INSTITUTE FOR DEFENSE ANALYSES

Trustworthy Autonomy: A Roadmap to Assurance

Part I: System Effectiveness

Daniel Porter
Michael McAnally
Chad Bieber
Heather Wojton
Rebecca Medlin, Project Leader

May 2020

Approved for Public Release.
Distribution Unlimited.

IDA Document P-10768-NS

Log: H 2019-000369

INSTITUTE FOR DEFENSE ANALYSES
4850 Mark Center Drive
Alexandria, Virginia 22311-1882

About This Publication
This work was conducted by the Institute for Defense Analyses (IDA) under a Separate Contract, Task C9082, Cross-Divisional Statistics and the Data Science Working Group. The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Acknowledgments
The IDA Technical Review Committee was chaired by Mr. Robert R. Soule and consisted of Mark R. Herrera and Stacey L. Allison from the Operational Evaluation Division, John C. Biddle and David A. Sparrow from the Science Technology Division, James M. Gilmore from the Systems Evaluation Division, Laura Freeman and Thad Polk, Non IDA Staff.

For more information:
Daniel Porter, Operational Evaluation Division
dporter@ida.org • (703)578-2869

Robert R. Soule, Director, Operational Evaluation Division
rsoule@ida.org • (703) 845-2482

INSTITUTE FOR DEFENSE ANALYSES

IDA Document P-10768-NS

Trustworthy Autonomy:
A Roadmap to Assurance

Part I: System Effectiveness

Daniel Porter
Chad Bieber
Michael McAnally
Heather Wojton
Rebecca Medlin, Project Leader

Executive Summary

The Department of Defense (DoD) has invested significant effort over the past decade considering the role of artificial intelligence and autonomy in national security (e.g., Defense Science Board, 2012, 2016; Deputy Secretary of Defense, 2012; Endsley, 2015; Executive Order No. 13859, 2019; US Department of Defense, 2011, 2019; Zacharias, 2019a). However, these efforts were broadly scoped and only partially touched on how DoD will certify the safety and performance of these systems. More recent work has done this big-picture thinking for the test and evaluation (T&E) community (e.g., Ahner & Parson, 2016; Haugh, Sparrow, & Tate, 2018; Porter et al., 2018; Sparrow, Tate, Biddle, Kaminski, & Madhavan, 2018; Zacharias, 2019b). In parallel, individual programs have been generating their own working-level solutions for their own particular use cases and challenges. The framework proposed in the current work bridges the gap between the big picture policy recommendations already made and individual program needs. It is meant to serve as a framework that the T&E community can follow in order to provide evidence that artificial intelligence (AI)-enabled and autonomous systems function as intended. At times we echo broad policy recommendations made by others as they will also enable T&E activities. In other places we make more specific recommendations relating to test planning and analysis.

In this document, we present part one of our two-part roadmap. We discuss the challenges and possible solutions to assessing system effectiveness. A future part two will deal with test efficiency, simulation, and infrastructure.

The recommendations below are summarized for an executive level. The body of the text goes into significantly more detail. Due to the scope of this project, even the main body only provides a survey of the challenges and our proposed solutions. However, this roadmap serves as an outline to a future series of technical papers covering these topics in detail for working-level testers and analysts.

Recommendations

- **Testers need to identify the features of autonomous systems that will (and will not) cause traditional test methods to misinform decision-makers about risk.** We need to identify when, why, and how testing will need to be different for AI-enabled systems. Overarching definitions of AI or autonomy often exclude some systems that would be difficult to test, and programs are not self-identifying as involving such risks. Other definitions suffer from disagreement over the meaning of words. In this paper, we define AI and autonomy as anything that makes decisions based on environmental information within the constraints of a specific task. We identify three types of

decisions—setting goals or constraints, defining the current situation, and choosing the next action—to help identify what does and does not change about testing. To avoid ambiguity, these definitions are grounded in a technical theory of decision-making.

- **Testers need more transparency in decision-making systems.** Transparency is important for end-users, but also for testers. Black-box systems prevent testers from making inferences about untested scenarios. Before we can confidently test system performance, we must understand how the system makes its decisions. This transparency can be built-in at the drawing board, or, as a less desirable option, the lack of transparency in design can be mitigated during early testing. We make recommendations for how to obtain, verify, and validate models of what causally drives system decision-making.
- **Testers need rights to system decision-making and learning processes and data generated by these systems.** In addition to benefits such as enabling modularity and reusability across systems, gaining ownership rights to the decision software is critical to testing. Proprietary concerns can cause an otherwise transparent system to be a black box to testers, as has already happened with several systems.
- **The DoD should consider adopting common, modular cognitive architectures to enable testing.** Many have discussed how modular cognitive architectures benefit system development, performance, and sustainment. Here we discuss how they facilitate efficient and effective T&E as well.
- **Research into and dissemination of methods for evaluating decision-making are needed.** These include metrics to quantify intermediate mission success, methods to qualitatively evaluate overall decision processes, novel calculations of classification accuracy for multi-categorical fuzzy groups, and ways to quantify a system's ability to learn.
- **Decision-making systems that have a built-in infrastructure for recording data (BIRD) become easier to certify.** We recommend a BIRD to enable testing, but it would serve many different needs. By having systems record data about themselves, by themselves, and by providing an infrastructural pipeline to securely collate, store, and disseminate these data, stakeholders can harvest data from a variety of previously inaccessible venues such as exercises and operational missions. These harvests can support many activities like T&E, operator and commander decision-making, and post-fielding fleet-wide learning.
- **Testers should use a strategy of Graded Autonomy with Limited Capability Fielding for difficult-to-certify systems.** Some systems are too dangerous to test live, but too difficult to simulate credibly. These systems should be tested like we do with medical residents. Train all skills, and then certify and field their least risky capability for use under supervision. While acting in realistic situations in the field or exercises,

have systems evaluate what they *would* have done with more risky capabilities. Use these data to spiral upward through risk and down through supervision levels as systems demonstrate safe competence.

- **Testers should characterize system flexibility as well as system performance.** Decision systems can achieve greater performance on a specific task by over-optimizing, which can create downstream costs and consequences when trying to upgrade, change, learn, or transfer to a related task. Testing should evaluate to what extent programs have made this tradeoff.
- **Testers need environments where different autonomous agents, including humans, can be tested together for emergent behavior.** When autonomous agents interact, you can get emergent behavior (EB). EB can be expected or unexpected, and it can be desirable or undesirable. Testers need to confirm that expected, desirable EB (such as teaming or synergy) functions correctly, while minimizing the probability of unexpected, undesirable EB. This must be tested under live as well as simulated environments. Centralizing test responsibility for EB can overcome a number of simulation challenges, while having a regular joint exercise would provide such a live test venue for validation while also helping troop readiness for existing and emerging technology employment.
- **Testers still need to emphasize human-system interaction for autonomous systems.** Even in fully autonomous systems, a human will be involved in some part of their decision-making chain, even if it is just issuing initial orders. These interactions must be fluid and minimize error to ensure responsible employment, and testers must evaluate this. Additionally, the acquisition community should assess whether warfighters will have *appropriately calibrated* trust of their systems.
- **Testers should adapt existing methods for evaluating human teams for the T&E of human-machine teams.** Though not all AI-human system relationships will truly involve teaming, systems that do will require a different approach to testing. The starting point for these evaluations should be the methods already created by the behavioral sciences and sports statisticians.
- **Testers should assess adversarial exploitation generational cycles.** Cyber and tactical exploitation is a never-ending, constantly evolving battle in learning systems. This may require a cultural shift away from testing against static, well-defined exploitation requirements. Testers should attempt to quantify how quickly adversaries can develop exploitations of our decision systems versus the speed at which we can re-counter them. Having a faster friendly than adversary cycle will likely be critical to meaningfully field these systems. At first this will be a test of industrial agility, though in time in may be a metric of systems' live behavioral flexibility.

Contents

1. Introduction

The world is witnessing a proliferation of systems that act independently from humans. These systems are beginning to permeate every aspect of our lives, aiding us with tasks as mundane as household chores or as consequential as nuclear power regulation (e.g., Wood, Upadhyaya, & Floyd, 2017). Whether they tangibly interact with the physical world or exist purely in cyberspace, these systems promise to revolutionize the way we live. However, this promise comes with a host of hazards. As machines take over decisions traditionally made by humans, and as these decisions become more meaningful, the potential for catastrophic consequences only expands. Nowhere is this truer than in national security applications of artificial intelligence (AI) and autonomy (Defense Science Board, 2012, 2016; Executive Order No. 13859, 2019; US Department of Defense, 2019). Testers must understand how these systems make decisions across different operating conditions if they are to provide stakeholders with appropriate levels of trust in autonomous or AI-enabled capabilities (Wojton, Porter, & Lane, 2020). In this paper, we propose a framework to facilitate the test and evaluation (T&E) of military systems enabled by AI or autonomy.

Achieving assurance for autonomous systems requires solving a number of challenges. For example, we need methods for generalizing our findings from test and training environments to conditions that have not been explicitly tested (e.g., Micskei, Szatmári, Oláh, & Majzik, 2012). We need better techniques to assess and ensure realism in our training data, simulations, and test environments (e.g., Ahner & Parson, 2016). We need better metrics to describe the effectiveness of decision-making and methods to differentiate brilliance from blunders, especially as systems become more sophisticated (e.g., Ilachinski, 2017). We must structure tests to allow assessment of appropriately calibrated trust in the systems (e.g., Culley & Madhavan, 2013). Cutting across all of this, we need methods to conduct more efficient tests, as the public, law, and warfighters themselves likely will hold autonomous systems to higher standards and require a greater burden of evidence (e.g., Defense Science Board, 2016). Any framework addressing the T&E of autonomous systems must grapple with these and other challenges to achieve success.

> The heart of our framework is identifying where and why our standard test processes are likely to fail, how to mitigate those failures, and how the lifecycle of testing needs to be structured to implement the new approach.

Previous work on the challenges of designing, developing, and testing autonomy formed the starting point for this framework (e.g, Defense Science Board, 2012; Defense Science Board, 2016; Endsley, 2015; Roske, Kohlberg, & Wagner, 2012). These efforts were invaluable for setting the stage. However, they were often scoped for a broader picture than just T&E, and primarily made high-level recommendations for testing, calling for the relevant authorities to develop more specific methods (Defense Science Board, 2016; Deputy Secretary of Defense, 2012). Other work has begun to make high-level recommendations for the T&E community (e.g., Ahner & Parson, 2016; Haugh et al., 2018; Porter et al., 2018; Tate & Sparrow, 2015; Zacharias, 2019b) or is focusing on more immediate needs for individual programs that may not apply to other systems (e.g., Kwashnak, 2019; Lowrance, Herman, Schneider, & Kasemer, 2019). Our framework grounds itself in high-level recommendations and attempts to bridge the gap to working-level solutions that can be executed across the world of testing.

We advocate that the community maintain a solution-oriented mindset. Many discussions about the T&E of AI-enabled technology end up focusing on why a proposal will not work without offering solutions to those challenges or providing an alternative, and as a result are unproductive. With many autonomous or AI-enabled systems already fielded, and even more on the horizon, we cannot wait for perfect solutions. As a community, we must recognize that acquisition is not slowing down despite the lack of test methods, and to meet this challenge we must provide solutions alongside our critiques. Criticism alone will not solve the problem.

Testers of these systems should have at least a basic grounding in the world of AI, autonomy, and machine learning. While it is common to see these terms used interchangeably, testers should be aware of both the overlap and differences between them. A complete discussion of these topics is beyond the scope of this framework, and a number of good references already exist (e.g., Russell & Norvig, 2009; Zacharias, 2019a). We assume that readers already have this grounding and understand basic taxonomic distinctions in the field such as narrow/weak vs. strong/general approaches to AI; symbolic vs. sub-symbolic approaches to information representation or learning, and supervised, unsupervised, or reinforcement machine learning techniques.

At the highest level, our framework does not change the essence of testing. As with our standard systems, the goal of testing is to provide assurance that a system works as advertised, and to do so, testers must identify (1) the system's use-case tasks, (2) measurable outcomes for those tasks, (3) conditions expected to affect those outcomes, and (4) test points that enable inferences about performance across those conditions (e.g., Montgomery, 2019). AI and autonomy do not change these basic steps; they change the effectiveness of the standard techniques we use to plan and execute each step.

In the rest of this paper, we (1) discuss how to identify system features of concern, and (2) describe the fundamental challenges of and proposed solutions to performance evaluation for these systems. This document is part one of two in our roadmap, and notably it does not address test efficiency—especially the challenges to and enablers of using simulation or test automation for AI-enabled technologies—or the infrastructural demands of test. Furthermore, as the topics under consideration are nuanced, the following discussions will be more specific than the previously mentioned studies on T&E of autonomous systems, but less detailed than the subsequent technical papers to come.

2. The Systems That Challenge Us

A. Identifying Systems of Concern

RECOMMENDATION

Testers need to identify the features of autonomous systems that will (and will not) cause traditional test methods to misinform decision-makers about risk. We need to identify when, why, and how testing will need to be different for AI-enabled systems.

The past few decades have seen a robust discussion on what it means for a military system to involve AI or to be autonomous (e.g., Deputy Secretary of Defense, 2012; Roske et al., 2012; Zacharias, 2019a). While this conversation has been fruitful for some fields, there is not yet consensus on the definitions of these terms. For example, a recent call by the Joint Artificial Intelligence Center (JAIC) for acquisition programs to identify if they involved AI yielded zero responses; the JAIC's own count is several hundred (Trent, 2019). Although budgetary or political concerns may have influenced this result, there is a real need for testers to be able to identify systems of interest.

Overarching definitions of AI or autonomy often exclude some systems that would be difficult to test, and programs are not self-identifying as involving such risks. Other definitions suffer from disagreement over the meaning of words. In this paper, we define AI and autonomy as anything that makes decisions based on environmental information within the constraints of a specific task. We identify three types of decision—setting goals or constraints, defining the current situation, and choosing the next action—to help identify what does and does not change about testing. To avoid ambiguity, these definitions are grounded in a technical theory of decision-making.

The goal of our definitions is not to be proscriptive but to provide common language to discuss these systems. At every stage of the acquisition pipeline, we have observed systems of concern where the DoD's normal approach to testing would likely mischaracterize both performance and risk across the range of operating conditions. Others have proposed definitions of autonomy and artificial intelligence that would exclude many of these systems of concern (e.g., AFRL's definition; Overholt & Kearns, 2013). There is a legitimate debate to be had over what it means for something to be truly autonomous or be a ‘real’ AI, but we are not here to contribute to that discussion. Rather, we seek a definition that will identify the systems of concern for T&E regardless of how those systems are categorized elsewhere. We propose a definition of these systems that we will use throughout our paper. However, if the DoD chooses the pursue another definition, we recommend that it at least meet a few key requirements.

B. Requirements of a Definition

Definitions should serve a purpose; our purpose is to enable T&E activities. We are not proposing a complete, holistic definition of autonomy or AI, and there will be characteristics important to other domains that we do not address. In the context of T&E, any definition should provide programs, as well as oversight, with clear ways of assessing whether aspects of a system require different test methods and a mechanism by which to identify optimal test strategies. For our definition, we provide both detailed and intuitive definitions of these types of systems, and we recommend alternative definitions do the same.

T&E AI definitions should:

- Enable T&E activities
- ID T&E-hindering features
- Be agnostic to labels
- Describe continuous dimensions
- Leverage formal theories

Many proposed definitions of autonomy are useful in some ways, but for testing purposes create ambiguous categories. For example, DoD Directive 3000.09 on Autonomy in Weapon Systems[1] differentiates between autonomous and semi-autonomous systems based on the degree of human supervision. *Meaningful human control* is an important concept for AI&A (e.g., Cook, 2019; Horowitz & Scharre, 2015; Santoni de Sio & van den Hoven, 2018), and the DoD directive could be useful for system employment. However, DoDD 3000.09 makes distinctions based on a Concept of Operations (CONOPS) for intended use rather than a system’s capabilities, making the definition less helpful for testing purposes. Under the DoD policy, whether the system is deemed autonomous or semi-autonomous depends on its current operating state. For our systems of concern especially, the CONOPS will not be known in advance (e.g., Haugh et al., 2018; Zacharias, 2019a); whether a system should be allowed to operate independently depends on how much it should be trusted under different conditions. This is not information we know before testing, and it is essential, because human-supervised and

[1] Dated November 21, 2012, updated

independently operating systems require differently structured tests.[2] Any T&E definition of these systems of concern should base itself on capabilities (can do) and not CONOPS (should do).

Definitions should avoid ambiguous words and consider avoiding contested terminology. For example, many definitions rely on the word 'intelligent' to differentiate autonomy and automation (Overholt & Kearns, 2013). However, intelligence is itself an ambiguous and hotly contested definition, making it difficult to use for reliably identifying systems of concern. Furthermore, despite years of effort, consensus on at least the need for common terminology, and a myriad of proposals by authors in many different camps, there is no agreement on what truly constitutes AI or autonomy. These camps can be territorial regarding what kinds of systems receive certain labels, and this debate can derail otherwise productive conversations. Whether our systems of concern are 'truly' AI or autonomous is irrelevant to T&E, and authors and policy makers should consider avoiding these terms if they are similarly unnecessary for their own purposes. However, because the T&E community commonly references the systems of concern as involving AI or autonomy, in this paper we will be referring to them collectively as AI&A. We make no claim that they are 'truly' AI or autonomous though, and we could just as easily label them as *Artificially Implemented Decision Engines* (AIDEs) or *Artificial Cognition Enabled Systems* (ACES).

We recommend that definitions move away from a holistic categorical approach and toward a continuous, dimensional approach to defining AI&A (Defense Science Board, 2012), and that definitions should focus on identifying features rather than holistic categories. These features should identify both what does and does not need to be tested differently in autonomous systems. In some cases, the poorly delineated categories used by other frameworks are the result of binning along what are really continuous information dimensions. We recommend that any T&E definition of AI&A should avoid ambiguous words and base itself on capabilities (can do) and not CONOPS (should do). We propose language and definitions that allow us to identify *aspects* or *features* of systems that require different approaches to testing.

Furthermore, DoD should create specific, technical definitions of these dimensions and features to minimize confusion. We have experienced and read a large amount of unnecessary debate because some people feel a certain word implies something that someone else does not. Using technical definitions instead of implied or assumed ones can reduce this part of the debate. However, in order to incorporate specific, technical definitions in test framework, DoD would need to adopt a specific technical theory that describes the processes of problem solving or decision making.

[2] For example, tests with a human intervening will not inform us about the system's capability without an operator, and tests without a human will not inform us about the operator's ability to intervene.

C. An Enabling Theory of Problem Solving

While we provide colloquial definitions of AI&A alongside our technical ones, ambiguous, idiosyncratic, or field-specific usage of common words often leads to disagreement about whether, to what extent, or in what way a system involves autonomy. By defining autonomy in the context of a particular theory, we can avoid these confusions. To this end, we have adopted Allen Newell and Herbert Simon's (1972) problem space hypothesis (PSH) as our principal guiding theory. To understand the technical distinctions we are making in our definitions, readers must have at least a basic grounding in the PSH. Here we have simplified it to those aspects relevant to our definitions.

> **Tasks** are represented by a **problem space**, which is composed of **problem states** and **procedures**. A problem state could be considered a description or representation of the **environment**, while a procedure is something that causes a **transition** between states. The problem space for a task is composed of all the states and procedures that could potentially be reached or used.

While other theories of problem solving or decision making exist, we chose the PSH because of several beneficial features. In particular, it (a) discretely represents the critical steps of decision making, (b) allows tunable granularity in scoping a problem or task, (c) is useful for differentiating ground truth and perception, and (d) explicitly represents where in the task an agent is at any given moment. While Colonel John Boyd's Observe-Orient-Decide-Act (OODA) Loop is fantastic at emphasizing decision-making's temporal component, we argue it is less helpful with the (b) and (d) advantages of the PSH.

