Dependable Learning-Enabled Multiagent Systems

Xiaowei Huang [a], Bei Peng [a] and Xingyu Zhao [a]
[a] *Department of Computer Science, University of Liverpool, Liverpool, U.K.*
E-mails: xiaowei.huang@liverpool.ac.uk, bei.peng@liverpool.ac.uk, xingyu.zhao@liverpool.ac.uk

Abstract. We are concerned with the construction, formal verification, and safety assurance of dependable multiagent systems. For the case where the system (agents and their environment) can be explicitly modelled, we develop formal verification methods over several logic languages, such as temporal epistemic logic and strategy logic, to reason about the knowledge and strategy of the agents. For the case where the system cannot be explicitly modelled, we study multiagent deep reinforcement learning, aiming to develop efficient and scalable learning methods for cooperative multiagent tasks. In addition to these, we develop (both formal and simulation-based) verification methods for the neural network based perception agent that is trained with supervised learning, considering its safety and robustness against attacks from an adversarial agent, and other approaches (such as explainable AI, reliability assessment, and safety argument) for the analysis and assurance of the learning components. Our ultimate objective is to combine formal methods, machine learning, and reliability engineering to not only develop dependable learning-enabled multiagent systems but also provide rigorous methods for the verification and assurance of such systems.

Keywords: Dependability, Automated Verification, Reinforcement Learning, Learning-Enabled Systems, Multiagent Systems

1. Introduction

A multiagent system consists of a set of interacting agents, each of which performs intelligently and autonomously by continuously receiving information, reasoning about the information, and taking actions. Multiagent systems are pervasive, and many complex real-world systems can be modelled as multiagent systems. In a multiagent system, individual agents have their own goals, and in the meantime a group of agents may have a collective goal. To implement their goals, the agents may need to form coalitions, share information, and find individual and group strategies. The strategies can be obtained through either *logic reasoning* or *machine learning*, depending on the available information. If the external interactions of the agent can be correctly modelled, logic reasoning can be conducted to synthesise the strategies. On the other hand, if the interactions cannot be modelled due to the lack of information, but there are positive and negative examples of the interactions, machine learning techniques can be applied to learn a strategy by optimising a learning model's performance on the example interactions.

Example 1. *The inspection of offshore energy assets [1], such as wind farms and oil platforms, can benefit from having dependable and autonomous multiagent systems, considering that they are in extreme environments which can be dangerous for human inspection. An approach that has been increasingly adopted is to employ an advanced ground control station and a set of unmanned vehicles (drones, surface vehicles, or underwater vehicles). The ground station is to determine mission goals, select key waypoints, and decide on mission characteristics such as risk tolerance and timing constraints. Then, the unmanned vehicles will, either individually or collectively, determine their strategies for the completion of the missions. The strategies may include e.g., a collective route plan that the vehicles may co-operate to complete the mission, the emergency plan to counter the failures of some vehicles, and the vehicles' individual motion plans to safely move from one waypoint to another without clashing to each other.*

Example 2. *In a smart factory, we may consider coordinated multiagent object transportation [2], in which there are multiple robots that aim to move an object to a base station, without clashing into (possibly moving) obstacles. The robot cannot move the object by itself, and only when more than one robot grasps the object simultaneously, can the object be lifted up. Moreover, once lifted up, the object cannot be moved without the robots moving towards the same direction. This task requires the robots to have a collaborative plan on reaching, lifting, and then moving the object. The object can be fragile and of significant value, so in addition to ensure the dependable execution of a robot, the dependability of their collaboration is also valuable.*

Example 3. *A complex, intelligent agent itself can also be a multiagent system where perception component, high-level planning component, motion planning component, and low-level control component can all be autonomous agents themselves. Typically, the perception component is implemented with convolutional neural networks, and the motion planning component can be implemented with deep reinforcement learning. However, other components such as the high-level planning component may be more suitably implemented with symbolic methods. The dependability of the intelligent agent will reply on not only the dependability of the component agents but also the interaction between the components. The interaction between components may affect how the failure of a component propagates through the rest of the system.*

As depicted in Fig. 1, our research revolves around the development of *dependable* multiagent systems. Various settings of multiagent systems have been studied, including systems with a learned agent and its attacker, systems with a learned agent and several cooperative agents, systems with multiple symbolic agents, and systems with multiple learning agents. The developed techniques – span across the fields of machine learning, verification and validation, and reliability engineering – are to enhance, evaluate, and/or assure the dependability of the systems. A set of dependability properties have been considered, including robustness, safety, security, interpretability, reliability, cognitive trust, and social norm, and we will extend to consider e.g., privacy and fairness in the future.

Our ultimate objective is to provide technical solutions (including theories, algorithms, and tools) for the development of *learning-enabled multiagent systems* to ensure both performance and dependability with provable guarantees. While many techniques have been developed, the foundations of them are based on either formal methods or machine learning. For example, traditional multiagent systems are mainly based on formal methods, with various specification languages and formal verification methods but without a machine learning component. On the other hand, multiagent reinforcement learning is mainly based on machine learning without rigorous specification and verification methods considered. We believe that a well-founded multiagent system with both performance and dependability achieved at the same time can only be built on three pillars: formal methods (which provides rigorous means for dependability), machine learning (which provides means for performance), and reliability engineering (which provides principled methods for the prediction, prevention, and management of failures). Our technical development for the ultimate objective will base on these three foundations.

We remark that, this is different from the neuro-symbolic computation [3, 4], which focuses on various combinations of symbolic methods and machine learning to achieve a good performance. Different from them, we require the consideration of dependability properties and not only the construction of a better performed system but also rigorous techniques to evaluate and to assure the dependability of a system with provable guarantees.

2. Research Agenda

We categorise a research agenda (sketched out at high-level in Fig. 1) in the following according to the settings of the multiagent systems under discussion. It is noted that, when mentioning agents, they can be either software agents or physical agents, because in most systems physical agents are driven by software systems.

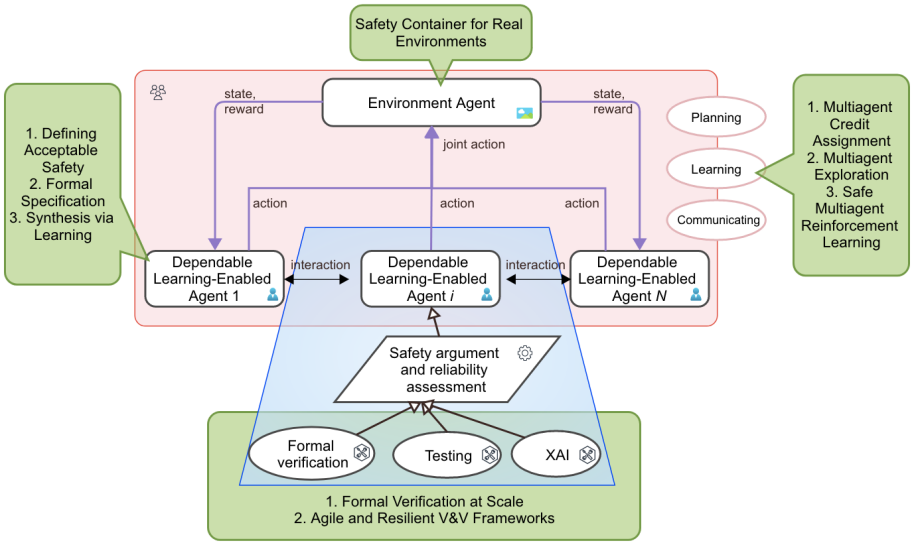

Fig. 1. An overview of the research agenda in dependable learning-enabled multiagent systems. Implemented approaches are shown in pink and blue boxes, while open challenges identified are presented in the green boxes associating with implemented topics.

