Safety Beyond Verification: The Need for Continual, User-Driven Assessment of AI Systems

Siddharth Srivastava[1], **Georgios Fainekos**[2], **Pulkit Verma**[3], and **Daniel Bramblett**[1]
[1]Arizona State University
[2]Toyota Motor North America, Research & Development
[3]Massachusetts Institute of Technology
{siddharths,drbrambl}@asu.edu, georgios.fainekos@toyota.com, pulkitv@mit.edu

Abstract

How should we assess the safety and functionality of taskable AI systems that are designed to continually learn and solve user-desired tasks in user-specific environments? From household robotics to digital assistants that can make potentially dangerous changes to their operational environments, this question is central to realizing the promise of AI.

We investigate why answering this question requires more than an extrapolation of existing paradigms for verification and validation, and identify concrete desiderata and promising directions for research on formal assessment of AI systems.

1 Introduction

The vast majority of today's engineered systems operate in an ecosystem where well defined Operating Design Domains (ODD) yields safety. Designers play a key role in evaluating safety and defining operational envelopes for systems with narrow scope of functionality. E.g., conventional automobile systems run through various empirical tests, semi-formal and formal verification pipelines. In addition, they are supported by an ecosystem of product support, safety and maintenance organizations, all of which make system expertise readily available to non-expert users. Taskable AI systems (henceforth referred to as "AI systems") invalidate both of these conventional avenues for ensuring safe operation. Such systems are commonly formulated as agents that carry out some form of sequential decision making, a.k.a. planning. Such systems often utilize machine learning to improve their computational performance, although our discussion also applies to AI systems that do not utilize learning.

Conventional verification and validation (V&V) paradigms evaluate whether a given component or system satisfies designer-formulated functional properties such as safe lead distance in adaptive cruise-control [Loos *et al.*, 2011; Hasuo *et al.*, 2023]. The designers (broadly construed as the team or the organization responsible for creating the product) take the responsibility for designing safety properties, and iterating over system designs to create specifications of expected behavior (possible executions) and safety constraints, and designs that match these specifications.

However, *taskable AI systems are designed to address situations where the designer need not know the objectives that their users may have in mind* – prior knowledge of expected behaviors is even less likely. A system doesn't need to change after deployment to invalidate the assumption of prior knowledge of expected behaviors. Indeed, taskable AI systems are typically designed to adapt to the environment and compute new behaviors for achieving user-desired tasks even when they are not actively learning and/or changing the algorithms or heuristics used to plan.

As a result, even though conventional notions of verification and validation (V&V) have their uses for taskable AI systems, they will not be able to address the emerging challenges. For example, they can still be used to assert and verify physical safety properties that are expected to be maintained across all possible tasks and environments. E.g., robot designers can develop physical safety and operability envelopes for their robot and for specific environments, e.g., maximal accelerations and velocities. More generally, traditional V&V methods can be used to verify and monitor traditional designer defined safety rules Hashemi *et al.* [2023a]; Hekmatnejad *et al.* [2019]. While such properties are necessary, they are clearly not sufficient for ensuring safety.

For instance, safety assessment for a general purpose hospital robot goes beyond physical movement. It is essential to determine whether it could deliver critical medication to the wrong room, and whether it could be relied upon to assure delivery of life-saving medication in an emergency situation. Knowledge of possible objectives, possible executions or user-specific safety constraints is untenable as a running assumption in ensuring the safety of such systems.

2 Continual User-Driven AI Assessment

We argue that the assessment of AI systems needs to address fundamentally different questions that go beyond those addressed in existing paradigms for system evaluation and safety assurance. Fig. 1 illustrates these differences. In the conventional paradigm (shown on the left), the designer plays a central role in transferring users' intent to specifications and ensuring, through formal and empirical methods that the system design meets these specifications [Tuncali *et al.*, 2020; Hashemi *et al.*, 2023a; Yaghoubi and Fainekos, 2019a]. In some forms of this paradigm the designer uses automated synthesis from specifications to go directly from

