BeliefNet: A neurosymbolic model for context-based traversability predictions in complex environments

Tom Scott, Argyrios Zolotas and Yang Xing
Centre for Autonomous and Cyber-Physical Systems, Cranfield University, UK

Abstract. Knowing how to traverse complex unstructured environments is a difficult and multivariate challenge, but one which humans can achieve through logic, reasoning, and experience; yet some of the most beneficial use cases for autonomous systems require them to operate effectively in complex environments without regular or significant human intervention. Furthermore, for machines to support humans in some of the more critical use cases, trust in decision making will be crucial, ensuring operators have confidence to deploy the capabilities. Despite its importance, enabling autonomous agents to navigate effectively and reliably in complex terrain is a difficult and unsolved challenge. Advances in neurosymbolic AI present an opportunity to significantly enhance performance in complex, explainable, and uncertain decision making, such as autonomous traversability analysis, by drawing together symbolic reasoning with the learning capability of neural networks. The challenge of complex environments is complicated by its non-deterministic nature; terrain will adapt and change through domains, and its properties can adapt rapidly based on external factors like weather or objects that are in proximity, which is true for one location on one day, will not persist. This paper presents a new neurosymbolic model structure that was designed specifically for this task. It uses experience to build a world model, similar to that of a neural network, but with some key delineating features such as full explainability, through life adaptation or evolution, and zero-shot capability, enabling it to perform as both a reasoning engine and a memory representation for an autonomous system. This provides the reasoning backbone for an autonomous agent to determine the level of risk each object presents based on its context and therefore determine the best possible route.

Keywords: Neurosymbolic AI, Machine learning, Knowledge-based learning, Autonomous systems, Complex environments

1. Introduction

Autonomous systems present an opportunity to transform the way humans complete some of the most dangerous, unpleasant, or persistent tasks, especially within domains such as Defence or Search and Rescue. These use cases present some of the greatest beneficiaries of autonomous systems, but have some of the most demanding requirements, most notably the ability to operate reliably in very complex terrain and dynamic domains, whilst maintaining a high degree of trust by their operators to complete their task at hand. Robustly operating in complex environments requires platforms to operate in both unstructured and uncertain terrain, where clear transition points between features may not exist, with high variation in slope, roughness and unpredictable terrain features like holes or depressions [67], [70], [71]. Furthermore, the characteristics of an object cannot be determined effectively without understanding the context in which it is found. Navigating requires inductive and deductive reasoning, an understanding of the environmental conditions, probabilistic judgement, and the ability to handle uncertainty. When considering autonomous agents, neither a symbolic nor a neural approach replicates them all sufficiently. Neural approaches generally fail to reason effectively and suffer from a lack of explainability but can be adaptive to out-of-distribution data, whilst symbolic approaches can reason but require a significant upfront knowledge base and

cannot effectively generalise. Fundamentally, performing these activities within an autonomous platform is not a simple extrapolation of either approach. The traversal of complex environments remains an outstanding challenge in the field of autonomous systems [17], [74].

Navigating complex terrain can be considered across a number of fields of research, such as perception, localisation, cognition and motion control [57]. This paper focuses on cognition and specifically on how to enable an agent to determine the traversability of a target object. When considering the prediction of traversability for a given object within complex environments, the continuous, layered structure of individual objects means that the assessment of an object in isolation is insufficient for making an accurate prediction. The images in Figure 1 show an example of two separate examples of the same trail object, both of which have separate performances, caused as a result of the adjacent objects. Furthermore, terrain characteristics are not consistent in all domains and environments; the performance of grass changes if it has rained in the last hour, and in winter, this may be the last day. As a result, such environmental information is the context and parameters of the overall prediction. Consideration must also be made to the rapid domain evolutions and inconsistencies required when operating within such an environment, meaning traversability predictions must have a broad generalisation capacity, enabling routine handling of previously unseen situations.

Vehicles will operate in close proximity to humans who, for both safety and functional reasons, need confidence in actions and to understand why a decision is to be made. As a result, operator trust must be considered within any agent cognition. This constraint makes any potential for context prediction within a conventional neural network-based solution [30], [77], [20], [76] a barrier to practical deployment, as any decision justification would be concealed within the black-box nature of the model, inhibiting explainability. Conversely, explicit reasoning and logic are easier to interpret than neural architectures [78], as reasoning transcends the originator; it is communicable and can be understood externally [52]. Through explicit reasoning, operators can interpret decisions and understand errors, making the prediction more deterministic and increasing trust [26].

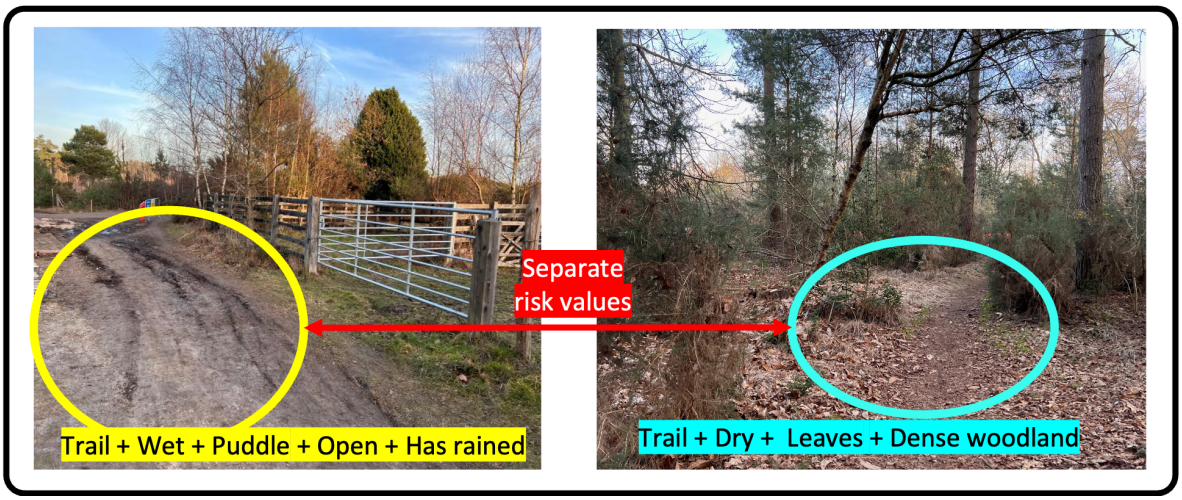


Fig. 1. An example of two instances of trail objects with their surrounding context, resulting in separate risk assessments

This concept paper builds on the concept of the World Model [42], using neurosymbolic methods [34] to develop a human-like approach to solving the challenge of autonomy in complex environments. It presents a model structure which enables an agent to make traversability predictions that account for an object's context, learn dynamically with new experiences, and use causal relationships to generalise across evolving domains. This paper outlines BeliefNet, a model designed to support explainable context-based prediction for complex environments.

BeliefNet uses a symbolically built neural architecture to form experience-based beliefs, overtime generating causal relationships between a target object, its context and traversability risk. The model is designed to train through life, learning with an agent's experience, enabling adaptation to new environments. The model seeks to extrapolate

causal relationships, enabling it to generalise effectively to different domains. It uses belief-based inference to form deterministic and explainable predictions, improving prediction accuracy, domain adaptation, and enhancing operator trust.

The contributions of this paper are as follows.

- The proposal of BeliefNet, a Neuro[Symbolic] model structure for context-based traversability prediction for autonomous systems.
- A demonstration of the performance of BeliefNet in an adapted version of the Yamaha CMU [79] dataset, to increase the performance of agent cognition in complex environments.
- Comparison of context-based terrain traversability prediction and object-based prediction.
- A high-level traversability taxonomy for ground platforms based on risk and speed.

2. Existing work

2.1. Traversability assessment

The field of traversability has received significant attention in recent years, leading to the development of three primary approaches emerging to conduct traversability assessment: terrain mapping, terrain classification, and end-to-end solutions [6]. Lidar analysis has been used extensively in traversability mapping approaches, both in direct obstacle avoidance [72][40], or in more complex feature segmentation [24] [2] [80]. Whilst delivering promising results, such approaches are spatial in nature, *potentially over simplifying the traversability calculus by ignoring the environmental and situational semantics*. Furthermore, the active nature of Lidar presents challenges in use cases where light emissions have negative secondary effects.

Terrain classification presents a method to incorporate semantics. Advances in computer vision, with the introduction of models such as YOLO [63] and approaches such as vision transformers [16] and panoptic segmentation [81], have made this increasingly feasible, allowing real-time inference on edge-based devices. The use of computer vision enables terrains to be segmented into constituent objects, from which semantic labels and classes can be subsequently assigned. As the terrain classification of complex environments is non-trivial, resulting from the discontinuous nature of objects, feature overlap, and environmental conditions [49], this continues to be an area of active research [12], [18], [25], [11], [75] [2], [71], [19]. Vision and Lidar modalities have been combined to integrate visual semantics with the spatial representations of Lidar [51].

Terrain classification is formed of two distinct components, first detecting and isolating a specific object within the scene, then assessing the traversability of the object. Although some, such as [2] have integrated both components, most of the research focuses mainly on accurately determining the object, not assessing the traversability. One challenge in this approach is that it can *neglect the need to consider the environment and context of a specific object*, ignoring that some objects will directly impact the traversability of others. Without such context, it can be challenging to make an accurate and reasoned determination, which is exacerbated as the complexity of the environment increases.

End-to-end deep learning approaches have had success in classifying the traversability of an image [77], [43]. Self-supervised approaches, in which a platform trains a model based on self-extracted features to predict the traversability of the terrain [66] [68], reduce the volume requirements for labelled data to some degree. Such approaches can be limited in generalisation performance and crucially limit explainability due to the conventional neural architecture of end-to-end deep learning approaches. The use of image segmentation, coupled with self-supervised learning, presents a method to increase explainability, but the *computational requirements are prohibitive and the explicit impact of object context is not explicit and unclear* [30]. Although research into traversability assessment for complex environments has been significant, it remains an open area of research and one in which significant advances are required to enable autonomous systems to complete the desired tasks.

2.2. Agent cognition

Agent cognition within the field of robotics to model actions with high levels of uncertainty has had significant success using probabilistic methods. Markov Logic Networks (MLN) in event modelling have been used successfully to classify events from images based on their context [33], but suffer in complex environments due to cross-domain adaptation, handling partial rule activation, and probabilistic complexity [13], [35]. Markov Decision Processes (MDP) have also been used in path planning and path trajectory, with Partially Observable MDPs (POMDP) used to handle increasing levels of uncertainty and complex environments [41]. Belief states have been integrated into POMDP to extend planning horizons and mitigate the impact of partial observability [31], [21]. Applications in robotics and autonomous systems focus primarily on action selection. In which state estimation is used to determine the best action to take across a given planning horizon, formulation of the reward for a given action (such as its predicted traversability value) is often adjacent to the model. Probabilistic programming using tools such as ProbLog has sought to overcome the complexity limitations faced by MDPs and MLNs, such as probability modelling and handling uncertainties in predictions [56], [73], but *requires a firm logical foundation, which may not be easily available* in complex environments.

The concept of a belief within Bayesian epistemology considers that beliefs are not consistent, and the degree to which a belief is believed is adapted to the available body of evidence [45]. Pearl emphasises the centrality of causality in beliefs, expanding beyond correlation [58]. Pearl introduces a three-layer model that supports causality in the concept of machine learning: association, intervention, and counterfactual; progression through these layers supports the classification of causal information and improves the degree of confidence in a given belief [59]. Graph structures have presented a method to effectively communicate causal relationships between entities, noting that such relationships are not hierarchical or linear [46]. The application of causality to machine learning has recently been identified as a method to significantly increase generalisation and cross-task adaptation. It presents a complex challenge due to the nature of feature extraction from data. Scholkopf et al. presented a number of potential approaches, such as self-supervised learning and reinforcement learning. This research outlined the importance of observation and intervention in learning causal relationships [65]. The applications of causality have been applied to learn causal relationships and apply them within inference [82], [62], [84]. Causality within AI presents significant promise, though limited within this challenge by the data volumes required by approaches such as autoencoders. The application of a priori knowledge to autoencoders has been applied by Komanduri et al. as a method of reducing upfront data requirements [38].

Advances in probabilistic modelling and decision making have had a significant impact on robotics, but the high levels of variability in object class, prediction confidence, object separation, and adaptive domains seen in complex environments make logical grounding and the application of finite rule sets insufficient to effectively model the complexity. The consideration of belief states and partial observability is very relevant to context-aware traversability. The state estimation is considered not as the state of the agent given an action, but estimation of the state of beliefs held about an object, it's context, and the resulting traversability. The ability to make predictions with partially observed inputs and an incomplete understanding of object interactions, whilst being able to update understanding when new information is available, is very pertinent to complex environments and domain adaptation. The concepts of beliefs and causality are also of benefit to this problem set, supporting an agent to both *learn from new experiences, and supporting generalisation when facing uncertainty*, both of which are common in complex environments.