Systems with a learned agent and its attacker. The first topic is to investigate the safety, security, and interpretability of a learned agent, or a trained machine learning model. Machine learning has been the de facto technique for the implementation of perception and motion control functionalities of a physical agent. Notable examples of physical agent include e.g., autonomous car, autonomous underwater vehicle, and drug target-controlled infusion. The perception component processes sensory input (such as camera imagery input, LiDAR signal, etc) to understand the appearance of the objects around the agent (such as pedestrian, obstacles, etc). The motion control component determines the behaviour of actuators according to the knowledge of the agent and its goals.

We concern the potential risks that may appear in any lifecycle stage of a machine learning model. The lifecycle of a machine learning model can be roughly split into three stages: data collection and preparation, model construction and training, and model deployment. Each lifecycle stage may be subject to attacks from either an adversarial agent (which conducts malicious behaviour) or a natural agent (which conducts random, but benign, behaviour). For example, in data collection and preparation stage, it may suffer from data poisoning attack. In model construction and training, it may be subject to backdoor attack. In model deployment stage, there might be robustness attacks, and various privacy attacks such as membership inference, model stealing, and model inversion.

It is of paramount importance to understand the existence of the attack, the extent to which the model is resistant to these attacks, and if a model can be improved against the attacks without compromising its performance. Other than the safety and security risks, it is equally important to understand why the machine learning model performs certain decisions (a.k.a. explainable AI) and whether a machine learning model can be made more interpretable.

Systems with a learned agent and several cooperative agents. As illustrated in Example 3, a complex intelligent system usually includes not only a learning agent (such as a perception component or a motion planning component) but also other cooperative agents. Despite the existence of safety and security risks on a component agent, it is known that the relevance – and severity – of the risks cannot be determined without considering the context, i.e., the whole autonomous system and its environment. To have a comprehensive understanding about this, there can be several perspectives. First of all, formalism and algorithms are required to understand how the failures of certain components may propagate or dissolve, and how they may affect the overall performance of the system. Second, considering that safety evidence (e.g., the probability of failures per demand) can be collected on individual components, it is imperative to estimate – with the support of a theoretical model – the reliability of the whole system based on the evidence.

Systems with multiple symbolic agents. In addition to multiagent systems with a learned agent as suggested above, intense research has been conducted on traditional multiagent systems, i.e., systems composed of multiple symbolic agents. Examples 1 and 2 can both be modelled with such systems. In such systems, agents have partial observation on the underlying system states. Also, agents may take different strategies, which are usually invisible to other agents. The partial observability has led to not only conceptual complexity (e.g., it is hard to comprehend nested reasoning such as "agent A_1 knows agent A_2 knows ... agent A_n knows if agent B is playing some strategy") but also computational complexity. For such systems, research interests are mainly on reasoning about agents' knowledge, strategy, collaboration, and their normalised behaviour. Computational complexity of reasoning about them is usually NP-hard.

Nevertheless, the main research topics on these systems are to develop various logic languages (or modalities) to enable the reasoning, and based on this, to consider the axiomatisation of the logic and the complexity of model checking or satisfiability. Related to the dependability, fundamental limits may be studied, e.g., given infinite memory and perfect reasoning ability, it is important to determine whether an agent or a group of agents can rightly conclude that the system can still achieve its goal even if other agents act adversarially.

Systems with multiple learning agents. Other than the aforementioned research issues, we focus on the crucial topic of learning in multiagent systems. Specifically, we focus on developing deep reinforcement learning algorithms, which combine reinforcement learning (RL) and deep learning, in multiagent systems. RL is a machine learning paradigm where an agent aims to solve a control problem by directly interacting with an unknown environment. The goal of the RL agent is to learn a sequence of actions or a policy that maximises its long-term expected total reward. In deep RL, deep neural networks are used as a function approximator, representing the policy and/or value function. This enables RL to scale to problems with high-dimensional state and/or action spaces, making it a promising approach to solving complex real-world problems, e.g., enabling a set of unmanned vehicles to automatically learn an optimal coordination strategy to complete the mission in Example 1 illustrated above.

Multiagent RL (MARL) has recently gained significantly more interests due to advances in single-agent deep RL techniques. In a MARL setting, multiple agents learn how to interact optimally in a shared uncertain environment to achieve specified long-term goals. Compared to learning in single-agent settings, learning in multiagent settings is inherently more challenging due to multiagent pathologies, e.g., the non-stationarity problem [5], curse of dimensionality [5], and relative overgeneralisation [6]. It is thus critical to study how to better overcome these issues, when developing deep MARL methods, to enable more efficient and scalable learning among agents within a multiagent system. These deep MARL approaches can potentially be applied to industrial applications such as collaborative robots in manufacturing, autonomous driving, and traffic control systems.

Fully decentralised policies are often used in MARL, due to partial observability, practical communication constraints, or as a way to deal with an intractably large joint action space. However, when training in simulation or under controlled conditions we may have access to additional information, and agents can freely share their observations and internal states. Exploiting these possibilities can greatly improve the efficiency of learning. The paradigm of centralised training with decentralised execution (CTDE) [7] has thus recently attracted considerable attention in the MARL community. However, in this setting, many critical challenges remain open. Our research focuses on addressing challenges surrounding 1) how to fully take advantage of the centralised training phase, 2) how to better extract decentralised policies that are fully consistent with their centralised counterpart, and 3) how to better represent and learn the joint action-value function that most MARL methods learn.

Other than learning in multiagent settings with fixes groups of agents and entities, we also consider the multitask MARL setting where we aim to learn control policies for variable-sized teams of agents. Many real-world multiagent settings contain tasks across which an agent must deal with varying quantities and types of cooperative agents, antagonists, or other entities. This variability in type and quantity of entities results in a combinatorial growth in the number of possible configurations, aggravating the challenge of learning control policies that generalise. We focus on training simultaneously on multiple multiagent tasks and developing training paradigms for improving generalisation and knowledge transfer within and across tasks with varying agent quantities and/or types.

3. Main Approaches

Formal Verification for Deep Learning. We consider a deep learning model as a function $f : \mathbb{R}^n \to \mathbb{R}^m$ that maps an n-dimensional data instance to a probability distribution over m classes. Intuitively, $f(\mathbf{x})_j$ returns the predictive

probability of classifying instance $\mathbf{x}$ into class j. Moreover, we write $\hat{f}(\mathbf{x})$ for the predictive class. Most deep learning verification techniques, as discussed in our recent survey [8], consider the robustness property, i.e., given a norm distance $||\cdot||_p$ for $p \geqslant 0$ and an input instance $\mathbf{x}$, it is to determine if $\hat{f}(\mathbf{x}') = \hat{f}(\mathbf{x})$ for all $\mathbf{x}'$ such that $||\mathbf{x}' - \mathbf{x}||_p < \delta$, where δ is a constant. Intuitively, robustness requires that a decision making $\hat{f}(\mathbf{x})$ needs to be invariant with respect to the input perturbations within a norm ball (of radius δ).

A number of formal verification techniques [9–14] have been developed by the group on the robustness verification of deep learning. This includes methods based on exhaustive search [9], global optimisation [10–12], game-based methods [13, 14], and symbolic interval analysis [15–17]. These methods ensure both the completeness and the soundness of the results. Moreover, in addition to the pixel perturbations measured with norm distances, we are also looking into real-world perturbations such as geometric and spatial perturbations [18].

One of the key features of our verification methods, as compared with others, is that many of them [10–14, 18] are black-box, i.e., they do not rely on the internal architecture of the neural networks. Theoretically, this brings a significant advantage that the computational complexity of the verification problem is NP-complete with respect to the number of input features, while white-box verification methods are NP-complete with respect to the number of hidden neurons. While the complexity class does not change, the number of hidden neurons can be an unlimited number of times more than that of input features, considering the current trend of deep learning on training deeper and larger networks. Also, the black-box verification means that we can work with neural networks of any scale.