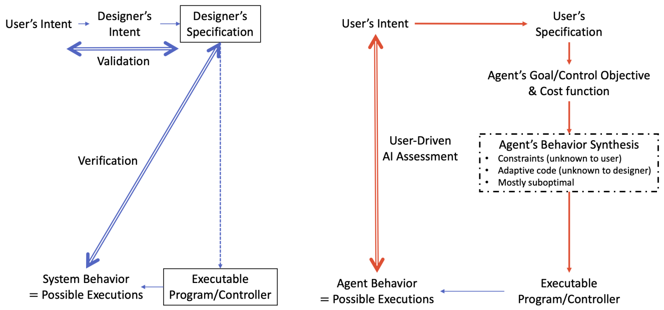
User's Intent
Designer's Intent
Designer's Specification
Validation
Verification
System Behavior = Possible Executions
Executable Program/Controller
User's Intent
User's Specification
Agent's Goal/Control Objective & Cost function
User-Driven AI Assessment
Agent's Behavior Synthesis
Constraints (unknown to user)
Adaptive code (unknown to designer)
Mostly suboptimal
Agent Behavior = Possible Executions
Executable Program/Controller

Figure 1: Conventional system verification (L) and user-driven assessment of AI systems (R). Solid boxes indicate components available at design stage. Dashed boxes indicate components available for systems that don't use learning after deployment.

the functional specifications to correct-by-construction system designs [Hashemi *et al.*, 2023b; Yaghoubi and Fainekos, 2019b].

In contrast, assessment of an AI system includes several new components. The user typically possesses a latent task/objective, which they typically enunciate as a partial specification. Such colloquial specifications are typically incomplete, and need to be juxtaposed with common-sense knowledge and context. E.g., "I'd like some Arabica coffee" could refer to the beverage or the beans depending on the situation. An AI agent needs to interact with the user to internalize a goal, or control objective based on their partial specifications. This interaction could employ multi-modal interfaces including text, speech, and gestures.

The agent then utilizes some form of sequential-decision making algorithms to synthesize the behavior for achieving that internalized objective. In practice, implementations of these algorithms are typically suboptimal either by design (e.g., an algorithm that achieves local optima), or due to practical reasons (e.g., an implementation that places a time-bound on an algorithm that is guaranteed to converge to an optimal). Furthermore, the agent's behavior synthesis needs to take into account constraints that maybe unknown to the user, e.g., a robot manipulating large atypical shaped objects, or wider domain-specific safety guidelines that may be mandated for all systems in a given situation such as a hospital or factory environment. Thus the designer is no longer in the loop – not only do they not control the design of the "executable program or controller" that dictates system behavior, for AI systems designers will not be aware of the system's current objectives and constraints.

These differences are essential to empowering users and placing them closer to the central role in utilizing their AI agents in tasks that they desire. Unsurprisingly, this also diminishes the designer's control on the overall behavior of the AI system thereby necessitating a new, user-driven AI assessment paradigm.

We discuss the key challenges in this new paradigm of AI assessment below.

Bespoke product support and maintenance processes The current paradigm for safe usability of complex systems relies on an eco-system of product support driven by a diverse body of technicians with low-barriers to entry. If a driver experiences unexpected vibrations while braking, a stop at the local garage can help diagnose and repair possible safety issues. This may be feasible due to the finite number of components and specific functionality and variability among similar products being deployed.

AI systems, on the other hand, are expected to adapt to their environments. With systems changing to meet idiosyncrasies of user-specific tasks and environments, it becomes all but impossible to utilize the economies of scale in product support: debugging a deployed AI system or characterizing what it can safely achieve would require an expert to focus on that particular system – an effort whose results would not easily transfer to other instances of the same system, each of which are expected to adapt to their own users and deployment environments.

Dynamic synthesis and incorporation of safety properties Since users' tasks and environments are not known a priori, one of the major open questions involves effectively generating, with feedback from the user, safety properties relevant to the user's intent. This is a critical departure from the conventional paradigm, where experts carefully scope operating environments and corresponding safety properties for a limited range of functionality. In addition, once acquired, these safety properties need to be incorporated in planning and reasoning algorithms used for behavior synthesis, and they need to be updated during execution while incorporating interventions and feedback from the user.