2.3. Neurosymbolic AI

Neurosymbolic AI is a promising area of research in machine decision making, explainability, and reasoning [8]. This area of research presents architectures to integrate the reasoning performance of symbolic reasoning and the learning power of sub-symbolic, connectionist or neural network-based approaches [34], based on the system-1 / system-2 approach defined by Kahneman [32]. The field is still growing and there remains diversity in approaches, but all have in common the structure of perception, integrated with existing knowledge [69], and their explainability and reasoning performance make them particularly beneficial in use cases with high levels of human-machine interaction [4]. Within the broader field of autonomous system navigation, neurosymbolic architectures have been used

to integrate physics rules into a neural network to determine the vehicle dynamics required to traverse a given path [85]. Neurosymbolic architectures are commonly represented by six core approaches [15], [34], two of particular relevance to this article are defined as NEURO;SYMBOLIC and NEURO[SYMBOLIC] [37].

NEURO;SYMBOLIC represents a system in which a *symbolic and neural systems work in concert with each other*, communicating and passing information between them, to achieve a common objective [34]. Examples of this are knowledge graph integration with neural networks [15] that allow a neural network to query, input to, and validate symbolic knowledge graphs. NSNnet, which passes between neural and symbolic modules in an attempt to solve hand-written Sudoku challenges, presents a unique perspective that maps both input and output to a non-symbolic output, with a central symbolic reasoning engine [1]. Both examples are dependent on a core level of symbolic reasoning. The Neuro-Symbolic Concept Learner (NCSL) is designed to unify text and visual concepts through the learning of image and question-answer pairs [50]. This model presents an interesting advance as it enables symbolic concepts to be learned, without implicit knowledge being defined upfront.

In contrast, the NEURO[SYMBOLIC] system is one in which a *neural network learns to reason about relationships between neural entities [34] [39], in effect forming a neural network of symbolic entities*. This is perhaps the most complex and least mature area of research within the field. Logic Tensor Networks (LTN) and Logic Neural Networks (LNN), which form networks from symbolic relationships and enable weighted training of the relationships using back-propagation based on a set of first-order logic statements [3] [64]. The pLogicNet model mostly precedes the core definitions of neurosymbolic AI and represents a method similar to Logic Tensor Networks based on the application of Markov Logic Networks [61]. The LTN and pLogicNet are designed to improve, validate, or deconflict a set of a priori logical statements. The challenge with these approaches when applied to an agent-based approach is that they require upfront knowledge that may not be practical to achieve. Models such as the Neuro Symbolic Reinforcement Learner, INSIGHT, by Luo et al. use a neural network to learn symbolic policies that support the agent in its decision-making, enabling reasoning to be learned from the environment [48].

Neurosymbolic systems have shown significant promise in vision and multimodal tasks, such as visual question answering and scene graph generation [44], [5], [29]. Despite success in these areas, multimodal neurosymbolic systems remain challenged in *ensuring consistency between modalities* as the deployment domain evolves [47]. From an autonomous system perspective, this might have the most impact at the point that multiple sensors are combined into a single collaborative neurosymbolic architecture, such as the camera and Lidar, as shown previously, a common approach. Although this remains an open challenge, it could *limit the breadth of neurosymbolic architecture application in autonomous vehicles.*

The current state of neurosymbolic AI presents significant advances in both reasoning and explainability, the NEURO[SYMBOLIC] concept of a single neural network which encapsulates symbolic reasoning presents an opportunity to represent an agent's world model. As with probabilistic approaches, current methods often rely on a set of a priori logic statements, leading to similar constraints on domain adaptation. As a result, BeliefNet has taken the concept of a symbolic network trained using sub-symbolic approaches, but in a manner that reflects the domain learning capabilities of models such as INSIGHT or the NSCL in which beliefs can be inferred from training. In the generation of beliefs, in opposition to rules, BeliefNet provides the ability to learn continuously from an agent's experience, avoiding the constraint of domain-specific logic that fails to support out-of-distribution inference.

3. Approach

3.1. Overview

BeliefNet is a graph-based directed network in which nodes represent symbolic information, and unlike a neural network, the edges are not fully connected, but instead *form relationships based on observation and counterfactual evidence*. The network nodes and edges then act as neurons and connections in a neural network supporting weight optimisation. This structure enables the model to make traversability adaptations with very small amounts of data when compared with a conventional neural network, whilst retaining absolute explainability in the model's deduction. The relationship between a given set of input predicates and output results represents a belief within the model. Beliefs are something the system has some degree of confidence in being true [55], based on its own experiences.

Conceptually, human beliefs continually evolve and adapt to our experiences and our current domain, and we learn through life. When we face something unknown, we find the set of closest beliefs, use them to make a prediction, then create a new belief that captures the separation between the prediction and the truth, often captured within the concept of Predictive Coding [54]. It is this function that the BeliefNet model looks to model; conventional neural networks struggle with this approach requiring full validation after each evolution. In contrast, the graph structure of BeliefNet makes domain adaptation and counterfactual generation a deterministic function of the model throughout its use.

The BeliefNet model is designed to operate post perception, so it can be agnostic to the object classification model, or even the modality. It is also capable of integrating new predicates into the model, which means new classes can be added to a perception model, and these will be incorporated into BeliefNet as they are experienced. The model is *built logically before training*, in which connections between objects, their context, or existing relationships (in the case of counterfactuals) are generated dynamically. After which, a forward pass through the model is made, followed by optimisation and back-propagation using a conventional loss function. This can be achieved using a conventional upfront training set and continued further as the agent experiences its environment, providing an intervention mechanism overtime generating causal beliefs. Combined with the symbolism retained within each node, the structure provides the ability to *activate only relevant sections of the model* during inference, aiding explainability and providing reasoning in unknown situations. This approach acts as a zero-shot domain adaptation model, without the need for the high data volumes conventionally required through existing zero-shot approaches.

The model is designed for human interaction. The symbolic nature of the nodes and deliberate relationships means that any *prediction can be traced through the model* directly and that contributing nodes can be clearly identified. This enables operators to interact with the agent's cognition in novel methods, which are likely to significantly enhance trust. Operators can clearly determine why a decision was made, and can actively correct the result and use this to directly train the model. Furthermore, if they hold logic that had not yet been experienced by the model, this can be *integrated as testimonial knowledge*, within the network directly. As a result, BeliefNet presents an approach capable of context-based traversability prediction and the ability to generate trust between the agent and the operator.

3.2. High level structure

The model is formed of a number of components, some of which are adaptations of existing deep-learning approaches and some which are specific to BeliefNet. At a high level, the model should be considered as *post-processing of a perception model*, it initialises by taking the perception output and transforming this into a graph structure, known as an instance graph. The instance graph is generated as the output of a semantic segmentation model, such as YOLO [28]. The predictions are further enhanced through a depth perception model [7], estimated 3D separation between objects, and augmented with environmental tags that represent the weather, light, and domain. The *instance graph is a dense symbolic representation of a given image*. During training the instance graphs are converted to a series of context graphs, representing a target object and the surrounding objects, distances, and environmental tags for a given object for which a prediction is made. Context graphs are passed to the building algorithm, which is a custom training method designed to *extract causal relationships between objects, context, and a traversability value*. This will occur even during inference, enabling new relationships to be formed as they are identified. This forms the basis of the network; each node has an activation function, bias parameter, and each edge has a connection weight. To make a prediction, the predicates within the context graph are activated and propagated through relationships in the graph, generating values at the output nodes. When training or when provided with feedback, the optimisation step occurs, which *uses conventional back-propagation*, such as the Adam algorithm [36] to adapt the weights in a supervised manner. It is the combination of the logical build process before the back-propagation which provides the reasoning capacity and explainability of the structure. The architecture in Figure 2 visually shows how these components fit together within the model.

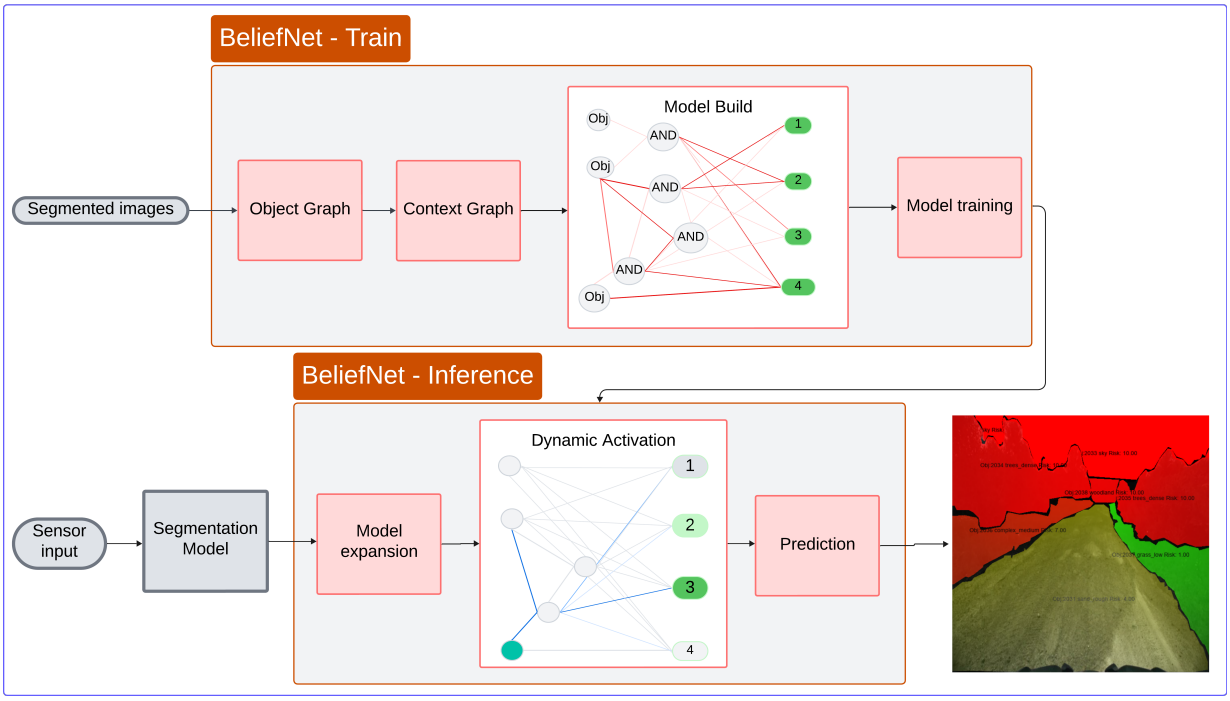


Fig. 2. Model Architecture, the high level architecture of the model is based on the structure of a neural network, but with adaptations to enable the symbolism to be retained throughout training and inference.

3.3. Data structures

3.3.1. Context graph

The context graph represents a target object for which a prediction is to be made, the relevant object, and environmental tags detected in proximity to the target object (the context) and how they each relate in proximity and position. It is the context graph which acts in effect as the input data to the core belief-net model. Within a given instance, there may be multiple objects about which a traversability assessment may want to be made. For each of these, a context graph (G) is generated, representing all objects (V) with relationships (E) to the target, such that $G = (V, E)$. Captured as a subgraph of the overall instance, it captures the target object(t), context object(c), relationship type (r) and the strength(s). For the traversability use case, the relationship is the positional relationship of the two objects, and strength represents the 3-dimensional Euclidean distance which is generated as post-processing from semantic segmentation. To ensure that this remains a subgraph, a relationship threshold (τ) is established. The relationship threshold and category ranges are parameters that can be tuned within the model.