Simulation-Based Verification of Deep Learning. Unlike formal verification methods which are designed to be exhaustive, simulation-based verification, following software testing methods, generates a set of test cases and uses the generated test cases to improve our confidence in the deep learning model, e.g., the in-existence of counterexamples for 10^5 test cases may contribute as a strong evidence for the satisfiability of certain property.

There are three major technical issues. The first is to design test adequacy criteria. Test adequacy criteria determine when the generation of test cases can terminate, and may also be used to guide the test case generation. Towards this, we have designed several test coverage metrics for feedforward neural networks [19] and recurrent neural networks [20], respectively. The second is to design test case generation methods. Some generation methods have been developed and tested on different neural networks, including concolic testing [21], genetic algorithm [20], and importance sampling [22]. The third is to design the test oracle, which automatically determines if a test case fails with respect to a property.

The test cases need to be generated with respect to the data distribution, and in some cases, a sampling from the unknown data distribution is impossible. To deal with this, we consider recent efforts on complexity measure [23], which determines an upper bound of the test performance based on the architecture and weights of the neural network, under the support of some deep learning theory such as the PAC Bayesian theory.

Formal Verification for Multiagent Systems. Components of a complex, autonomous system can be modelled with e.g., reactive modules [24] or knowledge-based program [25], each of which has a set S of states, a set O of observations, a transition relation T to update the states, and an observability function R that maps states to observations. It is not hard to see that a deep learning component can also be modelled in this way, such that $O = \mathbb{R}^n$ and $S = \mathbb{R}^{n+m+1}$ with m the number of classes. Intuitively, a state $s = (o, c, h)$ is a triple, where o is the input instance, c is a probability over classes, and h is a flag representing the status of the component (either processing or waiting). For two states (o, c, h) and (o', c', h'), we have $((o, c, h), (o', c', h')) \in T$ if either (1) $o' = o$, $c' = \hat{f}(o)$, $h = 1$ (processing), and $h' = 0$ (waiting), or (2) $h = 0$, $h' = 1$, and o' is on the data distribution.

The interactions of components can be modelled through either (synchronous) hand-shaking [26] or (asynchronous) message passing [27]. If the environment is known, the interactions of the system with the environment can also be modelled in this way. This can also be extended to the most general case where multiple agents run in an environment.

This formalism of modelling multiagent systems enables the logic-based formal verification of many properties, including epistemic properties [28, 29], probabilistic epistemic properties [30], strategy [31–33], diagnosability [34], social norm [35], reconfigurability [36], correlated equilibrium [37], and cognitive trust [38, 39]. We also study their computational complexity [40–45], the automatic synthesis of programs according to their specifications [46, 47], the impact of communications between agents [48], and their applications to e.g., pursuit-evasion games [49, 50].

We also consider the verification of practical learning-enabled robotics systems, focusing on the state estimation system [51], vehicle tracking system [52], and deep reinforcement learning enabled mobile robots [53].

Explainable AI (XAI). Deep learning models are highly complex non-linear functions with algorithmically generated (rather than engineered) coefficients. They are effectively "black-box", so it is difficult to explain their prediction behaviours. The goal of XAI is to create artefacts that provide a rationale for why a deep learning model generates a particular prediction for a given input. This is argued to enable stakeholders to understand and to appropriately trust deep learning models.

XAI methods can be classified by various criteria [54, Chpt. 2.2], such as model-specific vs model-agnostic or local (instance-wise) vs global (entire model). Readers are referred to [8, 55] for a comprehensive review. Our previous work [56] focuses on the class of XAI methods using local surrogate models for explaining individual predictions. Specifically, we develop a novel Bayesian extension to the most popular method in this category, Local Interpretable Model-agnostic Explanations [57], called BayLIME. BayLIME shows improved *consistency* in repeated explanations of a single prediction and *robustness* to kernel settings. Meanwhile, higher explanation *fidelity* is also expected in many settings. Arguably, these three properties are among the most desirable ones for any XAI method to have. BayLIME utilises a *Bayesian* linear regressor as the local surrogate model, which we show analytically is a "Bayesian principled weighted sum" of the prior knowledge and estimates based on new samples. The weights consist of parameters with dedicated meanings that can be either automatically fitted from samples (via Bayesian model selection) or elicited from application-specific prior knowledge (e.g., V&V methods). Our experiments show that BayLIME is superior than several state-of-the-art XAI methods in terms of those 3 desirable properties aforementioned.

Inspired by statistical fault localisation (SFL) methods in traditional software engineering, we also develop a XAI tool called DeepCover [58] that ranks the pixels using four well-known SFL measures (Zoltar, Ochiai, Tarantula and Wong-II) based on the results of running test suites constructed from random mutations of the input image. This ranking is then used by DeepCover to efficiently construct an explanation for the prediction made on the image.

Reliability Assessment and Safety Argument. Two key activities in the assurance of critical systems are: how to rigorously *assess* the risk and how to convincingly *demonstrate* the rigour of the assessment. In this regard, our group has been exploring reliability assessment modelling and safety arguments for machine learning components.

Deep learning models are subject to robustness concerns, reliability models without considering robustness evidence are not convincing. Reliability, as a user-centred property, depends on the end-users' behaviours [59]. The operational profile (OP) information (quantifying how the software will be operated [60]) should therefore be explicitly modelled in the assessment. Prior to our work [22], there is no dedicated reliability assessment model taking into account both the OP and robustness evidence. In [61], we propose a safety case framework tailored for deep learning, in which we describe an initial idea of combining robustness verification and operational testing for reliability claims. In [22], we implement this idea as a reliability assessment model, inspired by partition-based testing [62], operational-profile testing [63, 64], and deep learning robustness evaluation [65, 66]. It is *model-agnostic* and designed for pretrained deep learning models, yielding upper bounds on the *probability of miss-classifications per input* (*pmi*)[1] with confidence levels. Our reliability assessment model essentially follows the conceptualised equation of:

$$\textit{Deep Learning Reliability} = \textit{Generalisability} \times \textit{Robustness}.$$

It says, when assessing the deep learning reliability, we should not only concern how it generalises to a new datapoint (according to the future OP), but also the local robustness around it.

For certification and regulation purposes, it is equally important to *demonstrate* if the required dependability has been satisfied. As for traditional software-based systems, the emerging consensus within both industry and academia is to use safety cases for this purpose. Typically safety cases support claims of reliability, in support of safety, can be viewed as a structured way of organising arguments and evidence generated from safety analysis and reliability modelling activities. While such assurance activities are traditionally guided by consensus-based standards developed from vast engineering experience, deep learning poses new challenges due to the characteristics of deep learning models. In [67], we first propose an overall assurance framework for learning-enabled systems, presented in Claims-Arguments-Evidence (CAE) assurance cases [68]. While inspired by [69], ours is with greater emphasis on arguing for *quantitative* safety requirements. This is because the unique characteristics of machine learning

[1] *pmi* is similar to the conventional reliability metric of probability of failure per demand (*pfd*), but retrofitted for deep learning classifiers.

increase apparent non-determinism [70] that explicitly requires *probabilistic claims* to capture the uncertainties in its assurance [61, 71]. To demonstrate the overall assurance framework as an *end-to-end* methodology, we also consider important questions on how to derive and validate (quantitative) safety requirements and how to break them down to functionalities of learning components for a given learning-enabled system. A comprehensive case study based on autonomous underwater vehicles that carry out survey and asset inspection missions (cf. Examples 1 and 3) is conducted, with a video demo at https://youtu.be/akY8f5sSFpY.