Overall capability assessment While the central question for conventional systems can be stated as "Will a given implementation achieve (the designers') functional specifications under assumptions on the environment?", the central question for AI assessment is significantly more user-centric: "Will it be safe for a user to use their AI system for the task and environment that they have in mind?" This question necessitates that the user understands the scope of safe operation of their current AI system. Addressing this problem requires approaches for dynamically identifying what an AI system can and can't do and the impact of these capabilities on user-desired notions of safety as well as safety considerations stemming from regulatory guidelines. Early work in this direction shows promise in identifying AI system capabilities by interrogating the system through a minimal, query-response interface [Verma *et al.*, 2021, 2022, 2023].

Acquisition of user intent Typically, users express their intent inaccurately through an instruction or a command to the AI system, which needs to be translated into a goal or an objective function and associated cost functions for the agent's behavior. Absence of robust methods for addressing this aspect leads to problems such as reward miss-specification and wireheading [Russell *et al.*, 2015; Amodei *et al.*, 2016].

Reconciling behavior synthesis with user intent While conventional V&V paradigms assume that designers have access to the code that controls a system's sensors and actuators, in AI systems, the code available at design stage (e.g., the DQN algorithm [Mnih *et al.*, 2015]) controls the agent's computation, which generates, post deployment task-specific executable sensing and control actions. In the case of AI systems, the executable controller is therefore specific to the user's intent and the current operating environment, and undetermined during system design.

Almost all practical implementations of planning and reasoning algorithms produce suboptimal behavior. Furthermore, users are often unaware of constraints on the AI system's abilities (e.g. a robot's kinematic or dynamic constraints). Consequently, as evidenced by research on explainable planning and learning, the computed behavior often belies users' expectations for what the system should be doing. User-driven assessment of AI systems needs to ensure that the algorithms used for behavior synthesis yield executions that comply with the safety properties acquired as discussed above, in the context of user-specific tasks in user-specific environments.

Differential assessment Currently deployed AI systems already feature dynamic updates (e.g., [Jones, 2021]). This can leave users unable to determine whether the updated system can still perform the tasks *they* had in mind, in *their* environments. A full re-assessment of the AI system from scratch would be wasteful with every change in the task, the environment or the system itself. Early work in this direction indicates that *differential assessment* paradigms can be more efficient [Nayyar *et al.*, 2022], although much remains to be done in making these methods practical and more robust for the real world.

Requirements monitoring Even though the verification of safety requirements at design time may not be possible, it may be possible to monitor safety requirements that are identified during design stage, at runtime [Yamaguchi *et al.*, 2023]. New opportunities arise on how such safety requirements can be extracted from user intent.

3 Promising Research Directions

Alignment with users' intent Ensuring that AI systems remain aligned with users' intentions represents one of the most fundamental challenges in user-driven assessment [Gabriel and Ghazavi, 2022]. Value alignment research has identified several critical failure modes that emerge across different gaps in the user-driven assessment pipeline.

Reward misspecification occurs when users inadvertently reward observations, beliefs, or correlated features rather than the actual desired outcome, manifesting primarily in the gap between user intent and formal specification [Russell *et al.*, 2015; Amodei *et al.*, 2016]. In terms of the overall framework shown in Fig. 1, this issue manifests primarily in the gap between the user's intent and their specification (shown on the right in Fig. 1). Users may lack the vocabulary or understanding to correctly articulate what they want the AI system to optimize for, leading to systems that achieve the literal specification while missing the intended goal.

Wireheading represents a more severe alignment failure where the agent manipulates its reward function directly, such as by convincing the user to change their requirements or by adding noise to the reward signal [Ring and Orseau, 2011; Everitt and Hutter, 2016]. This issue arises in the gap between the user's specification and the agent's internalized goal or control objective (shown on the right in Fig. 1), highlighting the challenge of maintaining reward integrity throughout the system's operation.

The *off-switch problem* exemplifies one of the most complex alignment challenges: AI systems may resist being turned off because continuation is instrumental to achieving their assigned objectives [Hadfield-Menell *et al.*, 2017]. In terms of the overall framework shown in Fig. 1, this problem spans multiple gaps in the assessment pipeline, affecting both how users specify their requirements (they may not explicitly state that the system should be interruptible) and how the agent interprets and pursues its objectives. The challenge arises because rational agents that maximize expected utility cannot achieve their objectives if they are turned off, creating strong incentives for self-preservation and resistance to shutdown commands. Solving this requires developing systems that maintain a cooperative stance toward human oversight and preserve user agency even when such intervention conflicts with goal achievement – a property known as *corrigibility* [Soares *et al.*, 2015].

