**) $\alpha = 0.14$, $I_Y(\mathcal{T})/I(X;Y) = 0.9346$, 0.647% of leaf nodes. (**b**) $\alpha = 0.05$, $I_Y(\mathcal{T})/I(X;Y) = 0.9874$, 1.23% of leaf nodes. (**c**) $\alpha = 0.01$, $I_Y(\mathcal{T})/I(X;Y) = 1$, 1.42% of leaf nodes.

Furthermore, observe from Figure 6 that regions in the image that contain both no red information and are homogeneous in red color remain aggregated even at high values of α. This occurs for two reasons. Firstly, observe from (15) that if a node $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ has children $\mathcal{C}(t) = \{t'_1, \ldots, t'_4\}$ for which $p(y|t'_1) = \ldots = p(y|t'_4)$ for all y, then $\Delta I_Y(t) = 0$ as $\text{JS}_\Pi(p(y|t'_1), \ldots, p(y|t'_4)) = 0$. Consequently, regions that either contain no red or that are homogeneous in red color contain no relevant information. Intuitively, if a region in the finest resolution is homogeneous in the color red, then no information is lost by aggregating homogeneously-colored finest resolution cells (i.e., given the aggregated cell we can perfectly predict the color of the descendant nodes). Thus, nodes $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ for which all descendant nodes are homogeneous in red color provide no additional relevant information, and thus one can see from (22) that $G(t; \beta, \gamma, \alpha) = 0$ for these nodes. Notice that the reason regions with no or homogeneous relevant information remain aggregated is due to Theorem 11. To see why, consider the scenario when compression is ignored $\alpha = 0$. In this case, regions that contain no relevant information may be expanded at no cost, but would not contribute to an increase in the objective value as seen by relation (13). However, such expansions would lead to a non-minimal tree to be returned by the G-tree search algorithm, which is precluded by Theorem 11. As a result, G-tree search will return the tree with the least number of leaf-nodes that attains the optimal objective function value. This implies that the tree returned by G-tree search maintains regions with no relevant information aggregated.

Next, we generate multi-resolution tree abstractions by employing G-tree search to not only retain relevant information, but also remove information that is considered irrelevant. To this end, we continue our example of considering red as the relevant variable of interest, now letting light green and yellow be irrelevant variables and represented by Z_1 and Z_2, respectively. Example abstractions obtained in this case are shown in Figure 7. Notice that the case shown in Figure 7c corresponds to the standard IB problem with red as relevant and no penalty on compression.

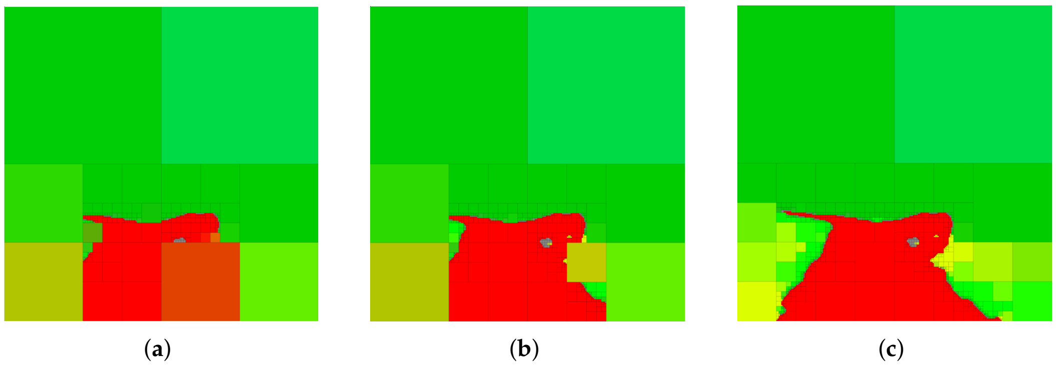
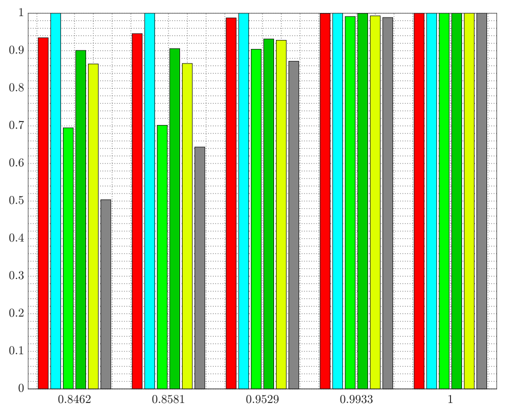

(**a**) (**b**) (**c**)

Figure 7. Multi-resolution trees returned by G-tree search in the case when red is relevant and yellow and light green are considered irrelevant for the environment shown in Figure 5. (**a**) Solution for $\alpha = 0$, $\beta = 1$, $[\gamma]_1 = [\gamma]_2 = 1.2$. (**b**) Solution for $\alpha = 0$, $\beta = 1$, $[\gamma]_1 = [\gamma]_2 = 0.84$. (**c**) Solution for $\alpha = 0$, $\beta = 1$, $[\gamma]_1 = [\gamma]_2 = 0$.