Multiagent Deep Reinforcement Learning. Regarding learning in multiagent systems, a number of deep MARL algorithms have been developed by the group to enable more sample-efficient [72], robust [73, 74], scalable [75, 76], and stable [77] learning among agents in cooperative tasks. These methods are evaluated in a variety of cooperative multiagent tasks, including cooperative matrix games, multiagent particle environments [78], and the challenging multiagent StarCraft benchmark tasks [79].

Our work studies and addresses a variety of problems within MARL. For instance, value function factorisation [80] has been widely employed in value-based MARL algorithms, not much work, however, has studied the open question of understanding the representational power of these methods. We illustrate how the representational limitation of existing value function factorisation methods can prevent them from solving cooperative tasks that require significant coordination within a given timestep, and develop new deep value-based MARL algorithms to resolve these limitations in theory and in practice [74]. Our key insight is that, if we ultimately care only about the greedy optimal policy, it is more important to accurately represent the value of the optimal joint action than the sub-optimal ones. We can thus incorporate some weighting schemes into the learning framework to place more importance on representing the value of better joint actions. With the novel weighting scheme, our approach is proved to converge to the optimal policy in an idealised tabular setting, while existing method might fail to recover the optimal policy. It is also shown to be able to scale to more complex deep RL settings, with improved ability to cope with different types of environments and improved robustness to the amount of exploration performed.

Overestimation in Q-learning is a critical problem that has been extensively studied in single-agent reinforcement learning, but has received comparatively little attention in the multiagent setting. We show how overestimation in MARL can be more severe than previously acknowledged and can lead to divergent learning behaviour in practice [77]. To tackle this issue, we propose a novel regularisation-based update scheme that penalises large joint action-values that deviate from a baseline and demonstrate its effectiveness in stabilising learning. Furthermore, we propose to employ a softmax operator, which we efficiently approximate in a novel way in the multiagent setting, to further reduce the potential overestimation bias. We formally prove that our approach can reduce the overestimation bias of existing deep multiagent Q-learning algorithms. We empirically show that our approach, when applied to existing deep multiagent Q-learning algorithms, simultaneously enables stable learning, avoids severe overestimation, and improves learning performance. Our approach is also general and and can be applied to any Q-learning based MARL algorithms. This work shed light on how to design better value estimation in MARL.

To enable more efficient and scalable learning on cooperative multiagent tasks, we propose a novel deep multiagent actor-critic method that uses a centralised but factored critic and a new centralised policy gradient [76]. Compared to existing methods that learn a centralised and monolithic critic [78, 81], learning a factored critic has two main benefits. First, it can potentially scale better to tasks with a larger number of agents and/or actions. Second, compared to value-based methods [82], factoring the critic allows for a more flexible factorisation as the critic's design is not constrained. In addition, our approach uses a new centralised gradient estimator that optimises over the entire joint action spaces, rather than optimising over each agent's action space separately as in existing methods [78]. This can enable learning of more coordinated behaviour among agents, as well as the ability to escape sub-optimal solutions. While Lyu et al. [83] recently show that merely using a centralised critic does not necessarily lead to better coordination between agents, our centralised gradient estimator re-establishes the value of using centralised critics. Our approach is also general and can be readily applied to any multiagent actor-critic algorithms that learn centralised critics.

An alternative approach to achieving more scalable multiagent learning is role-based learning, in which the complex multiagent task is decomposed using roles. Each role in the multiagent task is associated with a certain sub-task and a corresponding policy. Agents taking the same role collectively learn a role policy for solving the sub-task by sharing their learning. The key question to address in role-based learning is how to come up with a set of roles to effectively decompose the task. Previous work typically predefines the task decomposition and roles. However,

this requires prior knowledge that might not be available in practice and may prevent the learning methods from transferring to different environments. To solve this problem, we propose a novel framework to automatically learn an appropriate set of roles [75]. Our key insight is that, instead of learning roles from scratch, role discovery is easier if we first decompose joint action spaces into restricted role action spaces by clustering actions according to their effects on the environment and other agents. We can then learn a role selector based on the learned effect-based action representations, which improves generalisability of role policies across actions.

Our research has also tackled some other important problems in multiagent systems such as relative overgeneralisation [6]. Relative overgeneralisation is a game-theoretic pathology that can arise when the optimal joint action's utility falls below that of a sub-optimal joint action, which prevents agents from learning the optimal joint policy. We show that this problem can be overcome by improving the joint exploration of all agents during training [73]. Specifically, we propose a novel deep MARL algorithm that learns a set of related tasks simultaneously and uses the policies of previously learned related tasks to help improve the joint exploration of all agents. Our approach is based on the key idea that, even when a target task we are interested in solving exhibits relative overgeneralisation, there may be similar tasks that do not. If their optimal actions overlap in some states with the target task, executing these simpler tasks can implicitly weight exploration to overcome relative overgeneralisation. Our empirical results show that our approach significantly outperforms current state-of-the-art deep value-based methods on target tasks that exhibit strong relative overgenralisation and in zero-shot generalisation [84] to other reward functions.

Continuous Multiagent Learning Environments. Although some multiagent benchmark environments for continuous control exist [78, 85], few environments specialise in cooperative control and even fewer model partial observability. Moreover, existing benchmarks, like the popular multiagent particle environments [78], are not complex enough to meaningfully compare methods intended for robotic control. Our prior work introduces Multiagent MuJoCo (MAMuJoCo), a new, comprehensive benchmark suite that effectively captures the nature of cooperative robotic manipulation tasks and allows the study of decentralised continuous control [76].

Based on the popular single-agent MuJoCo benchmark [86], MAMuJoCo features a wide variety of novel robotic control tasks in which multiple agents within a single robot have to solve a task cooperatively. This design offers important benefits. It facilitates comparisons to existing literature on both the fully observable single-agent domain [87], as well as settings with low-bandwidth communication [88]. More importantly, it allows for the study of novel MARL algorithms for decentralised coordination in isolation (scenarios with multiple robots may add confounding factors such as spatial exploration), which is currently a gap in the research literature. MAMuJoCo also includes scenarios with a larger and more flexible number of agents, which takes inspiration from modular robotics [89, 90]. Compared to traditional robots, modular robots are more versatile, configurable, and scalable. A large number of fundamental continuous MARL research can thus be conducted in MAMuJoCo. We believe the lack of diverse continuous benchmarks is one important factor limiting progress in continuous MARL and MAMuJoCo can potentially stimulate more progress in this research direction.

4. Open Challenges

As natural extensions of the main approaches implemented, we identify key open challenges (shown in the green boxes of Fig. 1) to the dependable learning-enabled multiagent systems as what follows.

4.1. Formal Methods for Deep Learning Enabled Multiagent Systems

One of the priorities of the group will be on establishing the foundations for a closer integration of formal methods with the deep learning-enabled multiagent systems. Formal methods and machine learning are two research fields with drastically different foundations and philosophies. Formal methods utilise mathematically rigorous techniques for the specification, development and verification of software and hardware systems. Machine learning is focused on pragmatic approaches to gradually improve a parameterised model by observing a set of training data. While historically the two fields lack communication, this trend has been changed in the past few years – with the group being one of the key players – with an outburst of research interests on the robustness verification of neural networks. Nevertheless, the connection is too narrow, with only one specific property (i.e., robustness) extensively studied. A few aspects of research will be needed to push the research forward.