Each edge $e \in E$ within the context graph is defined as:

$$e = (t, c, r, s) \tag{1}$$

By way of an example, consider a target object $t = \texttt{grass\_low\_1}$, at position $pos(t_1) = (1, 1, 0)$. This is within the context of two proximal objects:

$$c_1 = \texttt{tree}_1, \quad pos(c_1) = (100,\ 50,\ 300),$$
$$c_2 = \texttt{puddle}_2, \quad pos(c_2) = (11,\ 4,\ 0),$$

And has the environmental tags of:

$$c_3 = \texttt{season:winter},$$
$$c_4 = \texttt{weather:light\_rain},$$

The graph vertices become:

$$V = \{\texttt{grass\_low\_1}, \texttt{tree\_1}, \texttt{puddle\_1}, \texttt{season:winter}, \texttt{weather:light\_rain}\}$$

Each edge is computed as follows; note the strength and relations are not relevant to tags:

$$e_1 = \big(\texttt{grass\_low\_1},\ \texttt{tree}_1,\ r_1 = \text{"high right"},\ s_1 = \|(1,1,0) - (100,50,300)\|_2 = 320\big),$$
$$e_2 = \big(\texttt{grass\_low\_1},\ \texttt{puddle}_1,\ r_2 = \text{"low left"},\ s_2 = \|(1,1,0) - (11,4,0)\|_2 = 10\big),$$
$$e_3 = \big(\texttt{grass\_low\_1},\ \texttt{season:winter}_1,\ r_3 = \text{"etag"},\ s_3 = 1\big),$$
$$e_4 = \big(\texttt{grass\_low\_1},\ \texttt{weather:light\_rain}_1,\ r_4 = \text{"etag"},\ s_4 = 1\big).$$

Edges and vertices are only included in the context graph if the relationship strength is below the relationship threshold. If $\tau = 30$:

$$G = \big(V = \{c_2, c_3, c_4\}, E = \{e_2, e_3, e_4\}\big)$$

3.3.2. Data labels

BeliefNet is fundamentally a supervised model, relying on labelled samples from which to learn. Context graphs are labelled with a traversability index value. Such a value could be infinitely complex and very specific to an individual agent's performance characteristics. To increase generalised performance, a level of abstraction was selected which outlined the behavioural impact, rather than physical or mechanical. The developed traversability index categorises expected speed (relative to an agent's default) and the level of caution the agent will require in their traversability. The traversability risk analysis framework proposed by [17], in which multiple metrics such as risk of collision, slippage, and contact loss are combined into a single risk measure, serves as the basis for a unitary caution value. Although traversability risk can be a regression problem [27], discrete values are required for classification. Through the abstraction of metrics into behavioural categories, 11 distinct values were defined enabling relative traversability across platforms to be compared. These values are shown in the diagram in Figure 3.

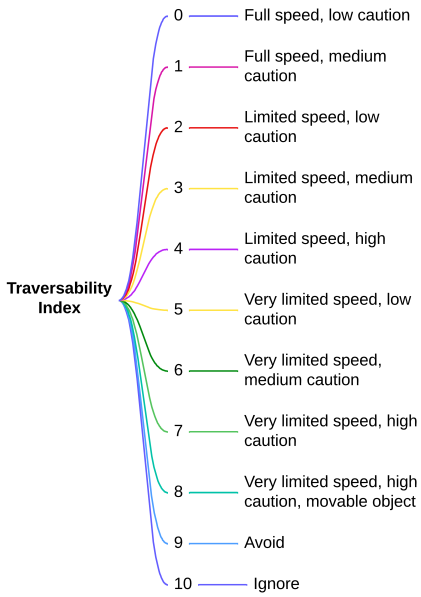
Labels are assigned to a context graph in two ways, depending on the phase of training. Firstly, human labelling enables a foundational training set to be developed, in which the target objects are assigned a relevant label based upon their context. This is used for initial supervised learning, where a large dataset is of value. Secondly, the agent can self-label the target objects based upon direct traversability experience. BeliefNet provides a prediction of the behaviour expected when traversing a given object. Once traversed, using methods such as those outlined by Zhao et al. [85], the separation from expected behaviour is used to generate appropriate labels for the context graph of objects. This method provides the data structures to support through-life learning of the model and the domain adaptation.

3.4. Model structure

3.4.1. BeliefNet nodes

BeliefNet is fundamentally based on a graphical structure of nodes and edges. As with a conventional neuron[60], each node $n \in N$ has both an input value from each edge $in_e \in In$ and a weight $w_e \in W$; there is a bias term b and an activation function to account for nonlinearity act. That is, the output value of a belief node is:

$$n' = act((\sum In * W) + b) \tag{2}$$



Speed potential	Caution	Object mobility
Full the platform can traverse the terrain upto the full speed of the platform	**Low** unlikely that an unpredicted dynamic response will be required, extended gaps between frame analysis	**Immovable** the object cannot be moved by the platform
Limited the platform should inhibit its speed upto 60% of full	**Medium** an unpredicted dynamic event is likely to occur, but infrequent, small gaps between frame analysis	**Movable** the object can be moved by the platform to increase traversability
Very limited the platform should limit its speed up to 20% of full speed	**High** an unpredicted dynamic event is likely to occur and should be expected on a frequent basis, analyse each frame	
Avoid the platform should not attempt to traverse the area		

Fig. 3. Traversability Index, there are 11 discrete traversability components, which increase in complexity as defined by the variables in the right hand table. These are categories that dictate the relative speed, level of caution the platform requires and the mobility of an object. In this context caution can be represented as the frequency of cognitive analysis, low caution objects can hold a greater frame separation between detailed processing, than high caution in which every frame may be analysed. They are assessed based on the individual perception of a single platform, therefore these cam be considered relative to the performance characteristics of an specific platform.

There are three types of nodes within the model, which are loosely equivalent to a single neuron within a neural network: an input node $n_{input} \in I$, $n_{belief} \in B$ and $n_{output} \in O$. An n_{input} represents an atomic predicate, it can only exist once within the model and only holds outgoing relationships to a set of n_{belief}. The belief nodes represent a logical grouping of context predicates C, akin to an AND relationship. The belief nodes act as the connection between the predicates and the output nodes. When combined into belief nodes, predicates and their logical relationship are retained within the name of the node. This enables individual nodes to be referenced directly and enables the contributing predicates to be directly identified.

$$\begin{aligned} &\text{Let } c_1 = grass\_low, \quad c_2 = hardcore\_smooth, \\ &\text{If } \langle c_1, e, n^1_{belief} \rangle and\ \langle c_2, e, n^1_{belief} \rangle, \\ &n^1_{belief} = grass\_low \wedge hardcore\_smooth \end{aligned} \tag{3}$$

The belief nodes can have relationships with other belief nodes, indicating counterfactual or divergent beliefs. Overtime, this component enables complex reasoning and causal relationships to emerge. If a third predicate c_3 was identified with a different output value when combined with the previous predicates, the following would be true.

$$\begin{aligned} &n^2_{belief} = (c_1 \wedge c_2) \wedge c_3, \\ &\quad E = \{\langle c_1, e, n^1_{belief} \rangle, \langle c_2, e, n^1_{belief} \rangle, \langle c_3, e, n^2_{belief} \rangle, \langle n^1_{belief}, e, n^2_{belief} \rangle\} \end{aligned} \tag{4}$$

The output nodes represent a specific output categorisation. Output nodes are combined into layers, in which each node represents a traversability index value, and a the layer is indexed to the object being classified. This provides the model with the ability to classify multiple different objects with the same model backbone. As they are a multiclass classification output, each output layer is combined with a Softmax function [10]. It is important to note that the Softmax only applies to the specific prediction object output layer, not all outputs. This approach also sets the foundation for cross-task generalisation, in which separate layers can exist for multiple tasks. Currently, it uses for object traversability risk layers; however, this could be more granular, with layers for variables like speed, roughness, and traction, each using the common model backbone.

3.4.2. The model

The model M can be represented as a combination of nodes N and edges E, in line with any conventional graph. The nodes are formally grouped into layers, based on their type. The input layer L_x represents all possible atomic predicates, the output layers are three-dimensional, with separate output layers for each prediction object or prediction task tn, between there are beliefs, as a result:

$$L_{input}, L_{belief}, L_{output} = I, (B + E), \{O_{tn^1}, O_{tn^2}, \ldots, O_{tn^n}\} \tag{5}$$

It is important to node that L_{belief} is not a linear layer, as relationships frequently exist between belief nodes, forming chains. There are no constraints to the chain length of these relationships, and there will be varying chain lengths throughout the model, since such B are logically grouped into a 3D layer for simplicity.

By way of an example, suppose that the input layer is a set of three atomic predicates:

$$I = \{n_{input}(\texttt{grass\_low}), n_{input}(\texttt{puddle}), n_{input}(\texttt{tree})\}$$

The model build process may then define two belief nodes and two output nodes:

$$\begin{aligned} B &= \{n_{belief}(\texttt{grass\_low AND puddle}), n_{belief}(\texttt{grass\_low AND puddle AND tree})\} \\ O &= \{n_{output}(1), n_{output}(5)\} \end{aligned} \tag{6}$$

Then edges are defined between the nodes (details explained in the model build section).

$$\begin{aligned} e_1 &= (n_{input}(\texttt{grass\_low});_{belief}(\texttt{grass\_low AND puddle})) \\ e_2 &= (n_{input}(\texttt{puddle});_{belief}(\texttt{grass\_low AND puddle})) \\ e_3 &= (n_{belief}(\texttt{grass\_low AND puddle});_{belief}(\texttt{grass\_low AND puddle AND tree})) \\ e_4 &= (n_{input}(\texttt{tree});_{belief}(\texttt{grass\_low AND puddle AND tree})) \\ e_5 &= (n_{belief}(\texttt{grass\_low AND puddle});_{output}(1)) \\ e_6 &= (n_{belief}(\texttt{grass\_low AND puddle AND tree});_{output}(5)) \end{aligned} \tag{7}$$

4. Model training and inference

4.1. Model build process

The model is designed to be persistent and adaptive throughout the lifecycle of an autonomous agent, meaning that it can be trained from no beliefs or use new instances, gained through experience, to update existing beliefs;

both use the same build methodology. Conventionally, neural networks have an initialised architecture that remains constant throughout the life cycle of the model, enabling the use of matrix multiplication. However, this inhibits adaptability and explainability. As a result, BeliefNet integrates a build phase prior to weight optimisation, in which relationships between predicates, beliefs, and outputs are dynamically formed, based upon presence within a supplied context graph. This occurs in the presentation of each context graph, which means that bulk training or experiential interventions retain the same capability. x

The build process uses individual context graphs or instances $ins_m \in Ins$, formed of a set of the target object tn, context objects (C^{ins_m}) and a set of target labels $\{\phi_{tn}^{(\text{ins}_m)}\}$, where the target label is a subject object and value, representing the traversability of the object about which a prediction was made. The model first establishes that each $\phi_{tn}^{\text{ins}_m} \in \Phi$ and that all $c_x^{ins_m} \in I$, otherwise new predicate nodes are created. Then it seeks to identify an existing belief that matches the exact context in which $b_y \equiv C^{\text{ins}_m}$, if found, it confirms that $\phi_{tn}^{\text{ins}_m} \in W_{b_y}$, otherwise it creates a new relationship $w_{b_y}, \phi_{tn}^{\text{ins}_m}$. If no direct match is found, the function searches for existing beliefs that host partial matches such that $b_y \subset C^{\text{ins}_m}$, it then creates a new belief b_z formed of $w(b_y, b_z)$ and $w(C', b_z)$ where $C' = C^{\text{ins}_m} \setminus b_y, b_z$. If no partial matching beliefs are found, it creates the belief from the relevant input nodes directly. For each relationship, the parameters are randomly initialised to prevent biasing the model to a local minimum. By way of an example, this process is demonstrated in Algorithm 1. For an instance in which partial matches were not required, this is shown as:

$$
\begin{aligned}
&ins_m = (tn_1, \{c_1^{ins_m}, c_2^{ins_m}, c_3^{ins_m}\}, \phi_{(tn^1)}^{\text{ins}_m}) \\
&I = \{c_1^{ins_m}, c_2^{ins_m}, c_3^{ins_m}\} \\
&O_{tn} = \{\phi_{(tn^1)}^{\text{ins}_m}), \ldots\} \\
&B = \{b_1\} = c_1^{ins_m} \wedge c_2^{ins_m} \wedge c_3^{ins_m} \\
&E = \{\langle c_1^{ins_m}, e, b_1\rangle, \langle c_2^{ins_m}, e, b_1\rangle, \langle c_3^{ins_m}, e, b_1\rangle, \langle b_1, e, \phi_{(tn^1)}^{\text{ins}_m})\rangle\}
\end{aligned}
\tag{8}
$$

However, in the event that a partial belief node already existed, it would create a edges with the existing partial node, such that:

$$
\begin{aligned}
&b_1 = c_1^{ins_m} \wedge c_2^{ins_m} \\
&b_2 = (c_1^{ins_m} \wedge c_2^{ins_m}) \wedge c_3^{ins_m} \\
&B = \{b_1, b_2\} \\
&E = \{\langle c_1^{ins_m}, e, b_1\rangle, \langle c_2^{ins_m}, e, b_1\rangle, \langle c_3^{ins_m}, e, b_2\rangle, \langle b_1, e, b_2\rangle, \langle b_1, e, \phi_{(tn^2)}^{\text{ins}_m})\rangle\ \langle b_2, e, \phi_{(tn^1)}^{\text{ins}_m})\rangle\}
\end{aligned}
\tag{9}
$$

The model build can be augmented with a priori knowledge during the build phase, where testimonial knowledge can be represented in effect in first-order logic. Relationships between specific predicates can be unified as knowledge with a direct relationship to the output node. This alone would not be sufficient to capture knowledge; therefore, knowledge nodes are initiated with high default parameter values for the weights and biases, often 1, this value has obvious impact on the model, so the value must be tested based on the domain. These parameters can be included or excluded from the optimiser, meaning they can be fixed or adapt with back-propagation. This represents the fact that knowledge could be permanently infallible, which is useful for human defined 'red lines', or could be feasibly disproved by future evidence. Both are viable options within the model. This feature enables the model to draw on some of the benefits of tools such as the LTN [3], which reasons over a corpus of provided knowledge, while allowing the system to add or adapt this knowledge based on induction. Unlike comparative models, this is optional and not a pre-requisite, the model can be very performant without the addition of knowledge.