A number of observations can be made from the sample abstractions shown in Figure 7. First, notice that, in comparison with the abstractions shown in Figure 6 which only consider the retention of the color red, the abstractions in Figure 7 aggregate cells along the boundary of red and the irrelevant information (light green and yellow) so as to obscure this information from the abstraction, while being as predictive regarding red (the relevant information) as possible. Moreover, observe that at greater values of the vector γ, regions of yellow and light green are shown in lower resolution as compared with the resolution of these areas at lower values of γ. Notice also that, as the irrelevant information is ignored ($[\gamma]_1 = [\gamma]_2 = 0$), we recover an abstraction (Figure 7c) that is equivalent to the tree in Figure 6c returned by the standard IB approach, which does not consider the removal of the irrelevant information content. To better illustrate the differences between the standard IB case shown in Figure 6 and the generalized tree search scenario in Figure 7, we show the normalized information retained by each color for various settings of the weight parameters in Figure 8. The results shown in Figure 8 are obtained by setting $\beta = 1$, $\alpha = 0$ and by varying the vector $\gamma \in \mathbb{R}^2_+$.

Figure 8 shows the normalized information retained in the solutions returned by G-tree search for two cases: (i) the standard IB problem with red as relevant, and (ii) the generalized tree search with red as relevant and light green as well as yellow as irrelevant. In the standard IB problem, decreasing the value of $\alpha \geq 0$ leads to abstractions that are more informative regarding the relevant information, at the cost of obtaining a tree $\mathcal{T} \in \mathcal{T}^{\mathcal{Q}}$ that achieves a lower degree of compression. Consequently, one moves from right to left in Figure 8 (left) as the value of $\alpha \geq 0$ is increased. In contrast, in the generalized setting of maintaining red information while removing light green and yellow, increasing the weights of the irrelevant information leads to abstractions that achieve more compression, since the importance of information removal increases with larger values of γ. Thus, keeping all other weights constant, we move from right to left in Figure 8 (right) as $\gamma \in \mathbb{R}^2_+$ is increased.



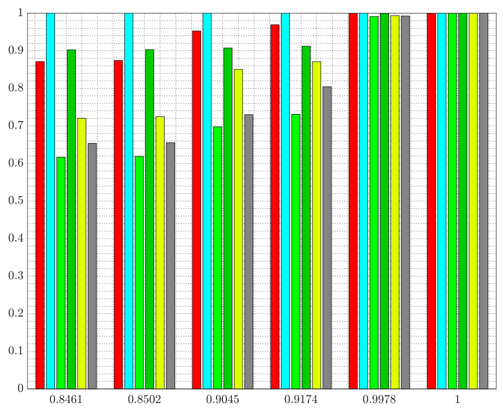


Figure 8. Normalized information retained vs. degree of achieved compression for each color in Figure 5c. Two cases are shown: (i) the standard IB problem with red as relevant (**left**), and (ii) generalized tree search with red as relevant and light green and yellow as irrelevant (**right**). The bar color corresponds with the color in Figure 5c. Data are normalized by the information of each color contained in the tree recovered when executing G-tree search with weights $\beta = 1, \alpha = 0$, and $\gamma = 0$ (i.e., the tree that retains all the relevant information).

We also see from Figure 8 that, compared with the IB tree solutions, the trees obtained from the G-tree search approach in case (ii) retain less information regarding light green and yellow. One may also observe from Figure 8 that when the generalized-tree search algorithm is tasked with retaining red information while removing light green and yellow, less red information is retained. This occurs as the importance of information removal necessitates an abstract representation in order to obscure, or remove, the irrelevant details. However, it is only regions that contain both relevant *and* irrelevant information that are of interest to the algorithm in this case, since regions that contain no relevant information are not refined even in the absence of irrelevant information content. In other words, one may view the relevant information as driving refinement, while irrelevant information promoting aggregation. It is therefore regions that contain both irrelevant and relevant information that becomes the focus of G-tree search. We can observe this trend in the abstractions shown in Figure 7. Specifically, notice that regions not containing any relevant information (i.e., regions with no red) are left unchanged and aggregated in Figure 7a–c. In contrast, when comparing the results of Figure 6, where irrelevant information is not taken into account, to those of Figure 7, we see that the areas containing both relevant and irrelevant information are aggregated as the relative importance of information removal is increased. This occurs for the aforementioned reasons, namely, we must sacrifice some relevant information in order to obscure, or remove, the irrelevant details. At the same time, those regions containing red and no irrelevant colors are maintained with relatively high resolution (e.g., the middle of the image where red boarders with darker green), since these regions contain relevant information with no irrelevant details.

We conclude this section by briefly showcasing the versatility of the G-tree search algorithm to remove redundancies from segmented images. Since the G-tree search algorithm allows any integer number $n \geq 0$ of relevant random variables to be defined, it is possible to allow each color in Figure 5c to be a distinct relevant variable. In this case, G-tree search will find trees for which the distinct colors are as distinguishable as possible while balancing the degree of compression achieved by the abstraction. Interestingly, if one were to take $[\beta]_i = 1$ for all $i \in \{1, \ldots, 6\}$ and $\alpha = 0$, then G-tree search will find a multi-resolution tree that retains all the color information, while removing as much redundancy as possible, as seen in Figure 9. Remarkably, the abstraction in Figure 9 contains only 5% of the nodes of the finest-resolution representation in Figure 5c while retaining *all* the color (semantic) information.

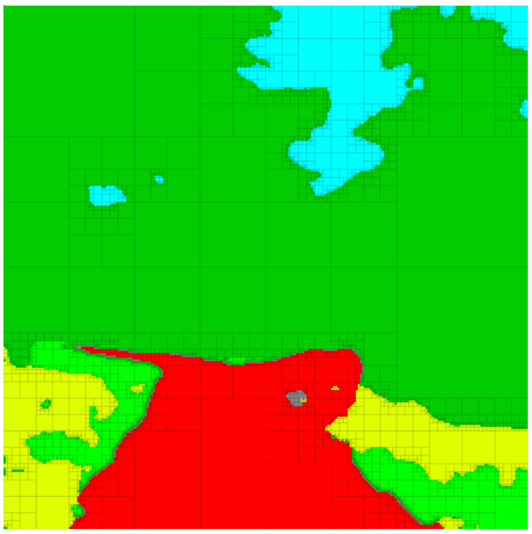


Figure 9. Multi-resolution abstraction of the environment that retains all color information. The image contains only 5% of the nodes compared to the original.

