RESEARCH STATEMENT

KISHOR JOTHIMURUGAN
keyshor.github.io

My research vision is to **enhance machine learning tools and techniques using formal methods in order to enable building of reliable, interpretable and intelligent systems.** Machine learning (ML) has become ubiquitous in all areas of computer science and has led to immense progress in many disciplines including artificial intelligence (AI), robotics and computer vision. Despite this progress, modern machine learning, especially deep learning (DL), suffers from a myriad of shortcomings preventing us from realizing its true potential. For example, ML/DL techniques often rely on large amounts of data, causing researchers to focus primarily on domains where the required data is readily available or easily generated—e.g., code generation models trained on data from online repositories. Furthermore, neural networks (NN) are not human interpretable which makes it challenging, sometimes even impossible, to explain the reasoning behind the decisions made by such models. It is often the case that ML models simply mimic the patterns in the training data without any understanding of the logic behind the observed patterns. The lack of formal specifications further exacerbates the difficulty in understanding and verifying ML models. Motivated by the many successes of formal methods in greatly improving key areas of engineering such as distributed systems, cyber-physical systems (CPS), hardware design and verification, I envision that many limitations of ML can be overcome by incorporating ideas from **formal methods** (FM) and **programming languages** (PL).

As a first step, my research so far has focused on applications of formal methods in **reinforcement learning** (RL). In RL, one typically specifies the learning objective using a reward function (e.g., providing higher rewards for desirable outcomes) and the aim is to synthesize a policy (typically an NN) to maximize the expected reward. Recent research has primarily focused on scaling RL to high-dimensional control systems with complex dynamics such as robotic arms and autonomous cars. However, many issues arise when attempting to synthesize policies for **complex specifications** using existing RL algorithms. My dissertation research tackles these issues using FM techniques and can be summarized as follows.

RL from logical specifications. Complex tasks are often challenging to express using reward functions. This line of work focuses on designing RL algorithms for learning to perform tasks expressed in logical specification languages such as Linear Temporal Logic (LTL). I have contributed to the theoretical foundations of RL from LTL specifications [1] by showing impossibility results regarding reward generation and PAC learning from LTL objectives. Furthermore, I have designed a specification language (based on LTL) for specifying robotics tasks [3] along with practical RL algorithms to learn to satisfy such specifications (in the finite-horizon setting) [4, 5].

Compositional RL. This research direction aims to build RL algorithms for learning to perform complex tasks by decomposing the given task into simpler subtasks. I have designed RL algorithms that use decompositions obtained from user provided abstract states [6] or from the structure in a given specification [4]. I have also desinged a compositional RL algorithm which enables training policies that generalize to multiple tasks [7].

Verification of NN controllers. Verifying safety of neural policies trained using RL is a challenging problem and current techniques do not scale well to long horizons and complex tasks. I have developed a compositional verification framework that leverages existing techniques and inductive reasoning to scale verification to long (potentially infinite) horizons [2].

I have published my work in premier **FM and verification** venues (CAV, LICS, EMSOFT) as well as in flagship **ML and AI** venues (NeurIPS, AISTATS). I believe that I am uniquely suited to drive interdisciplinary research facilitating collaborations across multiple areas including robotics, CPS, FM, PL, AI, ML and optimization.

Reinforcement Learning from Logical Specifications

One key shortcoming of RL is that the user must manually encode the desired task using a real-valued reward function, which can be challenging for several reasons. First, for complex tasks with multiple objectives and constraints, the user must manually devise a single reward function that balances different parts of the task.

Second, the state space must often be extended to encode the reward—e.g., adding indicators that keep track of which subtasks have been completed. Third, oftentimes, different reward functions can encode the same task, and the choice of reward function can have a big impact on the convergence of the RL algorithm. A primary research direction of mine focuses on designing RL algorithms that can learn policies directly from temporal specifications. Such algorithms eliminate the need for handcrafted reward functions and, more importantly, encourage the users to write formal specifications, which are useful for explaining the learning objective.

Hardness results. I have worked on analyzing the theoretical limitations in obtaining RL algorithms when using logical specifications instead of reward functions [1]. I showed that, in the infinite horizon setting, even for simple reachability specifications, there *do not* exist reward functions such that maximizing the expected value of discounted-sum rewards corresponds to maximizing the probability of reaching the goal. Furthermore, I proved that it is *impossible* to obtain a probably approximately correct (PAC) RL algorithm for LTL specifications without more knowledge about the environment than what is typically assumed in RL. This is due to the fact that LTL specifications are *non-robust*—i.e., small changes in the transition probabilities of the environment can lead to drastic changes in the nature of optimal policies.