Algorithm 1 Node and Relationship Update Procedure

Require: Features $F = \{grass\_low, puddle, tree\}$, Label $L = 1$

```
Let N ← grass_low ∧ puddle ∧ tree
if Exists(N) then
    if Exists(Relation(N, e, L)) then
        Do nothing
    else
        CreateRelation(N, e, L)
    end if
else
    Let P ← grass_low ∧ puddle
    if Exists(P) then
        Create(N = P ∧ tree)
    else
        Create(P = grass_low ∧ puddle)
        Create(N = P ∧ tree)
    end if
    CreateRelation(N, e, L)
end if
```

Knowledge predicates are defined:

$$KP = \{c_1, c_2, \ldots, c_n\}$$

A relationship is formed between the knowledge predicates and the output node:

$$\langle KP, e_k, \phi_{t,n} \rangle \tag{10}$$

The relationship weight is dictated by the knowledge type:

$$w_{e_k} = \begin{cases} w_m, & \text{if knowledge is mutable, trainable} = \text{True} \\ w_n, & \text{if knowledge is immutable, trainable} = \text{False} \end{cases} \quad \text{where } w_m < w_n$$

4.2. Dynamic activation

The concept of relevant beliefs is also a separation from conventional ML, which has been seen in neurosymbolic AI through the freezing of specific input nodes and network dissection [53]. The input layer is considered to be all atomic beliefs (those of the lowest fidelity) from a given context graph; only the atomic beliefs represented in the graph are activated; this is propagated through the network. Conventionally, layers in a model are defined by depth; however, as each union of predicates adds additional information to a belief, this is referred to as the fidelity of a belief. Activated atomic beliefs are combined recursively to activate or partially activate higher-fidelity beliefs. Any node that has been activated or partially activated can be considered a relevant belief. In the output layer, all relevant beliefs are passed to the activation function.

First the atomic predicates are activated in the input layer:

For $a_i \in A$,

$$A_{input} = \begin{cases} s, & \text{if } a_i \in C \\ 0, & \text{if } a_i \notin C \end{cases}$$

Higher fidelity beliefs are recursively activated:

$$\forall b \in B, \quad \text{Activation}(b) = w_b \sum_{c \in b} \text{Activation}(c) \quad \text{if} \quad b \cap C \neq \emptyset \tag{11}$$

Resulting in a set of Relevant Beliefs that are used to calculate the output node values:

$$R = \{b \in A \cup B \mid \text{Activation}(b) > 0\}$$

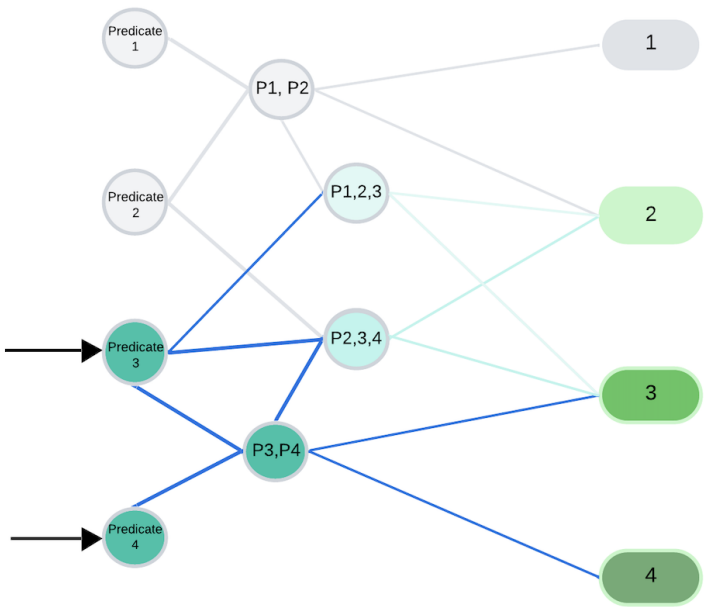


Fig. 4. An example of the dynamic activation based on relevant beliefs, and how this propagates through the model

4.3. Optimisation

Once the model is built, the model weights are then optimised using conventional back-propagation techniques. Relevant beliefs are activated by passing a scaled distance value, represented within the context graph, where $i_n \equiv c_x^{ins_m}$, then propagating the resulting values to the output nodes. The truth value is the target label, which is compared with the output values such that:

$$\text{loss}(t) = \text{Cross Entropy}\left(\phi_{tn}^{\text{ins}_m}, \max(\phi_t)\right) \tag{12}$$

The loss is then propagated using an optimisation algorithm, such as Adam [**?**], against the parameters existing within the nodes and edges. As only the relevant nodes were activated, the gradients outside these nodes will be zero, and therefore not affected. The model can hold multiple output targets (represented as output layers per target), but the individual forward pass through the model is assessed against a single target object, as such the loss is taken from ϕ_{tn} and not Φ. The model, as with all learning networks, is heavily influenced by the learning rate, which is managed by a scheduling function. This approach to optimisation is applied regardless of whether this is an initial bulk supervised training, self-analysis of traversal of a predicted object or a manually labelled sample by a operator to correct erroneous behaviour.

4.4. Prediction formulation

The generation of an output also has some key separations from a conventional neural network. As mentioned above, no matrix multiplication is conducted as part of the inference process. Although this could have a performance impact, this is offset by the overall sparsity of the model; for a given inference, there may only be a small proportion of the overall model activated at any time. However, the output nodes still need to draw on the precursor nodes to formulate an output. This is done through recursive activation of nodes, in which each node calls back through the network, extracting the output of the relevant beliefs $rb \in RB$ and calculating the output of the node. This function is called each time inference is run, similarly to a conventional predict function. In integrating this function, the model is able to account for new predicates, beliefs and relationships to be integrated into the model. Uniquely, BeliefNet uses an output layer per prediction object, which provides the model with its generalisation performance. Each layer has weighted relationships, and means that new output layers can be integrated into the model without having any direct experience of an object and make generalised assessments. The algorithm detailing how the outputs are generated is shown in 2. The output formulation for a given target tn is such that:

$$\begin{aligned} O_{tn} &= \{\phi_{(tn^1)}, \phi_{(tn^2)}, \dots \phi_{(tn^n)}\} \\ \phi_{(tn^n)} &= \sum_{n}^{1} n'(rb^n_{\phi_{(tn^n)}}) \\ tn' &= \max(O_{tn}) \end{aligned} \tag{13}$$

4.5. Explainability

A key feature of the model structure is the inherent traceability through the model to determine the factors that have led to a given prediction. This can be advantageous in highly regulated domains or environments where human-machine collaboration may be high. The traceability is a direct by-product of avoiding fully connected layers, meaning that an individual belief or input node can be simply and deterministically assessed for its contribution to a given output. The model nodes retain their previous outputs in state, meaning a critical path to prediction can be traced from each output node to the input node by recursively presenting the highest n contributors. This has been integrated directly into the model as an explainability function. A representation of this can then be visualised, as shown in 5. The contribution Cn for a given node i to a subsequent node j is defined as the combination of the output ϕ and the weight in the node w:

$$Cn_{ij} = \phi_i \times w_{ij} \tag{14}$$

5. Experimentation

To test the BeliefNet approach, we applied the model to a traversability scenario in which it was presented with a pre-segmented and labelled image, and sought to correctly classify the traversability of specific objects within the image. Within this scenario, we sought to test three factors:

Algorithm 2 Output Generation in BeliefNet

```
function GENERATEPREDICTION
    for all layers t ∈ T do
        for all nodes n in layer t do
            if φ_tn has predecessors then
                for all predecessor nodes p of φ_tn do
                    if φ_tn not computed yet then
                        φ_tn ← CALCULATENODEOUTPUT(p)
                    else
                        φ_tn ← cached value
                    end if
                end for
            end if
        end for
        if t is the prediction object then
            φ_t ← SOFTMAX(φ_t)
            O ← ARGMAX(n ∈ t)
            return O
        end if
    end for
end function
```

- Terrain classification comparison: How does BeliefNet compare to a static value approach, a graph neural network approach, and a random forest classifier?
- Data size comparison: How does BeliefNet compare to a graph neural network and random forests as the size of the training data increases?
- Activation function comparison: How does the model adapt with different combinations of activation functions across the model layers?

A comparative test to an end-to-end model was not conducted, due to a reliance upon both Lidar and imagery for most approaches, and the comparison of a segmented classification and pixel/voxel classification is not a simple translation.

5.1. Dataset

The experimentation data set is made up of a layered ontology used to label the Yamaha CMU data set [79] to better reflect the complexity of the environment. This enables object class, environmental meta-data, and class properties to be analysed by BeliefNet. The ontology is hierarchical, with classes, subclasses, and types, increasing the overall class numbers from 11 to 72. Enabling fidelity such as `vegetation_grass_low`, `vegetation_complex_tall` and `trail_hardcore_smooth`. Instances were generated from this dataset, and a baseline traversability value (as seen in Figure 3) was assigned to each object class. This served as a baseline, as it accurately represents the current terrain classification approach to the assessment of traversability, by directly assigning a value to a given class. To build a ground truth dataset, 300 images were re-labelled with human assessments of the traversability value, enabling humans to extract the image context and make a reasoned assessment on the relative risk associated with each object. This ground truth data is used as the basis for training BeliefNet. The re-labelled samples are then randomly split to provide a training and a test set, with all test metrics being completed by evaluating the test set. This established the framework from which the experimentation was conducted.

5.2. Methodology

The test was targeted at generated responses for the 'grass', 'hardcore', 'soil', 'sand', 'paved' ontology objects, which are the primary traversable objects. Multiple instances existed per training image, meaning that in total there

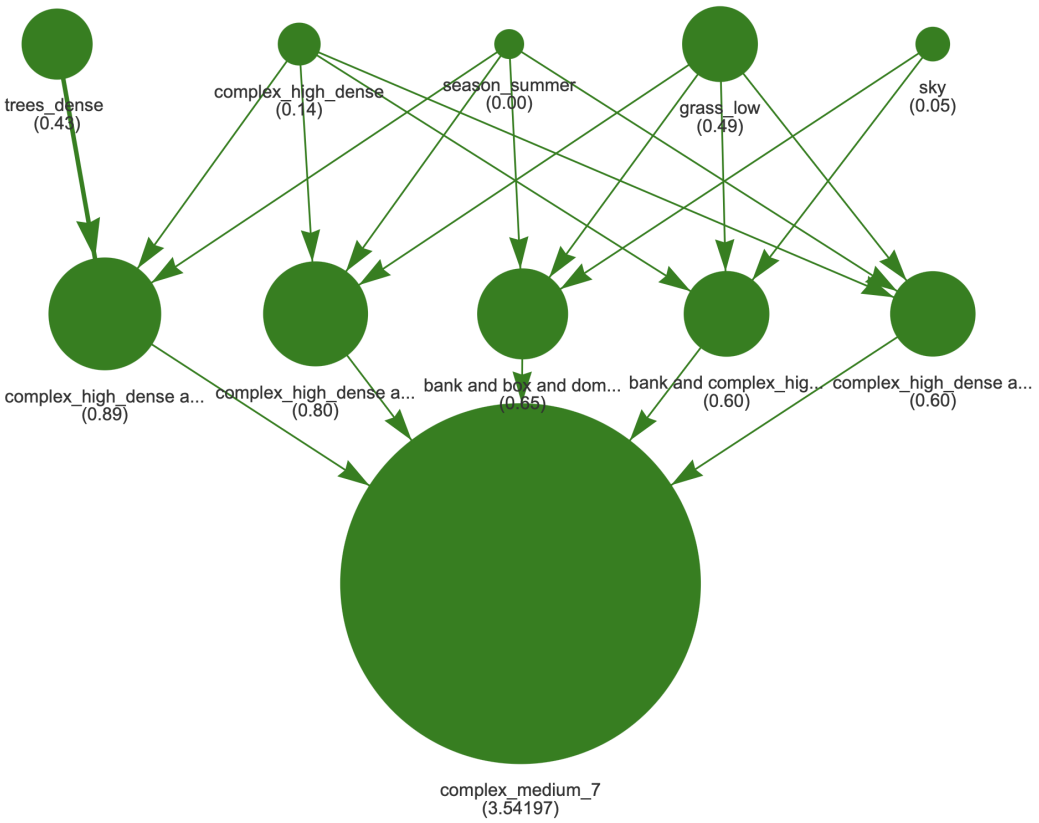


Fig. 5. The models graph explanation function showing the top 5 critical path contributors to the overall output, this is visualised graphically. Contributions are calculated recursively, with each layer showing the contribution to the subsequent node.

were c.350 training samples. This is a small amount for a traditional complex network, but represents a reasonable amount of *varied terrain data that an autonomous system could realistically gather about a given domain*. It enables us to test the ability of the model to adapt to smaller perturbations in the domain and data. The test set was extracted as 20% of the overall training set. At all points in the test, this was used to ensure comparability. The random samples were then taken from the training set in increasing increments from 25 samples to the entire data set, and the models for each set were trained. Each model was then tested against the test set and the accuracy was judged on the correct categorisation of the risk value against the human adjusted value. This was repeated 15 times and averaged for each model, with a new random test set identified for each iteration. The data holds large variability; due to its size, randomly selecting test data through multiple iterations ensures a broad set of complex challenges, especially zero-shot prediction, are represented in the test.