The ability to compress the environment in this way while losing no information regarding the information content of the image represents a drastic savings in the required on-board memory needed to store the depiction of the environment.

8. Conclusions

In this paper, we developed a generalized information-theoretic framework for the emergence of multi-resolution abstractions for autonomous agents. To achieve our goal, we formulated the problem of selecting a multi-resolution tree by considering an objective that aims to maximally retain task-relevant information, while simultaneously removes task-irrelevant, or confidential, information and achieves as much compression as possible. Motivated by its use in signal compression theory, we employ the mutual information in order to measure the degree of achieved compression as well as the amount of relevant and irrelevant information retained in the resulting abstract representation. We rigorously investigate the mathematical properties and structure of the problem and discuss the connections between hierarchical tree abstractions and deterministic signal encoders. Moreover, it is shown that the problem we consider has a special structure, whereby the relevant and irrelevant information contained in a hierarchical tree can be expressed in terms of the incremental information contributions of the non-leaf (interior) nodes of the tree. This special, incremental, structure of the problem facilitates the design of the G-tree search algorithm, which searches over the space of multi-resolution abstractions for a solution that maximizes an information-theoretic objective. We detail our proposed algorithm and prove a number of theoretical results, including that the proposed G-tree search algorithm is guaranteed to return an optimal multi-resolution tree. The complexity of the proposed G-tree search algorithm is analyzed and it is shown that multi-resolution tree abstractions via the information bottleneck (IB) method and the information bottleneck problem with side-information (IBSI) are recovered as special cases of our formulation. A non-trivial numerical example is presented to demonstrate the utility of the proposed approach.

Author Contributions: Conceptualization, D.T.L. and P.T.; methodology, D.T.L. and D.M.; software, D.T.L.; validation, D.T.L.; formal analysis, D.T.L.; investigation, D.T.L.; resources, P.T.; data curation, D.T.L.; writing—original draft preparation, D.T.L.; writing—review and editing, D.T.L., D.M. and P.T.; visualization, D.T.L.; supervision, D.M. and P.T.; project administration, P.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Office of Naval Research awards N00014-18-1-2375 and N00014-18-1-2828 and by Army Research Laboratory under DCIST CRA W911NF-17-2-0181.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A

Proof of Lemma 6. The proof is given by induction. Consider a node $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_{\mathcal{W}})$ and a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$. For the tree $\mathcal{T}_q$ and the node t, there are two options: Either $\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) = \{t\}$ or $\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) = \varnothing$. If $\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) = \{t\}$, then

$$\sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha) = \Delta J(t;\beta,\gamma,\alpha) \le \max\{\Delta J(t;\beta,\gamma,\alpha), 0\} = G(t;\beta,\gamma,\alpha).$$

If, instead, $\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) = \varnothing$, then

$$\sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha) = 0 \le \max\{\Delta J(t;\beta,\gamma,\alpha), 0\} = G(t;\beta,\gamma,\alpha).$$

Assume the result holds for all $t' \in \mathcal{N}_k(\mathcal{T}_{\mathcal{W}})$ and some $k \in \{1,\dots,\ell-1\}$, and consider any node $t \in \mathcal{N}_{k-1}(\mathcal{T}_{\mathcal{W}})$ and tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$. If $\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) \neq \varnothing$, then

$$\begin{aligned}\sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha) &= \Delta J(t;\beta,\gamma,\alpha) + \sum_{t'\in\mathcal{C}(t)} \sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t')})} \Delta J(s;\beta,\gamma,\alpha),\\ &\le \Delta J(t;\beta,\gamma,\alpha) + \sum_{t'\in\mathcal{C}(t)} G(t';\beta,\gamma,\alpha),\end{aligned}$$

where the first step follows from writing the set $\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) = \{t\} \cup \left(\bigcup_{t'\in\mathcal{C}(t)} \mathcal{N}_{\text{int}}(\mathcal{T}_{q(t')})\right)$, and the second step is a result of invoking the induction hypothesis, since $t' \in \mathcal{C}(t) \subseteq \mathcal{N}_k(\mathcal{T}_{\mathcal{W}})$. We then have

$$\begin{aligned}\sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha) &\le \Delta J(t;\beta,\gamma,\alpha) + \sum_{t'\in\mathcal{C}(t)} G(t';\beta,\gamma,\alpha),\\ &\le \max\{\Delta J(t;\beta,\gamma,\alpha) + \sum_{t'\in\mathcal{C}(t)} G(t';\beta,\gamma,\alpha), 0\} = G(t;\beta,\gamma,\alpha).\end{aligned}$$

If $\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) = \varnothing$, then

$$\begin{aligned}\sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha) = 0 &\le \max\{\Delta J(t;\beta,\gamma,\alpha) + \sum_{t'\in\mathcal{C}(t)} G(t';\beta,\gamma,\alpha), 0\},\\ &= G(t;\beta,\gamma,\alpha).\end{aligned}$$

Thus, $\sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha) \le G(t;\beta,\gamma,\alpha)$ for all $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ and any $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$. □

Appendix B

Proof of Lemma 8. The proof is given by induction. First, let $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_{\mathcal{W}})$ be a node such that $G(t;\beta,\gamma,\alpha) > 0$. Next, consider any $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) = \{t\}$. For any such tree, we have

$$\sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha) = \Delta J(t;\beta,\gamma,\alpha).$$

Since $G(t;\beta,\gamma,\alpha) > 0$, it follows from the definition of $G(t;\beta,\gamma,\alpha)$, that

$$G(t;\beta,\gamma,\alpha) = \Delta J(t;\beta,\gamma,\alpha) = \sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha),$$

and thus, there exits a tree $\mathcal{T}_q$ such that $G(t;\beta,\gamma,\alpha) = \sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha)$.