In the PSH, problems or tasks are represented by a *problem space* composed of *problem states* and *procedures*[3] (Newell & Simon, 1972). We use these words throughout the framework and assume the reader understands them. A problem *state* could be considered a description or representation of the environment, while a procedure is something that causes a transition between states. For a given problem, a decision agent begins in an *initial state* and is attempting to get to some *goal state*. The *problem space* for a task is composed of all the states and procedures that could potentially be reached or used. However, while all continuous variations of states and procedures could comprise the space, almost invariably an agent will compress functionally identical information into a single representation and eliminate what is believed to be irrelevant

[3] In the actual PSH, procedures are called *operators*. However, operators is already a term of art in the military. To avoid confusing our intended audience, we have replaced this word, but readers should note that this usage is unique to our framework.

information, resulting in a much reduced informational representation of the problem space (Abel et al., 2019; Barlow, 1961). This compression can occur to different degrees depending on context and capability (Howes, Lewis, & Vera, 2009).

Autonomy for a task has three aspects:
(1) **Executive autonomy**: setting a goal (including any sub-goals and path constraints)
(2) **Perceptual autonomy**: defining the current problem state
(3) **Procedural autonomy**: selecting the next procedure or sequence of procedures to move toward the goal

For a *problem state*, a set of variables describes the entire environment, and the agent is trying to represent the values of the task-relevant subset of such variables in a way that it can process. The problem state representation of real-world problems will almost necessarily be imperfect, either because of ignored variables or because of error in the value assignment process. For example, an airline pilot might represent the problem state using information like pressure-altitude, airspeed, and angle-of-attack but not the earth's rotational speed. In games like chess or Go, an AI's state representation might include just the configuration of pieces on the board; humans, however, might include information like the strategy they believe their opponent is pursuing.

To move between problem states, agents must apply a procedure. The agent will represent only a subset of possible procedures, either due to ignorance, irrelevance, or parsimony. These procedures can be represented at varying levels of abstraction, and prototypically these will be actions the agent can take: pull back on the yoke, do a barrel roll, or invade Russia during winter. A procedure does not require direct action. For example, "waiting" transitions the state to a new time value and allows other agents or environmental effects to alter the problem state. Whatever their nature, the agent must select procedures for non-random reasons related to the attempt to solve a problem or complete a task.

To solve a problem, the agent must be trying to satisfy some goal or goals.[4] In the PSH, the agent is attempting to reach a *goal state*. Complex problems might require the sequential achievement of multiple sub-goals, and the assignment of these sub-goals is one way to describe planning (Laird, 2012).[5] Agents might also have goals to avoid certain states, thus constraining the path they could take to the goal. When choosing between different procedures, the agent must evaluate which one best achieves its goals.

[4] For simplicity, we refer to a goal state, its sub-goals, and any path constraints from the PSH interchangeably as *goals*.

[5] There are many different ways that an agent can create and/or represent these goal sets, and the specific implementation used is less relevant than its existence.

In order to navigate through a problem space, the agent must loop through several steps: (1) have goals, (2) represent the current problem state, and (3) select a procedure or sequence of procedures to move toward the goal. As we move into our definitions of autonomy, these three steps are relevant for differentiating subtypes of autonomy.

D. A Definition of Autonomy for Test

In the T&E context, the authors define autonomy for a task as making the following decisions: (1) setting a goal (including any sub-goals and path constraints), a decision we refer to as 'executive autonomy'; (2) defining the current problem state, which we term 'perceptual autonomy'; and (3) selecting the next procedure or sequence of procedures to move toward the goal, or 'procedural autonomy'.[6] More colloquially, AI&A make decisions based on environmental information, and while in some contexts a decision could imply agency or awareness, in this framework it simply refers to the selection among alternatives. AI&A that possess all three subtypes of autonomy for a task would be described as fully autonomous for that task. Some systems might make only some of these types of decisions, but rather than labeling them broadly as "partially autonomous," testers should identify specifically which types of decisions the system makes.

Furthermore, we define autonomy within the constraints of a specific task rather than as a system attribute (Scharre & Horowitz, 2015), and we recommend that testers should evaluate AI&A at the task level. Any mission or task can be decomposed into a set of smaller tasks. The agent might not have autonomy at a higher level but might have it for some subtasks. For example, a mission could be to clear a minefield, and for this an agent would need to (a) pick a path to clear, (b) identify whether individual objects are mines or not, and (c) disarm those mines. After a human's order initiates the mine-clearing mission, the system could be free to make decisions about tasks (a) and (b), but not have freedom to decide (c), whether to remove the mines.[7] It is more useful for designing tests to discuss for *which tasks* the system has *what kinds of* autonomy, rather than whether the whole system is autonomous.

In the technical PSH context, a system has *executive* autonomy if it assigns itself goals or can alter how intermediate problem states are valued. Colloquially, one can think of this as making "should" decisions. For example, "I should turn left at the next street" is a goal statement, as is "I should avoid hitting pedestrians." The first provides a goal state that the agent can evaluate having reached, and the latter provides constraints on all states that the agent should visit. This is the type

[6] These distinctions and labels are a variation of a basic distinction between processes in cognitive psychology (e.g., Newell, 1990), and others have proposed related breakdowns for autonomy (e.g., Parasuraman, Sheridan, & Wickens, 2000; Roske et al., 2012). We make these distinctions because systems might perform some of these decisions while humans make the other ones, and which of these steps a system performs has implications for how it should be tested.

[7] For example, if the system cannot make collateral damage risk assessments, the CONOPS might call for a human to make that judgment.

of autonomy that often rightly concerns people the most (e.g., "I should shoot this person"). For the foreseeable future, it is very likely that people will design systems so that humans provide higher-level goals to the system and executive autonomy is implemented as sub-goaling within a well-defined mission (Trent, 2019). In fact, most machine learning today has none of this at all: the goals are chosen by designers and the system is only trained to select procedures to meet those pre-defined goals.

With *perceptual* autonomy, a system makes decisions (i.e., has alternatives or degrees of freedom) in how it represents the current problem state. Colloquially, these are "is" or "are" decisions. Most machine learning classifiers are captured in this type of autonomy, and it is likely that virtually all autonomous systems in the physical world possess some degree of perceptual autonomy. For example, a machine might be given the gaze angle and pupil dilation of a person as direct input, but it might use this information to decide "Is this person looking at me: Yes/No?" and represent the state only with the inferred variable. For sophisticated systems, there may be many layers of this perceptual autonomy between raw sensor feeds and the representation of the problem state on which procedural decisions are made.

Procedural autonomy simply means that the system can select its next procedure. Colloquially, these might be "how" or "what next" decisions. Systems such as autopilot have procedural autonomy, and processes which oversee action execution would similarly count. We have successfully tested many systems that have procedural autonomy, and by and large we know how to test these kinds of systems. If procedural autonomy works well, then the system will accomplish its goals. Procedural performance metrics are often the same as those we collect for systems anyway (e.g., probability of hit), and the way we currently structure tests is adequate to characterize decision quality for a system with only procedural autonomy. Developing high-quality procedural autonomy usually involves solving more challenges than demonstrating that it works. There are not many intellectual challenges to testing whether an autopilot maintains a certain airspeed and altitude—building a system that can do that might not be as easy. Where testing procedural autonomy becomes unwieldy is when large problem spaces are solved only with moment-to-moment procedural decisions, but this is primarily a quantity-of-testing challenge. Because these systems generally do not compress their problem spaces, they must be trained through extensive problem space exploration (e.g., OpenAI or AlphaGo), and so they also must be tested roughly as extensively. However, what should be measured is typically clear (e.g., games won or lost). For these reasons, our framework only minimally discusses testing this kind of autonomy.

Testers should avoid the heuristic that executive, perceptual, and procedural autonomy are organized hierarchically by complexity or risk. While these may be correlated in the systems we design, conceptually they are distinct. For example, one could try to solve a complex task just through procedural autonomy, and many current machine-learning efforts take this approach (Bathaee, 2018). A system might be trained to complete a complex task (e.g., OpenAI Five in Dota 2; (OpenAI, 2018)), but if it is trained only on a single signal (e.g., win the game) then this system may make only moment-to-moment procedural decisions. If the entire problem space is explored well enough, a complex task can be solved without executive autonomy, choosing sub-goals along the way.[8] However, executive autonomy can simplify the training and analysis of a problem space. For example, an AI&A system might make the executive decision to set a sub-goal "to take the north-side outpost." From there, its procedural decisions are made by how well they accomplish that goal.[9] There are differences in how one tests a system that sets goals versus one that just makes procedural decisions. For example, a goal-setting system can be evaluated in smaller discrete chunks, whereas a pure procedural one would need more exhaustive end-to-end testing (see Section 3 for more details). Furthermore, although executive autonomy with military tasks typically carries significant risk, military problems involving perceptual autonomy (e.g., misidentifying a target) may carry as much or more risk.

These definitions intentionally capture systems ranging from the ridiculously unsophisticated to the futuristically complex. A landmine meets our definition of an autonomous system (it has perceptual autonomy), but we are not suggesting that revolutionary methods are needed for a simple pressure-based mine. The type of decisions a system makes—executive, perceptual, and procedural—are one feature of AI&A that influences what we would try to quantify and how we would structure our tests to do it.

The more transparent the system's decision-making is, the smaller the amount of explicit testing we need. When systems make limited decisions in simple ways, it is much easier to achieve this transparency, and evaluating their performance might not require quantitatively measuring their performance. For a victim-activated device, the concern is its ability to discriminate a valid target. A simple pressure-based sensor and a complex multi-spectral data fusion landmine are both making the same decision, and what we need to measure to quantify that performance is the same. That they make the decision without an operator is what makes them autonomous. The difference is where we need to measure it. We understand very well what causes the pressure mine to make its decision, and that understanding lets us make inferences about performance across a range of conditions without directly measuring it.

Testing is about providing assurance, and sometimes assurance does not require exhaustive direct measurement. The better we understand a system, the more inferences we can make, and the fewer measurements we require. As we climb the sophistication gradient however, we begin to encounter more challenges to achieving this assurance. This feature of autonomy—the type of decision the system makes—affects what we measure. In the next section we discuss how decision type can affect testing, and we identify further features of autonomous systems that generate challenges when evaluating decision-making systems.

[8] E.g., Markov Decision Processes

[9] Humans and systems can have multiple goals, sub-goals, and path constraints active at any time.

3. Evaluating Performance for AI&A

A fundamental goal of testing is to provide assurance that a system works as intended (US Department of Defense, 2015). However, the purpose of this assurance is not to provide an across-the-board, binary “adequate/inadequate” evaluation. The purpose is to evaluate conditionally: to understand where the system performs better and where it performs worse. Depending on the stakeholder, this conditional understanding serves different purposes. For developers, this helps locate flaws to be fixed. For leadership (and ultimately the public), it informs whether there is enough overlap between where the system *is* good and where it *should be* good to spend taxpayer treasure on it. For the actual users, understanding where the system is effective and ineffective may be most important: it informs them how to appropriately and safely employ their system.

In general, tests to achieve this conditional assurance are designed by identifying (1) the system’s use cases, (2) relevant outcomes, and (3) the conditions of those use cases which might influence outcomes, and then (4) selecting test points across these conditions that allow us to make inferences about system performance. The processes the DoD uses for these steps need to be adapted to work with AI&A, and even after a functional test design is identified, a number of challenges remain.

We have scoped this section around our proposed solutions to a set of previously-identified core challenges to providing assurance to different stakeholders for an AI&A system (e.g., Ahner & Parson, 2016; Defense Science Board, 2012, 2016; Endsley, 2015). In some cases, these general challenges are not unique to, but may be exacerbated in AI&A. In no strong order, the challenges are:

- **Challenge #1:** Standard test designs assume that varied test factors are causal and that valid inferences can be made across the test dimensions, but black-box systems cannot guarantee this causality, thus limiting interpolating between or extrapolating beyond our test cases.
- **Challenge #2:** Measuring effectiveness of AI&A requires assessing the appropriateness of its decisions, and DoD standard metrics will be insufficient alone.
- **Challenge #3:** Diagnosing the causes of a decision requires data about, at minimum, the inputs received, and ideally about the intermediate processing the system performs.
- **Challenge #4:** The best assurance comes from realistic testing, and the DoD will need new practices to ensure safe but operationally relevant tests.
- **Challenge #5:** Battlespaces are integrated, and a system’s effectiveness is co-determined by its ability to work with others and others’ ability to work with it, and unexpected behaviors can emerge from these interactions.

- **Challenge #6:** Evaluations of AI&A effectiveness must be tempered by evaluations of adversaries' ability to exploit the system both tactically and virtually.

We have not provided an exhaustive treatment of these issues here. Our solutions may spawn challenges of their own, to which yet further solutions exist, which could pose further challenges, and so on ad infinitum. We have delved into these issues enough to create a roadmap for a series of papers that describe in more detail the implementable technical methods. How to politically or organizationally achieve these solutions is beyond the scope of this roadmap. Furthermore, this paper focuses on acquiring *effective* assurance. A second roadmap, currently under development, examines the policies and procedures related to test *efficiency*.

Finally, we do not wish to imply by making a recommendation that no one is doing it that way. In some cases, these recommendations are standard practice for certain groups. However, we have noted that adoption is not universal, and so we include these good practices for the awareness of the entire community.

A. Designing a Test of AI&A, Broadly

The broad strokes of designing a test for AI&A systems will be familiar. We believe it is worth explicitly discussing where processes will overlap, because it provides a common starting point for T&E, reinforces the attitude that these problems are solvable, and clarifies where differences do exist.

Tests are still designed by **identifying** (1) the system's **use cases**, (2) **relevant outcomes**, and (3) the **conditions** of those use cases which might influence outcomes, and then (4) selecting **test points** across these conditions that enable inferences about system performance.

As described in Section 2, the core of AI&A is decision-making, and tests of these systems should revolve around decision-making as well. Many of these systems will involve physical effectors or actuators which must be tested too (e.g., an AI&A aircraft still needs its flight performance tested), but how to select missions, factors, and levels of those factors, and then distribute test points in the physical-domain is understood relatively well in the DoD. What is needed are *a priori* and *post-hoc* methods to identify the missions, factors, and levels relevant to decision-making.

The first stage of decision-centric test design should be identifying, for each of its missions, what decisions the system makes. We recommend testers use task decomposition techniques such as hierarchical task analysis to break down missions into their component tasks and sub-tasks. The level of detail needed in this breakdown will likely depend on the maturity of the system under test, with earlier testing requiring more granular task analysis. Testers should then take these tasks and identify for which tasks the system has autonomy, and what type of autonomy (executive, perceptual, and/or procedural) it has on those tasks. Those autonomous tasks should be the primary focus of testing the AI&A components of a system.[10]

The goal of the decision-centric test is to evaluate whether the system is making appropriate decisions, and so the next stage of test design should be to define, for each task, what an appropriate decision is and select metrics. These metrics will differ by the type of autonomy even within a task. Procedural autonomy selects the next action to take to achieve a goal, and so it can generally be assessed by how well it accomplishes that goal. For example, for an autonomous system that is simply meant to eliminate human specified targets, like our current missiles or some concepts for UAVs, applying traditional metrics like probability of hit or kill will likely be sufficient. The same cannot necessarily be said for executive or perceptual autonomy however, and we discuss those challenges in more detail later in the section.

Testers also must identify the space across which these metrics should be collected—the factors we expect would change the outcomes. For decision-making, these are both the information dimensions that change what the correct decision is *and* the conditions that make that decision difficult for the system to make. At minimum, the former demands subject matter experts (SMEs) of the task and the latter SMEs of the system. Ideally, cross-functional teams involving both SMEs and technical analysts would select the test factors. Though we advocate that this identification should at least happen partly during system design and early test conception, empirical data will be needed to identify some factors. Due to the nature of AI-enabled systems, these factors may not be known in advance and might require experimentation to discover (e.g., Ahner, Parson, Thompson, & Rowell, 2018; Defense Science Board, 2012; Sparrow et al., 2018).

Traditionally, testers would take the identified factors, select levels across those dimensions that will be varied in test, and then use formal or informal techniques to select combinations of factors and levels as the test points. These points can be spread sparsely, because analysts can make inferences between the measured performance points while quantifying their uncertainty in the result. In these traditional test designs, however, risk is driven by uncertainty in measurement, not uncertainty in the causality of test factors. For example, aerospace engineers are very confident that the factors varied in testing are what causally drive an aircraft's flight performance. Uncertainty comes from potential imprecision in data collection, test execution, or factors not recorded, but this is not the sole source of uncertainty in a decision-space for AI&A. The information dimensions that define a good choice are often confounded with other irrelevant information. We lack confidence that the important information dimensions which were varied in test are what actually causally drive decision-making for a given system. Traditional methods of test design and uncertainty quantification do not account for this possibility, and so how to explore the operational space becomes the first major break with tradition when testing AI&A. However,

[10] In OT the focus should still be brought out to the overall mission, but the test should be designed to evaluate the impact of the AI&A decisions on that mission.

the goal is to alter test strategies ways that allow us make valid inferences, so that the T&E community *can* leverage existing uncertainty quantification techniques like statistical analysis and design of experiments, rather than abandon them entirely. In the next section, we will discuss how to enable inference in AI&A.

B. Core Challenge #1: Generalizing Understanding

> RECOMMENDATION
>
> Testers need more transparency in decision-making systems. Before we can confidently test system performance, we must understand how the system makes its decisions. Black-box systems prevent testers from making inferences about untested scenarios.
>
> Testers must obtain, verify, and validate models of what causally drives system decision-making.

If the national security community wants operators to make informed decisions about system employment in environments in which they were not literally trained and tested, then those operators need to understand how the factors in their environment are likely to affect the system's behavior—they need to have a mental model of its decision-making. However, they need to have the right model, and they are not the only ones who need one: testers also need to ensure that the test factors they vary are the ones that actually matter to the system. To enable these activities, testers first have to obtain a decision model, and then put in work to verify, validate, and accredit that model. This is easier for some types of systems than others.

1. The Challenge

To provide assurance across the operational space,[11] there are two basic options: exhaustively test all of the scenarios across this space, or test some of them and generalize those results to the rest (Fisher, 1935; Montgomery, 2019). Decision spaces can grow astronomically large, making exhaustive testing infeasible (e.g., Clarke, Klieber, Nováček, & Zuliani, 2012; Haugh et al., 2018).

[11] The range of situations a system is expected to encounter.

However, the quality of assurance we can provide through generalization depends fundamentally on the quality of our understanding of how the system works, or what causes its behavior to change across points in the operational space (Sparrow et al., 2018). The less one understands, the shorter the inferential jumps one can validly make, and the more data one needs to achieve a desired confidence level. This tension sets up generalization—making informed inferences about performance in situations we have not explicitly tested—as one of the critical challenges of testing AI&A.

Whenever we make inferences, we are making predictions based on a model—our understanding of the underlying factors that affect outcomes. Models do not have to be completely true to be useful; they just have to enable useful predictions (Box, 1976, 1979). Whether instantiated as formal statistical or computational processes, or as informal mental shorthand, when one generalizes, one is asserting that one understands how a change to the conditions would alter the outcome. Valid generalization is not possible without a model.

When it comes to many physical processes, we have a robust and sophisticated understanding of the factors that influence outcomes. These strong models enable us to run efficient tests of our standard systems. Traditional tests are structured to examine performance under some set of conditions and then interpolate between and extrapolate beyond them (Montgomery, 2019). For example, our model of aerodynamics allows us to predict where the edges of an aircraft's flight envelope would be. This lets us test near the edges of the operational space and make inference of safety between them and failure beyond, rather than needing to test every point of the space up to and including failure (Federal Aviation Administration, 2018).