Formal Specification. Existing efforts [91, 92] on specifying systems with learning components extend the temporal logics to include constructs for the description of neural network's external behaviour. These, however, might not be sufficient for the expression of safety and security properties [93] of machine learning models. A more suitable language has to consider not only the interactions of components and agents (i.e., the external behaviour of the neural network) but also the interactions of deep learning components with adversarial attacks and the training process (by considering the training dataset and the posterior distributions of the trained model). The two-player game between deep learning and adversarial attack has led to intensive studies in the past years, and many safety and security risks have been discovered, including data poisoning, backdoor, membership inference, model stealing, and model inversion. See [93] for more discussions. Up to now, these risks are studied in an ad hoc way, without a systematic research to connect them. It will be interesting if we are able to find atomic propositions and operators, based on which a set of properties can be defined for the risks. A coherent specification language, as many other specification languages such as LTL and CTL, will enable the development of a verification algorithm that can work with any properties expressible with the language.

For the consideration of safety risks and the training process, unlike traditional multiagent systems whose specification languages are based on Boolean atomic propositions, a specification language for the deep learning components might require random atomic propositions, i.e., an atomic proposition may represent a (high-dimensional) probabilistic distribution. For example, we may consider either data distribution $\mathcal{D}$ or posterior distribution $P(W|\mathbf{d})$, where $\mathbf{d}$ is a set of training instances. If so, a direct consequence of this will be the evaluation of atomic propositions, which will be non-trivial and computationally intractable. A recent attempt is included in [94].

Formal Verification at Scale. The other main technical barrier for the formal verification to be applied to real-world deep learning is the size of the neural network. As discussed earlier, we have suggested the consideration of black-box verification, which can avoid the consideration of internal architecture and therefore deal with large-scale neural networks. Nevertheless, black-box verification algorithms, currently relying on global optimisation techniques, are still hard to deal with problems with thousands of input features. More scalable methods will be needed.

Synthesis via Learning. Once we have a set of desirable properties expressed with a specification language, a direct question will be to automatically synthesise a system (or a partial program) that satisfies the properties. Unlike the program synthesis, as we have done in [46, 47], for learning-enabled multiagent systems, no prior method has been known. On the other hand, in machine learning, a popular research direction is to design better training schemes (e.g., training objectives, training algorithm) to achieve improved training results. This includes efforts on adversarial training for robust generalisation, such as [23, 95] where we consider different definitions of correlation between weights for the improvement of both robustness and generalisation, and on the training enhanced with uncertainty estimation [96]. Nevertheless, these methods can only deal with specific properties, instead of any properties expressible in a specification language. Besides, they may improve the learned model with respect to the properties, but do not provide a guarantee.

Moreover, we may consider methods from control community such as [97] which utilises Lyapunov-based method to remove undesirable actions from a reinforcement learning agent.

4.2. Safety Assurance on Learning Enabled Multiagent Systems

While formal methods provide rigorous foundations for construction and verification of dependable systems, their reliance on the (fidelity of) modelling leads to limitations (including the challenges we identified above). Reliability engineering supplements this with methods for the prediction, prevention and management of failures, and methods for the argumentation of safety and dependability.

Defining Acceptable Safety. No systems can be claimed as perfectly safe, thus safety cases normally start with a top claim that the system is "acceptably safe". Since the term "acceptably safe" is abstract and vague, we need first define the *acceptable safety* for the given system in the target operational environment. For traditional systems, we may argue that all safety requirements are satisfied implies being safe enough, while those safety requirements are derived from well-established safety standards and conventional safety analysis methods (identifying hazards, causes and mitigations, e.g., HAZOP [98, 99] and STAMP [100]). Given the lack of validated safety standards/policies

and mature safety analysis methods for disruptive technologies of AI/ML [101], it is challenging to derive safety requirements for learning-enabled systems, neither quantitatively nor qualitatively. The interaction dynamics of multi agents only makes the challenge harder, e.g., how to link the safety claims of individual agents to the safety claims of all agents? To this end, we are investigating new safety analysis methods dedicated for learning-enabled multiagent systems, taking into account ML characters while accommodating the inherent complexity of multi-agents.

Safety Container for Real Environments. It is normal that learning-enabled agent cannot achieve an acceptable level of safety. Even if it achieved a certain safety performance in simulation environments (given the availability of a massive amount of synthetic/simulated data), it will be hard to maintain the safety performance in real environments (where the data is sparse and expensive to obtain). A challenge is then on how to increase the safety in the real world to a level where stakeholders can sufficiently rely on to support safety cases. We believe safety containers (e.g., predefined leveraging simulation results) that are monitoring any behaviour outside of the container can help.

Heterogeneous Arguments based on Agile and Resilient V&V. Formal verification and statistical methods (operational/field testing or "proven-in-use") are two developed areas for the dependability of traditional systems, that fail to talk to each other. Facing the new challenges of assuring learning-enabled multiagent systems, a critical frontier of research for societal and industrial needs is to create agile and resilient V&V frameworks combining both areas to cover their own drawbacks, e.g., the scalability and misuse of abstraction of formal verification, while statistical methods cannot deal with constantly learning agents and evolving environment and become computationally expensive to achieve sufficient confidence in claims. Such new V&V frameworks may then form the evidence required by heterogeneous assurance arguments for safety claims.

4.3. Multiagent Deep Reinforcement Learning

Finally, we discuss some important open challenges for MARL the group will focus on for future research. MARL provides well performed solutions to the multiagent system construction, and we will enhance existing research with the consideration of dependability, and moreover seek novel methodologies for the integration of formal methods and reliability engineering into MARL, to achieve not only performance but also rigorous guarantees on dependability.

Multiagent Credit Assignment. In cooperative settings, joint actions typically generate only global rewards shared by all agents within the system, making it difficult for each agent to estimate its own contribution to the team's success. This is called the multiagent credit assignment problem. To tackle this, most recent work [81, 102] in MARL focuses on developing multiagent actor-critic methods that learn a centralised critic with decentralised actors, using the CTDE paradigm. However, these methods are typically limited to tens of agents. Alternative learning approaches need to be developed to handle large-scale systems with hundreds and thousands of agents. In addition, while learning a centralised critic can help address the non-stationary issue of MARL, it still has some problems. It is recently shown that, compared to learning decentralised critics, learning a centralised critic results in higher variance updates of the decentralised actors, making the policy learning less stable [83]. Furthermore, while the centralised critic has access to the information of the optimal solution available during the centralised training phase, this information is typically not effectively utilised to form a cooperative policy, leading to poor coordination behaviour among agents. How to fully take advantage of the centralised training phase is still an open problem. Therefore, we believe exploring how to reduce variance in policy updates and how to make better use of centralised training can be promising future directions.

Multiagent Exploration. In RL, exploration is crucial for gathering sufficient informative data to infer a good control policy. While there has been a lot of work on developing new exploration techniques for single-agent RL, the exploration problem is largely unstudied in MARL. Compared to single-agent settings, exploration in multiagent scenarios is more challenging, as it is more difficult to incentivise the agents to try and visit novel state-action pairs when the state and action spaces grows exponentially with the number of agents. The complex interactions and influences between different agents typically also need to be taken into account to better boost exploration. Most popular MARL methods [78, 81, 82] use simple noise-based exploration, i.e., the exploration policy is a noisy version of the actor policy. It is recently shown that, these classical exploration techniques are sub-optimal in a

MARL setting [103], which can result in slow exploration and sub-optimal solutions in complex environments, especially when rewards are sparse. How to explore effectively in general MARL settings remains an open problem. Our prior work addresses the exploration problem in MARL to some extent, by easing exploration in the joint action space via learning action effect based roles [75], or using the policies of previously solved related tasks in the harder target task to help improve the joint exploration of all agents [73]. For future research, we are interested in developing more exploration methods for MARL that consider interaction and coordination among multiple agents, to better identify states and/or actions that are worth exploring.