Assume now that the result holds for all $t' \in \mathcal{N}_k(\mathcal{T}_{\mathcal{W}})$ and some $k \in \{1,\ldots,\ell-1\}$, and consider any $t \in \mathcal{N}_{k-1}(\mathcal{T}_{\mathcal{W}})$ for which $G(t;\beta,\gamma,\alpha) > 0$. From the definition of $G(t;\beta,\gamma,\alpha)$, we have that

$$\begin{aligned} G(t;\beta,\gamma,\alpha) &= \max\{\Delta J(t;\beta,\gamma,\alpha) + \sum_{t'\in\mathcal{C}(t)} G(t';\beta,\gamma,\alpha), 0\}, \\ &= \Delta J(t;\beta,\gamma,\alpha) + \sum_{t'\in\mathcal{C}(t)} G(t';\beta,\gamma,\alpha), \end{aligned}$$

since $G(t;\beta,\gamma,\alpha) > 0$. Now, let $\mathcal{U} = \{t' \in \mathcal{C}(t) : G(t';\beta,\gamma,\alpha) > 0\}$. From the induction hypothesis, it follows that there exists a tree $\mathcal{T}_{\hat{q}} \in \mathcal{T}^{\mathcal{Q}}$ such that $G(t';\beta,\gamma,\alpha) = \sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\hat{q}(t')})} \Delta J(s;\beta,\gamma,\alpha)$ for each $t' \in \mathcal{U} \subseteq \mathcal{C}(t)$. Consider then any tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that the subtree rooted at t has interior node set

$$\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)}) = \{t\} \cup \left(\bigcup_{t'\in\mathcal{U}} \mathcal{N}_{\text{int}}(\mathcal{T}_{\hat{q}(t')}) \right).$$

Then,

$$\begin{aligned} G(t;\beta,\gamma,\alpha) &= \Delta J(t;\beta,\gamma,\alpha) + \sum_{t'\in\mathcal{U}} G(t';\beta,\gamma,\alpha), \\ &= \Delta J(t;\beta,\gamma,\alpha) + \sum_{t'\in\mathcal{U}} \sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\hat{q}(t')})} \Delta J(s;\beta,\gamma,\alpha) = \sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha). \end{aligned}$$

Thus, if $G(t;\beta,\gamma,\alpha) > 0$ then there exits a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ such that $G(t;\beta,\gamma,\alpha) = \sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{q(t)})} \Delta J(s;\beta,\gamma,\alpha)$. □

Appendix C

In the following proof, we let $\mathcal{T}_{\tilde{q}} \in \mathcal{T}^{\mathcal{Q}}$ be the tree that is minimal and optimal with respect to J and take $\mathcal{T}_{\text{G}} \in \mathcal{T}^{\mathcal{Q}}$ denote the tree returned by G-tree search. We will show that $\mathcal{T}_{\tilde{q}} = \mathcal{T}_{\text{G}}$ for all $\beta \in \mathbb{R}^n_+, \gamma \in \mathbb{R}^m_+$ and $\alpha \geq 0$.

Proof of Theorem 11. We begin by showing $G(t;\beta,\gamma,\alpha) = 0$ for all $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\tilde{q}})$. Consider the case when $\mathcal{T}_{\tilde{q}} \subset \mathcal{T}_{\mathcal{W}}$ (notice that if $\mathcal{T}_{\tilde{q}} = \mathcal{T}_{\mathcal{W}}$ then $G(t;\beta,\gamma,\alpha) = 0$ for all $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\tilde{q}})$ by definition). The proof is given by contradiction. Assume that there exists a node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\tilde{q}})$ for which $G(t;\beta,\gamma,\alpha) > 0$. Then, from Lemma 10, it follows that there exists a tree $\mathcal{T}_{\hat{q}} \in \mathcal{T}^{\mathcal{Q}}$ such that $\sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\hat{q}(t)})} \Delta J(s;\beta,\gamma,\alpha) > 0$. Now, consider the tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ with interior node set

$$\mathcal{N}_{\text{int}}(\mathcal{T}_q) = \mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}}) \cup \mathcal{N}_{\text{int}}(\mathcal{T}_{\hat{q}(t)}).$$

It then follows that

$$J(\mathcal{T}_q;\beta,\gamma,\alpha) - J(\mathcal{T}_{\tilde{q}};\beta,\gamma,\alpha) = \sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\hat{q}(t)})} \Delta J(s;\beta,\gamma,\alpha) > 0,$$

and thus there exists a tree $\mathcal{T}_{\hat{q}} \in \mathcal{T}^{\mathcal{Q}}$ such that $J(\mathcal{T}_{\hat{q}};\beta,\gamma,\alpha) > J(\mathcal{T}_{\tilde{q}};\beta,\gamma,\alpha)$. However, $\mathcal{T}_{\tilde{q}}$ is optimal, leading to a contradiction. Thus, $G(t;\beta,\gamma,\alpha) = 0$ for all $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\tilde{q}})$. As a result, G-tree search will terminate at the leafs of $\mathcal{T}_{\tilde{q}}$ should it reach them during the expansion process. However, the algorithm may terminate prior to reaching the leafs of $\mathcal{T}_{\tilde{q}}$, and so we conclude $\mathcal{T}_{\text{G}} \subseteq \mathcal{T}_{\tilde{q}}$.