> To understand how a system will behave in a new situation, we need to understand what causally drives its behavior. We cannot do that for black-box systems.

When it comes to decision making, the models we need are similar: we need to understand the factors that causally influence outcomes. However, rather than physical processes, the factors of concern for decision models will be the dimensions of information that influence decisions. In the world of the PSH, this model would be understanding how the system arrives at and represents problem states (perceptual autonomy); what its goals are, how it selects new ones, and how it represents the problem space (executive autonomy); and the process it uses to identify procedures that best meet those goals (procedural autonomy). We even have fairly strong models of human decision making, though we rarely think of them in this way. "Common sense" is really just the set of mental models we assume humans use to navigate and interact with the world (Fletcher, 1986). Yet while we can assume the aerodynamics model will extend from one aircraft to another, the analogous statement cannot be made for decision models. The "physics of decision making" can be totally different between systems, and we assume at our own risk that human decision models ("common sense") hold true for AI&A (Defense Science Board, 2016; Gunning, 2018).

The systems where we have these models—where we understand the information dimensions that drive decisions—are the systems that do not concern us as much in test. A sentinel robot designed to shoot any enemies that come too close is not performing a fundamentally different task than a landmine. We are not concerned with how to test the landmine or developing techniques, tactics, and procedures (TTPs) for its employment because we have an extremely strong understanding of its decision model. We can easily infer that its model would be incapable of discriminating valid from invalid targets. How does the sentinel robot discriminate targets, though? What criteria does it use? How does it detect their proximity? The answers to these questions are a part of its model, and when we know them, we can make better inferences about its performance under different conditions. These inferences let us certify performance and develop TTPs.

A commonly voiced concern about generalization in AI&A is that systems could demonstrate "discontinuities" or large changes in behavior where we have not tested them (Haugh et al., 2018; Sparrow et al., 2018); we advocate that testers use these unpredicted behaviors to test and understand the system's decision model. It is functionally guaranteed that systems will demonstrate behaviors that are divergent from our past experience with them, and we agree that testers should try to find these. Some have advocated that failure modes are a good way to find discontinuities (e.g., Deonandan, Valerdi, Lane, & Macias, 2010; Giampapa, 2013; Luna, Lopes, Tao, Zapata, & Pineda, 2013), and others that behavioral consistency would be efficient (e.g., Harikumar & Chan, 2019; Zhou & Sun, 2019). However, based on our conversations[12] about this topic, the DoD community appears to interpret the search for discontinuities as a performance description process: a search for *undesirable* behavior—discontinuities in space where performance goes from good to bad. The assumption seems to be that in earlier situations, information was affecting decision-making appropriately, whereas at the discontinuity, it is not. While this is possible, especially in specific, rule-based symbolic processing, in sub-symbolic systems, which are commonly universal function approximators (Funahashi, 1989), it is more likely that the system processed all of the situations in the same way. We argue that rather than looking for undesirable behavior, we should be looking for *unpredictable* behavior—discontinuities in space where predictions go from accurate to inaccurate. This will happen when our assumptions about the system's decision model are wrong—either because we are modeling insufficient dimensions or interactions, or because we are assuming the wrong function across that space. From the standpoint of inference, good performance we could not predict hurts our confidence in our

The inability to predict behavior may be more problematic for operations in the long term than merely observing bad but explainable behavior.

[12] E.g., At the Army Test & Evaluation Center's workshop series on AI T&E, the DoD Human Factors Engineering Technical Advisory Group discussion, and various personal conversations.

generalizations just as much as bad performance. Furthermore, CONOPS can be developed around behavior that is bad but predictable. When testers observe a discontinuity in a system's behavior along a dimension or dimensions, they should use it to test their assumptions about the decision model.

When testing decision models, it is critical to differentiate whether the discontinuity arose from the system using the right information in the wrong way, or by using a different information dimension altogether. In the real world, signals of causal factors are often confounded with irrelevant noise. Our testing may make it appear that the system bases its decisions on this causal signal, when in reality it uses the noise. If you move to a part of the problem space where that noise is absent, behavior will change in a way that the causal-signal model would not have predicted. This is not a discontinuity in its behavior;[13] it is evidence invalidating the decision model we believe is in use. For example, in many parts of the world, the edge of the road is correlated with a rise in elevation.[14] If an autonomous car is trained and tested in that environment, it may learn to use this rise and appear to detect the edge of the road perfectly well. Yet when confronted with a cliff, it would detect no rise in elevation and happily drive itself off. This is not a discontinuity—this is consistent with how the system processes the world. We advocate that testers frame using "discontinuities" to invalidate the understood decision model, rather than describing where they exist.

Unlike with humans, in most cases we do not have models of AI&A decision processes. The problem is compounded by a growing trend in AI&A development to use black box machine learning to solve complex problems (Bathaee, 2018). These systems are too complex to test exhaustively (Defense Science Board, 2016), but we also have virtually no understanding of how they work (Bathaee, 2018; Gunning, 2017). However, some methods of creating white-box systems do not necessarily solve the problem either. Testers must evaluate whether these specific rules encompass the entirety of the operational space, which quickly outstrips humans' ability to evaluate as complexity rises. Without a general model that can be used to draw inferences, the entire space needs to be covered, and the system's decision processes still may not be intuitive to humans. If defense developers design systems in these ways, the T&E community must either provide an honest assessment of the level of assurance we can obtain for a given cost, or generate a way to overcome this challenge.

We will need to obtain models of system decision-making.

If a primary challenge to providing assurance is that we lack models that allow us to generalize our findings, then the obvious solution is that we should have them. The problem spaces in which AI&A operate grow rapidly as complexity rises, and the starting point for testing these systems cannot be the type of deterministic testing to which many engineers are accustomed

[13] This is a continuous response in the correct, noise-based model.

[14] E.g., from a curb, sidewalk, or berm.

(Haugh et al., 2018). Testing these systems cannot be about performance alone; it must also be about process. A key goal of testing must be providing assurance that *the way* the system processes information is reasonable and likely to succeed across its employment domains.

To provide the assurance we need, testing must have a model of decision making so that we can generalize, and testing must also provide confidence that the model we have is sufficiently correct—through both systematic and realistic examination—so that we can minimize the likelihood of "discontinuities." We must obtain, verify, validate, and accredit (OVVA) models of system decision-making. Ultimately, we argue that the road to assurance is not through just testing more, but through process-based assurance—testing to understand.

2. OVVA Part One: Obtain a Model

Testers can use data-driven methods to obtain sub-symbolic decision models, and more design-oriented evaluation for more symbolic systems.

How to obtain a model depends fundamentally on how the system is designed. Testers must identify where along two dimensions an AI&A system lies: the extent to which information processing or representation is symbolic (that is, processing content is meaningful, e.g., some traditional computer languages) vs. sub-symbolic (human-non-meaningful processing such as through parallel distributed activation, e.g., neural networks), and the extent to which input-output relationships are monolithic (all inputs occur in parallel, all possible outputs in a parallel layer) vs. modular (outputs of one process act as inputs to another). In practice, these are not orthogonal dimensions, where monolithic systems are more likely to be sub-symbolic (e.g., deep neural networks) and symbolic systems tend to be more modular (e.g., cognitive or software architectures; branching logic). Furthermore, these are dimensions, not binary descriptors, and a system can take a hybrid approach such as having smaller neural network modules take symbolic input and transform it to different symbolic information that will act as input to other neural network modules (Simen & Polk, 2010). As a trend, more modular and/or symbolic systems are easier to test, whereas more sub-symbolic and/or monolithic systems can be easier for developers to optimize.

In general, when an AI&A system is more symbolic and/or modular, models can be obtained by examining designers' choices, whereas more sub-symbolic and/or monolithic systems demand data-driven methods to uncover the decision model (see Figure 1). The ease and quality of the current state-of-the-art for these approaches vary significantly. The next sections discuss how to obtain decision models through data-driven, design-driven, and hybrid methods.

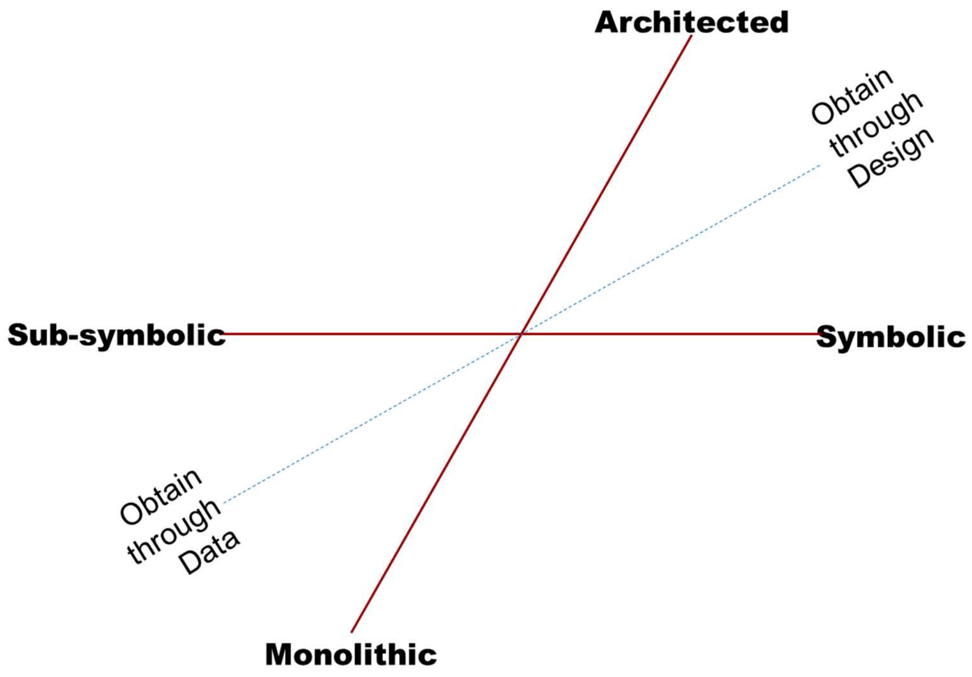


Figure 1. Design choices affect OVVA methods.

a. Obtaining Through Data

A black box is black because we do not understand its causal processes—how inputs are transformed into outputs—but it is possible to demystify those processes (Google, 2019; Gunning, 2017; Mueller, Hoffman, Clancey, Emrey, & Klein, 2019). This is easier said than done, however, and while some methods already exist that aid this endeavor, there are a number of more specific hurdles that must be overcome to find success.

When demystifying a black box, one option is to pick techniques from under the umbrella of what the Defense Advanced Research Projects Agency's (DARPA) Explainable AI (XAI) program are calling "model induction" (Gunning, 2017). Generally in model induction, the goal is to use systematic testing to create a map of inputs[15] to outputs. For example, salience mapping in vision attempts to break down components or super-components of the stimulus to provide a value for how influential each component was for the ultimate categorization made (Erhan, Bengio, Courville, & Vincent, 2009; Simonyan, Vedaldi, & Zisserman, 2014). This input-to-output mapping gives us the beginnings of a model that might let us predict future behavior (and therefore generalize). Extensions of this basic input-output map start to integrate intermediate processing to achieve an input-to-node/layer-to-output map. For example, adding labeled sub-features to images and statistically linking those to activation of intermediate layers or nodes in a neural

[15] Or combinations of inputs, or sub-aspects of inputs

network can allow you to try to better understand what environmental features matter to that part of the network (Gunning, 2019).[16]

However, model induction as a test strategy creates another problem: obtaining data to obtain the model. While there is a good deal of important technical distinction between different techniques under the model induction umbrella, one assumption they currently share is that testers have large quantities of valid, operationally relevant inputs to feed the system. For any sub-symbolic system, there is already tension between using data for training versus testing, and model induction will only exacerbate this problem by introducing new testing demands. Failing to keep data for performance validation has well-documented issues (Raschka, 2018), but to what extent model induction can or cannot reuse training data is a matter of speculation currently. Based on the current model induction methods, we speculate that if it can be reused, it will at least require huge efforts to relabel.

While data quantities are a major hurdle, testers must also contend with ensuring that the data are valid and operationally representative, which will be particularly challenging for systems with full autonomy[17] that are embedded in real-world physical systems. Right now, the main solutions to this problem are either using real pre-recorded mission data (Pinelis, 2019) or simulating environments with high fidelity (Kwashnak, 2019). When systems only possess passive perception (e.g., machine-learning classifiers), feeding pre-recorded data is perfectly reasonable—it is literally operationally representative because it comes from real operations. However, even sensors of the same class can have proprietary or idiosyncratic data formats, or different qualities and resolutions. Sub-symbolic networks can be sensitive to these differences (Hazan, Papandreou, & Tarlow, 2016), and responses to these stimuli may be different from responses to the real environment from a system's own sensors. Common or universal formats could help mitigate some problems (e.g., Defense Science Board, 2016; Endsley, 2015), but whenever possible, if the system is being fed pre-recorded input, testers should do their best to match the data source to the system's sensor. However, even sensor-matched pre-recorded data can be problematic in fully autonomous embedded systems because these systems can change the information they acquire—a drone that is having trouble recognizing an ambiguous object could move to get a better angle or train a different active sensor on it. That decision cannot be made in pre-recorded data, preventing us from obtaining that part of the decision model that way.

It is not clear how testers will – or if they even can – safely and efficiently obtain the inputs for data-driven methods when systems are fully autonomous.

[16] This work is still early in development. At the moment, it is extremely labor intensive and results in mixed success.

[17] Systems that can make executive, perceptual, and procedural decisions within a task.

To obtain models of fully autonomous systems, they must be tested in an environment where they can make all of their decisions; that requires either a behavioral simulation environment or live testing. This is not problematic for purely virtual systems like video game bots or cyber-operations AIs, because their valid live testing can be computer simulated. However, for real-world embedded systems, this creates a catch-22 under current technological limitations. Simulations of the physical world do not represent all aspects of reality, and in order to VV&A them, we must determine whether they sufficiently represent the important factors for the topic at hand (Wojton et al., 2019). With a black box system, however, the reason we need to obtain a model is that we *do not know* what these important factors are yet, and thus we are prevented from legitimately verifying the simulation. One could try to collect these data via live, real-world testing, but this is prohibitively expensive and begs the question of how we will certify these systems as safe for testing on the range. The current consensus is that testers will use behavioral simulations in order to get limited safety releases for live testing (e.g., Ahner et al., 2018; Kwashnak, 2019).

The workarounds to getting valid data are extremely resource intensive. Some are using very high-fidelity simulations built from real locations (e.g., LIDAR mapping an entire range or course; Kwashnak, 2019), ensuring all environmental features at that location are represented. However, this takes a great deal of effort to set up, and sim runs may be slower-than-real-time under realistic computational constraints (Kwashnak, 2019). Furthermore, although a location may have high fidelity, our ranges only cover a limited set of the environmental variability that exists in the world, limiting test representativeness. Alternatively, one can accept some risk and proceed through live testing at a slower rate,[18] but this is likely impractical given the competition that already exists for budgets and range times.

Finally, though the XAI program is making promising progress developing model induction techniques, if DoD intends to pursue these harder-to-certify black-box decision-making systems, then we recommend that investment in this area be significantly increased. There remain many open questions for both basic and applied research (Gunning & Aha, 2019; Mueller et al., 2019), and our workforce does not have sufficient expertise, tools, or training to successfully execute what can be highly technical analysis across the ever-growing body of systems with autonomy (Ahner et al., 2018; Gil & Selman, 2019; Zacharias, 2019b). Model induction can help testers demystify a black box and continue on to the next stage of testing; however, developers, program managers, testers, and evaluators should all be aware that this process can be extremely time- and resource-intensive, and may be outside of their workforce's capability.

[18] Civilian applications have taken this approach, but mitigate the problem through massively parallel testing/training in real environments, e.g., autonomous car fleets driving on the highway (Templeton, 2019). This is not really an option for military systems.

b. Obtaining Through Design

At the other end of the spectrum, one can obtain a decision-model by examining the choices the designer made. When systems are more symbolic, the cause-and-effect relationship between information and behavior is more explicit. When systems are more modular, how information moves and is transformed is clearer. Looking at these choices describes how the system makes decisions. This does not imply that the overall evaluation task is trivial—white-box symbolic systems can still be highly complex and it can be difficult to intuit how they will behave across the entire operational envelope—but the point is that we do not need testing to uncover the causal factors that drive behavior. These are explicitly represented and obtainable by looking.

However, fully symbolic architectures are difficult to optimize and typically fail to achieve the same level of performance seen in sub-symbolic systems—at least when it comes to the more task-specific AI&A that are developed today (Hernández-Orallo, 2016). Though symbolic systems are far easier to obtain models for, the current trend is to adopt at least some level of sub-symbolic processing. If this trend continues, testers may rarely have the ability to obtain models through design examination alone.

c. Obtaining Through Hybrid Methods

Because the sub-symbolic/symbolic and monolithic/modular are dimensional distinctions, designers can adopt a hybrid approach between these endpoints, which also enables a hybrid approach to obtaining decision models. In a hybrid architecture, modules might take in symbolic information and transform out different symbolic content, but do this transformation through sub-symbolic processing (Simen & Polk, 2010). What inputs it processes and outputs it produces are known, though exactly how it does so is not. They might also mix module types, e.g., use a monolithic neural network for perceptual decisions and a fully symbolic system for goal setting. Though tradeoffs of different hybrid approaches are a robust conversation in the field of AI (Laird, 2012), we make no recommendations regarding how or what level of detail at which to specify these architectures. However, developers might choose a hybrid approach when they want more control of the overall processing strategy while still enabling the use of optimizable sub-symbolic processing within a module. For example, a notional computer vision system meant to perform viewpoint-invariant recognition of 3D manmade objects might be designed as a series of modules that transform information into increasingly abstract components.[19]

> Hybrid architecture models can be obtained by examining the flow of information, and then initially verified by injecting synthetic input first into individual modules and then into successively longer cascades of sequential modules.

[19] This is a simplified illustrative example, not a design recommendation.

The first module might be a neural network trained to take symbolic content in the form of RGB pixel values and output the endpoints of continuous edges. The next module might take those symbolic edges and build vertices, pass vertices to a module that builds shapes out of them, then pass those shapes to a module that associates shapes to a single object. The final module might try to recognize objects as spatial arrangements of shapes, while an entirely separate module identifies how far away it is.

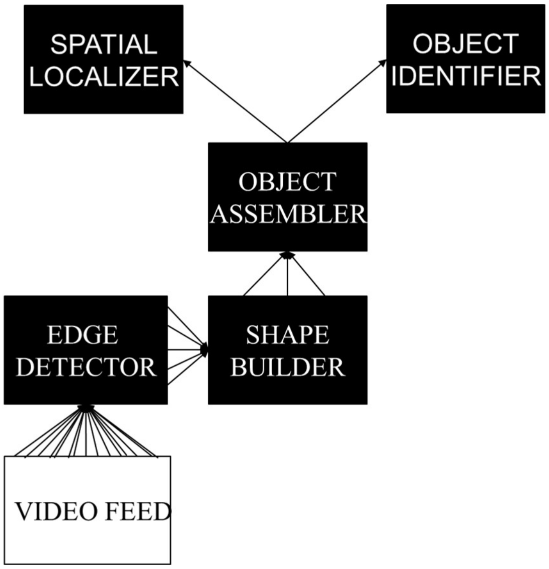


Figure 2. A toy example of a hybrid architecture. Black boxes take and transform symbolic content into more abstracted, compressed information before passing it to the next module.