Safe MARL. Safety is a critical concern in many real-world multiagent systems, such as air traffic control and autonomous driving. Unfortunately, most existing MARL methods do not have safety guarantees. The convergence process of MARL algorithms is inherently stochastic, making it problematic when applied to safety-critical scenarios. Therefore, to push forward the application of MARL on safety-critical domains, it is of paramount importance to account for the safety constraints during learning and deployment. When developing and deploying MARL algorithms, we need to constrain the agent behaviours such that no unsafe states and/or actions are ever visited. The goal of the learning agents is to maximise the team-average long-term expected return, while subjecting to all safety constraints. To achieve this, we are interested in combining our work on formal verification with MARL algorithms to learn good reliable control policies in safety-critical domains.

5. Conclusion

In this article, we have reviewed some research activities that were developed in, or brought to, the University of Liverpool in the past few years on the multiagent system research, including formal verification, reliability assessment, multiagent deep reinforcement learning, explainable AI, and others. These research directions nicely form a holistic view towards the dependable learning-enabled multiagent systems, with various settings of multiagent systems considered. Based on them, we have identified that, the ultimate solution for the construction of learning-enabled multiagent systems with both performance and dependability guarantees requires a close communication of three disciplines: formal methods, machine learning, and reliability engineering.

References

[1] M. Fisher, R.C. Cardoso, E.C. Collins, C. Dadswell, L.A. Dennis, C. Dixon, M. Farrell, A. Ferrando, X. Huang, M. Jump, G. Kourtis, A. Lisitsa, M. Luckcuck, S. Luo, V. Page, F. Papacchini and M. Webster, An Overview of Verification and Validation Challenges for Inspection Robots, *Robotics* **10**(2) (2021).

[2] F. Badeig, F. Balbo, G. Scemama and M. Zargayouna, Agent-based coordination model for designing transportation applications, in: *2008 11th International IEEE Conference on Intelligent Transportation Systems*, IEEE, 2008, pp. 402–407.

[3] A.S. d'Avila Garcez, M. Gori, L.C. Lamb, L. Serafini, M. Spranger and S.N. Tran, Neural-symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning, *Journal of Applied Logics* **6**(4) (2019), 611–632.

[4] M.K. Sarker, L. Zhou, A. Eberhart and P. Hitzler, Neuro-symbolic artificial intelligence, *AI Communications* **34**(3) (2021), 197–209.

[5] P. Hernandez-Leal, B. Kartal and M.E. Taylor, A survey and critique of multiagent deep reinforcement learning, *Autonomous Agents and Multi-Agent Systems* **33**(6) (2019), 750–797.

[6] L. Panait, S. Luke and R.P. Wiegand, Biasing coevolutionary search for optimal multiagent behaviors, *IEEE Transactions on Evolutionary Computation* **10**(6) (2006), 629–645.

[7] F.A. Oliehoek, C. Amato et al., *A concise introduction to decentralized POMDPs*, Vol. 1, Springer, 2016.

[8] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu and X. Yi, A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability, *Computer Science Review* **37** (2020), 100270. doi:https://doi.org/10.1016/j.cosrev.2020.100270.

[9] X. Huang, M. Kwiatkowska, S. Wang and M. Wu, Safety verification of deep neural networks, in: *Computer Aided Verification*, LNCS, Vol. 10426, Springer International Publishing, Cham, 2017, pp. 3–29. ISBN 978-3-319-63387-9.

[10] W. Ruan, X. Huang and M. Kwiatkowska, Reachability Analysis of Deep Neural Networks with Provable Guarantees, in: *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence, IJCAI-18*, 2018, pp. 2651–2659.

[11] W. Ruan, M. Wu, Y. Sun, X. Huang, D. Kroening and M. Kwiatkowska, Global Robustness Evaluation of Deep Neural Networks with Provable Guarantees for the Hamming Distance, in: *Proc. of the 28th Int. Joint Conf. on Artificial Intelligence, IJCAI-19*, 2019, pp. 5944–5952.

[12] P. Xu, W. Ruan and X. Huang, Quantifying safety risks of deep neural networks, *Complex & Intelligent Systems* (2022).

[13] M. Wicker, X. Huang and M. Kwiatkowska, Feature-guided black-box safety testing of deep neural networks, in: *Tools and Algorithms for the Construction and Analysis of Systems*, D. Beyer and M. Huisman, eds, LNCS, Vol. 10805, Springer, Cham, 2018, pp. 408–426.

[14] M. Wu, M. Wicker, W. Ruan, X. Huang and M. Kwiatkowska, A game-based approximate verification of deep neural networks with provable guarantees, *Theoretical Computer Science* **807** (2020), 298–329.

[15] J. Li, J. Liu, P. Yang, L. Chen, X. Huang and L. Zhang, Analyzing Deep Neural Networks with Symbolic Propagation: Towards Higher Precision and Faster Verification, in: *SAS2019*, Springer, 2019, pp. 296–319.

[16] R. Li, J. Li, C.-C. Huang, P. Yang, X. Huang, L. Zhang, B. Xue and H. Hermanns, PRODeep: A Platform for Robustness Verification of Deep Neural Networks, in: *Proc. of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2020, ACM, New York, NY, USA, 2020, pp. 1630–1634.

[17] P. Yang, J. Li, J. Liu, C.-C. Huang, R. Li, L. Chen, X. Huang and L. Zhang, Enhancing Robustness Verification for Deep Neural Networks Via Symbolic Propagation, *Form. Asp. Comput.* **33**(3) (2021), 407–435.

[18] F. Wang, P. Xu, X. Huang and W. Ruan, GeoRobust: Evaluating Geometric Robustness of Neural Networks with Provable Guarantees, in: *submitted*, 2022.

[19] Y. Sun, X. Huang, D. Kroening, J. Sharp, M. Hill and R. Ashmore, Structural Test Coverage Criteria for Deep Neural Networks, in: *ICSE'19–Companion*, IEEE Press, Piscataway, NJ, USA, 2019, pp. 320–321.

[20] W. Huang, Y. Sun, X. Zhao, J. Sharp, W. Ruan, J. Meng and X. Huang, Coverage-Guided Testing for Recurrent Neural Networks, *IEEE Transactions on Reliability* (2021), 1–16. doi:10.1109/TR.2021.3080664.

[21] Y. Sun, M. Wu, W. Ruan, X. Huang, M. Kwiatkowska and D. Kroening, Concolic Testing for Deep Neural Networks, in: *ASE'18*, ACM, 2018, pp. 109–119.

[22] X. Zhao, W. Huang, A. Banks, V. Cox, D. Flynn, S. Schewe and X. Huang, Assessing the Reliability of Deep Learning Classifiers Through Robustness Evaluation and Operational Profiles, in: *AISafety'21 Workshop at IJCAI'21*, Vol. 2916, 2021.

[23] G. Jin, X. Yi, L. Zhang, L. Zhang, S. Schewe and X. Huang, How does Weight Correlation Affect the Generalisation Ability of Deep Neural Networks, in: *NeurIPS'20*, 2020.

[24] R. Alur and T.A. Henzinger, Reactive Modules, *Formal Methods in System Design* **15**(1) (1999), 7–48.

[25] R. Fagin, J.Y. Halpern, Y. Moses and M.Y. Vardi, Knowledge-based programs, *Distributed Computing* **10**(4) (1997), 199–225.

[26] R. Milner, *A Calculus of Communicating Systems*, Springer, 1980.

[27] C.A.R. Hoare, Communicating Sequential Processes, *Commun. ACM* **21**(8) (1978), 666–677–. doi:10.1145/359576.359585.

[28] X. Huang and R. van der Meyden, The Complexity of Epistemic Model Checking: Clock Semantics and Branching Time, in: *ECAI 2010*, Vol. 215, H. Coelho, R. Studer and M.J. Wooldridge, eds, 2010, pp. 549–554.