The comparison models selected were a random forest and a graph neural network (GraphSAGE) [22] combined with an XGBoost [14] classification head. The methodology for selecting these models was that they *separately present a neural and a symbolic approach* fit the problem, and in initial testing against a broad set of models demonstrated the most potential for extracting sufficient information from a context-based data structure, when compared to models such as a linear neural network or XGBoost alone. A key contributor to this was the *volume of data the experiment was constrained to*; this was a conscious choice to ensure any outcomes represented how the model could feasibly be used on a platform. Existing end-to-end approaches favour a continuous traversability classification, rather than discrete, making direct comparison not feasible. The input to BeliefNet is segmented objects from a detection model; for a conventional terrain classification model using state-of-the-art segmentation capabilities such

as YOLO architectures [28] or vision transformers, this object would be allocated a direct traversability value; this is *directly equivalent to the baseline traversability value*. As a result, the baseline comparison value represents the performance of such approaches.

5.3. Metrics

As the overall classification metrics in this case are risk-based and incremental, performance can also be assessed by assessing the distance in separation between the predicted and actual values. A model that gets its predictions closer to the actual classification performs better than a model that is further away. To capture this, we will look at both an absolute classification and a fuzzy accuracy, which assesses the score as +/- 1 of the absolute.

5.4. Variables

The baseline accuracy is using the default values for an object based on its ontological class and value, compared with the human-edited values; this would be heavily skewed by sampling, so a consistent baseline from the full dataset was taken as 23% absolute accuracy and 43% with fuzzy accuracy.

In addition to the baseline values, we tested three additional approaches:

- BeliefNet model as described in this paper.
- A random forest classifier [9], which was chosen to as a comparator due to its reasoning capacity with small datasets, and its ability to explain its results, making it the most similar in output to BeliefNet
- A graph neural network, GraphSAGE which uses an LSTM based architecture [23] to learn and generate context graph embedding and then passes the embedding to an XGBoost algorithm, acting as a classification head, to conduct supervised classification [14].

5.5. Results

When BeliefNet was trained to predict the outputs of the traversable object classes in the ontology (grass, hardcore, soil, sand, complex, rock), using the full dataset it achieved 47% absolute and 81% fuzzy accuracy; this did not include a priori knowledge. When scaled with the dataset, this is performed as shown in figure 7. This test was repeated with only grass objects, as these present the highest proportion of the data set and are terrain characteristics with the highest variation in the traversability index within the class; the results of which are shown in figure 8. The average results for each model are shown in Table 5.5. The comparison between baseline and BeliefNet against multiple prediction objects can be seen in figure 6, noting that the number of samples is not consistent between object types, which is related to the increased variance in some objects over others; for example, the low grass distribution is significantly lower than tall grass. In the more challenging object, tall grass, due to the higher variation, *BeliefNet outperformed the baseline in both absolute and fuzzy accuracy*.

The graph embedding model failed to learn effective patterns within the data; this is likely due to the additional abstraction generated by the embeddings and the small amount of data for a given prediction, preventing the model from being able to generalise effectively. This resulted in the model returning the same value for instances of a given terrain and not identifying any factors that would shift the risk. Even after training using the full dataset, the model returned an absolute score of 33% and a fuzzy score of 52%. The GraphSAGE model is the comparator to a conventional neural network; the inability to converge on a solution demonstrates the importance of a neurosymbolic approach in a complex reasoning task.

The random forest was more successful and was able to make comparable predictions in both actual and fuzzy accuracy, with the full training data achieving 79% fuzzy accuracy, compared to 81% for BeliefNet, as shown in Figure 8. Furthermore, random forests present two additional downsides compared to the BeliefNet model.

The nature of random forests means that it is challenging for them to *form predictions across multiple classifications and classification objects*. As a result, each classification object, such as grass_low, required its own model. Whilst this is a standard practice, it comes with a number of drawbacks; firstly, it prevents generalised concepts from being formed across multiple terrain types, in effect reducing the training data available to each model, and

Model	Prediction objects	Absolute accuracy	Fuzzy Accuracy
Baseline	All	23%	43%
GraphSAGE + XGBoost	All	33%	52%
Random Forest	Grass	35%	72%
	All	**49**%	79%
BeliefNet	Grass	**35**%	**75**%
	All	47%	**81**%

Table 1

The summary results using a full dataset over 15 iterations with random test sets for each of the test models.

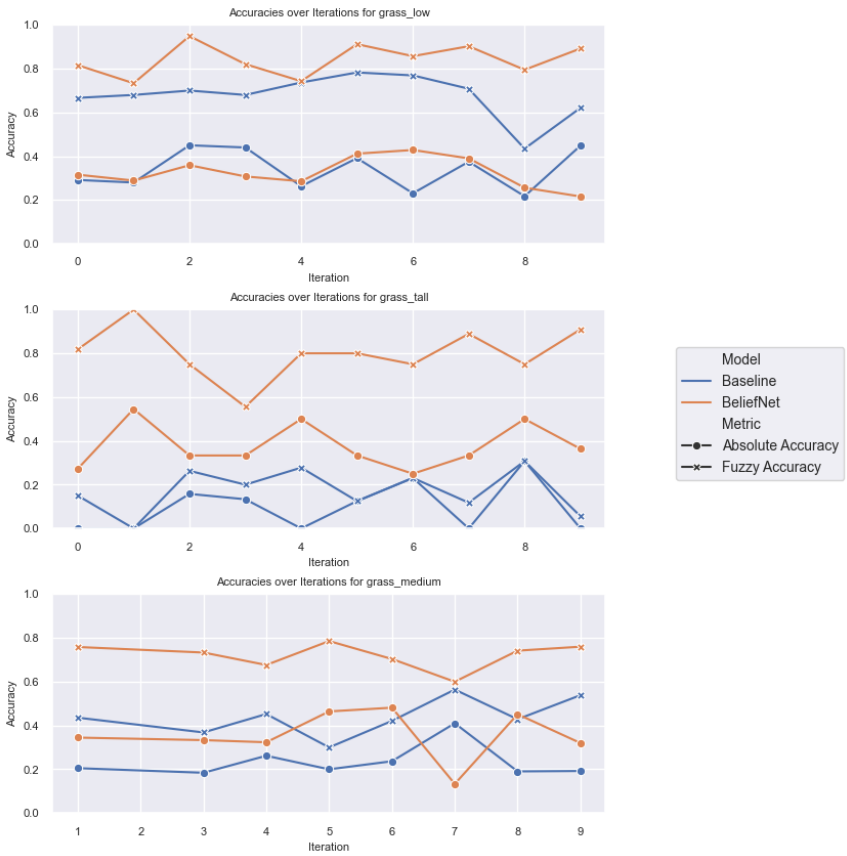


Fig. 6. Experiment comparison of BeliefNet and the baseline absolute and fuzzy accuracy for individual grass prediction objects, the variance in prediction value increases in objects left to right.

this will impact domain adaptation. Secondly, in practice, there will be an i/o cost to loading new models, which could be a bottleneck in situations with more than one traversable object. Given the volume of assessments required in a continuous terrain classification, this will have a significant cost. Some instances in the dataset had five target objects, meaning 5 separate models would need to be loaded for 1 image. In contrast, *BeliefNet is capable of having multiple output layers simultaneously for a single model backbone*. This means that the model is able to draw generalised concepts rather than terrain-specific ones, which provides significant advantages when the domain ontology adapts. This can be seen in the data; a random forest was trained for each object, meaning that throughout the training it has always seen a representation of the object previously, whereas it is possible that BeliefNet makes classifications with no prior knowledge of an object. In all evaluation runs, *BeliefNet would make a prediction on at*

least one class that was not in its training distribution. This represents a trade-off between accuracy and generalisation and is demonstrated clearly by the separation between absolute accuracy in all prediction objects. Although this is a separation of 3%, it is likely that this is the benefit of having a specific model for each class. Whilst this is beneficial, this is outweighed significantly by the model being able to make predictions on unseen data classes, as the BeliefNet demonstrates.

An additional advantage of BeliefNet over a random forest is related to the fixed inputs required for a random forest. The input data for the model are a fixed shape array with each item in the array reflecting a possible context object and the distance from that object. This has two drawbacks; firstly, in an ontology such as the one used in this model, with more than 70 objects, this results in a *very sparse set of input data*, which can lead to overfitting [83] and may be a contributor to the flat learning profile. Secondly, the fixed nature means that the model cannot adapt to new objects identified within the domain. If a new object was identified, based on a new or adapted sensor classifier, the model would require retraining. In contrast, BeliefNet has a dynamic input length *requiring only the predicates that are sensed to be passed*, and it is designed to be extensible, and when a new predicate is identified, this can be directly integrated into the model. In this case, weights are initialised with a default value, but can then be fine-tuned, but in a manner which constrains the adaptation only to the relevant predicates, as only they are activated. This prevents having an adverse impact on existing and unrelated concepts. This flexibility and adaptive structure is core to BeliefNet's domain generalisation and establishes it as a through-life model, which grows with the agent's understanding of the world.

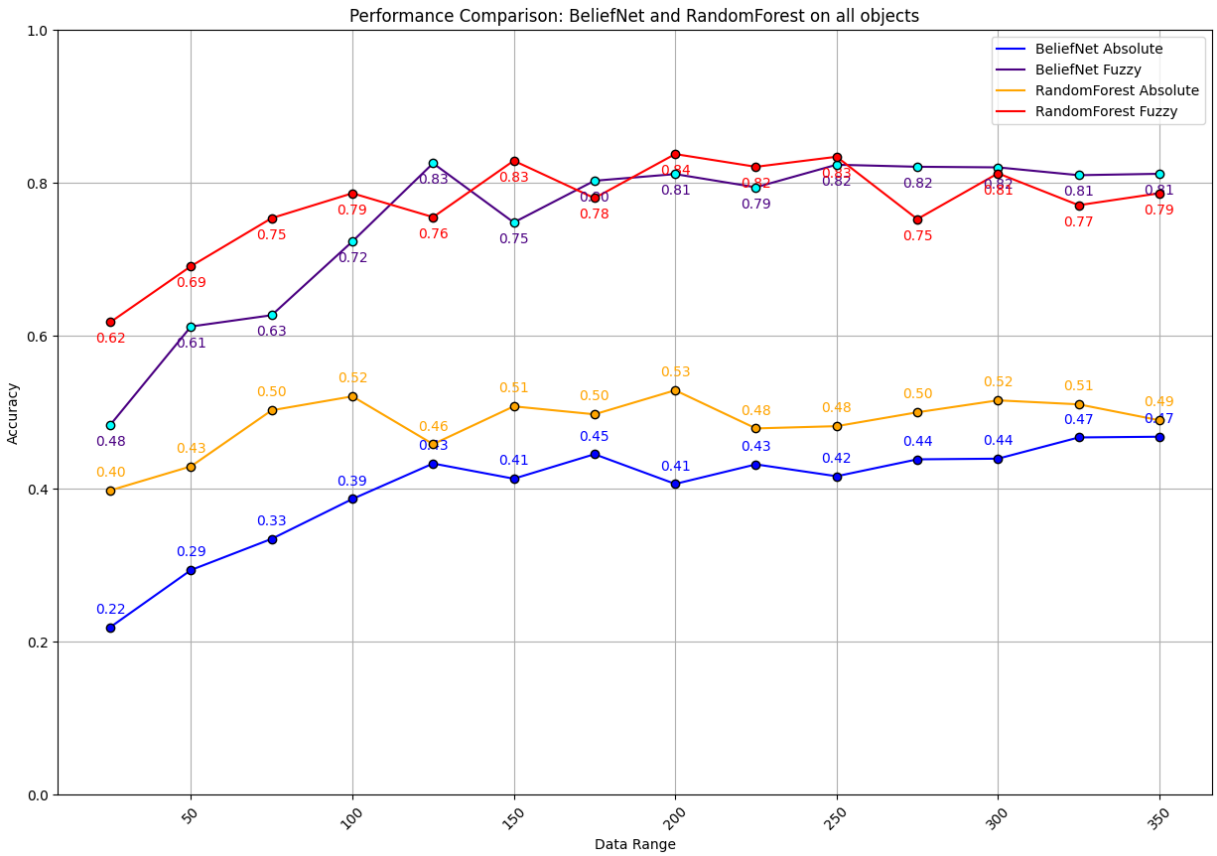


Fig. 7. Experiment comparison of BeliefNet and a Random Forest, with a scaled dataset comparing classification of objects 'grass', 'sand', 'hardcore', 'complex', 'soil' risk classifications