Hybrid architecture decision models can be obtained through what we are calling cascading compositional verification. A design-driven evaluation examines the flow of symbolic information to understand which information dimensions drive the system's behavior, while a data-driven evaluation examines whether the information transformations happen in the intended way. As needed, testers might use model induction to learn more about specific modules. Because we know the symbolic content that modules take, it is much easier to supply the necessary data to obtain the model: the information dimension is known, so values can be synthetically injected from across the range of possibilities. If testers know what the output should be, the individual module's performance can be assessed. In standard compositional verification, each of the component modules would be tested separately, and if they function, the whole integration is assumed to function.

Because AI&A decision processes will likely rely on emergent properties (e.g., the whole is greater than the sum of its parts—see Challenge #5 for an in-depth discussion), may involve feedback or recursive processes, and may be non-deterministic, we do not recommend pure compositional verification. Instead, after individual modules are verified, testers can start adding modules by cascading through the processing chain from both the top down and the bottom up. The top-down method works by starting at the final module and sequentially adding modules backwards, while the bottom-up method adds modules to the processing chain in a feed-forward manner.

How exhaustive the cascading portion of testing needs to be will likely be a program-specific call based on acceptable risk, but it can be aided by analytic tools. Because modular architectures will operate on garbage-in, garbage-out (GIGO) principles, errors in one module can propagate and compound as you proceed through the processing stream. Compositional verification would let you estimate the performance of a module with some amount of uncertainty;[20] however, if analysts could propagate that statistical uncertainty from earlier modules to the evaluations of downstream ones, it could help estimate uncertainty across larger segments even all of the entire architecture, and help scope testing. The details of how to execute this analysis are an example of the type of content IDA is looking to address in the technical papers that will follow from this framework, and though this is not a fully solved problem, the statisticians on IDA's Test Science team have developed a method that can enable this kind of evaluation across individually tested modules.

d. Proprietary Hurdles

RECOMMENDATION

Testers need rights to system decision-making and learning processes and data generated by these systems. In addition to benefits such as enabling modularity and reusability across systems, gaining ownership rights to the decision software is critical to testing. Proprietary concerns can cause an otherwise transparent system to be a black box to testers, as has already happened with several systems.

Obtaining models through either data- or design-driven techniques can be hindered by proprietary concerns, and it may be necessary for DoD to own rights to the decision systems themselves and data they produce. Programs the authors have encountered to date have refused to

[20] I.e., statistical error

share information with testers about the system's decision making. What may be a white box to a developer might be a total black box to the T&E community, and this would still prevent evaluators from making valid inferences across test dimensions. Without rights to the decision processes, design-driven methods for obtaining a model cannot work. These same programs also prevented the use of data-driven techniques by refusing to allow their systems to be instrumented in order to protect their algorithms. The authors fully acknowledge it will be a fight to get developers to provide rights to their systems and/or data. It is against their self-interest, and DoD has historically failed at this battle. How to muster the political capital and will for this fight is beyond the scope of this framework. However, we argue this fight is absolutely critical in a way that has not been true before.

Having a decision model for AI&A is critical for providing feasible assurance for them. The only alternatives are to test exhaustively or to blindly accept unknown amounts of risk. The former is functionally impossible. The latter will result in warfighter deaths if adopted as practice at the enterprise level. The authors find policies that essentially give up on the idea of independent test and evaluation unacceptable and cannot recommend them.

3. OVVA Part 2: Initial Model Evaluation

After obtaining a model, testers will still need to put in more work to show, through both systematic and realistic testing, that this is actually a sufficiently correct model—they must verify, validate, and accredit (VV&A) it. However, these stages take time and resources, and a preliminary evaluation of the system's decision-making might obviate the need for these expenditures. When discussing the 'quality' of a decision model, testers could be referring to two distinct things: the correctness or adequacy of the model's prediction of the system's behavior (what we need to VV&A), or whether the method of decision-making described by the model is useful for the problem (what we ultimately want to evaluate). It requires much less evidence to show that something obviously will fail to meet high performance standards than it does to show that it meets them, and so prior to VV&A, there should be an early evaluation of whether the way the system appears to make decisions is likely to succeed across the system's expected operations.

> After obtaining a model, there should be an initial evaluation of it to filter obvious failures.

Regardless of the methods used to obtain it, understanding a system's decision model does not guarantee that way of making decisions is actually effective for its task. With our discovered model, we can start to generalize our findings, but the generalization might be that it would fail in most of the scenarios outside of its training data. For example, the XAI model induction community reported that some of the strategies they uncovered in their game playing AIs obviously will not transfer beyond the training set (Gunning, 2019), and computer vision systems often fail to transfer to more complex environments (e.g., Kheradpisheh, Ghodrati, Ganjtabesh, &

Masquelier, 2016; Kosiorek, Sabour, Teh, & Hinton, 2019; Owens, 2020). We predict that preliminary evaluations of decision models will reveal many failing strategies, especially as we move farther from narrow/soft/weak AI that operates in relatively constrained and better defined tasks.

When evaluating a model, analysts should assess whether what the system has learned (or been programmed to do) is a useful abstraction that is likely to succeed across most of the situations it will encounter. This evaluation might involve SME judgment (Goerger, 2004), comparison to systems that have already demonstrated robust performance (Caseley, 2018), or explicit examination of model predictions under different conditions. Which of these strategies or combinations thereof are necessary will be a case-by-case decision and depend to some extent on system design choices. In some cases, an SME might be all one needs. For example, if testers obtained a model of autonomous car perception that showed it defines the edge of the road using painted lines on the road, it is not hard to infer that this will fail on a dirt or gravel road. If the system were meant to operate extensively in those situations, it would be fair to say that the system is not going to be effective and needs to be redesigned before further testing.

To define the operational space against which the model will be evaluated, we reiterate our earlier recommendation that testers deconstruct the system's intended tasks into the information dimensions relevant to decision making. These information dimensions also form the critical factors that must be varied in test, and for some systems are the processing stream designers built the system around. This is not to say that every last dimension must be human specified—there is value in allowing the system to learn stochastic regularities in the environment of which humans are at least not consciously aware.[21] From the standpoint of assurance, the purpose of decomposition is to consider what the safety, ethical, and performance-critical information dimensions *are*, so that we can gain not only some surety that they are processed at all, but moreover that they are processed in a strategy that should work across most of the system's operational conditions.

a. Symbolic Challenges

When white-box, symbolic rule-based system development struggles, it is often because designers created large sets of domain- or scenario-specific rules that do not generalize to other situations, and it is exceptionally difficult to create rules for all situations the system might encounter. Preliminary evaluation of these specific rulesets might focus on the extent to which they cover (and at first blush perform across) the likely operational space. Throughout the history of AI, there have been attempts to manually craft rules and knowledge sets, derived from experts, into a computer (e.g., MYCIN, Shortliffe & Buchanan, 1975). While this approach can and has worked in well-defined problems,[22] most of DoD's imagined use-cases do not meet the

[21] Or simply forgot to specify.

[22] In the formal definition of well-defined, e.g., Newell & Simon, 1972.

requirements of a well-defined problem (Defense Science Board, 2016; US Department of Defense, 2019). Humans fundamentally do not have access to all of the information that goes into their decision-making (Nisbett & Wilson, 1977), and attempts to extract and apply expert rules to ill-structured problems have typically met with more limited success (Zacharias, 2019b). Early space-covering exploration, using automated tools, of how these rulesets perform can catch problems when it is still early enough to fix them (Haugh et al., 2018).

b. Sub-symbolic Challenges

When black-box, sub-symbolic systems fail, often it is because they have been trained on unrepresentative data (Haugh et al., 2018). This unrepresentativeness usually comes—broadly speaking—in two non-exclusive forms: training on tasks or signals that are not the actual desired use case, or using data that do not represent the true distribution of possibilities.

> Training data validation is highly desirable, even necessary, but there is no clear path forward for DoD to scale this activity in the way civilian enterprises do.

The necessity of clean, representative data for machine learning is by now a commonly accepted requirement (Jafferjee, 2019), yet the difficulty of achieving this goal results in regular reports of failures in real-world systems (e.g., Dastin, 2018; Narla, Kuprel, Sarin, Novoa, & Ko, 2018). For real world tasks, it is impossible to have training data that comprises the entire spectrum of possibilities. In machine learning, we expose a system to a sample of data from the operational space and hope what it learns can be generalized to the rest of the space. However, if there are biases in that sample, the system will learn them. Furthermore, systems can over-fit themselves to training data, achieving higher performance there at the cost of transferability (Roelofs, 2019). Because environments (and data sets selected from them) are stochastic, there are many strategies that would work in a single sample of that environment but are bound to fail on average across samples.

Some are advocating that training data should therefore itself undergo V&V (Haugh et al., 2018). While we agree in principle, there are a number of gaps that must be addressed before testers can achieve this in practice: chief among them, scalability. DoD is already wrestling with the question of how its workforce will initially label data for training, and civilian solutions like crowdsourcing are not options for sensitive or classified material[23] and open up the potential for data poisoning (Owens, 2020). Adding a V&V requirement to this already stretched workforce seems unlikely to succeed. Some are suggesting that this V&V process could be automated (Micskei et al., 2012), but that creates a "*quis custodiet*"[24] problem of its own: who will validate

[23] E.g., Will SIPRNet now have a classified intelligence community CAPTCHA to log in?

[24] "*Quis custodiet ipsos custodes?* — Who will guard the guards?

the validator? Furthermore, if we already had a system we trusted to categorize these data, it begs the question why we do not just use that instead. In some cases, the automated validator may be more computationally demanding than is feasible for an embedded system,[25] and this is a way to validate less demanding ones, but at this point we are edging towards describing a process for achieving AI singularity. While the creation and validation of such a system would involve significant investment, given the importance of having clean, valid data, we recommend that DoD investigate the feasibility and potential return on investment of such an automated data validation system.

Evaluators can at least examine training signal or task adequacy.

At a minimum for training validation, however, analysts can at least evaluate whether the task a sub-symbolic AI&A system was trained on was of real operational complexity. It is important to train and evaluate systems on the task one actually wants performed, rather than a simpler abstraction of it.[26] Strategies learned on simpler problems are less likely to transfer to more complex ones than the reverse. For example, the computer vision community continues to relearn that systems that are trained on and work well for the perception of static, 2D images often fail at viewpoint invariant recognition, whereas systems trained to do dynamic 3D perception do not suffer from the same transfer challenge (e.g., Kosiorek et al., 2019). The 2D-trained systems that do not suffer at viewpoint invariant recognition appear to mimic a 3D-trained decision model anyway, such as that of a human (Kheradpisheh et al., 2016). It is not impossible for simpler training to transfer upward, just unlikely, and so part of the reason to obtain a decision model is to assess whether this transfer is plausible.

c. Evaluating Adequacy

Initial model evaluations can include comparisons to certified systems, SME judgment, and low fidelity simulations.

Part of the evidence for whether a decision model will work robustly across the operational space can be the extent to which it matches another decision model that is known to work robustly across that same operational space. When this argument is advanced, typically proponents are referring to reusing modules that have already undergone extensive testing (e.g., Caseley, 2018; Zacharias, 2019b), and therefore reusing the evidence that the capability works. While we support the pursuit of common, reusable modules (e.g., it's preferable not to create and validate 20 different passive video perceptual systems), and the reuse of evidence, at the moment we do not have this historical body of evidence for AI systems. Instead, the only robust decision models we possess

[25] E.g., creating Size, Weight, and Power (SWaP) problems (Sparrow et al., 2018).

[26] Simple abstractions can be popular because it is easier to collect data and/or create simulations for these tasks.

are those for biological organisms. Over the past century-and-a-half, scientists have made great strides toward uncovering the underlying structure of the human[27] mind (Newell, 1990). Low-level structures are relatively common across species and evolutionarily tested for robust performance in common environments (Dunbar & Shultz, 2007; Heyes, 2012; Singh, Lewis, Barto, & Sorg, 2010). By taking advantage of this knowledge, we not only become more likely to develop good systems (Defense Science Board, 2016; Krichmar, 2012; Krichmar, Severa, Khan, & Olds, 2019; Kurup & Lebiere, 2012; Neema, 2019; Zacharias, 2019a),[28] such as with dynamic 3D perception (Kheradpisheh et al., 2016; Kosiorek et al., 2019), we also gain a head start on assurance by providing evidence that the system's decision model matches one we know works well for this task. As we move forward and begin to find and validate robust, non-biological information processing strategies, we can add these strategies to our comparison library.

A major part of the evaluating whether a decision model will generalize is assessing how the problem is represented—what problem state perceptual autonomy creates upon which procedural decisions are made. Having a representation that matches the procedural algorithm is critical for problem solving (Pretz, Naples, & Sternberg, 2003). In spatial navigation for example, a strategy that should mostly work across all environments would be to make procedural decisions based on a four-dimensional representation of where objects are and will be in space. Perceptual autonomy built around that process could feed procedural autonomy that chooses where to move to minimize collisions.[29] Evaluating the adequacy of the problem state representation can be either holistic SME judgment or a more quantitative comparison of the task decomposition dimensions to those represented by the system.

Often the adequacy of a decision model will be ambiguous and demand more than SME judgment; however, higher-fidelity examinations move away from the resource-saving purpose of the preliminary evaluation. We recommend the use of lower-fidelity space exploring tools to perform this early assessment. For rule-based systems, tools such as the Range Adversarial Planning Tool (RAPT) or the Autonomous Systems Test Capability (ASTC) can help to efficiently explore this space without massive computational overhead (Kwashnak, 2019; H. Miller, 2019). However, these systems are not necessarily intended for sub-symbolic AI&A, and different

[27] And non-human

[28] Many of the major breakthroughs in AI research have been finding ways to implement biological processing strategies in our technology. Whether it was the first artificial neural networks, or AlexNet implementing visual-cortex-inspired processing on a GPU, many leaps forward are biomimetic.

[29] For an illustrative description of a simplified toy perceptual model: the perceptual basis of knowing where objects are located is (1) the capability to distinguish distinct objects in the environment. If we have distinct objects, we can (2) infer where they are relative to each other. If we know where distinct objects are, we can now (3) track them over time with memory. If we can track over time, we can (4) infer velocity. If we have distinct objects' location and velocity, we can (5) integrate information to construct the 4D map. From that 4D map, we can (6) train to make procedural decisions that meet the goal of minimizing collisions. Just one of many ways to implement this in a hybrid architecture would be to use supervised learning mechanisms separately at each of those processing stages, instead of just at the end of the task for successful navigation.

techniques are needed there. If the decision model obtained comes in the form of a mathematical equation predicting how values of information alter behavioral probabilities, one can compute and evaluate these (Hernández-Orallo, 2016). Other times, one might need to simulate the system in an environment to check its behavior. Agent-based modeling (ABM) is a useful tool for lower-fidelity descriptions of cognitive agent decision-making (Greer, 2013; Ilachinski, 2017), but these also require some VV&A, even if it is a lower burden.

Early in system development, there will probably be many of these obtain-evaluate cycles before an apparently useful solution is found, especially if developers rely on sub-symbolic machine learning processes. However, it is highly likely that some of these design failures will not be found until later in what would traditionally be development or operational test. The entire acquisition community should be prepared for iterative development[30] and test (Ahner et al., 2018; Defense Science Board, 2016; Gunning & Aha, 2019; McLean, Bertram, Hoke, Rediger, & Skarphol, 2016; Trent, 2019), and the likelihood of failure; but by integrating these T&E processes early in the cycle with lower threshold fidelity, we are more likely to catch duds before they become too big to fail.

4. OVVA Part 3: Verifying, Validating, & Accrediting a Decision Model

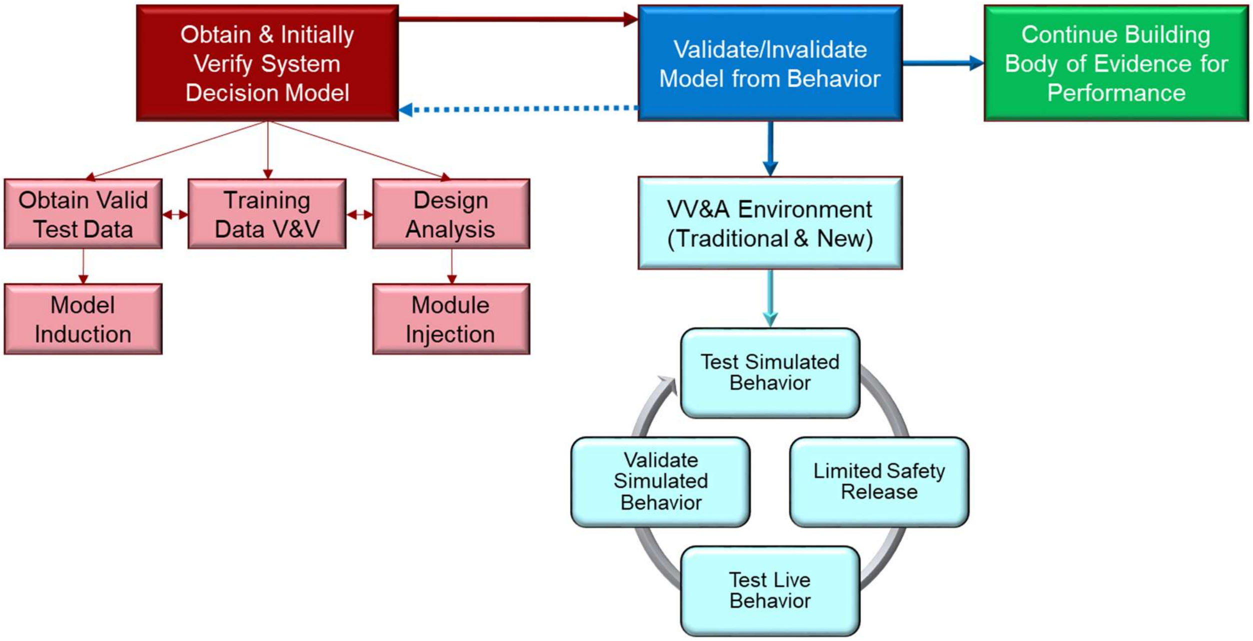
Decision model VV&A will need to try to invalidate the obtained model based on system behavior. This will need to be an iterative process before traditional T&E.

We cannot rest on our laurels after the preliminary evaluation of our obtained model—we must verify that this model adequately explains system behavior, and validate that the model meets these expectations in realistic conditions. Failing to do so invites the "discontinuities" that worry many testers. The purpose of the model is to know how information changes behavior. This allows us to manipulate these dimensions in test and have confidence when generalizing our results. In the first step, testers should generate predictions from the decision model about how the system will behave[31] or respond under certain conditions. From there, the goal is to hunt for discontinuities and use them to falsify our believed decision model: we test the model's predictions against the system's actual behavior in increasingly realistic scenarios. This requires testing the system's actual decision processes (e.g., symbolic branching logic; deep neural network) in either a simulated environment or live testing. As the behavioral simulation itself will need to undergo VV&A, we recommend that model testing spiral through risk—use the simulation both to test the

[30] And we do not just mean fake agile (Defense Innovation Board, 2018).

[31] Though colloquially behavior is assumed to be an overt physical action, in this paper we use the term broadly to cover the result of a decision. An intelligence, surveillance, and reconnaissance (ISR) image classifying system might not have any physical actions, but the act of labeling an object and forwarding that information somewhere would be its behavior.

model and to obtain limited safety releases for live testing. These live test points can be used both to validate the decision model and the behavioral simulation environment (see Figure 3). As with the preliminary model evaluation, the acquisition community should be prepared for iterative and adaptive testing. The model VV&A should not be a single shot demonstration. We should expect to find behaviors that invalidate our decision model and require us to refine it—either because the information dimensions are wrong or the model is insufficiently complex to adequately describe the system.