[29] X. Huang, C. Luo and R. van der Meyden, Improved bounded model checking for a fair branching-time temporal epistemic logic, in: *AAMAS2010*, W. van der Hoek, G.A. Kaminka, Y. Lespérance, M. Luck and S. Sen, eds, 2010, pp. 1403–1404.

[30] X. Huang, C. Luo and R. van der Meyden, Symbolic model checking of probabilistic knowledge, in: *TARK-2011*, 2011, pp. 177–186.

[31] X. Huang, Bounded planning for strategic goals with incomplete information and perfect recall, in: *AAMAS 2013, Saint Paul, MN, USA, May 6-10, 2013*, 2013, pp. 885–892.

[32] X. Huang and R. van der Meyden, Symbolic Model Checking Epistemic Strategy Logic, in: *AAAI2014*, C.E. Brodley and P. Stone, eds, 2014, pp. 1426–1432.

[33] X. Huang, Bounded model checking of strategy ability with perfect recall, *Artif. Intell.* **222** (2015), 182–200. doi:10.1016/j.artint.2015.01.005.

[34] X. Huang, Diagnosability in concurrent probabilistic systems, in: *AAMAS 2013*, 2013, pp. 853–860.

[35] X. Huang, J. Ruan, Q. Chen and K. Su, Normative Multiagent Systems: A Dynamic Generalization, in: *IJCAI2016*, 2016, pp. 1123–1129–.

[36] X. Huang, Q. Chen, J. Meng and K. Su, Reconfigurability in Reactive Multiagent Systems, in: *IJCAI 2016*, 2016, pp. 315–321.

[37] X. Huang and J. Ruan, ATL Strategic Reasoning Meets Correlated Equilibrium, in: *IJCAI 2017*, 2017, pp. 1102–1108.

[38] X. Huang and M.Z. Kwiatkowska, Reasoning about Cognitive Trust in Stochastic Multiagent Systems, in: *AAAI 2017*, 2017, pp. 3768–3774.

[39] X. Huang, M. Kwiatkowska and M. Olejnik, Reasoning about Cognitive Trust in Stochastic Multiagent Systems, *ACM Trans. Comput. Log.* **20**(4) (2019), 21:1–21:64. doi:10.1145/3329123.

[40] X. Huang, K. Su and C. Zhang, Probabilistic Alternating-Time Temporal Logic of Incomplete Information and Synchronous Perfect Recall, in: *AAAI 2012*, 2012.

[41] X. Huang and C. Luo, A logic of probabilistic knowledge and strategy, in: *AAMAS 2013*, 2013, pp. 845–852.

[42] X. Huang and R. van der Meyden, A Temporal Logic of Strategic Knowledge, in: *KR2014*, C. Baral, G.D. Giacomo and T. Eiter, eds, 2014.

[43] X. Huang, Q. Chen and K. Su, The Complexity of Model Checking Succinct Multiagent Systems, in: *IJCAI 2015*, 2015, pp. 1076–1082.

[44] X. Huang and M. Kwiatkowska, Model Checking Probabilistic Knowledge: A PSPACE Case, in: *AAAI 2016*, 2016, pp. 2516–2522.

[45] X. Huang and R. van der Meyden, An Epistemic Strategy Logic, *ACM Trans. Comput. Log.* **19**(4) (2018), 26:1–26:45. doi:10.1145/3233769.

[46] X. Huang and R. van der Meyden, Symbolic Synthesis of Knowledge-based Program Implementations with Synchronous Semantics, in: *TARK2013*, B.C. Schipper, ed., 2013.

[47] X. Huang and R. van der Meyden, Symbolic Synthesis for Epistemic Specifications with Observational Semantics, in: *TACAS 2014*, E. Ábrahám and K. Havelund, eds, 2014, pp. 455–469.

[48] X. Huang, Q. Chen and K. Su, Strengthening Agents Strategic Ability with Communication, in: *AAAI 2016*, 2016, pp. 2509–2515.

[49] X. Huang and R. van der Meyden, Synthesizing Strategies for Epistemic Goals by Epistemic Model Checking: An Application to Pursuit Evasion Games, in: *AAAI 2012*, 2012.

[50] X. Huang, P. Maupin and R. van der Meyden, Model Checking Knowledge in Pursuit Evasion Games, in: *IJCAI2011*, T. Walsh, ed., 2011, pp. 240–245.

[51] Y. Sun, Y. Zhou, S. Maskell, J. Sharp and X. Huang, Reliability Validation of Learning Enabled Vehicle Tracking, in: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9390–9396. doi:10.1109/ICRA40945.2020.9196932.

[52] W. Huang, Y. Zhou, Y. Sun, J. Sharp, S. Maskell and X. Huang, Practical Verification of Neural Network Enabled State Estimation System for Robotics, in: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 7336–7343. doi:10.1109/IROS45743.2020.9340720.

[53] Y. Dong, X. Zhao and X. Huang, Dependability Analysis of Deep Reinforcement Learning based Robotics and Autonomous Systems through Probabilistic Model Checking, in: *IROS2022*, 2022.

[54] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.*, E-book on leanpub.com, 2020.

[55] A. Adadi and M. Berrada, Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), *IEEE Access* **6** (2018), 52138–52160.

[56] X. Zhao, W. Huang, X. Huang, V. Robu and D. Flynn, BayLIME: Bayesian local interpretable model-agnostic explanations, in: *Proc. of the 37th Conference on Uncertainty in Artificial Intelligence*, C. de Campos and M.H. Maathuis, eds, UAI'21, Vol. 161, PMLR, 2021, pp. 887–896.

[57] M.T. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, in: *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, ACM, New York, NY, USA, 2016, pp. 1135–1144.

[58] Y. Sun, H. Chockler, X. Huang and D. Kroening, Explaining Image Classifiers Using Statistical Fault Localization, in: *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox and J.-M. Frahm, eds, Springer International Publishing, Cham, 2020, pp. 391–406. ISBN 978-3-030-58604-1.

[59] B. Littlewood and L. Strigini, Software reliability and dependability: A roadmap, in: *Proc. of the Conference on The Future of Software Engineering*, ICSE '00, ACM, New York, NY, USA, 2000, pp. 175–188. ISBN 1-58113-253-0. doi:10.1145/336512.336551.

[60] J. Musa, Operational profiles in software-reliability engineering, *IEEE Software* **10**(2) (1993), 14–32.

[61] X. Zhao, A. Banks, J. Sharp, V. Robu, D. Flynn, M. Fisher and X. Huang, A Safety Framework for Critical Systems Utilising Deep Neural Networks, in: *Computer Safety, Reliability, and Security*, A. Casimiro, F. Ortmeier, F. Bitsch and P. Ferreira, eds, LNCS, Vol. 12234, Springer, 2020, pp. 244–259.

[62] D. Hamlet and R. Taylor, Partition testing does not inspire confidence, *IEEE Tran. on Software Engineering* **16**(12) (1990), 1402–1411.

[63] L. Strigini and B. Littlewood, Guidelines for Statistical Testing, Technical Report, City, University of London, 1997. http://openaccess.city.ac.uk/254/.

[64] X. Zhao, K. Salako, L. Strigini, V. Robu and D. Flynn, Assessing safety-critical systems from operational testing: A study on autonomous vehicles, *Information and Software Technology* **128** (2020), 106393.

[65] N. Carlini and D. Wagner, Towards Evaluating the Robustness of Neural Networks, in: *IEEE Symp. on Security and Privacy (SP)*, IEEE, San Jose, CA, USA, 2017, pp. 39–57. doi:10.1109/SP.2017.49.

[66] S. Webb, T. Rainforth, Y.W. Teh and M.P. Kumar, A statistical approach to assessing neural network robustness, in: *7th Int. Conf. Learning Representations (ICLR'19)*, OpenReview.net, New Orleans, LA, USA, 2019.