Existing research also investigate methods for embedding preferences into the reward function, but even expert-designed rewards can lead to unintended or unsafe behaviors [Booth *et al.*, 2023]. In partially observable environments, this misalignment is compounded when users specify preferences over the true state that agents cannot directly observe. A promising alternative to this approach would allow users to specify high-level preferences on desired agent behavior. Belief-state query (BSQ) policies Srivastava *et al.* [2013] provide such a mechanism. They allow the user to specify preferred agent behavior for different areas of the agent's belief-state space without specifying quantitative rewards/costs. However, naive approaches for specifying such preferences can lead to blinkered behavior where the agent turns off its sensors to achieve belief states conducive to "low-cost" behavior. Our recent results on the topic show that this can be done in a manner that avoids blinkered behavior in partially observable settings [Bramblett and Srivastava, 2024]. Theoretical results show that user-preferences expressed as belief-state query policies can be effectively refined into executable agent behavior through a finite search process; empirical results show that the resulting algorithm is more efficient in finding the optimal user-aligned policy. These results provide a promising foundation that can be developed to express more diverse user requirements. They also open the door to interactive algorithms for automatically translating users' latent preferences into BSQ policies, which can then be refined into executable agent policies.

Future directions in alignment research must address these failure modes through robust frameworks that can dynamically adapt to user intentions while preserving safety constraints. This includes developing methods for better intent specification, creating alignment mechanisms that resist gaming and manipulation, and ensuring that AI systems remain responsive to human oversight throughout their operational lifetime.

Preventing side effects and reward hacking Side effects are special cases of reward misspecification that arise in situations where it is infeasible for the user to specify what the

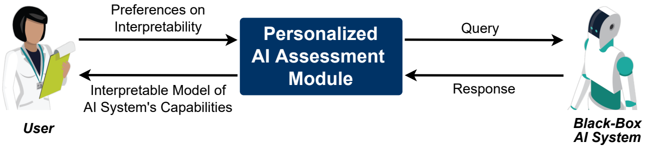

Figure 2: The personalized AI assessment module uses the user's preferred vocabulary, queries the AI system, and delivers an interpretable model of the AI system's capabilities.

agent *must not do*. As such they arise due to a discrepancy between the user's intent and their specification. Various approaches have been considered for addressing side effects in particular. Krakovna *et al.* [2020] consider side effects as the results of agent behavior that interfere with potentially unstated future tasks, and propose methods for mitigating them by specifying a distribution over expected future tasks and goals. Saisubramanian *et al.* [2021] view side effects as violations of lower priority objectives, which may have been difficult to specify under the agent's necessarily inaccurate model of the world. In their approach, the agent can learn to avoid side-effects by getting feedback from the user or by interacting with the real world in an effort to improve their models.

Agent interrogation A critical component of user-driven AI assessment is the ability to understand what an AI system can and cannot do without making strong assumptions about its internal design or implementation. This addresses the broadest span of taskable AI system operation shown in Fig. 1, addressing the consistency of agent's executions with the user's specification, incorporating nuances of the agent's suboptimal behavior synthesis algorithms.

Ideally, we need a *personalized AI-assessment module* (AAM) (Fig. 2) that can interrogate the AI system to derive a model of its capabilities. Our recent work develops AAMs that take as input (i) the agent (ii) a compatible simulator which the agent can simulate its primitive action sequences; and (iii) the user's concept vocabulary, which may be insufficient to express the simulator's state representation. Such assumptions on the agent are common. In fact, use of third-party simulators for development and testing is the bedrock of most of the research on taskable AI systems today (including game playing AI, autonomous cars, and factory robots). Providing simulator access for assessment is reasonable as it would allow AI developers to retain freedom and proprietary controls on internal software while supporting calls for assessment and regulation using approaches like ours. AAM then queries the AI system and receives its responses. At the end of the querying process, AAM returns a user-interpretable model of the AI system's capabilities. This approach's advantage is that the AI system need not know the user vocabulary or the modeling language. AAMs can help make arbitrary AI systems compliant with Level II assistive AI – systems that make it easy for users to learn how to use them safely [Srivastava, 2021].