Obtain & Initially Verify System Decision Model
Validate/Invalidate Model from Behavior
Continue Building Body of Evidence for Performance
Obtain Valid Test Data
Training Data V&V
Design Analysis
Model Induction
Module Injection
VV&A Environment (Traditional & New)
Test Simulated Behavior
Validate Simulated Behavior
Limited Safety Release
Test Live Behavior

Figure 3. Overview of the OVVA Process

Ultimately, having a decision model serves two purposes: having confidence in the dimensions across which we are making inferences, and also providing a lower-fidelity—but critically, still adequate—method to make those inferences. As discussed here and elsewhere (e.g., Defense Science Board, 2016; Sparrow et al., 2018; Wegener & Bühler, 2004), we cannot cover all of the possibilities in AI&A systems. Our general recommendation for testing is to use lower-fidelity methods to try to cover more operational space and higher-fidelity testing to verify and validate those simpler techniques (e.g., Harikumar & Chan, 2019; Laverghetta, Leathrum, & Gonda, 2018; Visnevski & Castillo-Effen, 2010). For this purpose, the decision model is one way to let testers cover a large amount of space with fewer resources. Behavioral simulations can let testers systematically though not exhaustively visit a large part of the operational space, while live testing can be used to validate the simulation. The idea is to test efficiently cover the space while still reducing risk.

VV&A may need to cycle through levels of test fidelity and realism in order to manage risk.

There may not be a single unified decision model for a system for all time. The type, fidelity, complexity, and predictive precision one wants for a model depends on its purpose and the level of error one is willing to tolerate. Different stakeholders are going to have different needs and make different choices for the model. A tester who needs to be confident when interpolating or extrapolating test results requires a different model than a warfighter deployed to a conflict zone who needs to make broadly safe real-time decisions to employ the system or not. Early testing may call for more granular models than capstone testing before a full-rate production decision. In this section, we focus on the decision model that leads to system behavior—how executive, perceptual, and procedural decisions interact to produce operationally relevant actions. However, the OVVA process is not limited to this, and readers should be aware that they may need to have more detailed models of individual decision types for their systems.

Testers should keep in mind a fundamental tenet of testing: we cannot prove, only falsify. As a result, we should try to maximize the value of our test points in terms of how informative they are about our model. Exhaustively retesting the centroid of the performance envelope may be less valuable than looking for brittle edges or discontinuities, and testing points that do not allow us to differentiate between two competing decision models may not be particularly helpful. Alternative Design of Experiments (DOE) techniques can be of use here. DOE attempts to optimize the value of test points to some criteria (Montgomery, 2019). DoD is familiar with DOE techniques that optimize to statistical uncertainty (error) and our ability to detect effects (Director Operational Test & Evaluation, 2009), but these are not the only criteria testers can use. For example, some fields use Shannon entropy to optimize their DOE, while others might use the Akaike Information Criterion (AIC) to optimize for model selection (Nowak & Guthke, 2016). Which DOE techniques are best for the OVVA process is another candidate for a technical paper to emerge from this framework.

Furthermore, because OVVA testing will involve a large amount of exploratory testing, testers will need tools that help them adaptively plan test points over time, rather than monolithically designing large test events years in advance, as is often the case now. Here, sequential DOE is a tool that can help testers adaptively, efficiently, and effectively execute the OVVA process. However, though sequential DOE has been used by industry for decades, there remain methodological challenges to adapting it to DoD AI&A T&E (Ahner & Parson, 2016; Hess & Valerdi, 2010), such as how to meaningfully perform factor screening on categorical variables. IDA's Test Science statisticians are also working on solutions to this problem.

Testers will need to be able to diagnose the causes of unexpected behavior in order to invalidate the decision model. If for instance we get unexpected behavior because a sensor or actuator failed, this does not really constitute evidence against the decision model. Furthermore, GIGO problems can make it difficult to assess the decision model overall: our understanding of the executive and procedural systems may be correct, but because we do not adequately understand the perceptual system, we could have difficulty predicting the system's behavior. However, diagnosis as a challenge is not limited to OVVA (Greer, 2013; Sparrow et al., 2018), and is a complex enough topic that we devote Challenge #3 in this paper to the problems and our proposed solutions.

This diagnosis problem leads to a need to describe whether our model or models are adequate in a more continuous and conditional way. An across-the-board, thumbs-up/thumbs-down evaluation will not help testers identify how the model can be improved. One—though not the only—method of doing this is to use what we are calling "Bayesian discrepancy modeling" as a method for simulation validation. In this paradigm, we use a Bayesian framework to try to estimate, based on our live test data, the continuous probability[32] that the simulation results would acceptably[33] match reality. A further advantage of this method is that it gives different continuous probabilities across the different test conditions.

5. Summary of OVVA

To make valid inferences, testers will need to obtain, verify, validate, and accredit a model of the system's decision making process. To identify what test methods are appropriate for obtaining the model, testers should identify whether the system is designed to be more modular or monolithic, and whether its processing is more symbolic or sub-symbolic. After a model is obtained, testers should perform an initial evaluation to weed out obviously bad systems. From there, testers should examine the system's actual behavior against the model's predictions to see whether the model sufficiently explains the system. Testers can use a spiral of simulation and live testing to perform this step.

The model is a key part of assurance, but though we may believe when choosing it that the strategy is effective, belief is not proof. It may be the case that the designer's choice of decision strategy was wrong, that the architectural implementation of the strategy was incorrect, or that the system failed to optimize the processing steps sufficiently. Any of these could lead the system to still be ineffective despite possessing the system's decision model. The beginnings of assurance come from the decision model representing a reasonable, robust strategy across the operational space—the rest must come from test.

While testers might be tempted simply to evaluate the accuracy of perception or the appropriateness of behavior as outcomes alone, some argue this will not be sufficient (Sparrow et al., 2018), and this high-level approach misses the point of model-based assurance. Classification accuracy in computer vision, for example, does not confirm the system would work outside of our test cases—the goal of the test procedures described is to try to falsify that assertion that the factors *we believe* the system uses to make decisions are the factors the system *actually* uses.[34] By stepping systematically through the processing stages, we test whether those stages actually do what we believe, and we quantify their performance. Having a verified and validated model of the

[32] 0-100 percent

[33] As defined by the level of risk a stakeholder is willing to accept and/or by operational significance.

[34] The model does not need to be complete or comprehensive, just sufficiently useful and correct.

information dimensions the system uses to make decisions allows us to vary those dimensions during performance or certification testing and have confidence when we make inferences between and beyond our test points.

Having a model of system decision-making and being able to generalize our test findings is a major step on the road to assurance. This allows us to use the basic test philosophy of identifying factors and testing across some operational space without being exhaustive. However, it is not the only step. As we proceed through the other challenges to assurance for AI&A, we will refer back to many of the ideas discussed here, and in some cases expand upon them. We frequently referenced assessing "decision appropriateness" as a part of validating a system's model without providing information on what that would mean. In the next section, we discuss what it means for AI&A to be effective and the challenges of demonstrating that effectiveness therein.

C. Challenge #2: Measuring AI&A Effectiveness

RECOMMENDATION

Research into and dissemination of methods for evaluating decision-making are needed. These include metrics to quantify intermediate mission success, methods to qualitatively evaluate overall decision processes, novel calculations of classification accuracy for multi-categorical fuzzy groups, and ways to quantify a system's ability to learn.

In order to systematically evaluate decision-making systems, we need ways to measure the appropriateness of the systems' decisions. This needs to include both what to measure and how to go about measuring it. Quantifying decision-making will be a critical step on the path to writing testable and verifiable, but operationally relevant, requirements specification, which many have identified as a major hurdle to T&E. For executive decisions, this requires evaluating whether goals chosen were useful to the mission. Having intermediate outcome metrics for missions will make this easier, and examining the reasons underlying goal decisions will allow more holistic assessment. For perceptual systems, testers will need better methods for defining error and accuracy, but also better training in these more complex analytic techniques. Procedural autonomy will likely be the easiest to measure simply through the achievement of the goals those decisions are supposed to support. Finally, systems will be expected to adapt to changing battlefield expectations, either over the long term or in real time, and testers need to develop methods for measuring the capacity of systems to learn. This would enable system comparison for benchmarking or source selection.

1. Basic Test Designs

Human decision-making certification methods should be the starting-point, but not end-point, of AI&A certification.

The T&E community and public at large is experiencing great consternation over how we will certify AI&A. These concerns are a mix of legitimate challenges and ill-informed doom-saying. Many authors' work on the former contributed to the development of this roadmap (e.g., Defense Science Board, 2016; Endsley, 2015; Helle, Schamai, & Strobel, 2016; Macias, 2008; Roske et al., 2012), but this document is as much as anything a response to the latter pessimism. While it is absolutely true that failing to develop the new test processes will limit the credibility of assurance provided for an AI&A system (Defense Science Board, 2016), those who claim we have no idea how to test these systems are mistaken. The fundamental difference in AI&A is that these systems make decisions, and the appropriateness of these decisions is a critical factor in their effectiveness. When decisions are appropriate, regardless of whether they are executive, perceptual, or procedural, they advance the full set of goals[35] relevant to the mission. However, how to certify the appropriateness of decision-making is not a problem that originated in AI&A.

The starting point for developing methods to certify artificial decisions should be our methods for certifying human decisions (Defense Science Board, 2016). Many of the certification challenges are not unique to AI&A, but are issues that we have already mitigated or solved with humans. Testers should start by clearly defining what decisions the system makes, and then look at how we gain trust that a human would make those decisions appropriately. In some cases, there may be literal analogues of certifying the decision made. For example, there are common missions that have many sub-tasks drawing on very different skills, that occur in environments that are too complex and varied to simulate reliably (so must be tested in real life), but pose significant risk to life if the task fails. How then would we safely test these systems? We note, though, that this is the exact same challenge we face in training and certifying human doctors. Using the graded autonomy process developed for medical residents (Halpern & Detsky, 2014) as a starting point

[35] Often when people theorize about inappropriate AI&A decisions, they end up describing programmers failing to explicitly define the full set of 'common sense' goals people intuitively assume.

for testing a robot surgeon would be beneficial.[36] These processes and metrics can form the basis, though not end-point, of our test strategies.

We do not advocate that testers blindly adopt human-certification techniques. Testers should use the general shape of testing as the starting point, not the amount or exact execution of it. These techniques take shortcuts because we can assume that—in humans—this evaluation of decision-making is just the next step in a long chain of implicit and explicit testing. We can have relatively anemic certification testing because these tests do not exist in a vacuum. A sixteen-year-old arriving for his or her driving test does not need a full shakedown of all cognitive abilities. Low-level perceptual and motor processes are common amongst humans and evolutionarily selected to be robust across domains (Kandel, Schwartz, Jessell, Siegelbaum, & Hudspeth, 2012; Smith, Zakrzewski, Johnson, Valleau, & Church, 2016), so we are confident they are using perceptual and motor processes that should work for driving. Life itself has also acted as operational testing of these capabilities: surviving to that age helps assure those processes are functioning. Explicit tests of other skills (e.g., school) have also examined those abilities. We do not need rigorous testing there—a brief confirmatory eye-exam will suffice, and the skills portion of the test becomes more about whether they have successfully translated their domain-general motor control[37] to the specific domain of driving. We have less, but still some confidence in a human's executive autonomy at that age (Best & Miller, 2010). We assume they probably are pursuing reasonable high-level goals (e.g., "Don't die."), but a major part of the evaluation is whether their risk-taking behaviors appear well-matched to that goal (e.g., not making turns into oncoming traffic). These are assumptions we can make about human drivers that we cannot make about self-driving cars. We do not know that the AI&A car's perceptual autonomy uses robust strategies; we do not know that its effectors are capable, and we do not know what goals it is pursuing and whether these are reasonable. These must be tested.

Even when we cannot adopt the certification techniques we already use, examining them for what we assume *does not* need testing in humans strongly informs the capabilities we must explicitly test in AI&A. Techniques like hierarchical task analysis or decomposition can aid in this breakdown (Stanton, 2006). Testers should still use human techniques as the starting point, identify which gaps our assumptions about human skills create, and then explicitly test the system's performance on those skills.

2. Metrics for Effectiveness

Testers will need to measure AI&A performance using metrics that are rarely employed by the DoD T&E community. What makes a decision appropriate depends heavily on the type of decision being made and the context of that decision. There will not be a one-size-fits-all solution, and the metrics for executive, perceptual, and procedural autonomy may be very different from

[36] We will expand on this certification strategy later in the document.

[37] I.e., part of their procedural autonomy

each other. However, metrics within those broader categories will likely share a number of features, and T&E will benefit from developing common processes for the development and selection of metrics.

a. Executive Autonomy

Evaluating executive autonomy requires assessing whether the system set goals for itself that enabled its success. There are a number of challenges to this: it is not obvious how to connect goal-setting to mission outcomes; we lack reliable, objective metrics to quantify goal appropriateness; goals have auto-correlated effects—future option-availability depends to some extent on earlier decisions made; and goals *not* pursued may be as important as what was chosen.

There are two basic options for testing executive decisions: try to quantitatively tie individual decisions made to task outcomes, or qualitatively assess the *process* by which decisions are made (as we do in humans). However, as we will expand on in this section, each of these options requires work before they are implementable. We make two recommendations to overcome their problems:

- T&E should develop standard processes for programs to use to select or create intermediate mission-outcome metrics.
- In parallel, making AI&A at least transparent, and ideally explainable, can aid evaluation by enabling SMEs to evaluate the system's overall goal-selection process.

1) Intermediate outcome metrics

Ultimate mission outcomes are multiply determined, and having multiple causal factors makes it difficult to evaluate the effect of individual executive goal decisions at that level. A mission might fail not because the goal chosen was wrong, but because the system failed to execute pursuit of that goal effectively. Additionally, executive decisions might still be optimal under the situation's constraints or available information even if they ultimately fail. Trying to disentangle the relationship between goal decisions and ultimate mission outcomes will be difficult for both interpretability and statistical reasons.

> Quantitative intermediate outcome metrics can help evaluate executive decisions. For example, tracking yards gained and points instead of just winning or losing the game in American football makes it easier to evaluate individual play calls.

The more indirect a relationship is, the harder it is to detect, and the distance between goal decisions and ultimate outcomes makes them difficult to model statistically. This distance could be defined many ways, such as the number of decisions that occur between two states, the degrees of freedom or number of possible options, or simply elapsed time. Though the goal choices closest to the ultimate outcome have the most direct (and therefore statistically detectable) connection to it, they might not have had the strongest influence. Early choices can open up or restrict the goals that can be pursued down the line, and those first choices can end up being the most important.[38] However, the problem space between those early goals and final results may be huge (and therefore hard to detect statistically). There are many degrees of freedom between most goal decisions and final outcomes, and the large number of degrees of freedom require larger and larger tests to detect relationships quantitatively.[39] Reducing the distance between decisions and outcomes they predict is desirable, but testers must do so in a way that preserves operational relevance.

To assess the effect of individual goal decisions, we recommend that testers not just track final outcomes, but also record intermediate progress along the way. This intermediate progress provides a metric closer to individual decisions, reducing the degrees of freedom so that their impact can be evaluated more easily. From a statistical standpoint, these metrics would ideally be continuous rather than binary, as this increases sensitivity (Altman & Royston, 2006). These metrics are not just for evaluation, however—they also provide signals on which the system can be trained.[40] Though more critical for AI&A, these metrics would help with operationally testing standard systems as well.

We recommend that intermediate mission success metrics become an area of systematic research effort. Creating observable metrics for ultimate and intermediate mission success would help develop and evaluate AI&A executive autonomy, but to the authors' knowledge at the time of writing there is no concerted effort to create these metrics in DoD. Industry and competition teams have intermittently pursued them (e.g., Silver et al., 2016; Wegener & Bühler, 2004), but the attempts have usually been one-off for specific purposes. Given the importance of this to virtually all AI&A development and testing, as well as the potential benefit to the operational testing of standard systems, and the need for resourcing, DoD should consider investing in this capability as soon as possible.

[38] For example, the play call in a third-and-long situation in American football is an executive decision that strongly determines whether the team will get a first down. However, one might only be in that situation because of attempting low probability passes on the first two downs.

[39] The degrees of freedom also make it more difficult to train the system in the first place, not just to evaluate it.

[40] One of the AlphaGo team's great insights was developing a way for the system to assign value to intermediate board states (Silver et al., 2016).

Leveraging interdisciplinary teams to develop processes for creating metrics, rather than specific measures, may provide more enterprise-level value at this time.

However, we recommend that at this time the first step be development of standard *processes* for *creating* or *selecting* metrics, rather than specific intermediate outcomes. We foresee three potential pitfalls in leaving metric development to occur organically for individual programs. First, the skillsets needed to work effectively in the AI world are both diverse and in short-supply within DoD (Zacharias, 2019a). Given the problem's difficulty and the expertise shortage, without guidance programs will likely develop low-quality metrics. Secondly, programs are incentivized primarily for their own success, and metrics may be developed that are usable for their own specific needs, but are not re-usable and are less advantageous at an enterprise level. Finally, the stovepiped nature of acquisition may hinder programs from discovering quality, reusable metrics that others have already developed. By initially focusing expertise on developing a *process* for metric creation, the T&E community can mitigate quality control problems and encourage development of metrics usable across a mission area, not just for individual systems. Furthermore, by providing a central knowledge-sharing mechanism (e.g., a repository in the JAIC), DoD can alleviate the problem of stovepiping.

We recommend that this metric-creation process be a multi-disciplinary collaboration between industry, academia, and the military. AI&A lives at the intersection of many disciplines (Endsley, 2015; Laird, 2012; Zacharias, 2019a), and these metrics must also be operationally relevant. A process developed only by one field may ignore the necessary insights from another (T. Miller, Howe, & Sonenberg, 2017). This is a brave new world, and we do not yet know what the right process looks like. Many proposals will need to be solicited and tried, and these proposals should be created from all the relevant AI&A fields. Historically however, these fields fail to interact (Steinberg, 2019; Zacharias, 2019a), and so some incentivization may be necessary. For example, one method that has succeeded in the academic world to foster collaboration between traditionally separate fields is to require that grant applications demonstrate they have representatives from all (or most) relevant fields (e.g., "MCubed Program Requirements", n.d.). These fields should include, at a minimum, the computer, cognitive or psychological, and statistical sciences; engineering; and operations research. Military SMEs will be indispensable for this research, and so whoever leads this systematic research effort should also work to foster increased partnerships between academic and military worlds. When these metrics deal with legal, moral, or ethical (LME) outcomes (e.g., fratricide, collateral damage, escalation) we recommend lawyers and philosophers be included as well.

2) Transparent or explainable AI&A

SMEs can evaluate the process or reasons behind executive decisions. This requires transparency or explainability in our AI&A systems.

The reason we do not already have these intermediate metrics appears to be that our methods for certifying human executive decision-making rely on SMEs qualitatively evaluating the decision process rather than the individual decisions (Simpson, 2019). For military tactics certification, for example, those SME evaluations look very similar to the type of model-based assurance discussed in the generalization challenge—examining how students represent the problem and the soundness of the reasoning (Simpson, 2019). While SME evaluations may include directly observable criteria, those criteria are not sufficiently diagnostic to be reliable on their own (Simpson, 2019). If a questionable decision is made by a student, evaluators ask that the decision be explained. The reasons underlying the decision are more useful for deciding whether that student will be able to generalize their performance to other situations.