[67] X. Zhao, W. Huang, V. Bharti, Y. Dong, V. Cox, A. Banks, S. Wang, S. Schewe and X. Huang, Reliability Assessment and Safety Arguments for Machine Learning Components in Assuring Learning-Enabled Autonomous Systems, arXiv, 2021. doi:10.48550/ARXIV.2112.00646.

[68] R. Bloomfield and P. Bishop, Safety and assurance cases: past, present and possible future – an Adelard perspective, in: *Making Systems Safer*, C. Dale and T. Anderson, eds, Springer London, London, 2010, pp. 51–67. ISBN 978-1-84996-086-1.

[69] R. Bloomfield, G. Fletcher, H. Khlaaf, L. Hinde and P. Ryan, Safety Case Templates for Autonomous Systems, *arXiv preprint arXiv:2102.02625* (2021).

[70] Johnson, C. W., The increasing risks of risk assessment: On the rise of artificial intelligence and non-determinism in safety-critical systems, in: *the 26th Safety-Critical Systems Symposium*, Safety-Critical Systems Club, York, UK., 2018, p. 15.

[71] R. Bloomfield and J. Rushby, Assurance 2.0: A Manifesto, *arXiv preprint arXiv:2004.10474* (2020).

[72] S. Iqbal, C.A.S. De Witt, B. Peng, W. Böhmer, S. Whiteson and F. Sha, Randomized Entity-wise Factorization for Multi-Agent Reinforcement Learning, in: *International Conference on Machine Learning*, PMLR, 2021, pp. 4596–4606.

[73] T. Gupta, A. Mahajan, B. Peng, W. Böhmer and S. Whiteson, Uneven: Universal value exploration for multi-agent reinforcement learning, in: *International Conference on Machine Learning*, PMLR, 2021, pp. 3930–3941.

[74] T. Rashid, G. Farquhar, B. Peng and S. Whiteson, Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning, in: *Advances in neural information processing systems*, 2020, pp. 10199–10210.

[75] T. Wang, T. Gupta, A. Mahajan, B. Peng, S. Whiteson and C. Zhang, Rode: Learning roles to decompose multi-agent tasks, in: *International Conference on Learning Representations*, 2021.

[76] B. Peng, T. Rashid, C. Schroeder de Witt, P.-A. Kamienny, P. Torr, W. Böhmer and S. Whiteson, Facmac: Factored multi-agent centralised policy gradients, in: *Advances in Neural Information Processing Systems*, 2021.

[77] L. Pan, T. Rashid, B. Peng, L. Huang and S. Whiteson, Regularized Softmax Deep Multi-Agent Q-Learning, in: *Advances in Neural Information Processing Systems*, 2021.

[78] R. Lowe, Y. Wu, A. Tamar, J. Harb, O.P. Abbeel and I. Mordatch, Multi-agent actor-critic for mixed cooperative-competitive environments, in: *Advances in Neural Information Processing Systems*, 2017, pp. 6379–6390.
[79] M. Samvelyan, T. Rashid, C.S. de Witt, G. Farquhar, N. Nardelli, T.G.J. Rudner, C.-M. Hung, P.H.S. Torr, J. Foerster and S. Whiteson, The StarCraft Multi-Agent Challenge, *arXiv preprint arXiv:1902.04043* (2019).
[80] D. Koller and R. Parr, Computing factored value functions for policies in structured MDPs, in: *Proceedings of IJCAI*, 1999, pp. 1332–1339.
[81] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli and S. Whiteson, Counterfactual Multi-Agent Policy Gradients, in: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
[82] T. Rashid, M. Samvelyan, C.S. Witt, G. Farquhar, J. Foerster and S. Whiteson, QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning, in: *International Conference on Machine Learning*, 2018, pp. 4292–4301.
[83] X. Lyu, Y. Xiao, B. Daley and C. Amato, Contrasting Centralized and Decentralized Critics in Multi-Agent Reinforcement Learning, in: *Proceedings of the 20th International Conference on Autonomous Agents and Multi-Agent Systems*, 2021.
[84] D. Borsa, A. Barreto, J. Quan, D. Mankowitz, R. Munos, H. Van Hasselt, D. Silver and T. Schaul, Universal successor features approximators, in: *International Conference on Learning Representations*, 2019.
[85] S. Liu, G. Lever, J. Merel, S. Tunyasuvunakool, N. Heess and T. Graepel, Emergent coordination through competition, *arXiv preprint arXiv:1902.07151* (2019).
[86] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba, Openai gym, *arXiv preprint arXiv:1606.01540* (2016).
[87] OpenAI, openai/baselines, OpenAI, 2020, original-date: 2017-05-24T01:58:13Z.
[88] T. Wang, R. Liao, J. Ba and S. Fidler, NerveNet: learning structured policy with graph neural networks, in: *6th International Conference on Learning Representations, ICLR*, 2018.
[89] M. Yim, Y. Zhang and D. Duff, Modular robots, *IEEE Spectrum* **39**(2) (2002), 30–34.
[90] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo and S. Murata, Distributed self-reconfiguration of M-TRAN III modular robotic system, *The International Journal of Robotics Research* **27**(3–4) (2008), 373–386.
[91] A. Balakrishnan, A.G. Puranic, X. Qin, A. Dokhanchi, J.V. Deshmukh, H. Ben Amor and G. Fainekos, Specifying and Evaluating Quality Metrics for Vision-based Perception Systems, in: *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019, pp. 1433–1438. doi:10.23919/DATE.2019.8715114.
[92] S. Bensalem, C.-H. Cheng, X. Huang, P. Katsaros, A. Molin, D. Nickovic and D. Peled, Formal Specification for Learning-Enabled Autonomous Systems, in: *FoMLAS2022*, 2022.
[93] X. Huang, G. Jin and W. Ruan, *Machine Learning Safety*, Springer, 2022.
[94] X. Huang, W. Ruan, Q. Tang and X. Zhao, Bridging Formal Methods and Machine Learning with Global Optimisation, in: *ICFEM2022*, 2022.
[95] G. Jin, X. Yi, W. Huang, S. Schewe and X. Huang, Enhancing Adversarial Training With Second-Order Statistics of Weights, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15273–15283.
[96] K. Cai, C.X. Lu and X. Huang, STUN: Self-Teaching Uncertainty Estimation for Place Recognition, in: *IROS2022*, 2022.
[97] Y. Chow, O. Nachum, E.A. Duéñez-Guzmán and M. Ghavamzadeh, A Lyapunov-based Approach to Safe Reinforcement Learning, in: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018, pp. 8103–8112.
[98] H. Lawley, Operability studies and hazard analysis, *Chem. Eng. Prog.* **70**(4) (1974), 45–56.
[99] Y. Qi, P.R. Conmy, W. Huang, X. Zhao and X. Huang, A Hierarchical HAZOP-Like Safety Analysis for Learning-Enabled Systems, in: *AISafety'22 Workshop at IJCAI'22*, 2022.
[100] N.G. Leveson, *Engineering a safer world: Systems thinking applied to safety*, The MIT Press, 2016.
[101] R. Bloomfield, H. Khlaaf, P.R. Conmy and G. Fletcher, Disruptive Innovations and Disruptive Assurance: Assuring Machine Learning and Autonomy, *Computer* **52**(9) (2019), 82–89.
[102] M. Zhou, Z. Liu, P. Sui, Y. Li and Y.Y. Chung, Learning implicit credit assignment for cooperative multi-agent reinforcement learning, in: *Advances in Neural Information Processing Systems*, 2020, pp. 11853–11864.
[103] A. Mahajan, T. Rashid, M. Samvelyan and S. Whiteson, Maven: Multi-agent variational exploration, in: *Advances in Neural Information Processing Systems*, 2019, pp. 7611–7622.