Systematic generation of rewards. Despite the hardness results, RL practitioners have managed to train policies for simple specifications such as reachability using handcrafted reward functions by restricting the problem to the finite-horizon setting. In order to reduce human effort involved in reward generation, I have designed a specification language, SPECTRL, for specifying robotics tasks in which one can express reachability objectives, safety constraints and compose specifications via temporal sequencing and disjunction operators.

Based on common practices in reward design, I have developed an algorithm to automatically generate shaped rewards from SPECTRL specifications [3]. As opposed to most existing techniques, this approach generates *non-sparse* rewards—i.e., meaningful rewards are provided at intermediate states for making progress towards the overall objective. The algorithm also constructs an automaton from the given specification which is used to monitor the progress of the RL agent and provide it additional information necessary for inferring optimal actions. The reward generation algorithm is available as an open-source tool [3] and experiments on continuous control environments show that it is quite effective at generating reward functions that guide RL algorithms towards optimal solutions.

Specifying behaviours of multi-agent systems. SPECTRL can also be used to formally specify the objective of each agent in a multi-agent system where the agents may have competing goals. In such cases, it is crucial to ensure that the policies of the agents form a Nash equilibrium, thereby preventing any agent from deviating from its policy. Restricting ourselves to the finite horizon setting, I have designed a multi-agent RL algorithm, HIGHNASH, that guarantees that the learned policies form an ϵ-Nash equilibrium with high probability [5]. Furthermore, it uses heuristics to learn policies that achieve high social welfare. HIGHNASH is the first RL algorithm for learning social welfare maximizing equilibria in unknown stochastic environments. Experiments on finite-state environments show that HIGHNASH is able to learn policies with higher social welfare as compared to policies trained using existing multi-agent RL algorithms.

Compositional Reinforcement Learning

Many RL algorithms are successful in learning to perform short-horizon tasks—i.e., tasks that can be completed within a small number of steps. However, they often do not scale well to long-horizon tasks due to a myriad of reasons including delayed rewards, failure to utilize temporal structure in the task and high cost of exploration. The key idea behind compositional RL is to decompose the overall task into multiple simpler subtasks.

Hierarchical RL using abstract states. In hierarchical RL, the decomposition is usually based on subgoals that are also learned by a high-level RL agent. In [6], I showed that, in the presence of user provided *abstract states* that denote regions of interest for exploration, one can obtain more sample-efficient hierarchical RL algorithms. In particular, under some conditions on the abstract states, we can show that *abstract interpretation* applied to value iteration using the interval abstract domain converges and leads to near-optimal policies. Hence, vanilla RL for learning to reach different abstract states combined with the high-level abstract value iteration procedure for planning yields a principled approach to hierarchical RL.

Compositional RL for SPECTRL specifications. A key advantage of using formal specifications as opposed to reward functions is that the structure in the specification provides valuable information during the search for optimal policies. In particular, I showed that SPECTRL specifications can be compiled to graph structures called *abstract graphs* in which edges denote subtasks and vertices denote abstract states. Utilizing this structure, I developed an algorithm, DIRL, that interleaves traditional planning over the abstract graph and RL for training subtask policies to synthesize policies that are structured and interpretable; the overall policy includes the high-level plan which is a path in the graph and the subtask policies which are NN controllers.

I implemented DIRL in an open-source tool [4]. Evaluation was performed on multiple continuous control environments, including realistic MUJOCO-based environments such as the robotic arm. Empirical results show that DIRL scales to complex long-horizon specifications whereas the performance of existing approaches degrades rapidly with increase in the complexity of the specification.

Generalizing to multiple tasks. In existing methods for compositional RL, subtask policies are often trained using vanilla RL either separately, or together with the aim of solving a specific overall task. However, in many scenarios, it is preferable to train subtask policies that can be used to perform a wide variety of tasks. In a recent work [7], I developed a method for training subtask policies that can be used to perform multiple tasks, where a task is a sequence of subtasks. The key insight is to model the problem as a two-player zero-sum game in which one player represents the learning agent and the other is an adversary who selects the overall task. The resulting game can be solved compositionally, leading to a sample-efficient procedure for robust multi-task RL.

Verification of Neural Network Controllers

Recent progress in verifying the safety of NN policies (trained using RL) in closed-loop systems has led to algorithms that overapproximate the set of states reached at different time steps. However, these approaches do not scale well to long horizons as the approximation errors grow with the number of time steps. I have proposed a compositional solution to this problem in which the overall verification problem is reduced to many short-horizon verification instances [2]. As before, such a decomposition can either be user provided or obtained from a formal specification. My solution also includes an algorithm to generate pre- and post-conditions such that the verification results corresponding to the smaller instances compose well to guarantee safety for the overall task. This approach can also be applied to the multi-task setting which enables the ability to run verification once to guarantee the safety of a given controller in multiple scenarios. I used this approach to verify that a controller trained to steer a small scale F1/10th autonomous car guarantees that the car will remain safe when deployed in *any* racing track that is constructed using five different kinds of track segments. This is the first application of NN verification to guarantee safety of a realistic (CPS) system for arbitrarily long horizons.