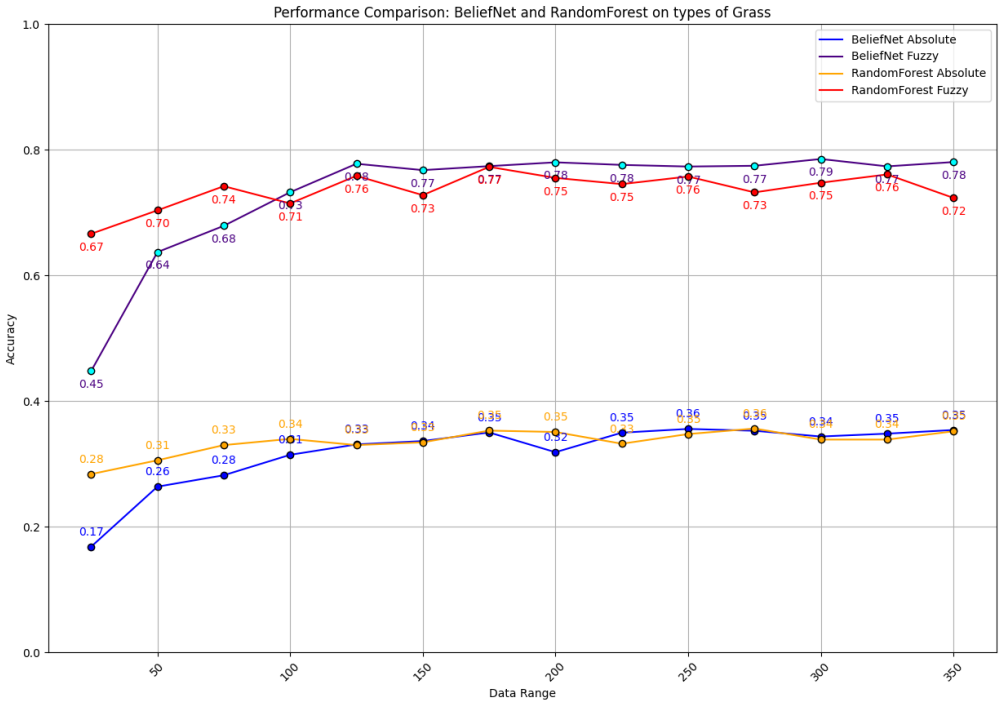


Fig. 8. Experiment comparison of BeliefNet and a Random Forest, with a scaled dataset comparing classification of Grass objects *grasslow*, *grassmedium*, *grasstall* risk classifications

To validate the performance characteristics of the model, we tested the grass sample set using a number of activation functions; in doing so, we were able to see how the model adapts over different combinations. Activation functions were assigned to the input layer and belief nodes separately, noting that they each had separate behaviours. Several functions were used:

- Leaky-Rectified Linear Unit.
- Linear activation, in effect the identity of the input.
- Hard-sigmoid.
- Learnable Rectified Linear Unit (ReLU), which was generated with a learnable scalar parameter p such that $\mathrm{ReLU}(x) * P$.

The experiment sought to identify any key variations in the results from the separate activation functions. Each combination was repeated 15 times and the mean results are shown in Figure 9, using a consistent learning rate of 0.001 and over 15 epochs of learning. The model performed consistently across the models. The best performing combinations were those that involved a linear function at the input layer. As the input is a function of distance, this suggests that the model benefits from retaining the symbolic information. The learnable activation functions performed well, but did not significantly outperform, suggesting that there are sufficient model parameters without the requirement to augment.

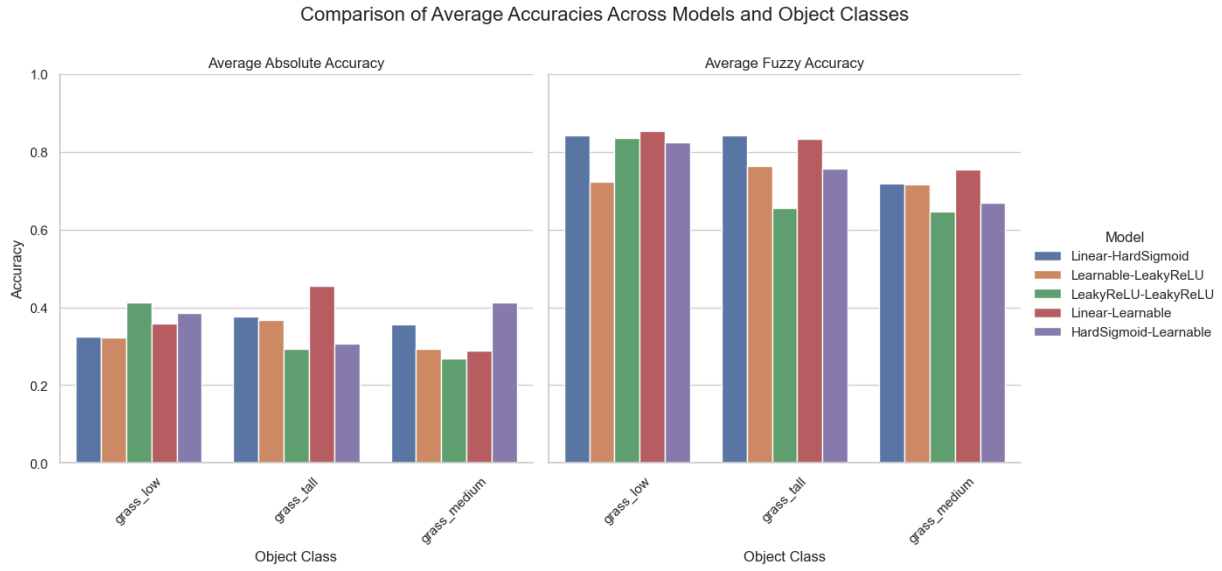


Fig. 9. Experiment comparison of BeliefNet using separate activation functions (input, belief node) when classifying the grass objects(*grasslow*, *grassmedium*, *grasstall*)

Table 2

Model	Fuzzy Acc Var	Abs Acc Mean	Abs Acc Var	Abs Acc Std Dev	Fuzzy Acc Mean	Fuzzy Acc Std Dev
Linear-HardSigmoid	0.003302	0.353033	0.000431	0.020770	0.801267	0.057464
Learnable-LeakyReLU	0.000445	0.327467	0.000934	0.030568	0.735500	0.021083
LeakyReLU-LeakyReLU	**0.007640**	0.324967	0.004002	0.063263	0.713000	**0.087407**
Linear-Learnable	0.001807	0.367900	**0.004646**	**0.068162**	**0.814167**	0.042511
HardSigmoid-Learnable	0.004052	**0.368600**	0.002058	0.045362	0.751433	0.063652

6. Discussion

BeliefNet presents an opportunity to provide a unified reasoning engine to support terrain traversal, in a manner which enables an agent to make an informed decision about risk and traversability. It is inherently extensible, meaning that it can use what it has learnt within one domain, and adapt this to unknown environments, and its inherent explainability means that operators can interpret, understand and impact decision making. This approach significantly increases performance when compared with the static value approach, and enhances the flexibility and explainability when compared to an end-to-end model. This paper demonstrated the application of the BeliefNet model within an autonomous agent traversability reasoning task; however, this model structure has the potential to be applied more widely across similar tasks with high complexity and underlying logic, which may not be immediately accessible. This could be particularly relevant to domains with high-assurance or regulatory requirements, which traditionally AI struggles to meet.

6.1. Deployment considerations

When the deployment of BeliefNet to an autonomous system is considered, there are a number of topics worthy of discussion. A key challenge of capability deployment to an autonomous system is that power and space are

finite and broadly shared, and there are competing priorities. As a result, the computational overhead of any single system must be considered in the context of the system as a whole. As a component of the navigational system, BeliefNet will be expected to have comparable performance to a perception system likely running at > 10 fps. Although explicit benchmarking of speed performance was out of scope for this research, the removal of matrix multiplication could have a negative impact on inference speed as the model scales. This was not seen within this experimentation, but could be mitigated by the set-based nature of nodes within the model, meaning that model size will grow logarithmically with experience. This could also be further enhanced through the addition of more complex conjunctives, such as NOT and OR, which could aggregate beliefs more succinctly.

When considering how BeliefNet fits within a deployed platform, it is valuable to consider the full information processing pipeline. This research explored the cognition element alone, but has dependencies on both the perception module and low-level control of the platform. BeliefNet does not require fixing to any given perception model, or require retraining if the perception model is. However, there is a critical dependency between the two models; for BeliefNet to make accurate context-based predictions, it depends on accurate classifications of objects within the scene. Classification errors could have a potentially greater impact on cognition output. Whilst vision was the focus of this research, BeliefNet was intended to work with any classification modality. Platform low-level control both depends on and provides to BeliefNet. It relies on abstracted traversability values to predict the platform kinematics required to effectively traverse an object of a given value. Furthermore, once traversing an object, the performance of the platform, vs expected, provides a valuable feedback mechanism that can be used directly to inform optimisation.

6.2. Further work

This research outlines the potential for BeliefNet in the domain of complex environment traversability, but there are opportunities for further development, which could enhance its applicability. Firstly, extending the solution further to include the connection of a single, or multimodality sensor module would be the next step towards platform integration. This would also provide an opportunity for Lidar/Vision combinations to test the capability of BeliefNet to collaborate across modalities. Secondly, considering the platform conversion of exteroceptive and interoceptive sensing outputs into a traversability assessment, thus creating a full learning loop for the agent, this could be another application of BeliefNet. Another area to be considered is experimentation with the learning rate for manual learning/human intervention such that learning is effective, without adversely skewing model outputs. Finally, this research into the model's performance was completed against a single objective function; expanding the research to support multi-objective optimisation would enable additional agency in more complex situations. For example, the ability for BeliefNet to support the risk/time trade-off when assessing tactical route planning.

7. Conclusion

In this paper, we have defined the challenge of traversability assessment for autonomous systems when operating in complex environments, demonstrated the importance of context within predictions, and detailed BeliefNet as a novel neurosymbolic model capable of generating context-based predictions for traversability. BeliefNet is capable of learning through life from the experience of an autonomous agent, providing a method to enhance domain adaptation, and uses causal beliefs to support predictions in unknown situations. By retaining a symbolic structure within the network, it remains explainable and provides operators with the ability to interact with model training directly, enhancing trust. BeliefNet presents an advance towards enabling autonomous system deployment and performance in complex, demanding environments.

References

[1] A. Agarwal, P. Shenoy and a. Mausam, End_to_End Neuro_Symbolic Architecture for Image_To_Image Reasoning Tasks, *arXiv* (2021). http://arxiv.org/pdf/2106.03121.