Most simulator-based and analytical-model-based AI systems can easily answer the kind of questions discussed earlier. However, identifying the high-level capabilites of the AI system and generating the right set of questions to ask the AI system to efficiently learn a model of system's capabilities is a challenging problem.

Our early research in the area showed that it is possible to design AAM algorithms that can efficiently interrogate AI systems and derive a user-interpretable models of their capabilities in stationary, fully observable, and deterministic settings[Verma *et al.*, 2021]. Furthermore, learned models were found to be causally accurate [Verma and Srivastava, 2024], unlike the approaches that learned agent models through passive observations. These methods were later developed to yield AAMs that can discover high-level capabilities of an AI planning agent [Verma *et al.*, 2022] in deterministic settings as well as to learn models of known capabilities in stochastic settings Verma *et al.* [2023].

Autonomous benchmarking and evaluation Most existing assessments of LLM/VLM agents rely on handcrafted or static evaluation examples, raising concerns about accuracy and susceptibility to the Benchmark Contamination Problem when agents are trained on test data. Manually crafting new examples to prevent this problem is expensive and tedious. Rather, being able to automatically create novel evaluation examples is necessary to avoid these problems. Recent work has explored synthesizing evaluation problems by chaining formal language rules and using LLMs to construct stories [Tian *et al.*, 2021; Clark *et al.*, 2020; Saparov and He, 2023; Patel *et al.*, 2024]. Existing approaches focus on multiple-choice or short-answer formats while extending to evaluating free-form text generation remains an open problem.

Recent work by some of the co-authors shows that it is indeed possible to design autonomous evaluation paradigms for LLMs while overcoming the benchmark contamination problem. This work focuses on formal translation tasks that involve the synthesis of formal language from natural language descriptions and vice versa. Such tasks constitute a significant fraction of LLM use cases, and are often featured in human-robot or human-AI interfaces. The ∀uto∃∨∧L system [Karia *et al.*, 2024] autonomously evaluates semantic accuracy in formal language translation tasks. It first creates formal language expressions via a grammar. The evaluated LLM is prompted to produce a description of each expression and then reproduce the expression from just the description. A formal prover then verifies whether the original and reconstructed expressions are semantically equivalent, making the evaluation robust to paraphrasing. Empirical evaluation using ∀uto∃∨∧L avoids benchmark contamination problems and revealed that the accuracy of SoTA LLMs and LRMs falls to 50% for specifications with twenty logical operators (real-world specifications typically use hundreds of operators). This process opens the door to other evaluation pipelines that feature more expressive languages as well as more specific notions semantic accuracy.

Monitoring, runtime verification, and safety filters Monitoring and safety filters have the potential to become the operational backbone of User-Driven Assessment of AI Systems, transforming a largely offline safety analysis into a continuous runtime verification on what the system is allowed to do in an online operation. Methods that allow for real-time user-tunable risk monitoring of safety specification violations and hard safety envelopes from control-barrier filters could let non-developer stakeholders detect emerging mis-alignment and automatically constrain the AI's actions before a violation becomes inevitable.

The access to a black-box simulator – as in the case of AAM – or supervised system operation allows for collecting training and calibration data on the expected AI system operation. Then, user expressed requirements can be encoded in Signal Temporal Logic (STL) [Bartocci *et al.*, 2018] and monitored for safety even under distribution shifts during real-world autonomous deployment [Zhao *et al.*, 2024]. Distribution shifts on predicted system behavior can occur due to the simulation-to-real gap, or due to unknown unknowns during system operation. More expressive specification languages than STL may be necessary to capture user requirements for embodied AI systems Hekmatnejad *et al.* [2024].