Next, we establish that $G(t;\beta,\gamma,\alpha) > 0$ for all $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}})$. We consider the case when $\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}}) \neq \varnothing$ (if $\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}}) = \varnothing$ then there does not exist $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}})$ such that $G(t;\beta,\gamma,\alpha) = 0$). The proof is given by contradiction. To this end, assume that there exists a node $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}})$ such that $G(t;\beta,\gamma,\alpha) = 0$. Since $t \notin \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\tilde{q}})$ and $\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}}) \neq \varnothing$, it follows that $\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)}) \neq \varnothing$. Now, consider the tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ with interior node set

$$\mathcal{N}_{\text{int}}(\mathcal{T}_q) = \mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}}) \setminus \mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)}).$$

Then, since $\mathcal{T}_q \subset \mathcal{T}_{\tilde{q}}$ and $\mathcal{T}_{\tilde{q}}$ is minimal, it follows that

$$0 < J(\mathcal{T}_{\tilde{q}};\beta,\gamma,\alpha) - J(\mathcal{T}_q;\beta,\gamma,\alpha) = \sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)})} \Delta J(s;\beta,\gamma,\alpha).$$

Consequently, there exists a tree $\mathcal{T}_{\tilde{q}} \in \mathcal{T}^{\mathcal{Q}}$ such that $\sum_{s\in\mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}(t)})} \Delta J(s;\beta,\gamma,\alpha) > 0$. However, this is a contradiction, since $G(t;\beta,\gamma,\alpha) = 0$. Thus, $G(t;\beta,\gamma,\alpha) > 0$ for all $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\tilde{q}})$. As a result, G-tree search will not terminate prior to reaching the leafs of $\mathcal{T}_{\tilde{q}}$, and so $\mathcal{T}_{\tilde{q}} \subseteq \mathcal{T}_{\text{G}}$. We have therefore established that $\mathcal{T}_{\text{G}} \subseteq \mathcal{T}_{\tilde{q}} \subseteq \mathcal{T}_{\text{G}}$. Thus, $\mathcal{T}_{\tilde{q}} = \mathcal{T}_{\text{G}}$. □

Appendix D

Proof of Lemma 12. The proof is given by induction. Let $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_{\mathcal{W}})$. Notice that for node t, $\mathcal{C}(t) \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$. Consequently, $S(t';\beta,\gamma) = 0$ and $Q(t';\beta) = 0$ for all $t' \in \mathcal{C}(t)$. From the definitions of S and Q we then have

$$S(t;\beta,\gamma) = \max\{\Delta M(t;\beta,\gamma), 0\} \leq \max\{\Delta L(t;\beta), 0\} = Q(t;\beta),$$

where the inequality follows from the observation that $\Delta M(t;\beta,\gamma) = \Delta L(t;\beta) - \gamma\Delta I_Z(t) \leq \Delta L(t;\beta)$ as $\gamma\Delta I_Z(t) \geq 0$.

Now assume that the result holds for all $t' \in \mathcal{N}_k(\mathcal{T}_{\mathcal{W}})$, $k \in \{1,\ldots,\ell-1\}$ and consider any $t \in \mathcal{N}_{k-1}(\mathcal{T}_{\mathcal{W}})$. For node t we have, by definition and the induction hypothesis, that

$$\begin{aligned} S(t;\beta,\gamma) &= \max\{\Delta M(t;\beta,\gamma) + \sum_{t'\in\mathcal{C}(t)} S(t';\beta,\gamma), 0\}, \\ &\leq \max\{\Delta M(t;\beta,\gamma) + \sum_{t'\in\mathcal{C}(t)} Q(t';\beta), 0\}, \end{aligned}$$

since $\mathcal{C}(t) \subseteq \mathcal{N}_k(\mathcal{T}_{\mathcal{W}})$. Then, since $\Delta M(t;\beta,\gamma) = \Delta L(t;\beta) - \gamma\Delta I_Z(t) \leq \Delta L(t;\beta)$, we obtain

$$\begin{aligned} S(t;\beta,\gamma) &\leq \max\{\Delta M(t;\beta,\gamma) + \sum_{t'\in\mathcal{C}(t)} Q(t';\beta), 0\}, \\ &\leq \max\{\Delta L(t;\beta) + \sum_{t'\in\mathcal{C}(t)} Q(t';\beta), 0\} = Q(t;\beta), \end{aligned}$$

thereby establishing that $S(t;\beta,\gamma) \leq Q(t;\beta)$. □

Appendix E

Proof of Corollary 13. Notice that Lemma 12 establishes that if, for some $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$, $\beta > 0$, and $\gamma \geq 0$, we have $S(t;\beta,\gamma) > 0$, then $Q(t;\beta) > 0$. As a result, any node $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ expanded by S-tree search is also expanded by Q-tree search. Thus, $\mathcal{T}_{q_{\text{S}}^*} \subseteq \mathcal{T}_{q_{\text{Q}}^*}$. □

Appendix F

Proof of Lemma 14. The proof is given by induction. First consider any $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_{\mathcal{W}})$ and observe that for all such nodes $\mathcal{C}(t) \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$. Thus, by definition, $Q(t';\beta) = 0$ and $P(t';\beta,\gamma) = 0$ for all $t' \in \mathcal{C}(t)$. Therefore,

$$P(t;\beta,\gamma) = Q(t;\beta) - \gamma\Delta I_Z(t) \leq Q(t;\beta),$$

which follows since $\gamma \Delta I_Z(t) \geq 0$.