[2] A. Agha, K. Otsu, B. Morrell, D.D. Fan, R. Thakker, A. Santamaria-Navarro, S.-K. Kim, A. Bouman, X. Lei, J.A. Edlund, M.F. Ginting, K. Ebadi, M. Anderson, T. Pailevanian, E. Terry, M.T. Wolf, A. Tagliabue, T.S. Vaquero, M. Palieri, S. Tepsuporn, Y. Chang, A. Kalantari, F. Chavez, B.T. Lopez, N. Funabiki, G. Miles, T. Touma, A. Buscicchio, J. Tordesillas, N. Alatur, J. Nash, W. Walsh, S. Jung, H. Lee, C. Kanellakis, J. Mayo, S. Harper, M. Kaufmann, A. Dixit, G. Correa, C. Lee, J. Gao, G. Merewether, J. Maldonado-Contreras, G. Salhotra, M.S.d. Silva, B. Ramtoula, Y. Kubo, S.A. Fakoorian, A. Hatteland, T. Kim, T. Bartlett, A. Stephens, L. Kim, C. Bergh, E. Heiden, T. Lew, A. Cauligi, T. Heywood, A. Kramer, H.A. Leopold, H.C. Choi, S. Daftry, O. Toupet, I. Wee, A. Thakur, M. Feras, G. Beltrame, G. Nikolakopoulos, D.H. Shim, L. Carlone and J. Burdick, NeBula: Quest for Robotic Autonomy in Challenging Environments; TEAM CoSTAR at the DARPA Subterranean Challenge, *CoRR* (2021), arXiv: 2103.11470. https://arxiv.org/abs/2103.11470.

[3] S. Badreddine, A. Garcez, L. Serafini and M. Spranger, Logic Tensor Networks, *Artificial Intelligence* (2022), 103649. doi:10.1016/j.artint.2021.103649. http://arxiv.org/abs/2012.13635.

[4] E. Barnes and J. Hutson, Natural Language Processing and Neurosymbolic AI: The Role of Neural Networks with Knowledge-Guided Symbolic Approaches, *DS Journal of Artificial Intelligence and Robotics* (2024), 1–13. doi:10.59232/AIR-V2I1P101. https://www.dsjournals.com/air/AIR-V2I1P101.

[5] J.J. Bauer, T. Eiter, N. Higuera Ruiz and J. Oetsch, Visual Graph Question Answering with ASP and LLMs for Language Parsing, *Electronic Proceedings in Theoretical Computer Science* (2025), 15–28. doi:10.4204/EPTCS.416.2. http://arxiv.org/abs/2502.09211v1.

[6] S. Beycimen, D. Ignatyev and A. Zolotas, A comprehensive survey of unmanned ground vehicle terrain traversability for unstructured environments and sensor technology insights, *Engineering Science and Technology, an International Journal* (2023), 101457. doi:10.1016/j.jestch.2023.101457. https://www.sciencedirect.com/science/article/pii/S2215098623001350.

[7] S.F. Bhat, R. Birkl, D. Wofk, P. Wonka and M. Müller, ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth, 2023. doi:2302.12288.

[8] B.P. Bhuyan, A. Ramdane-Cherif, R. Tomar and T.P. Singh, Neuro-symbolic artificial intelligence: a survey, *Neural Computing and Applications* **36**(21) (2024), 12809–12844. doi:10.1007/s00521-024-09960-z.

[9] L. Breiman, Random Forests, *Machine Learning* **45**(1) (2001), 5–32. doi:10.1023/A:1010933404324.

[10] J. Bridle, Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters, in: *Advances in Neural Information Processing Systems*, Vol. 2, Morgan-Kaufmann, 1989. https://proceedings.neurips.cc/paper/1989/hash/0336dcbab05b9d5ad24f4333c7658a0e-Abstract.html.

[11] F. Castaño, G. Beruvides, R.E. Haber and A. Artuñedo, Obstacle Recognition Based on Machine Learning for On-Chip LiDAR Sensors in a Cyber-Physical System, *Sensors* **17**(9) (2017), 2109, Number: 9 Publisher: Multidisciplinary Digital Publishing Institute. doi:10.3390/s17092109. https://www.mdpi.com/1424-8220/17/9/2109.

[12] R.O. Chavez-Garcia, J. Guzzi, L.M. Gambardella and A. Giusti, Learning Ground Traversability From Simulations, *IEEE Robotics and Automation Letters* **3**(3) (2018), 1695–1702. doi:10.1109/LRA.2018.2801794. https://ieeexplore.ieee.org/document/8280544.

[13] F. Chen, F. Weitkämper and S. Malhotra, Understanding Domain-Size Generalization in Markov Logic Networks, in: *Machine Learning and Knowledge Discovery in Databases. Research Track*, A. Bifet, J. Davis, T. Krilavičius, M. Kull, E. Ntoutsi and I. Žliobaitė, eds, Springer Nature Switzerland, Cham, 2024, pp. 297–314. ISBN 978-3-031-70368-3. doi:10.1007/978-3-031-70368-3$_1$8.

[14] T. Chen and C. Guestrin, XGBoost: A Scalable Tree Boosting System, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 785–794. ISBN 978-1-4503-4232-2. doi:10.1145/2939672.2939785.

[15] L.N. DeLong, R.F. Mir and J.D. Fleuriot, Neurosymbolic AI for Reasoning Over Knowledge Graphs: A Survey, *IEEE Transactions on Neural Networks and Learning Systems* **36**(5) (2025), 7822–7842. doi:10.1109/TNNLS.2024.3420218. https://ieeexplore.ieee.org/document/10603423.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2020. https://openreview.net/forum?id=YicbFdNTTy.

[17] D.D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick and A.-A. Agha-Mohammadi, STEP: Stochastic Traversability Evaluation and Planning for Safe Off-road Navigation, *Robotics: Science and Systems* (2021).

[18] P. Filitchkin and K. Byl, Feature-based terrain classification for LittleDog, in: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Vilamoura-Algarve, Portugal, 2012, pp. 1387–1392. ISBN 978-1-4673-1736-8 978-1-4673-1737-5 978-1-4673-1735-1. doi:10.1109/IROS.2012.6386042. http://ieeexplore.ieee.org/document/6386042/.

[19] L. Fritz, H.A. Hamersma and T.R. Botha, Off-road terrain classification, *Journal of Terramechanics* **106** (2023), 1–11. doi:10.1016/j.jterra.2022.11.003. https://www.sciencedirect.com/science/article/pii/S0022489822000751.

[20] M.V. Gasparino, A.N. Sivakumar and G. Chowdhary, WayFASTER: a Self-Supervised Traversability Prediction for Increased Navigation Awareness, in: *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 8486–8492. doi:10.1109/ICRA57147.2024.10610436. https://ieeexplore.ieee.org/document/10610436.

[21] T. Gürtler and B.L. Kaminski, Programming and Reasoning in Partially Observable Probabilistic Environments, arXiv, 2025, arXiv:2506.13491 [cs]. doi:10.48550/arXiv.2506.13491. http://arxiv.org/abs/2506.13491.

[22] W.L. Hamilton, R. Ying and J. Leskovec, Inductive representation learning on large graphs, in: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 1025–1035. ISBN 978-1-5108-6096-4.

[23] J. Hawthorne, Inductive Logic, in: *The Stanford Encyclopedia of Philosophy*, Spring 2021 edn, E.N. Zalta, ed., Metaphysics Research Lab, Stanford University, 2021. https://plato.stanford.edu/archives/spr2021/entries/logic-inductive/.

[24] M. Himmelsbach, F.v. Hundelshausen and H.-J. Wuensche, Fast segmentation of 3D point clouds for ground vehicles, in: *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 560–565, ISSN: 1931-0587. doi:10.1109/IVS.2010.5548059.

[25] T. Hodgdon, A. Fuentes, J. Olivier, B. Quinn and S. Shoop, Automated terrain classification for vehicle mobility in off-road conditions, Technical Report, Engineer Research and Development Center (U.S.), 2021. doi:10.21079/11681/40219. https://hdl.handle.net/11681/40219.

[26] M. Ingram, R. Moreton, B. Gancz and F. Pollick, Calibrating Trust Toward an Autonomous Image Classifier, *Technology, Mind, and Behavior* **2**(1) (2021). doi:10.1037/tmb0000032. https://tmb.apaopen.org/pub/calibrating-trust/release/2.

[27] H. Inotsume and T. Kubota, Terrain traversability prediction for off-road vehicles based on multi-source transfer learning, *ROBOMECH Journal* **9**(1) (2022), 6. doi:10.1186/s40648-021-00215-3.

[28] G. Jocher, Ultralytics YOLOv5, 2020. doi:10.5281/zenodo.3908559. https://github.com/ultralytics/yolov5.

[29] M. Junaid Khan, A. Masood Siddiqui, H. Saeed Khan, F. Akram and M. Jaleed Khan, MuRelSGG: Multimodal Relationship Prediction for Neurosymbolic Scene Graph Generation, *IEEE Access* **13** (2025), 47042–47054. doi:10.1109/ACCESS.2025.3551267. https://ieeexplore.ieee.org/document/10925205.

[30] S. Jung, J. Lee, X. Meng, B. Boots and A. Lambert, V-STRONG: Visual Self-Supervised Traversability Learning for Off-road Navigation, in: *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 1766–1773. doi:10.1109/ICRA57147.2024.10611227. https://ieeexplore.ieee.org/document/10611227.

[31] L.P. Kaelbling and T. Lozano-Pérez, Integrated task and motion planning in belief space, *The International Journal of Robotics Research* **32**(9–10) (2013), 1194–1227. doi:10.1177/0278364913484072.

[32] D. Kahneman, *Thinking, fast and slow*, Thinking, fast and slow, Farrar, Straus and Giroux, 2011. ISBN 978-0-374-27563-1 978-1-4299-6935-2.

[33] K. Kardaş, Ulusoy and N.K. Çiçekli, Learning complex event models using markov logic networks, in: *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2013, pp. 1–6. doi:10.1109/ICMEW.2013.6618413. https://ieeexplore.ieee.org/document/6618413/.

[34] H. Kautz, The Third AI Summer: AAAI Robert S. Engelmore Memorial Lecture, *AI Magazine* **43**(1) (2022), 105–125, Number: 1. doi:10.1002/aaai.12036. https://ojs.aaai.org/index.php/aimagazine/article/view/19122.

[35] T. Khot, N. Balasubramanian, E. Gribkoff, A. Sabharwal, P. Clark and O. Etzioni, Exploring Markov Logic Networks for Question Answering, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics. https://aclanthology.org/D15-1080.pdf.

[36] D.P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization., in: *ICLR (Poster)*, Y. Bengio and Y. LeCun, eds, 2015. http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#KingmaB14.

[37] H. Kokel, The 6 Types of Neuro-Symbolic Systems, 2020, Section: posts. https://harshakokel.com/posts/neurosymbolic-systems/.

[38] A. Komanduri, Y. Wu, W. Huang, F. Chen and X. Wu, SCM-VAE: Learning Identifiable Causal Representations via Structural Knowledge, in: *2022 IEEE International Conference on Big Data (Big Data)*, 2022, pp. 1014–1023. doi:10.1109/BigData55660.2022.10021114. https://ieeexplore.ieee.org/document/10021114.

[39] L.C. Lamb, A.d. Garcez, M. Gori, M.O.R. Prates, P.H.C. Avelar and M.Y. Vardi, Graph Neural Networks Meet Neural-Symbolic Computing: A Survey and Perspective, in: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence Organization, Yokohama, Japan, 2020, pp. 4877–4884. ISBN 978-0-9992411-6-5. doi:10.24963/ijcai.2020/679. https://www.ijcai.org/proceedings/2020/679.

[40] J. Larson and M. Trivedi, Lidar based off-road negative obstacle detection and analysis, in: *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 192–197, ISSN: 2153-0017. doi:10.1109/ITSC.2011.6083105.

[41] M. Lauri, D. Hsu and J. Pajarinen, Partially Observable Markov Decision Processes in Robotics: A Survey, *IEEE Transactions on Robotics* **39**(1) (2023), 21–40, arXiv:2209.10342 [cs]. doi:10.1109/TRO.2022.3200138. http://arxiv.org/abs/2209.10342.

[42] Y. LeCun, A Path Towards Autonomous Machine Intelligence Version 0.9.2, 2022-06-27, *OpenReview Archive* (2022).

[43] Y. LeCun, U. Muller, J. Ben, E. Cosatto and B. Flepp, Off-Road Obstacle Avoidance through End-to-End Learning., 2005.

[44] H. Li, G. Zhu, L. Zhang, Y. Jiang, Y. Dang, H. Hou, P. Shen, X. Zhao, S.A.A. Shah and M. Bennamoun, Scene Graph Generation: A comprehensive survey, *Neurocomputing* **566** (2024), 127052. doi:https://doi.org/10.1016/j.neucom.2023.127052. https://www.sciencedirect.com/science/article/pii/S092523122301175X.