Future Vision and Outlook

My long-term research vision is to incorporate formal reasoning in ML tools and techniques to enable building of reliable, interpretable and intelligent systems. I aim for a future where AI systems leverage the benefits of both symbolic reasoning and machine learning while minimizing their drawbacks. My research so far has made progress towards this goal by demonstrating the effectiveness of FM techniques such as formal specifications, abstractions and inductive reasoning in improving interpretability and reliability of RL while, at the same time, making RL easier to apply to new problems. In the future, I will continue my research on advancing RL using FM and also explore similar ideas in other paradigms of ML such as supervised and unsupervised learning while seeking collaborations with researchers working on topics such as robotics, computer vision and NLP. A few concrete research directions that I plan to pursue are discussed below.

Logical reasoning in RL. I will continue my work on leveraging FM to improve RL in order to achieve better sample complexity, reliability, interpretability and usability. In the line of work on RL from formal specifications, hardness results imply that existing specification languages such as LTL are not well suited for RL. I will work on developing specification languages that are expressive and user-friendly while simultaneously admitting RL algorithms with strong theoretical guarantees. I have already started analyzing different ways of defining a time-discounted semantics for LTL in this context. On the practical front, motivated by the promise shown by compositional approaches, I will explore algorithms for decomposing a given task into simpler subtasks that represent logical steps required to complete the whole task. This will involve considering a wide range of ways of specifying the overall objective and handling the lack of an exact model of the environment. Finally, I am also very interested in exploring statistical verification algorithms to provide guarantees about NN policies in the model-free setting; such approaches are more widely applicable to real-life scenarios. I believe that progress in the above directions will enable novel applications of RL in robotics, embedded systems and other domains.

Specification-guided learning. Traditional synthesis has always been associated with formal specifications such as LTL, Hoare triples and input-output examples. Viewing ML as a method for program synthesis (where the programs are ML models), it is often possible to write (partial) formal specifications for the model being trained. For instance, requiring adversarial robustness of a vision model at some input is an example of such a specification. Existing work, including my work on RL from logical specifications, provides evidence that one can leverage such specifications to improve learning. Looking beyond RL, I plan to develop learning algorithms that are guided by formal specifications in other domains. One such domain is large-language models for code generation where the user can often provide formal requirements for the generated code in the form of assert statements, in addition to a natural language prompt. I envision that this research direction will lead to new domain-specific learning algorithms that require fewer data and generate models that are interpretable and verifiable. I hope for a future where ML engineers think carefully about formal requirements of ML models.

As a **formal methods researcher**, I am quite excited to explore new and interesting applications of FM in ML. My research will involve collaborations with researchers in academia as well as in the industry working on various topics in FM/ML and I strongly believe that my research program will lead to a significant progress in AI.

References

[1] R. Alur, S. Bansal, O. Bastani, and **K. Jothimurugan**. A Framework for Transforming Specifications in Reinforcement Learning. *Springer Festschrift in honor of Prof. Tom Henzinger*, 2022.

[2] R. Ivanov, **K. Jothimurugan**, S. Hsu, S. Vaidya, R. Alur, and O. Bastani. Compositional Learning and Verification of Neural Network Controllers. In *International Conference on Embedded Software*, 2021.

[3] **K. Jothimurugan**, R. Alur, and O. Bastani. A Composable Specification Language for Reinforcement Learning Tasks. In *Advances in Neural Information Processing Systems*, 2019.

[4] **K. Jothimurugan**, S. Bansal, O. Bastani, and R. Alur. Compositional Reinforcement Learning from Logical Specifications. In *Advances in Neural Information Processing Systems*, 2021.

[5] **K. Jothimurugan**, S. Bansal, O. Bastani, and R. Alur. Specification-Guided Learning of Nash Equilibria with High Social Welfare. In *International Conference on Computer Aided Verification*, 2022.

[6] **K. Jothimurugan**, O. Bastani, and R. Alur. Abstract Value Iteration for Hierarchical Reinforcement Learning. In *International Conference on Artificial Intelligence and Statistics*, 2021.

[7] **K. Jothimurugan**, S. Hsu, O. Bastani, and R. Alur. Robust Option Learning for Compositional Generalization. In *Deep RL Workshop, NeurIPS*, 2022.