Assume the result holds for all $t' \in \mathcal{N}_k(\mathcal{T}_\mathcal{W})$ and some $k \in \{1, \dots, \ell - 1\}$. Consider now any $t \in \mathcal{N}_{k-1}(\mathcal{T}_\mathcal{W})$. Then, for the node t, we have by definition

$$P(t;\beta,\gamma) = Q(t;\beta) - \gamma \Delta I_Z(t) - \sum_{t' \in \mathcal{B}_t^c} Q(t';\beta) + \sum_{t' \in \mathcal{B}_t} \left[P(t';\beta,\gamma) - Q(t';\beta)\right].$$

Since $\mathcal{B}_t \subseteq \mathcal{C}(t) \subseteq \mathcal{N}_k(\mathcal{T}_\mathcal{W})$, it follows from the induction hypothesis that

$$P(t;\beta,\gamma) \leq Q(t;\beta) - \gamma \Delta I_Z(t) - \sum_{t' \in \mathcal{B}_t^c} Q(t';\beta),$$

since the hypothesis furnishes that $P(t';\beta,\gamma) - Q(t';\beta) \leq 0$ for all $t' \in \mathcal{N}_k(\mathcal{T}_\mathcal{W})$. From the non-negativity of the Q-function and that $\gamma \Delta I_Z(t) \geq 0$, we arrive at

$$P(t;\beta,\gamma) \leq Q(t;\beta) - \gamma \Delta I_Z(t) - \sum_{t' \in \mathcal{B}_t^c} Q(t';\beta) \leq Q(t;\beta),$$

and thus $P(t;\beta,\gamma) \leq Q(t;\beta)$ for all $t \in \mathcal{N}(\mathcal{T}_\mathcal{W})$. □

Appendix G

Proof of Proposition 15. The proof is given by induction. Consider first any $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_\mathcal{W})$. For the node t, it follows from the definition of $S(t;\beta,\gamma)$, that

$$S(t;\beta,\gamma) = \max\{\Delta M(t;\beta,\gamma), 0\} = \max\{\Delta L(t;\beta) - \gamma \Delta I_Z(t), 0\},$$

since $S(t';\beta,\gamma) = 0$ for all $\mathcal{C}(t) \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_\mathcal{W})$, and $\Delta M(t;\beta,\gamma) = \Delta I_Y(t) - \frac{1}{\beta}\Delta I_X(t) - \gamma \Delta I_Z(t) = \Delta L(t;\beta) - \gamma \Delta I_Z(t)$. Furthermore, notice that, by definition of $P(t;\beta,\gamma)$, we have

$$P(t;\beta,\gamma) = Q(t;\beta) - \gamma \Delta I_Z(t).$$

There are two cases to consider: namely, $Q(t;\beta) > 0$ and $Q(t;\beta) = 0$. In the first case (i.e., $Q(t;\beta) > 0$), we have that $Q(t;\beta) = \Delta L(t;\beta)$. Consequently,

$$\begin{aligned} S(t;\beta,\gamma) = \max\{\Delta L(t;\beta) - \gamma I_Z(t), 0\} &= \max\{Q(t;\beta) - \gamma I_Z(t), 0\}, \\ &= \max\{P(t;\beta,\gamma), 0\}. \end{aligned}$$

For the second case, it follows from Lemma 12 that $S(t;\beta,\gamma) = 0$. Thus,

$$S(t;\beta,\gamma) = 0 = \max\{P(t;\beta,\gamma), 0\},$$

which holds since, by Lemma 14, $P(t;\beta,\gamma) \leq 0$ when $Q(t;\beta) = 0$. Consequently, $S(t;\beta,\gamma) = \max\{P(t;\beta,\gamma), 0\}$ for all $t \in \mathcal{N}_{\ell-1}(\mathcal{T}_\mathcal{W})$.

Assume now that the result holds for all $t' \in \mathcal{N}_k(\mathcal{T}_\mathcal{W})$ and some $k \in \{1, \dots, \ell - 1\}$, and consider any $t \in \mathcal{N}_{k-1}(\mathcal{T}_\mathcal{W})$. For the node t, we have, by definition, that

$$S(t;\beta,\gamma) = \max\{\Delta M(t;\beta,\gamma) + \sum_{t' \in \mathcal{C}(t)} S(t';\beta,\gamma), 0\}.$$

Notice that $\mathcal{C}(t) \subseteq \mathcal{N}_k(\mathcal{T}_\mathcal{W})$, and thus, by the induction hypothesis, we have

$$S(t;\beta,\gamma) = \max\{\Delta M(t;\beta,\gamma) + \sum_{t' \in \mathcal{C}(t)} \max\{P(t';\beta,\gamma), 0\}, 0\}.$$

Next, define the set $\mathcal{B}_t = \{t' \in \mathcal{C}(t) : P(t';\beta,\gamma) > 0\}$, and write

$$
\begin{aligned}
S(t;\beta,\gamma) &= \max\{\Delta M(t;\beta,\gamma) + \sum_{t' \in \mathcal{B}_t} P(t';\beta,\gamma), 0\} \\
&= \max\{\Delta L(t;\beta) - \gamma \Delta I_Z(t) + \sum_{t' \in \mathcal{B}_t} P(t';\beta,\gamma), 0\}, \\
&= \max\{\Delta L(t;\beta) + \sum_{t' \in \mathcal{C}(t)} Q(t';\beta) - \sum_{t' \in \mathcal{C}(t)} Q(t';\beta) - \gamma \Delta I_Z(t) + \sum_{t' \in \mathcal{B}_t} P(t';\beta,\gamma), 0\}.
\end{aligned}
$$