[45] H. Lin, Bayesian Epistemology, in: *The Stanford Encyclopedia of Philosophy*, Summer 2024 edn, E.N. Zalta and U. Nodelman, eds, Metaphysics Research Lab, Stanford University, 2024. https://plato.stanford.edu/archives/sum2024/entries/epistemology-bayesian/.

[46] A.M. Lipsky and S. Greenland, Causal Directed Acyclic Graphs, *JAMA* **327**(11) (2022), 1083–1084. doi:10.1001/jama.2022.1816.

[47] Z. Lu, I. Afridi, H.J. Kang, I. Ruchkin and X. Zheng, Surveying neuro-symbolic approaches for reliable artificial intelligence of things, *Journal of Reliable Intelligent Environments* **10**(3) (2024), 257–279. doi:10.1007/s40860-024-00231-1.

[48] L. Luo, G. Zhang, H. Xu, Y. Yang, C. Fang and Q. Li, INSIGHT: End-to-End Neuro-Symbolic Visual Reinforcement Learning with Language Explanations, in: *International Conference on Machine Learning 2024*, 2024. https://arxiv.org/pdf/2403.12451.

[49] R. Manduchi, A. Castano, A. Talukder and L. Matthies, Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation, *Autonomous Robots* **18**(1) (2005), 81–102. doi:10.1023/B:AURO.0000047286.62481.1d.

[50] J. Mao, C. Gan, P. Kohli, J.B. Tenenbaum and J. Wu, The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision, in: *International Conference on Learning Representations 2019*, 2019, arXiv:1904.12584 [cs]. doi:10.48550/arXiv.1904.12584. http://arxiv.org/abs/1904.12584.

[51] D. Maturana, P.-W. Chou, M. Uenoyama and S. Scherer, Real-Time Semantic Mapping for Autonomous Off-Road Navigation, in: *Field and Service Robotics*, Vol. 5, M. Hutter and R. Siegwart, eds, Springer International Publishing, Cham, 2018, pp. 335–350. ISBN 978-3-319-67360-8 978-3-319-67361-5. http://link.springer.com/10.1007/978-3-319-67361-5_22.

[52] H. Mercier and D. Sperber, Why do humans reason? Arguments for an argumentative theory, *Behavioral and Brain Sciences* **34**(2) (2011), 57–74. doi:10.1017/S0140525X10000968. https://www.cambridge.org/core/product/identifier/S0140525X10000968.

[53] A. Mileo, Towards a neuro-symbolic cycle for human-centered explainability, *Neurosymbolic Artificial Intelligence* (2024), 1–13. doi:10.3233/NAI-240740.

[54] B. Millidge, Y. Song, T. Salvatori, T. Lukasiewicz and R. Bogacz, A Theoretical Framework for Inference and Learning in Predictive Coding Networks, in: *The Eleventh International Conference on Learning Representations*, 2023. https://openreview.net/forum?id=ZCTvSF_uVM4.

[55] L. Newman, Descartes' Epistemology, in: *The Stanford Encyclopedia of Philosophy*, Winter 2023 edn, E.N. Zalta and U. Nodelman, eds, Metaphysics Research Lab, Stanford University, 2023. https://plato.stanford.edu/archives/win2023/entries/descartes-epistemology/.

[56] D. Nitti, V. Belle and L. De Raedt, Planning in Discrete and Continuous Markov Decision Processes by Probabilistic Programming, in: *Machine Learning and Knowledge Discovery in Databases*, Vol. 9285, A. Appice, P.P. Rodrigues, V. Santos Costa, J. Gama, A. Jorge and C. Soares, eds, Springer International Publishing, Cham, 2015, pp. 327–342, Series Title: Lecture Notes in Computer Science. ISBN 978-3-319-23524-0 978-3-319-23525-7. doi:10.1007/978-3-319-23525-7$_2$0.

[57] P.K. Panigrahi and S.K. Bisoy, Localization strategies for autonomous mobile robots: A review, *Journal of King Saud University - Computer and Information Sciences* **34**(8, Part B) (2022), 6019–6039. doi:10.1016/j.jksuci.2021.02.015. https://www.sciencedirect.com/science/article/pii/S1319157821000550.

[58] J. Pearl, *Causality*, 2nd edn, Cambridge University Press, Cambridge, 2009. ISBN 978-0-521-89560-6.

[59] J. Pearl, The seven tools of causal inference, with reflections on machine learning, *Communications of the ACM* **62**(3) (2019), 54–60. doi:10.1145/3241036.

[60] M.-C. Popescu, V.E. Balas, L. Perescu-Popescu and N. Mastorakis, Multilayer perceptron and neural networks, *WSEAS Trans. Cir. and Sys.* **8**(7) (2009), 579–588.

[61] M. Qu and J. Tang, Probabilistic Logic Neural Networks for Reasoning, in: *33rd Conference on Neural Information Processing Systems*, arXiv, 2019, arXiv:1906.08495 [cs, stat]. doi:10.48550/arXiv.1906.08495. http://arxiv.org/abs/1906.08495.

[62] V. Rakesh, R. Guo, R. Moraffah, N. Agarwal and H. Liu, Linked Causal Variational Autoencoder for Inferring Paired Spillover Effects, in: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, ACM, Torino Italy, 2018, pp. 1679–1682. ISBN 978-1-4503-6014-2. doi:10.1145/3269206.3269267.

[63] J. Redmon and A. Farhadi, YOLO9000: Better, Faster, Stronger, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Honolulu, HI, 2017, pp. 6517–6525. ISBN 978-1-5386-0457-1. doi:10.1109/CVPR.2017.690. http://ieeexplore.ieee.org/document/8100173/.

[64] R. Riegel, A. Gray, F. Luus, N. Khan, N. Makondo, I.Y. Akhalwaya, H. Qian, R. Fagin, F. Barahona, U. Sharma, S. Ikbal, H. Karanam, S. Neelam, A. Likhyani and S. Srivastava, Logical Neural Networks, *Consortium for Reliability and Reproducibility (CoRR)* (2020). https://arxiv.org/abs/2006.13155v1.

[65] B. Schölkopf, F. Locatello, S. Bauer, N.R. Ke, N. Kalchbrenner, A. Goyal and Y. Bengio, Toward Causal Representation Learning, *Proceedings of the IEEE* **109**(5) (2021), 612–634. doi:10.1109/JPROC.2021.3058954. https://ieeexplore.ieee.org/document/9363924.

[66] J. Seo, S. Sim and I. Shim, Learning Off-Road Terrain Traversability with Self-Supervisions Only, *IEEE Robotics and Automation Letters* **8**(8) (2023), 4617–4624, arXiv:2305.18896 [cs]. doi:10.1109/LRA.2023.3284356. http://arxiv.org/abs/2305.18896.

[67] H. Seraji and A. Howard, Behavior-based robot navigation on challenging terrain: A fuzzy logic approach, *IEEE Transactions on Robotics and Automation* **18**(3) (2002), 308–321. doi:10.1109/TRA.2002.1019461. http://ieeexplore.ieee.org/document/1019461/.

[68] C. Sevastopoulos, K.M. Oikonomou and S. Konstantopoulos, Improving Traversability Estimation Through Autonomous Robot Experimentation, in: *Computer Vision Systems*, D. Tzovaras, D. Giakoumis, M. Vincze and A. Argyros, eds, Springer International Publishing, Cham, 2019, pp. 175–184. ISBN 978-3-030-34995-0.

[69] A. Sheth, K. Roy and M. Gaur, Neurosymbolic Artificial Intelligence (Why, What, and How), *IEEE Intelligent Systems* **38**(03) (2023), 56–62, Publisher: IEEE Computer Society. doi:10.1109/MIS.2023.3268724. https://www.computer.org/csdl/magazine/ex/2023/03/10148662/1NVf9V0YKze.

[70] D. Silver, J.A. Bagnell and A. Stentz, Learning from Demonstration for Autonomous Navigation in Complex Unstructured Terrain, *The International Journal of Robotics Research* **29**(12) (2010), 1565–1592, Publisher: SAGE Publications. doi:10.1177/0278364910369715.

[71] S. Siva, M. Wigness and J.G.R.H. Zhang, Robot Adaptation to Unstructured Terrains by Joint Representation and Apprenticeship Learning, in: *Robotics: Science and Systems XV*, Vol. 15, 2019. ISBN 978-0-9923747-5-4. p30.pdf.

[72] Y. Su, T. Wang, S. Shao, C. Yao and Z. Wang, GR-LOAM: LiDAR-based sensor fusion SLAM for ground robots on complex terrain, *Robotics and Autonomous Systems* **140** (2021), 103759. doi:10.1016/j.robot.2021.103759. https://www.sciencedirect.com/science/article/pii/S0921889021000440.

[73] T. Sztyler, G. Civitarese and H. Stuckenschmidt, Modeling and Reasoning with ProbLog: An Application in Recognizing Complex Activities, in: *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2018, pp. 259–264. doi:10.1109/PERCOMW.2018.8480299. https://ieeexplore.ieee.org/document/8480299.

[74] UK MOD, Defence Artificial Intelligence Strategy, 2022. https://www.gov.uk/government/publications.

[75] A. Valada, J. Vertens, A. Dhall and W. Burgard, AdapNet: Adaptive semantic segmentation in adverse environmental conditions, in: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Singapore, Singapore, 2017, pp. 4644–4651. ISBN 978-1-5090-4633-1. doi:10.1109/ICRA.2017.7989540. http://ieeexplore.ieee.org/document/7989540/.

[76] G. Vecchio, S. Palazzo, D.C. Guastella, D. Giordano, G. Muscato and C. Spampinato, Terrain traversability prediction through self-supervised learning and unsupervised domain adaptation on synthetic data, *Autonomous Robots* **48**(2) (2024), 4. doi:10.1007/s10514-024-10158-4.

[77] M. Visca, S. Kuutti, R. Powell, Y. Gao and S. Fallah, Deep Learning Traversability Estimator for Mobile Robots in Unstructured Environments, in: *Towards Autonomous Robotic Systems*, C. Fox, J. Gao, A. Ghalamzan Esfahani, M. Saaj, M. Hanheide and S. Parsons, eds, Springer International Publishing, Cham, 2021, pp. 203–213. ISBN 978-3-030-89177-0. doi:10.1007/978-3-030-89177-0$_2$2.

[78] W.J. von Eschenbach, Transparency and the Black Box Problem: Why We Do Not Trust AI, *Philosophy & Technology* **34**(4) (2021), 1607–1622. doi:10.1007/s13347-021-00477-0.

[79] W. Wang, Yamaha-CMU Off-Road Dataset - AirLab, 2021. https://theairlab.org/yamaha-offroad-dataset/.

[80] S. Xie, R. Song, Y. Zhao, X. Huang, Y. Li and W. Zhang, Circular Accessible Depth: A Robust Traversability Representation for UGV Navigation, *IEEE Transactions on Robotics* **39**(6) (2023), 4875–4891. doi:10.1109/TRO.2023.3308780.

[81] J. Yang, Y.Z. Ang, Z. Guo, K. Zhou, W. Zhang and Z. Liu, Panoptic Scene Graph Generation, in: *Computer Vision – ECCV 2022*, Vol. 13687, S. Avidan, G. Brostow, M. Cissé, G.M. Farinella and T. Hassner, eds, Springer Nature Switzerland, Cham, 2022, pp. 178–196, Series Title: Lecture Notes in Computer Science. ISBN 978-3-031-19811-3 978-3-031-19812-0. doi:10.1007/978-3-031-19812-0$_1$1.

[82] M. Yang, F. Liu, Z. Chen, X. Shen, J. Hao and J. Wang, CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models, in: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9588–9597. doi:10.1109/CVPR46437.2021.00947.

[83] J. Zhang and W. Lu, Sparse data machine learning for battery health estimation and optimal design incorporating material characteristics, *Applied Energy* **307** (2022), 118165. doi:10.1016/j.apenergy.2021.118165. https://www.sciencedirect.com/science/article/pii/S0306261921014380.

[84] Y. Zhao and Q. Liu, Causal ML: Python package for causal inference machine learning, *SoftwareX* **21** (2023), 101294. doi:10.1016/j.softx.2022.101294. https://www.sciencedirect.com/science/article/pii/S2352711022002126.

[85] Z. Zhao, B. Li, Y. Du, T. Fu and C. Wang, PhysORD: A Neuro-Symbolic Approach for Physics-infused Motion Prediction in Off-road Driving, in: *2024 International Conference on Intelligent Robots and Systems (IROS)*, arXiv, 2024, arXiv:2404.01596 [cs]. doi:10.48550/arXiv.2404.01596. http://arxiv.org/abs/2404.01596.