Among the reasons for DoD to pursue explainable AI (XAI) is that it can help evaluate executive autonomy. There are some who treat XAI as a bonus, an extra thing to be pursued, but there is evidence that explainability is fundamental to human problem solving, not an additional capability. Research suggests that problem solving is not a separate module in humans, but is an ability scaffolded on top of our natural language processes (Polk & Newell, 1995). Although it is extremely unlikely that the only way to achieve artificial flexible reasoning is to use language or semantic networks as the underpinning, these approaches may be a desirable path to pursue over others. Systems that are explainable—that can themselves explain their decisions to others—are desirable in and of themselves for end-users; explainability also strongly enables T&E of executive autonomy, and it may be the most-likely-to-succeed path forward to flexible problem-solving anyway. However, while explainability is desirable, transparency in system decision-making—knowing the causal relationship between information and behavior, i.e., having a decision model—could theoretically be sufficient for evaluation. If testers have a model of how the system sets goals, they can additionally provide that model to SMEs to attempt a subjective evaluation. This would be a mitigation at best, however—we still recommend that DoD pursue explainability in its systems.

b. Perceptual Autonomy

The goal when testing perceptual autonomy is to evaluate whether the system's representation of the problem state accurately reflects reality. For autonomous systems, perception is the basis of action, and so it becomes especially critical to demonstrate that systems sufficiently represent the situation currently facing them. In this section, we focus on the topic of categorization. Although perceptual autonomy is not limited to this, the potential for high regret through improper categorization and the number of fruitful research avenues make this a worthwhile topic.

Evaluating perceptual autonomy will require non-standard methods for defining error and evaluating accuracy.

In general, assessing accuracy requires asserting a ground truth and defining error in relation to that ground truth. This creates a few core problems when evaluating AI&A perceptual accuracy: ground truth is not always known, the dominant approach of using binary error terms will not always be appropriate, and context can change what accuracy means. We make three recommendations relating to these problems:

- Testers should invest more resources in accurately capturing ground truth during testing and proceed with extreme caution in situations where ground truth is subjective and/or lacks consensus.
- Testers should pursue the development and dissemination of non-binary accuracy metrics.
- Testers should use perceptual accuracy metrics that account for how context can change the correct categorization.

3) Establishing ground truth

Before we can assess accuracy, we must have ground truth. Accuracy is not absolute—it is an outcome relative to a target. If we do not have confidence in what we are asserting is the target, we cannot have confidence in our evaluation of the system's responses. If a system says there is a tank in those trees at 1309 local time, and if we cannot see that location ourselves, we would have to know the location of all tanks on the range at 1309 to evaluate whether it was correct. Although it is not universally true, testers often fail to sufficiently capture ground truth during their events to make these assessments, particularly during operational tests that might last for days at a time. This may be only minimally disruptive for certifying our standard systems, but greater care may be needed with AI&A. This will take more resources than we often currently expend, but if ground truth can be asserted objectively, then for AI&A it should be asserted from precise recorded data.

However, in many cases ground truth might not be objective or may lack consensus. There might be objective, measurable criteria that predict people's responses, but there is not universal consensus on what those criteria should be. More operationally relevant and most moral or ethical categorizations will fall under this subjective umbrella, and assertions by developers or testers can have significant LME consequences. We recommend that everyone proceed with extreme caution when asserting a subjective ground truth such as acceptable collateral damage or fratricide risk from which system perceptual accuracy will be assessed.

4) Defining error

Once ground truth is defined, testers must choose how to define errors, and there are multiple ways to do so. This decision can change what the estimate of a system's accuracy is, making the error definition possibly the most important one for perceptual autonomy. To pick an appropriate error term, testers should start by identifying whether the perceptual decision is best described as binary or fuzzy (is or is not versus a matter of degree), and discrete or multi-categorical.

Formal perception test methods have typically fallen under the rubric of Signal Detection Theory (SDT: Marcum, 1947; Peterson, Birdsall, & Fox, 1954; Tanner & Swets, 1954). In the dominant SDT approach, both ground truth and perception are assumed to be discrete-binary—e.g., the thing *is* either present or absent, and the system *represents* it as either present or absent (Green & Swets, 1966). For example, the aircraft is actually there (or not), and the radar either tracks it (or does not). This results in two basic flavors of error in a confusion matrix: the system believes the thing is present when it is in fact absent (a false alarm), or the system believes the thing is absent when it is actually present (a miss). Establishing ground truth to make this assessment might not be trivial or free of error, but it is clear at least what must be compared. While standard SDT will continue to be relevant for AI&A, it may require adaptation for certain perceptual decisions.

In the real world, not all errors are discrete-binary. Things can belong to multiple simultaneous categories (they are non-binary: Ashby & Townsend, 1986), and membership in those categories may be a matter of degree (they are fuzzy, not discrete: Parasuraman, Masalonis, & Hancock, 2000), and it may be both of these together (O'Connell, 2015). Categories can have different structures that influence error: they might be single-label non-binary, uniformly distributed or not (e.g., variants A, B, C); they might be hierarchical (e.g., taxonomies—a Persian[41] is a type of cat is a type of mammal is a type of animal); they might be orthogonal or correlated multi-label groups (e.g., male/female sex is largely unrelated to nationality), or have other structures. Standard SDT error terms are not fully applicable to fuzzy, multidimensional categories (O'Connell, 2015).

Some work[42] has examined how to process accuracy in fuzzy, multidimensional, or multidimensional-fuzzy categories (e.g., Ashby, 2000; O'Connell, 2015; Parasuraman et al., 2000), and some fields apply them, but the techniques are not as well-researched, widely known, or

[41] Categories can also be homographic or homophonic. Persian is both a cat breed and human ethnicity.

[42] For example, General Recognition Theory (GRT) is a multidimensional extension of SDT.

frequently applied. Binary SDT errors could be used here, but this can lose important information. Given that these metrics will be relevant for huge numbers of programs, leaving the development to individual teams will likely result in redundant parallel efforts. Bringing these efforts under a single roof that will invest in further research in extending SDT, workforce education in these techniques, and the development of standard procedures to define necessary precision and cost-function can help avoid these problems.

5) Context affects accuracy

The context in which decisions are made can change what the accurate categorization is, and the context in which decisions are tested can change how we compute that accuracy. Testers must account for these possibilities when evaluating perceptual autonomy.

The development and T&E communities should create a guide for defining necessary precision in perceptual decisions. What is correct or good enough in one situation might not be elsewhere. In a hierarchical taxonomy, one can be technically correct without having the necessary level of precision. For example, categorizing the Persian cat as a mammal is technically correct, but might be less precise than is operationally required. In orthogonal categories, one could be correct in one dimension but wrong in another, and these dimensions might not be weighted equivalently (Pinelis, 2019). For example, incorrectly labeling an ally as an enemy is worse on most missions than incorrectly labeling their sex. Fuzzy categories further complicate each of these problems. However, what precision or weighting is appropriate might not be stable across all missions: some factors may become more important as situations evolve. Most programs must grapple with these issues to have adequate testing, but currently it is left to individuals to recognize this challenge and develop their own solution to it. A common process could save resources and promote quality across program portfolios.

Furthermore, even using the appropriate techniques above will only correctly estimate accuracy if the test event had operationally representative event rates. We recommend that the entire acquisition community, including requirements writers, use Bayesian probability—not just traditional accuracy calculations—when evaluating system accuracy. In a standard confusion matrix, accuracy is calculated as the number of hits[43] plus the number of correct rejections[44] compared to the total number of observations (Green & Swets, 1966). However, this computed accuracy value will only be representative of the system's operational accuracy if the ratio of present to absent trials reflects the real world (Bayes & Price, 1763). Bayesian probabilities take this ratio into account, however, and should be used instead.[45] Tests do not have to be designed

[43] Actually Present : Represented Present

[44] Actually Absent : Represented Absent

[45] Take as an example a system whose mission is to move ahead of advancing warfighters and declare whether the route is free of mines. One might design a test with 100 test points where mines are actually present, and 100 when there are no mines at all. If the test showed the system correctly identified mines 99 times (99percent

around operationally realistic event ratios as long as analysts can use the base rates in their evaluations.[46] If base rates are unknown or volatile, then analysts can calculate the Bayesian probabilities across a range of possibilities.

c. Procedural Autonomy

We will only spend limited time discussing the evaluation of procedural autonomy. Though it is by no means necessarily easy to design, it is much less intellectually demanding to test. Evaluating procedural autonomy just requires evaluating how well goals are accomplished. Procedural autonomy selects the next action in pursuit of a goal. If provided an accurate and useful problem state by perceptual systems, a system with good procedural autonomy will have good outcomes *for that goal* and a system with bad procedural autonomy would have bad outcomes. Assessing procedural autonomy becomes more about ruling out the other autonomy types as the causes of a bad outcome, the process for which is described in more detail under Challenge #3. However, systems that solve complex problems through procedural autonomy alone will be time- and resource-intensive to test. Test efficiency is the challenge in those systems, not planning what data to collect.

1) Metrics in Requirements

The DoD acquisition system has struggled to write requirements that are both testable, verifiable hypotheses and translate to operational success (Ahner et al., 2018; Durst & Gray, 2014; Micskei et al., 2012). AI&A systems will pose even greater challenges. In line with Goodhart's Law (Goodhart, 1981),[47] history is replete with examples of AI-competition winners struggling or failing to transfer their victory to other domains or metrics (Kheradpisheh et al., 2016; Marcus, 2018). Current popular machine learning techniques are fundamentally about optimization, and when problems are well-defined, this optimization works well (Marcus, 2018; Soni, 2019). However, most of the applications of AI&A that DoD has indicated it wants to pursue are not well-structured,[48] and humans have historically been bad at selecting sets of metrics that actually define

sensitivity), and correctly rejected that there were no mines 95 times (95percent specificity), then a naïve analyst might declare that the system is (99+95/200) = 97percent accurate. However, in the real world, half of all existing routes are not mined, and these rates should not be weighted equally—they should be weighted by their probability of occurring. Every stakeholder would want to know what the probability is that a mine is actually there when the system says it is there. One can use Bayes' Theorem to calculate a system's accuracy under different assumptions about base rates: if one in ten routes are mined, the system's accuracy is only 69percent; if one in a hundred are, it drops to just 17percent.

[46] However, system designers should carefully consider base rates when choosing training data. Signals that are diagnostic in certain data mixes (e.g., 50percent mines instead of 1percent) may become useless in the real world, and designers should consider using operationally representative ratios.

[47] E.g., "When a measure becomes a target, it ceases to be a good measure." (Strathern, 1997)

[48] Well-structured problems have clearly defined initial states, goal states, and path constraints. When does an automated base defense system's task start? Do you know at the start what a battlefield's final desired state looks like exactly or only vaguely?

rather than just indicate success (Gray, 2015; Johnson, 1984). If a set of quantitative metrics are chosen, developers *will* be able to optimize performance to them, but there is a strong risk that the best performer at those metrics will not necessarily be the best performer of the mission. As a result, we recommend that quantitative metrics should be both redundant and blind, assessments of those metrics should happen robustly and under varied conditions; and holistic, mission-oriented requirements should be employed as well. Whatever the ultimate solution is, testers need methods to minimize the possibility that systems have "gamed" the evaluation.

DoD will need **testable, verifiable requirements**. Metrics defining these requirements should be **redundant**, **blind** to contractors, and tested under **varied** conditions. Requirements should also **include holistic, mission-level evaluations**.

Metrics should be redundant. Rather than selecting single indicators of performance, testers should break performance down into multiple aspects. For example, kill-chain analysis breaks the task of destroying a target into serial steps like detect, identify, track, locate, engage, and destroy. Then, for each aspect, testers should define multiple quantitative metrics that access that aspect in different ways. For example, one might want to select multiple categorization metrics (as described earlier) for the 'identify' stage of the kill-chain. This mitigates but does not eliminate the possibility that systems will be optimized to a bad representation of the mission.

Metrics should be blind. The tasks or missions, as well as the holistic mission criteria, should be very well defined, but to limit a competitor's ability to over-optimize, developers should not know the exact quantitative criteria used to judge performance.

Competitions should be robust and varied. Even if metrics are blind, it is possible that the best performer at a competition randomly selected the same set of criteria on which to train their system. Redundant metrics help with this, and testers could consider using a random subset of these metrics across source-selection competitions at multiple locations and environments. Furthermore, developers should not have access to test locations for training data. That is another way (in addition to metric optimization) where systems can perform well in test but fail to deliver good operational performance.

Requirements should also include holistic, mission-oriented evaluations. The call for this kind of requirement is not new, but the need for it in AI&A is stronger (e.g., Ahner et al., 2018). For traditional human-operated systems, it is not that this kind of holistic evaluation does not happen, but that it is not performed by the T&E community or contractually required. This evaluation traditionally happens in human training where there is no cultural phobia of holistic evaluation. It is the authors' opinion that since this holistic assurance is obtained outside of T&E, there has not been a forcing function to get holistic requirements in contracts. DoD will not have that luxury for AI&A, and though the problem is not any easier to solve, the defense community must find a way to write these requirements that mitigates risk to the military while being acceptable to contractors. The alternative is to break precedent in another way by getting very good at selecting quantitative metrics, which we believe is unlikely.

3. Learning as an Outcome

> Testers should find a way to measure a system's ability to learn and adapt to new situations as a performance indicator in and of itself.

One of the great unrealized promises of military AI&A is to have systems that can dynamically respond to novel tactics and technologies (e.g., US Department of Defense, 2011, 2019). While we are nowhere close to achieving the dream of real-time flexibility, systems will still need to adapt to the evolving nature of war over time. Adversaries will develop new strategies or assets, and systems cannot remain behaviorally static to these changes. Systems that can adjust[49] to these changes with less time, data, and resources than their competitors are desirable. We recommend that the speed at which systems can adapt to new situations become a metric in and of itself. In the future, this speed might be measured as real-time learning, but for the time being this will almost certainly be a measure of industrial rather than system agility.

We recommend that a formal part of AI&A evaluation be measuring how much time, money, and data it requires to acquire new adaptive behavior, as well as demonstrating that this new capability has not regressed original skills. Realistically, the DoD will have less data on emerging adversary capabilities compared to legacy systems. There might be strong operational need to react swiftly to adversary developments, so timelines may be compressed, and there may be many programs which must be updated, creating competition for budgets. However, a system can be made to *appear* to learn quickly by throwing more data, compute cycles, and money than is operationally realistic. One option for testing would be to give developers a specified amount of data with a limited timeline and budget and measure the system's capability at the new task by the end of it. If this process becomes standardized, it may even allow for benchmarking between systems. The capability under test could be a holdout tactic or behavior category that was planned to be developed at some point anyway (though this allows developers to prepare outside of the test, potentially skewing results), it could be something developed by a TTP red team (Zacharias, 2019b), or it could be a nonsense situation. The point of the test is simply to quantify, in an interpretable way, how well the system learns. Restricting the different resources (e.g., data, time, money) allows for a comparison beyond the resources a given developer was willing to apply to the problem at that time, and is more operationally representative of the types of challenges we will likely face in the future.

[49] Or be adjusted

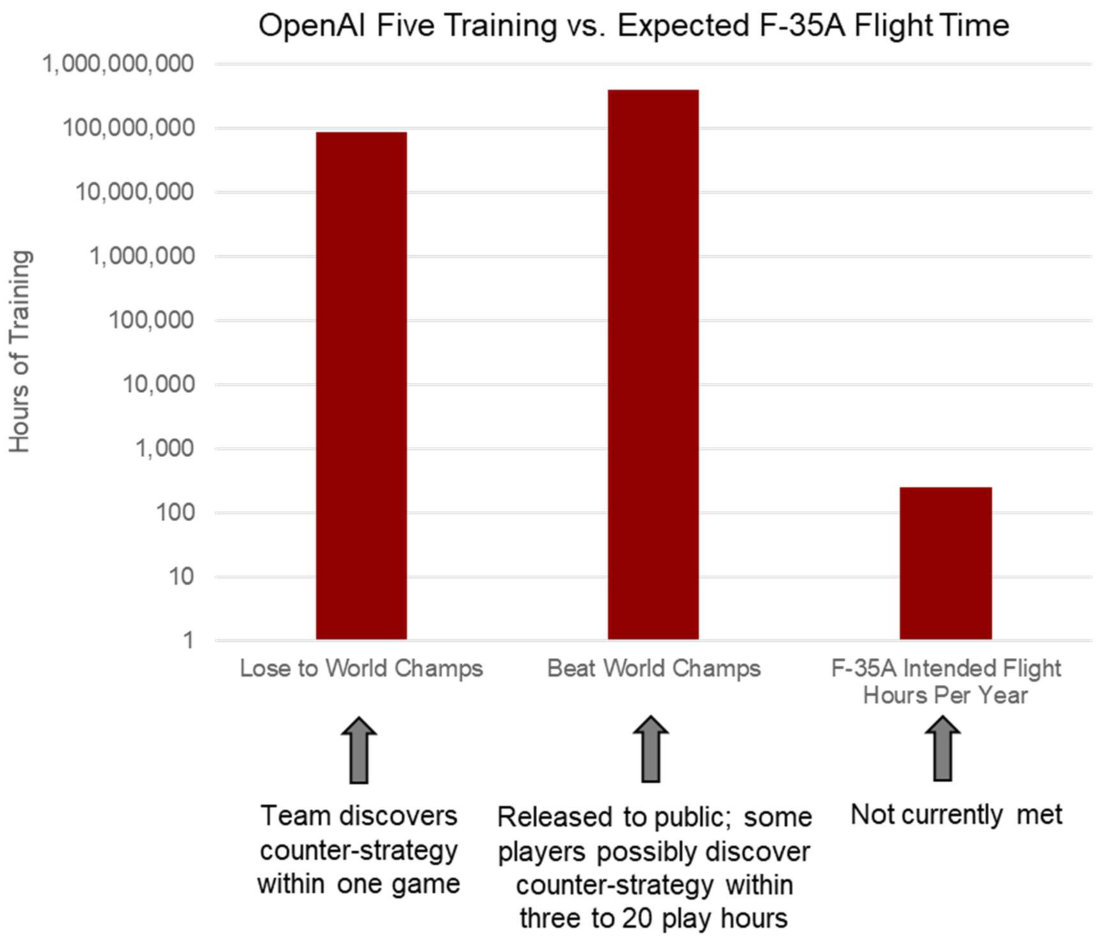
4. Continuous Learning Systems

We recommend that DoD *not* pursue systems that update or change their decision processes in real time in response to situations or data they encounter during operations—otherwise known as continuous learning systems—at this time. In short, the above challenges in measuring and testing AI&A will be amplified in continuous learning systems that demand continuous testing, and the results are unlikely to achieve the actual intent of the capability. While continuous learning sounds like a positive quality, this is a misnomer. What these systems do is continuously change, and that change could be for the better, for no effect, or for the worse. This demands continuous testing to confirm that the change is for the better as well as continuous regression testing to make sure performance on original requirements is not degraded. Finally, there is little guarantee that continuous learning will result in the desired capabilities. Current machine learning methods can be extremely inefficient compare to human learning. For example, to achieve a level of skill that would typically take 10,000 hours for a human, the OpenAI bots took 10,000 *years* of gameplay (OpenAI, 2018).[50] Though the OpenAI system was able to acquire those data much faster than real-time, real-time learning by definition does not acquire data faster than real-time (see Figure 4).

DoD should not pursue continuous learning systems at this time.

Continuous learning is often mentioned in response to a desire for tactical flexibility or to respond to new threats. We recommend, rather than using continuous learning systems, that new or novel experiences be recorded and saved for data validation. Such situations that actually do show a new tactic or threat can be used in periodic retraining or update training. This provides the adaptive capability desired in continuous learning, enables that learning to be shared throughout the fleet, and allows for appropriate human oversight and discrete test events to ensure the desired traits are learned and other capabilities are not degraded.

[50] OpenAI intentionally did not use the most efficient learning methods; the state-of-the-art would have offered improvement by a factor of two or three. Even this is eight to 10 orders of magnitude away from real-time relevance.



Team discovers counter-strategy within one game

Released to public; some players possibly discover counter-strategy within three to 20 play hours

Not currently met

Figure 4. The amount of operational data individual systems will collect is orders of magnitude less than needed to see meaningful tactical learning in sub-symbolic systems. Achievements in cooperative video games, while impressive, require massive amounts of training to improve or update while still remaining relatively tactically brittle.