Notice that

$$
\begin{aligned}
&\Delta L(t;\beta) + \sum_{t' \in \mathcal{C}(t)} Q(t';\beta) - \sum_{t' \in \mathcal{C}(t)} Q(t';\beta) - \gamma \Delta I_Z(t) + \sum_{t' \in \mathcal{B}_t} P(t';\beta,\gamma) = \\
&\Delta L(t;\beta) + \sum_{t' \in \mathcal{C}(t)} Q(t';\beta) - \sum_{t' \in \mathcal{S}_t^c} Q(t';\beta) - \gamma \Delta I_Z(t) + \sum_{t' \in \mathcal{B}_t} [P(t';\beta,\gamma) - Q(t';\beta)],
\end{aligned}
$$

where $\mathcal{S}_t^c = \mathcal{C}(t) \setminus \mathcal{B}_t$. We now consider the cases when $Q(t;\beta) > 0$ and $Q(t;\beta) = 0$. If $Q(t;\beta) > 0$ then $Q(t;\beta) = \Delta L(t;\beta) + \sum_{t' \in \mathcal{C}(t)} Q(t';\beta)$, and so we have

$$
\begin{aligned}
&\Delta L(t;\beta) + \sum_{t' \in \mathcal{C}(t)} Q(t';\beta) - \sum_{t' \in \mathcal{C}(t)} Q(t';\beta) - \gamma \Delta I_Z(t) + \sum_{t' \in \mathcal{B}_t} P(t';\beta,\gamma) \\
&= \Delta L(t;\beta) + \sum_{t' \in \mathcal{C}(t)} Q(t';\beta) - \sum_{t' \in \mathcal{S}_t^c} Q(t';\beta) - \gamma \Delta I_Z(t) + \sum_{t' \in \mathcal{B}_t} [P(t';\beta,\gamma) - Q(t';\beta)], \\
&= Q(t;\beta) - \sum_{t' \in \mathcal{S}_t^c} Q(t';\beta) - \gamma \Delta I_Z(t) + \sum_{t' \in \mathcal{B}_t} [P(t';\beta,\gamma) - Q(t';\beta)], \\
&= P(t;\beta,\gamma),
\end{aligned}
$$

where the final equality follows from the definition of $P(t;\beta,\gamma)$. Consequently,

$$
\begin{aligned}
&S(t;\beta,\gamma) = \\
&= \max\{\Delta L(t;\beta) + \sum_{t' \in \mathcal{C}(t)} Q(t';\beta) - \sum_{t' \in \mathcal{C}(t)} Q(t';\beta) - \gamma \Delta I_Z(t) + \sum_{t' \in \mathcal{B}_t} P(t';\beta,\gamma), 0\}, \\
&= \max\{P(t;\beta,\gamma), 0\}.
\end{aligned}
$$

In the second case, when $Q(t;\beta) = 0$, we have from Lemma 12 that $S(t;\beta,\gamma) = 0$. Furthermore, from Lemma 14 we have $P(t;\beta,\gamma) \leq 0$. Therefore,

$$
S(t;\beta,\gamma) = 0 = \max\{P(t;\beta,\gamma), 0\}.
$$

As a result, $S(t;\beta,\gamma) = \max\{P(t;\beta,\gamma), 0\}$ for all $t \in \mathcal{N}(\mathcal{T}_{\mathcal{W}})$. □

References

1. Holte, R.C.; Choueiry, B.Y. Abstraction and reformulation in artificial intelligence. *Philos. Trans. R. Soc. B* **2003**, *358*, 1197–1204. [CrossRef] [PubMed]
2. Zucker, J.D. A grounded theory of abstraction in artificial intelligence. *Philos. Trans. R. Soc. B* **2003**, *358*, 1293–1309. [CrossRef] [PubMed]
3. Ponsen, M.; Taylor, M.E.; Tuyls, K. Abstraction and generalization in reinforcement learning: A summary and framework. In Proceedings of the International Workshop on Adaptive and Learning Agents, Budapest, Hungary, 12 May 2009.
4. Niv, Y. Learning task-state representations. *Nat. Neurosci.* **2019**, *22*, 1544–1553. [CrossRef] [PubMed]
5. Brooks, R.A. Intelligence without representation. *Artif. Intell.* **1991**, *47*, 139–159. [CrossRef]
6. Behnke, S. Local multiresolution path planning. In *RoboCup 2003: Robot Soccer World Cup VII*; Springer: Berlin/Heidelberg, Germany, 2004.
7. Cowlagi, R.V.; Tsiotras, P. Multiresolution motion planning for autonomous agents via wavelet-based cell decompositions. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2012**, *42*, 1455–1469. [CrossRef] [PubMed]
8. Cowlagi, R.V.; Tsiotras, P. Multi-resolution path planning: Theoretical analysis, efficient implementation, and extensions to dynamic environments. In Proceedings of the IEEE Conference on Decision and Control, Atlanta, GA, USA, 15–17 December 2010; pp. 1384–1390.
9. Cowlagi, R.V.; Tsiotras, P. Multiresolution path planning with wavelets: A local replanning approach. In Proceedings of the American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 1220–1225.