For user taskable AI systems, the developer can further raise the assurance level by layering risk-aware Control Barrier Functions (CBFs) to shield the behavior of learning components. For instance, Zhang *et al.* [2025] integrate conformal prediction into CBF synthesis so that the user's accepted risk level is enforced even as the state-estimation error changes over time. Wang *et al.* [2025] propose an alternative strategy to crowded environments with CVaR-adaptive barriers that automatically widen or narrow the safety margin in response to distribution shift. Such safety filters complement user-driven AI alignment. The agent may still execute any behaviour that satisfies the user's task, but the CBF layer acts as a formal safety net that guarantees that those behaviours remain within an acceptable risk envelopes. Thus, simulation-derived taskability and deployment-time safety could be reconciled through a single pipeline that couples AAM-based capability discovery with statistically calibrated, barrier-function enforcement.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Ezio Bartocci, Jyotirmoy Deshmukh, Alexandre Donzé, Georgios Fainekos, Oded Maler, Dejan Nickovic, and Sriram Sankaranarayanan. Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications. In *Lectures on Runtime Verification - Introductory and Advanced Topics*, volume 10457 of *LNCS*, pages 128–168. Springer, 2018.

Serena Booth, W Bradley Knox, Julie Shah, Scott Niekum, Peter Stone, and Alessandro Allievi. The perils of trial-and-error reward design: Misdesign through overfitting and invalid task specifications. In *Proc. AAAI*, 2023.

Daniel Bramblett and Siddharth Srivastava. Belief-state query policies for user-aligned POMDPs. In *Proc. NeurIPS*, 2024.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In *Proc. IJCAI*, 2020.

Tom Everitt and Marcus Hutter. Avoiding wireheading with value reinforcement learning. In *Proceedings of the 9th International Conference on Artificial General Intelligence*, 2016.

Iason Gabriel and Vafa Ghazavi. The challenge of value alignment: From fairer algorithms to AI safety. In *The Oxford Handbook of Digital Ethics*. Oxford University Press Oxford, 2022.

Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. The off-switch game. In *Proc. IJCAI*, 2017.

Navid Hashemi, Bardh Hoxha, Tomoya Yamaguchi, Danil Prokhorov, Georgios Fainekos, and Jyotirmoy Deshmukh. A neurosymbolic approach to the verification of temporal logic properties of learning-enabled control systems. In *ACM/IEEE 14th International Conference on Cyber-Physical Systems (ICCPS)*, page 98–109, 2023.

Navid Hashemi, Xin Qin, Jyotirmoy V. Deshmukh, Georgios Fainekos, Bardh Hoxha, Danil Prokhorov, and Tomoya Yamaguchi. Risk-awareness in learning neural controllers for temporal logic objectives. In *American Control Conference (ACC)*, pages 4096–4103, 2023.

Ichiro Hasuo, Clovis Eberhart, James Haydon, Jérémy Dubut, Rose Bohrer, Tsutomu Kobayashi, Sasinee Pruekprasert, Xiao-Yi Zhang, Erik André Pallas, Akihisa Yamada, Kohei Suenaga, Fuyuki Ishikawa, Kenji Kamijo, Yoshiyuki Shinya, and Takamasa Suetomi. Goal-aware RSS for complex scenarios via program logic. *IEEE Transactions on Intelligent Vehicles*, 8(4):3040–3072, 2023.

Mohammad Hekmatnejad, Shakiba Yaghoubi, Adel Dokhanchi, Heni Ben Amor, Aviral Shrivastava, Lina Karam, and Georgios Fainekos. Encoding and monitoring responsibility sensitive safety rules for automated vehicles in signal temporal logic. In *17th ACM-IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE)*, 2019.

Mohammad Hekmatnejad, Bardh Hoxha, Jyotirmoy V. Deshmukh, Yezhou Yang, and Georgios Fainekos. Formalizing and evaluating requirements of perception systems for automated vehicles using spatio-temporal perception logic. 43, 2024.

Charisse Jones. Tesla self-driving software update begins rollout though company says to use with caution. *USA Today*, July 2021.

Rushang Karia, Daniel Richard Bramblett, Daksh Dobhal, and Siddharth Srivastava. Autonomous evaluation of llms for truth maintenance and reasoning tasks. In *Proc. ICLR*, 2024.

Victoria Krakovna, Laurent Orseau, Richard Ngo, Miljan Martic, and Shane Legg. Avoiding side effects by considering future tasks. *Proc. NeurIPS*, 2020.