However, testers do not get to decide what systems get developed, and if someone does insist that continuous learning systems will happen, then this capability does still need to be tested. In this case, we must try to at least mitigate the problem, and we would recommend adopting a similar tiered skill recertification as we have for our human operators. Programs would need to create different diagnostic tests that can be executed at different levels of expertise. A unit-level recertification would need to be relatively unsophisticated, and would check to make sure certain critical behaviors are operating within some set of defined parameters. These diagnostics could be run regularly by the units themselves. At a less frequent but regular interval, a battalion-level expert could be recertifying the decision systems with more detailed diagnostics. If history is any judge, this expert will probably be a field service representative (FSR). Finally, at rates driven by the program, all units can be realigned via formally tested capability upgrades.

D. Challenge #3: Diagnosing Decision Causes

RECOMMENDATION

Decision-making systems that have a built-in infrastructure for recording data (BIRD) become easier to certify. By having systems record data about themselves, by themselves, and by providing an infrastructural pipeline to securely collate, store, and disseminate these data, stakeholders can harvest data from a variety of previously inaccessible venues such as exercises and operational missions.

Even if the acquisition community makes every effort to design predictable systems, AI&A will almost inevitably produce unexpected behavior, and it is critical that developers and testers be able to diagnose the underlying causes (e.g., Haugh et al., 2018). This holds true regardless of whether the behavior met or missed its goal. As a rule[51] there is no decision strategy that will universally succeed or fail. Good, robust policies[52] are bound to come up short on occasion, and systems can occasionally find success despite bad solutions. In unexpected failures, testers must be able to tease apart whether this was an edge case for a good policy, or the predictable result of a bad one. If developers cannot identify why something went wrong, they cannot differentiate those possibilities. If failure is attributed to a bad policy, not having the underlying reason prevents identifying what needs to be fixed. One could throw more training data at the system, but this has little guarantee of solving the issue, or that it will not disrupt previously stable desirable behavior because data were not representative. Unexpected does not always imply undesirable though (Ferreira, Faezipour, & Corley, 2013): one of the dreams of AI is to have systems that find viable solutions that humans did not consider (US Department of Defense, 2011, 2019). Just because a human would not have made that decision does not mean it was a bad choice.

Testers need insight into systems' internal processing.

[51] There are always exceptions to rules.

[52] In the reinforcement learning sense.

Testers also need to be able to diagnose successful decisions to assess whether it was brilliance or randomly successful blunder (Ilachinski, 2017). Until XAI is realized however, testers will need to explain the decisions on their own, and to do so they will need data.

The barest minimum data needed to diagnose a decision are the inputs the system received and the behavioral output produced (e.g., Zacharias, 2019b). Testers need to know what the environmental conditions were so that they can attempt to replicate the behavior and/or to examine the relationships in other data statistically. However, the decision space from environmental factors to output behavior is enormous even for relatively modest AI&A (Clarke et al., 2012), and as discussed earlier, detecting those relationships quantitatively requires a massive amount of data.

Systems need to record data about themselves, by themselves.

Having testers record these data is not a viable option. At a practical level, having humans write down the environmental conditions and system responses at every test point would break our workforce's capacity (Trent, 2019). Indeed, having humans record the data is a bad idea even on a small scale. Human-level descriptions of the environment collapse or fuzz many input variables. Though systems might be deterministic with exact input matches, what humans would consider to be isomorphic situations are not what the system would consider to be equivalent. Behavior across the human-level description of the environment is better modeled as stochastic, making it even harder to replicate effects and assess relationships.

Furthermore, though these input-output pairs are the bare minimum, the desired level of detail includes information on intermediate processing (Haugh et al., 2018; Zacharias, 2019a). It would be better to localize where in the processing stream errors originate rather than just indicate that certain environmental conditions might lead to certain outcomes. If we have data from the intermediate processing steps, we can help isolate source versus propagation as discussed in Challenge #1 and Challenge #2. However, this intermediate data is fundamentally unobservable to human senses.

Systems need to be recording these data about themselves, by themselves (Haugh et al., 2018). Even if a system records no more than the inputs it received and the output it produced, having automatic data collection allows test and training data to be harvested from a variety of environments. Moreover, automated data collection opens up the ability to record internal system states, to implement what some are calling 'cognitive instrumentation' (Haugh et al., 2018).

For any cognitive instrumentation to be useful, the data to be recorded must be specified and intelligible, and specifying the processing flow when designing a hybrid cognitive architecture provides this scaffolding. In sub-symbolic systems, it is not clear what should be recorded beyond the initial inputs and ultimate output. A neural net is transforming information as it moves toward the final layer, but unless serious effort has gone into model induction, it is definitionally unclear in a black box what channels, layers, or nodes are important for the final output, or in what way.

However, when systems' processing streams are designed around intermediate symbolic production, the hooks for cognitive instrumentation are built right into the system. For example, supervised learning channels have clearly interpretable input and output, and unsupervised channels in an architecture at least have clear points to include hooks.[53]

In modular systems, testers can use the outcomes of individual processing stages to diagnose where problems seem to occur. If these stages produce intelligible output, this could even be assessed directly. Otherwise, as discussed in Challenge #1, testers can step backward through processing stages while systematically varying components of the recorded data to explore where in the decision process the error occurs.

The cognitive instrumentation should be scalable. In even relatively benign cases, there will be many variables fluctuating at high frequency over long periods that could be recorded. Early in testing, there will be limited understanding of which of these are most influential, and it is possible that testers will need to capture all of them. As our understanding grows more sophisticated however, we may be able to downscope the process, and eventually after fielding we may have a core set of channels that are recorded. However, as updates are pushed or mishaps are investigated, new or different variables may need to be recorded. This instrumentation should be designed from the beginning to enable this flexibility.

Cognitive instrumentation also serves as a scaffold for other system capabilities like live behavioral health monitoring, training modes, and explainability (Haugh et al., 2018; Trent, 2019). AI&A will need their own independent internal processes that can identify whether their primary processing is outside of safe bounds or is producing errors. Cognitive instrumentation does not just serve test purposes, but also would provide the data that could inform middleware meta-cognitive systems that assess whether the primary system is in a situation that exceeds its processing capability, or is under cyberattack. The instrumentation could also be reversed, injecting input instead of recording it, allowing these systems to safely but realistically participate in live, virtual, and constructive (LVC) training activities.[54] In addition, the data provided by the cognitive instrumentation for testers to identify the 'why' of a decision could also serve systems that would diagnose that 'why' themselves. This could serve as the basis for XAI of complex systems.

[53] More research and study is required to determine what and where to record in a system trained under a reinforcement learning paradigm.

[54] We acknowledge that this produces a potential adversarial attack vector. Whether the capabilities provided outweigh the potential risks will need to be assessed on a case-by-case basis.

Programs need **end-to-end** secure **data pipelines** to collect, record, collate, transmit, and store information generated by cognitive instrumentation.

The massive flows of information these systems can produce means we need built-in infrastructures for recording data (BIRDs). The first part of this infrastructure is the internal cognitive instrumentation to record system states and behaviors, but it must be much more than this. These data must be collected, recorded, collated, transmitted, and stored; and each of these processes must be done securely. Read access to these data could allow an adversary to at least partially reverse engineer our systems, and write access could allow them to poison training and/or evaluation (Casola & Ali, 2019). AI&A have the potential to generate enormous quantities of data, and every program will need end-to-end pipelines that can be maintained and scaled across the lifetime of the system. Non-AI software-intensive systems have already shown the acquisition community that leaving this as an ad-hoc process or waiting until system maturity to develop the pipeline courts chaos (see existence of Defense Innovation Board, 2018). The infrastructure for end-to-end data recording and management needs to be built into a program from the start.

A BIRD would support not just T&E, but would be critical for using post-fielding data to improve system functioning. Unless DoD wants each AI&A unit to have idiosyncratic capabilities and weaknesses, system learning updates should be batched. The expanded pool of training data must come from somewhere, and a BIRD would serve as that pipeline. Systems could use their cognitive instrumentation to record their fielded experiences, while the rest of the BIRD serves as the pipeline to get that data to developers and evaluators.

We recommend that both the BIRD's overarching structure and its individual components be made a fundamental requirement of virtually all AI&A systems. This instrumentation and pipeline is a critical enabler of diagnosing system behavior, the functions described above, and some of our proposals for testing certain kinds of systems. It is ultimately in developers' best interest to have a BIRD, but beyond that we argue it is in DoD's interest to ensure it is implemented as early as feasible in development, and that the easiest way to accomplish that is by including it in the formal requirements process.

E. Challenge #4: Safe, Secure Realism

RECOMMENDATION

Testers can use a strategy of Graded Autonomy with Limited Capability Fielding for difficult-to-certify systems. High-consequence, difficult to certify systems should be tested like we do with medical residents. Train all skills, and then certify and field their least risky capability for use under supervision. While acting in realistic situations, have systems record what they would have done with more risky capabilities, and use these data to decide to decrease supervision and/or increase task risk.

While realistic testing provides the most operationally applicable assurance, in many cases realism is not possible. Whether driven by safety concerns or a literal lack of realistic assets, there are many situations where test-isms trump realism. Many of the points that are most important to real operations are the ones we do not have the ability to test realistically. It is critical that we have methods to explore these scenarios for our AI&A systems.

1. Extremely Unsafe, Realism Required

Though virtually all military systems have operating modes that are unsafe to test realistically, we predict this will be much more frequent for AI&A. Furthermore, many of these systems may perform tasks that are unsuitable for heavy reliance on simulation. In particular, we are concerned about systems (a) that perform a variety of different tasks relying on different skills, (b) where failure has a high risk of real harm to humans, (c) where simulation would not provide enough assurance, and (d) which have not sufficiently demonstrated their capabilities in the real world to be trusted with humans. We will refer to these types of systems as Varied, Safety Critical, Autonomous Robotic Intelligences (Vari-SCARI) systems.

An AI&A system meant to fully replace a neurosurgeon would be a good example of a Vari-SCARI system. A neurosurgeon robot would have to diagnose its patients, develop a treatment or surgery plan, and then execute that plan. Failure at any of these steps poses grave risk to the patient. However, though there are large similarities across human brains, there is a good deal of idiosyncrasy in the structure, vasculature, and functional topology of individuals. Furthermore, pathologies can manifest in very different ways between patients. These plus the high-fidelity physics needed[55] combine to make a sufficient simulation of brain surgery extremely difficult to VV&A as a full substitute for live testing. How then could we ever certify a Vari-SCARI system for safe fielding? Some might answer that we should not even bother pursuing such systems, but this and other Vari-SCARI systems are often some of the most desirable use cases for complex autonomy. For example, having a robotic surgeon that can have its skills quickly replicated across hospitals could improve quality and consistency of care across the system and reduce the costs associated with continually training and certifying new individual doctors. We do not advocate for abandoning these systems at conception. Instead, we point out that this is a very similar challenge we face with the human residents who will one day become neurosurgeons.

We recommend that testers certify Vari-SCARI systems by using a graded autonomy paradigm as we do when training medical residents (Halpern & Detsky, 2014). In the human analog for these high risk, complex tasks, people are initially certified, but not immediately released into the wild with full independence. Typically, they begin to perform lower risk tasks by themselves, but under the watch of an experienced supervisor who is ready to intervene. They are not allowed to perform the higher risk tasks, but their supervisors may ask them what their decisions or actions *would* be in the current live situation.[56] As these novices demonstrate competence, they are authorized to act more autonomously. They are allowed to perform the lower risk tasks with less supervision, and start to perform the higher risk tasks under supervision. In this way, their decision-making and skills are tested under realistic scenarios while mitigating the risks of novices operating. Though graded autonomy in humans involves real-time learning, and this could be the case for AI&A as well, it is not a necessary feature in order to execute graded autonomy. Systems could have their decision-making software frozen and merely evaluate the responses that the static system chooses but does not execute. However, testers should be aware of the potential for certified decision processes to need

> Systems that need to be tested in real-world scenarios can be certified the same way we certify medical residents: after an initial test, field it under close supervision on lower risk tasks, and over time, progressively decrease supervision and increase task-risk.

[55] Millimeters matter in neurosurgery. Minor failures of collision physics or material interactions in a simulation could be the difference between life and death.

[56] Tesla is implementing something partially akin to this with their “shadow testing” (Templeton, 2019).

regression testing if decision processes which impact them are changed. We recommend that graded autonomy be adopted more or less wholesale for Vari-SCARI systems.

In order to expose military systems to these realistic scenarios, it may be necessary to combine graded autonomy with limited capability fielding (GALCF). Instead of trying to certify the entire system suite of capabilities and tasks at once, we recommend that testers begin with the operationally useful capability that carries the lowest risk, and formally test the system for this capability. The system would then be approved for fielding for that capability only. This can get at least some capability into the hands of the warfighter without needing to wait decades. For example, an autonomous multi-role fighter might be tested and certified for reconnaissance only. As it is performing these approved capabilities in operations or training exercises, it is simultaneously evaluating what it *would* have done with its higher risk capabilities. For example, if the next step of graded autonomy were electronic attack against materiel targets,[57] it would use its cognitive instrumentation to record what decisions it would have made with each capability in the situations it encounters in the field. Those data can then be evaluated much as the human resident's hypothetical answers are—to see if it is competent enough at the next risk level to be trusted under supervision. In the multi-role aircraft example, evaluators could examine the field data to initially approve the system to electronically attack targets, but require that a human must first approve each engagement for this first certification. This then spirals upward through the process of graded autonomy. However, this is different (though not exclusively) from incremental capability fielding. GALCF assumes that the next capability already exists in production-representative form, but simply has not been approved for use. This typically is not the case with incremental capability fielding.

[57] As permitted by DoD Directive 3000.09

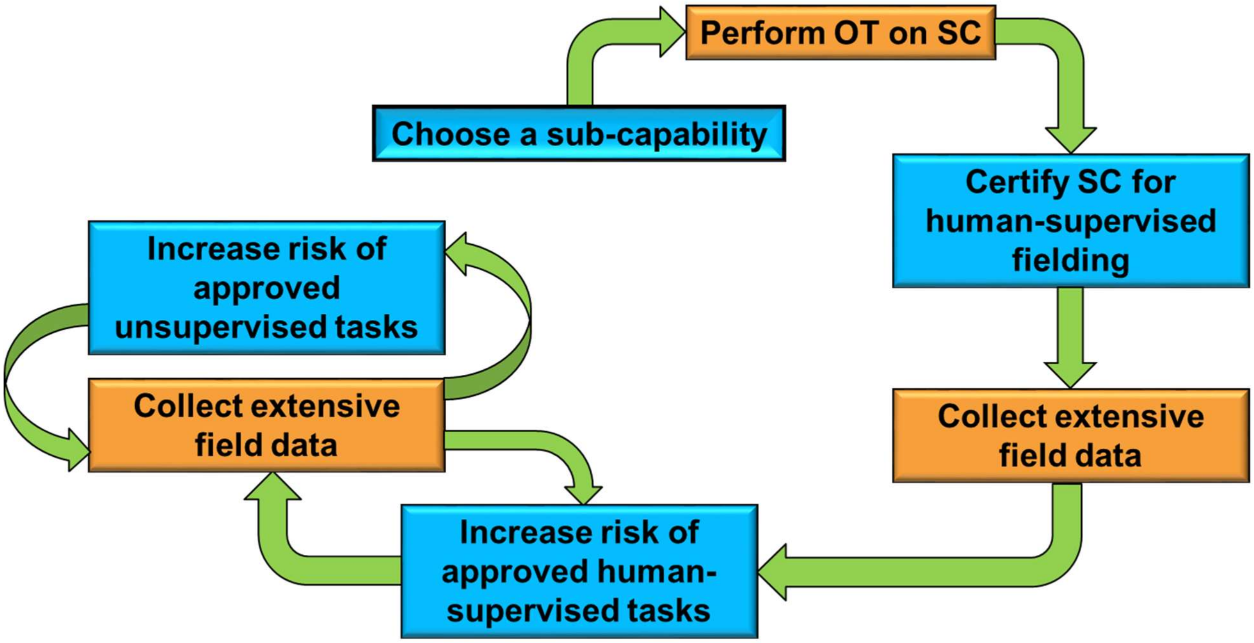

Perform OT on SC
Choose a sub-capability
Certify SC for human-supervised fielding
Increase risk of approved unsupervised tasks
Collect extensive field data
Collect extensive field data
Increase risk of approved human-supervised tasks

Figure 5. After initial operational testing (OT) on a useful, but less risk sub-capability (SC), systems can be fielded and allowed to make "shadow" decisions—record what they would have done with other capabilities in the current real, operational situation without being allowed to execute it. These data can be used to evaluate the adequacy of these riskier capabilities.

Finally, readers should note that GALCF is critically enabled by a BIRD, and BIRDs are in turn enabled by having a system architecture that has undergone OVVA, which in turn is facilitated by modular and at least hybrid sub/symbolic system design. In order to use field data, the system must—by itself—record its decisions and the conditions under which it made those decisions. There is no other feasible solution. If the system is monolithic, the hooks for cognitive instrumentation are much harder to discern. Without each of these, GALCF becomes much harder, and Vari-SCARI systems become much harder to certify.

a. Maintaining Safety During Test

Many systems will not perform the fundamentally unsafe tasks associated with Vari-SCARI systems, but may have uncertainty with respect to their safety. We recommend that developers and testers follow the example of the United States Air Force (USAF) and DARPA in this area. The USAF is developing middleware to safely test some of their autonomous systems with what they are calling Testing Autonomy in a Complex Environment (TACE; Thuloweit, 2019). The goal of any middleware like TACE will be to provide assurance in the form of a safety net during T&E or operation (Neema, 2019). If, for example, a UAV

Safety middleware systems can mitigate the risks of unverified autonomous capabilities.

enters conditions forbidden by the safety middleware,[58] the UAV control will revert from the UAV's less tested control software to the safety middleware. A challenge in testing with safety middleware is that early generation systems may not have the middleware available. Instead, the system could be tested with a biological middleware in the form of an operator on- or in-the-loop using a supervised remote capability. However, this will only work for systems and tasks that are actually amenable to meaningful human oversight (Arnold & Scheutz, 2018), as we discuss more in Challenge #5. When systems are designed in more modular ways, as described in Challenge #1, these safety overlays are much easier to implement.

2. Lack of Assets

In the process of testing AI&A, the T&E community will inevitably run into the problem of a lack of realistic assets to test the system against. This problem is not wholly unique to AI&A. Traditional weapons systems are regularly tested against stand-ins rather than true threats. For example, our 4th- and 5th-generation fighters can expect to fly against enemy stealth fighters if we fight a near-peer competitor. However, we are not going to have a squadron of Chinese J-30s and the pilots to fly them for use as red air during tests. Instead, we use the best facsimile available by employing F-22 or F-35 squadrons to pretend to be these enemy aircraft. However, where a human pilot can pretend easily, monolithic sub-symbolic AI&A cannot. Our understanding of how a concept like "F-35" is represented in the pattern of activation of nodes will be fuzzy at best after model induction, and it will be functionally impossible to reliably inject that activation into the middle of processing. If an autonomous multi-role fighter went up against an F-35 squadron during test and took no offensive action, or ignored a Smokey SAM[59] fired at it, it is hard to determine if these were malfunctions or operationally problematic. It may have identified correctly that F-35s are friendly assets or that the smoky SAM was not a real threat, for example. What we need are surrogates in test that are "good enough," but it is typically unclear what "good enough" is for an AI&A system, and even if we know, we might not possess those assets.

Hybrid or symbolic systems can be taught to "pretend" so that the lack of realistic assets is less of a problem for testing.