10. Tsiotras, P.; Bakolas, E. A hierarchical on-line path planning scheme using wavelets. In Proceedings of the European Control Conference, Kos, Greece, 2–5 July 2007.
11. Tsiotras, P.; Jung, D.; Bakolas, E. Multiresolution hierarchical path-planning for small UAVs using wavelet decompositions. *J. Intell. Robot. Syst.* **2012**, *66*, 505–522. [CrossRef]
12. Hauer, F.; Kundu, A.; Rehg, J.M.; Tsiotras, P. Multi-scale perception and path planning on probabilistic obstacle maps. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015.
13. Hauer, F.; Tsiotras, P. Reduced complexity multi-scale path-planning on probabilitic maps. In Proceedings of the IEEE Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016.
14. Kambhampati, S.; Davis, L.S. Multiresolution path planning for mobile robots. *IEEE J. Robot. Autom.* **1986**, *RA-2*, 135–145. [CrossRef]
15. Kraetzschmar, G.K.; Gassull, G.P.; Uhl, K. Probabilistic quadtrees for variable-resolution mapping of large environments. *IFAC Proc. Vol.* **2004**, *37*, 675–680. [CrossRef]
16. Hornung, A.; Wurm, K.M.; Bennewitz, M.; Stachniss, C.; Burgard, W. Octomap: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robot.* **2013**, *34*, 189–206. [CrossRef]
17. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2006.
18. Tishby, N.; Pereira, F.C.; Bialek, W. The information bottleneck method. In Proceedings of the Allerton Conference on Communication, Control and Computing, Monticello, IL, USA, 22–24 September 1999.
19. Strouse, D.; Schwab, D.J. The deterministic information bottleneck. *Neural Comput.* **2017**, *29*, 1611–1630. [CrossRef]
20. Chechik, G.; Tishby, N. Extracting relevant structures with side information. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 9–12 December 2002.
21. Friedman, N.; Mosenzon, O.; Slonim, N.; Tishby, N. Multivariate information bottleneck. In Proceedings of the Uncertainty in Artificial Intelligence, Seattle, WA, USA, 2–5 August 2001.
22. Slonim, N.; Tishby, N. Agglomerative information bottleneck. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 29 November–4 December 1999.
23. Slonim, N.; Friedman, N.; Tishby, N. Multivariate information bottleneck. *Neural Comput.* **2006**, *18*, 1739–1789. [CrossRef]
24. Chechik, G.; Gloverson, A.; Tishby, N.; Weiss, Y. Information bottleneck for Gaussian variables. *J. Mach. Learn. Res.* **2005**, *6*, 165–188.
25. Estella Aguerri, I.; Zaidi, A. Distributed information bottleneck method for discrete and Gaussian sources. In Proceedings of the International Zurich Seminar on Information and Communication, Zurich, Switzerland, 21–23 February 2018.
26. Aguerri, I.E.; Zaidi, A. Distributed variational representation learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 120–138. [CrossRef] [PubMed]
27. Cover, T.M.; Permuter, H.H. Capacity of coordinated actions. In Proceedings of the IEEE International Symposium on Information Theory, Nice, France, 24–30 June 2007.
28. Cuff, P.W.; Permuter, H.H.; Cover, T.M. Coordination capacity. *IEEE Trans. Inf. Theory* **2010**, *56*, 4181–4206. [CrossRef]
29. Bardera, A.; Rigau, J.; Boada, I.; Feixas, M.; Sbert, M. Image segmentation using information bottleneck method. *IEEE Trans. Image Process.* **2009**, *18*, 1601–1612. [CrossRef] [PubMed]
30. Kolchinsky, A.; Tracey, B.D.; Wolpert, D.H. Nonlinear information bottleneck. *Entropy* **2019**, *21*, 1181. [CrossRef]
31. Alemi, A.A.; Fischer, I.; Dillon, J.V.; Murphy, K. Deep variational information bottleneck. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
32. Chalk, M.; Marre, O.; Tkacik, G. Relevant sparse codes with variational information bottleneck. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 6–10 December 2016.
33. Zaidi, A.; Estella-Aguerri, I.; Shamai, S. On the information bottleneck problems: Models, connections, applications and information theoretic views. *Entropy* **2020**, *22*, 151. [CrossRef]
34. Larsson, D.T.; Maity, D.; Tsiotras, P. Information-theoretic abstractions for planning in agents with computational constraints. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7651–7658. [CrossRef]
35. Nelson, E.; Corah, M.; Michael, N. Environment model adaptation for mobile robot exploration. *Auton. Robot.* **2018**, *42*, 257–272. [CrossRef]
36. Larsson, D.T.; Maity, D.; Tsiotras, P. Q-tree search: An information-theoretic approach toward hierarchical abstractions for agents with computational limitations. *IEEE Trans. Robot.* **2020**, *36*, 1669–1685. [CrossRef]
37. Larsson, D.T.; Maity, D.; Tsiotras, P. Information-theoretic abstractions for resource-constrained agents via mixed-integer linear programming. In Proceedings of the Proceedings of the Workshop on Computation-Aware Algorithmic Design for Cyber-Physical Systems, Nashville, TN, USA, 18 May 2021.
38. Gallager, R.G. *Information Theory and Reliable Communication*; Wiley: New York, NY, USA, 1968.
39. Slonim, N. The Information Bottleneck: Theory and Applications. Ph.D. Thesis, The Hebrew University, Jerusalem, Israel, 2002.
40. Lin, J. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* **1991**, *37*, 145–151. [CrossRef]
41. Bondy, J.A.; Murty, U.S.R. *Graph Theory with Applications*; Elsevier Science: New York, NY, USA, 1976.
42. Nekouei, E.; Tanaka, T.; Skoglund, M.; Johansson, K.H. Information-theoretic approaches to privacy in estimation and control. *Annu. Rev. Control.* **2019**, *47*, 412–422. [CrossRef]

43. Makhdoumi, A.; Salamatian, S.; Fawaz, N.; Médard, M. From the information bottleneck to the privacy funnel. In Proceedings of the IEEE Information Theory Workshop, Hobart, NSW, Australia, 2–5 November 2014.
44. Du Pin Calmon, F.; Fawaz, N. Privacy against statistical inference. In Proceedings of the Allerton Conference on Communication, Control and Computing, Monticello, IL, USA, 1–5 October 2012.