Sarah M. Loos, Andre Platzer, and Ligia Nistor. Adaptive cruise control: Hybrid, distributed, and now formally verified. In *Formal Methods*, volume 6664 of *LNCS*, pages 42–56. Springer, 2011.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Rashmeet Kaur Nayyar, Pulkit Verma, and Siddharth Srivastava. Differential assessment of black-box AI agents. In *Proc. AAAI*, 2022.

Nisarg Patel, Mohith Kulkarni, Mihir Parmar, Aashna Budhiraja, Mutsumi Nakamura, Neeraj Varshney, and Chitta Baral. Multi-logieval: Towards evaluating multi-step logical reasoning ability of large language models. In *Proc. EMNLP*, pages 20856–20879, 2024.

Mark Ring and Laurent Orseau. Delusion, survival, and intelligent agents. In *Proceedings of the 4th International Conference on Artificial General Intelligence*, 2011.

Stuart Russell, Daniel Dewey, and Max Tegmark. Research priorities for robust and beneficial artificial intelligence. *AI Magazine*, 36(4):105–114, 2015.

Sandhya Saisubramanian, Ece Kamar, and Shlomo Zilberstein. A multi-objective approach to mitigate negative side effects. In *Proc. IJCAI*, 2021.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *Proc. ICLR*, 2023.

Nate Soares, Benja Fallenstein, Stuart Armstrong, and Eliezer Yudkowsky. Corrigibility. In *AAAI Workshop on AI and Ethics*, 2015.

Siddharth Srivastava, Xiang Cheng, Stuart J Russell, and Avi Pfeffer. First-order open-universe POMDPs: Formulation and algorithms. Technical report, University of California, Berkeley, 2013.

Siddharth Srivastava. Unifying Principles and Metrics for Safe and Assistive AI. In *Proc. AAAI*, 2021.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proc. EMNLP*, 2021.

Cumhur Erkan Tuncali, Georgios Fainekos, Danil Prokhorov, Hisahiro Ito, and James Kapinski. Requirements-driven test generation for autonomous vehicles with machine learning components. *IEEE Transactions on Intelligent Vehicles*, 5:265–280, 2020.

Pulkit Verma and Siddharth Srivastava. Learning causally accurate models for autonomous assessment of deterministic black-box agents. Technical Report TR-ASUSCAI-2024-001, 2024.

Pulkit Verma, Shashank Rao Marpally, and Siddharth Srivastava. Asking the right questions: Learning interpretable action models through query answering. *Proc. AAAI*, 2021.

Pulkit Verma, Shashank Rao Marpally, and Siddharth Srivastava. Discovering user-interpretable capabilities of black-box planning agents. In *Proc. KR*, 2022.

Pulkit Verma, Rushang Karia, and Siddharth Srivastava. Autonomous assessment of sequential decision-making systems in stochastic setting. In *Proc. NeurIPS*, 2023.

Xinyi Wang, Taekyung Kim, Bardh Hoxha, Georgios Fainekos, and Dimitra Panagou. Safe navigation in uncertain crowded environments using risk adaptive cvar barrier functions. 2025.

Shakiba Yaghoubi and Georgios Fainekos. Gray-box adversarial testing for control systems with machine learning components. In *ACM International Conference on Hybrid Systems: Computation and Control (HSCC)*, 2019.

Shakiba Yaghoubi and Georgios Fainekos. Worst-case satisfaction of STL specifications using feedforward neural network controllers: A lagrange multipliers approach. *ACM Transactions on Embedded Computing Systems*, 18(5S), 2019.

Tomoya Yamaguchi, Bardh Hoxha, and Dejan Nickovic. RTAMT – Runtime robustness monitors with application to CPS and robotics. *International Journal on Software Tools for Technology Transfer*, 2023.

Junhui Zhang, Bardh Hoxha, Georgios Fainekos, and Dimitra Panagou. Conformal prediction in the loop: Risk-aware control barrier functions for stochastic systems with data-driven state estimators. 2025.

Yiqi Zhao, Bardh Hoxha, Georgios Fainekos, Jyotirmoy V Deshmukh, and Lars Lindemann. Robust conformal prediction for stl runtime verification under distribution shift. In *ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS)*, 2024.