If we are going to use stand-in assets during test, we have to teach robots how to pretend. When a system is based on a hybrid architecture with intermediate symbolic output, pretending becomes much easier—one can change the label of what is passed up from the perceptual processes. An F-35 can be relabeled as a J-30 and passed to executive and procedural processes, a shipping container can be a T-72, or a mannequin can be a real live human being. Pretending—

[58] For example, using geo-fencing to keep a system in an area, or secondary control laws governing speed.

[59] A test asset meant to mimic the smoke plume of a surface-to-air missile (SAM).

whether done by artificial or biological agents—is not a perfect solution to realism, but it is much better than none at all. Plus the capability to pretend does not just benefit testing, it allows the system to participate in training activities believably.

F. Challenge #5: AI&A Emergence

RECOMMENDATION

Testers need environments where different autonomous agents, including humans, can be tested together for emergent behavior (EB). Centralizing test responsibility for EB can overcome a number of simulation challenges, while having a regular joint exercise would provide such a live test venue for validation while also helping troop readiness for existing and emerging technology employment.

A critical challenge in AI&A is that of interoperability writ large. Modern American doctrine emphasizes joint employment across services, domains, and systems to accomplish our military and geopolitical objectives (Joint Chiefs of Staff, 2018). This warfighting philosophy revolves centrally around the idea of emergence, synergy, or gestalt, of the whole being more than the sum of its parts.[60] Testing must therefore show that a new part contributes to and does not disrupt that whole. The need to show that systems can interact effectively on the battlefield is hardly unique to AI&A—our standard systems do not exist in a vacuum either—but the challenges of interoperability will be exacerbated in AI&A (e.g., Ahner et al., 2018). Autonomous systems respond to the environment, and when other systems can change that environment, behaviors can emerge from their interaction that would not have been expected from one system alone. As merely a part of a larger multi-domain battlefield, AI&A will have to work with other systems and agents (autonomous or not, and of artificial or biological origin) to successfully achieve the

[60] While often used broadly or imprecisely, emergence is formally defined as what happens when the interaction of parts produces effects that the individual components do not have on their own; emergence can be described as strong or weak.

commander's intent. Ensuring common formats and processes for communication will not be enough to ensure interoperability for AI&A—systems can and will mutually alter their behaviors (Defense Science Board, 2016). The system's effectiveness at contributing to a mission depends on its ability to work with those other systems, and those other systems' ability to work with it. Because AI&A will be behaviorally inflexible for the foreseeable future, system-to-system, human-to-system, and system-to-human interoperability will have to be explicitly tested (Defense Science Board, 2016).

Because there are competing formal definitions of emergence (for a history of emergence see O'Connor & Wong, 2012), and some conversations about emergent behavior define it imprecisely at best, with little connection to any formal definition, it is necessary to be clear about what we mean by emergent behavior. When used in this paper, emergence is when the interaction of parts produces effects the individual components do not have on their own (Ferreira et al., 2013; O'Connor & Wong, 2012). Emergent should not be used as a synonym for unexpected or undesirable[61]—emergent properties and behaviors can be any combination of expected/unexpected and desirable/undesirable (Ferreira et al., 2013). Finally, emergent behavior should be assessed by the extent to which it has an impact, rather than an absolute binary yes or no to its existence. The main goals of emergence testing in DoD should be to confirm that expected, desirable emergent properties or behaviors are functioning and to mitigate the probability that unexpected, undesirable emergence will occur.[62] In some cases there may be expected, undesirable properties addressed through CONOPS, and testers should assess the effectiveness of any procedures aimed at minimizing operational impact.

> Behaviors that would not have been expected from the individual elements alone will emerge when entities interact. These behaviors can be expected or unexpected, and desirable or undesirable.

1. Intra-system Emergence

Autonomous systems often have emergent properties even internally. Behaviors arise not from a simple summary output of subsystems, but come about from interactions across the system of systems (SOS; e.g., Zacharias, 2019a). It becomes very difficult to predict how adding or removing pieces of that SOS will alter behavior. A new capability might be modularized to

[61] Emergent behavior is also imprecisely used to describe unexpected effects that in retrospect were fully predictable from its base elements alone. It would not technically be emergence if a self-driving car that does not distinguish between valid and invalid driving surfaces took a "shortcut" through a cornfield. Failing to consider a part of the operational space during design or testing does not constitute emergence.

[62] Though unexpected but desirable emergent behaviors are possible (Baker et al., 2019; OpenAI, 2018), they can be rare and difficult to achieve. Development strategies predicated on using these to accomplish a task will likely encounter difficulties, and it is not an efficient use of test resources to try to hunt these down.

directly interact with only one other component of the network, but might indirectly affect everything downstream from that connection. Recursive processes further complicate the issue. Emergence can be delicate, and even small perturbations may eliminate a behavior we desire. Because we predict these interactive properties will be critical to the functioning of many systems,[63] we argue that testers will need to assess the extent to which behavior relies on intra-system emergence for a given system.

RECOMMENDATION

Testers should characterize system flexibility as well as system performance. Decision systems can achieve greater performance on a specific task by over-optimizing, which can create downstream costs and consequences when trying to upgrade, change, learn, or transfer to a related task. Testing should evaluate to what extent programs have made this tradeoff.

Testers should quantify the robustness of a system's decision-making. Systems will encounter situations where their subsystems are degraded for any number of reasons such as cyber-attack, battle damage, file corruption, or normal reliability problems. Decision-making resilience in the face of these conditions is important, and testing should inform stakeholders to what extent this is true. For example, it is a common practice for neural network developers to examine performance degradation by knocking out nodes or injecting fuzzed data as inputs (Meyes, Lu, Puiseau, & Meisen, 2019; Zhou & Sun, 2019). Testers should ensure that test plans include these practices and extend them to the entire architecture. Part of system simulation should be performance examination when entire modules are degraded (node knockout and/or input fuzzing) or removed.

Testers should also evaluate AI&A plasticity. There is a basic tradeoff in artificial and biological neural networks between flexibility and optimization (e.g., Huttenlocher, 1979;

[63] There is good evidence that human intelligence is not a complex module to itself, but is an emergent property of many different simpler modules (Friston, 2011). The capability of some of these modules is unmatched in other species (Smith et al., 2016), but even those unique levels of performance do not explain human intelligence alone.

Sternberg, 1996), sometimes referred to as plasticity versus crystallization. For example, biological sensory processing is exceptionally well tuned to the specific sensory organs of that individual organism. When those sensory networks are optimized (e.g., in adults), they can be retrained over time with small, incremental changes, but may never adapt to large sudden changes (Linkenhoker & Knudsen, 2002). This is acceptable for biological organisms because barring catastrophic problems, any changes to wiring in sensory organs will usually be gradual—they will not wake up one day with a totally new set of eyes.[64] We know this will not be true for AI&A, however; sensor upgrades or changes happen all the time, and what is acceptable crystallization for organisms may be over-tuned in AI&A. Tuning the entire network or architecture to a specific sensor will likely increase performance, but this may come at a cost if it is ever swapped out. Since this swapping or capability module additions are certain to occur during the system's lifecycle, testing should quantify the extent to which the AI&A's network is crystal or plastic. During testing, sensors could be swapped out to simulate a sensor upgrade, and performance changes tested. Furthermore, retraining time should also be evaluated. For example, the time, data quantity, and processing power required to retune the network back to its original performance levels could be quantified.[65] This evaluation matters because in a source down-selection, one system may demonstrate superior performance to the others, but it may have achieved this performance at a cost to plasticity. The better performer might not even be meaningfully different than the others—two systems may just exist at different points on the same tradeoff curve between performance and plasticity. Superior systems have better tradeoff curves, not just superior performance. Quantifying plasticity would provide more information for the source selection and help make informed decisions about whether performance increases are worth later upgrade costs and difficulties. Furthermore, having plasticity as an explicit part of evaluation incentivizes contractors to emphasize it in their systems.

Learning systems can **overfit** themselves to specific data, environments, or hardware, **trading global for local performance.** Testers should assess the extent to which systems have made this tradeoff.

Just like in all software, AI&A will need regression testing (Deputy Secretary of Defense, 2012). We cannot assume that new capability modules will not disrupt existing functions, especially if system functioning relies on emergent properties of the network. However, systems will vary in the fragility or robustness of these emergent properties. This is an area where we see

[64] Though glasses can improve vision, they do not change retinotopy.

[65] We cannot assume there will be as much data on which to train future capabilities as was available initially, and pre-existing data may not be appropriate for training the new module.

no clear path forward to accomplish this regression testing process robustly, cheaply, and quickly.[66] As is already the case in software-intensive systems (Director Operational Test & Evaluation, 2010), the amount of regression testing required for AI&A should be based on a risk assessment. However, incorporated into this risk assessment should be the fragility or robustness of the system's emergent properties.

2. System-System Interaction

Emergence is a concern not just within systems, but also between them (e.g., Ahner et al., 2018; Defense Science Board, 2016; Zacharias, 2019a), and though our doctrine is joint, our testing is not. Realistically, standard systems will be interoperating on the battlefield with a host of other systems, but for good reasons, formal test events rarely employ them all together. Requiring a full-scale joint exercise to test every non-AI&A system separately is prohibitively expensive, but also currently unnecessary. Systems are usually designed to serve a specific function on the battlefield, with humans dealing with the less tractable battlespace integration issues. Systems are evaluated on their specific functions, while humans are trained to execute our joint doctrine, and formal evaluation of each occurs separately. This T&E formulation breaks down when humans are removed from the loop.

Emergent behavior is possible when AI&A systems interact.

Humans have historically served as the lubricant to mitigate poor interoperability in our system-of-systems, but as we move from operator-in-the-loop to having them merely on-the-loop or out of it entirely, this lubricant will disappear (Endsley, 2015). The flexibility that humans provide will be replaced by the rigidity of early AI, and our testing must adapt to assess the effect of this loss. Testing will need to reveal to what extent AI&A is capable of integrating its behavior with other battlefield systems.

The potential for emergent behavior is of particular concern when AI&A take over more of the battlespace. The essential goal of DoD's joint doctrine is emergence: using synergistic interactions and coordination to act as force multipliers—the whole being greater than the sum of its parts. However, critical pieces of this emergent capability are abilities which humans have but AI&A do not (yet). When there are relatively few autonomous systems, they are less likely to interact directly, and the rest of the human-controlled systems may be able to flex to fit their behavior. In essence, humans add plasticity to the warfighting collective. However, as AI&A becomes more common, the systems may begin interacting not with humans, but other autonomous systems. A major concern here is of course the standard interoperability issues (formats, channels, etc.), but now without the facilitating effect of human cognition to smooth the rough edges.

[66] Even a "pick two" situation may be difficult to achieve.

Standard interoperability testing will not disappear—if anything it must receive even greater emphasis.

AI&A introduce the concern that these systems will alter each other's behavior. This does not require or imply that these systems communicate or are aware of each other—as one example, it could be as simple as the goals and procedures selected by System A creating problem states that were never explored when System B was trained, leading to unusual behavior in System B. If other inflexible systems rely on System B, emergent behavior (or failures) may cascade. Inflexibility is not the only concern though; flexible AI&A interactions will also produce unpredictable behaviors, possibly even more unusual.[67] In either case though, AI&A-to-AI&A interactions have the potential to disrupt emergent capabilities from joint warfighting, and the ability of these systems to interact with each other cannot be assumed and must be tested. If systems must be tested together, then the obvious questions become what, where, when, and how to test.

a. Efficient Coverage

The entire space of all system combinations across all conditions is intractable (Clarke et al., 2012). We need methods for efficiently covering as much of the space as we can, while delving into important conditions in more detail. At a general level, we recommend that testers use a strategy of lower-fidelity methods like agent-based modeling to cover space, and higher-fidelity methods like LVC simulations and operational tests or exercises to validate the adequacy of the lower-fidelity methods or to diagnose problematic system interactions. However, before this happens, it is necessary to identify the subset of systems that must be tested together (e.g., all of the AI&A systems that would participate in a Close Air Support mission). Optimal test designs should account for the probability that interactions will occur, how critical they are to the mission, and their flexibility (e.g., if they are mediated by a human or a brittle AI). We recommend that testers identify the relevant simplex or complex of systems to test through sequential use of multiple technical methods.

> There are too many possible combinations to test; the set of test points needs to be intelligently narrowed down.

We predict it will not be sufficient to test the pairwise combinations of systems independently—the entire system-of-systems that would interact on an operational mission should be tested together. Emergence can cascade—emergent behavior as a result of one pair of systems interacting can then go on to affect other systems' behavior too. Emergence may also be the result of higher-order interaction. Along with operational realism, these factors are why we predict a need to test many systems in their broader operational context.

[67] This may not always be bad—again, one of the desires for AI is to discover solutions previously unconsidered.

Testers can use network analysis to help determine system combinations for emergence testing.

A first technical method step could be to use statistical network analysis (Chiesi, 2015). The warfighting collective could be described as a sparse network of nodes, with each node representing a system, and labeled connections representing different kinds of interactions. Network analysis provides a method for assessing network structure and the influence nodes have on each other. If these connections are labeled with their probability of occurring, their criticality to operations, and the brittleness[68] of their interactions, then testers can use network analysis to prioritize system interactions. To give a simplistic example, the analysis might show that a second-degree connection is unimportant because it has a low probability to occur and is mediated by a human (so more flexible), while an entire fifth-degree connection chain may be very important because of the combination of criticality and inflexibility. Note that this analysis does rely on the network being accurately described. One could initialize the connection labels with SME knowledge and later update them as testers collect empirical data, and this network would be a cross-program test asset. However, compared to other formal statistical techniques, network analysis is relatively young. DoD has already successfully used network analysis for some operational tasks, but there remain many open statistical questions and predictive validation experiments that researchers need to address. We recommend that network analysis become an area of importance for method development efforts.

Once the set of relevant systems is identified, testers can begin to test scenarios using lower resolution simulation—but what is a sufficient level of resolution is currently unknown and likely to change over time. The goal of this resolution reduction is to get more runs for a given amount of computing power, trying to achieve faster-than-real time runs while still having credible results.[69] Moore's Law predicts computing power will grow, but simultaneously, a growing portfolio of AI&A systems will put increasing demand on those resources. Testers will need to strike a balance between the competing demands of simulation accuracy and resourcing.

LoFi simulation can cover large numbers of test points to find areas of greater interest.

To achieve this lower resolution, testers can sacrifice either the fidelity of the system and its decision processes, of the level of detail in the environment, or of both. Currently there is only speculation on which of these will work better for covering large amounts of the state-space when testing emergent behavior between systems. If the

[68] In the simplest case, whether it is human-human, human-machine, machine-machine, though there are many ways to improve the usefulness and accuracy of these labels.

[69] At the time of writing, a reasonably high-priority acquisition program, which requested that we not attribute them, informed us that high-fidelity simulations currently run slower than real time given their available computing power.

OVVA process is followed as recommended, it may be easiest to maintain the system's actual tactical software while abstracting the environment, as what is a sufficient environmental representation for the system will have already undergone some level of VV&A. The T&E community should continue to monitor ongoing AI&A acquisition programs to start building a better understanding of the tradeoffs made between these choices.

Testers can use the network analysis to pare down the system combinations that will be primarily examined. Next, testers can visit a larger part of the operational space of those combinations through lower resolution simulation to look for interesting cases, following up on those with higher fidelity simulation. Finally, testers can use the simulation results to choose the most important test points to collect during more expensive live testing (e.g., Robbins & Steffen, 2018). While this is one process to identify what should be tested, we still must describe the where and how.

b. Validating, Investigating, & Diagnosing in Higher Resolution

All of our emergence simulations need to undergo VV&A at some point, which will require live data from somewhere. This first means that the modules for the individual system's behavior need to be validated. If good data governance practices are maintained, it might be possible to reuse historical data for this part of the VV&A process. However, because we are trying to test behavior when systems are together, we also need data on the systems' behavior when together in a live setting. It is unlikely that programs will already have this data from earlier testing. Testers need a venue where these data can be harvested, but getting these assets together can be challenging, especially when they belong to different services.

HiFi simulations can investigate areas identified by LoFi methods.

To compromise between the need for live joint testing and the practical constraints, we recommend that the DoD create a regularly scheduled joint exercise where AI&A past certain milestones of maturity—including those already fielded—are invited to participate. There will not be a unique joint event for every system, which helps bring down costs, while still giving a venue for these systems to interact. As a regularly occurring event, it also affords the opportunity to continue testing emergent behavior in what will eventually be our legacy AI&A. New systems may not themselves behave oddly in the collective, but introducing them may create emergent behaviors in our old ones. The T&E community will need to keep up with this possibility across the system lifespan.

A joint live exercise would help VV&A the simulations.

Additionally, while we are proposing this joint exercise for the T&E community, it would be a combined venture for the COCOMs and P&R as well. We must evaluate how the systems interact with each other, but in order for the human element of our warfighting collective to flex to the new behaviors of AI&A, our warfighters must train with those systems. This venue would enable this too.

However, this test event would be much more observational than is traditional in DoD testing. Test points would not be planned in advance—there are too many systems participating for that to be realistic. Interactions would occur organically, and the T&E goal would be to validate, post-hoc, whether the simulations produce the same behavior from the observed interactions as the behavior that actually occurred.

The viability of this observational joint test/training event depends critically on all of our AI&A systems having a BIRD. It is completely impractical to have humans attempt to record all of the AI&A system interactions that occur as well as the conditions when they happen. Systems need to be recording those data about themselves, by themselves, and there needs to be a pipeline established to pull data from the individual units, collate it, and provide it to developers, program offices, analysts, and modelers. If the systems do have a BIRD, any exercise or fielded activity where AI&A interact can become potential evaluation data, not just the planned joint exercise.

The downside of finding unexpected behaviors in uncontrolled live exercises is that it becomes nearly impossible diagnose the exact causes of individual system behaviors—there are just too many degrees of freedom. However, the primary use we are suggesting for this event is to continue to validate the emergence simulations and look for aberrant behaviors that can be diagnosed by more detailed testing. More controlled joint system testing can occur as needed. Testers can use the network analysis, simulation results, and SME judgment to inform which system combinations and conditions would provide the most valuable test points for live testing.

c. Ensuring Extensibility

As new autonomous programs are added to the SOS that will interact on a mission, they can create emergent behavior in our legacy systems, necessitating that the whole set be retested together. If each environment or set of modules is a unique solution to that particular combination, we would need to redevelop and re-VV&A the set to incorporate this novel system. While this may not be an immediate problem on the two- to five-year horizon, this approach makes the number of simulations developed accelerate over time.

The number of AI&A systems, and therefore the **combinations** needed to test for emergence**, will grow over time**. Testers should make some effort to **ensure** the **extensibility** of emergence M&S to future systems.

Developers and testers will need standards to ensure that simulations are extensible and connectible without needing to rebuild the full set each time. However, it is an open question whether existing standards like the Test and Training Enabling Architecture (TENA) are sufficient for the challenge of emergence. At this juncture, AI&A involve many unknown unknowns. A possible path to developing these new standards (or reaffirming existing ones) is to let programs develop their own simulations for the time being, while in parallel convening a working group or other organization to work with the programs to find cross-cutting lessons-learned that can inform the standards that are needed.

These standards would need to also provide integration for human-controlled avatars. For the foreseeable future, humans will remain a part of the warfighting collective, and the simulation environment needs to include unpredictable human actions. This is especially true for systems intended to have a human supervisor. If we had a verified, validated, and accredited simulation of human decision making—a system we believed could make the same decisions a human would—we would have already solved the AI challenge. Until then, actual humans must provide that input to the simulation. Though they might not participate in every simulation, the environment would need this capability. Fortunately, this is another capability that serves multiple roles and is not simply for test: an environment with these hooks can also be used to support training and readiness for the services if planned correctly.

d. Responsibility for Emergence Testing

Emergence testing inherently involves multi-domain and inter-service interactions, raising two related critical questions: who is responsible for emergent behaviors and who is responsible for testing them? If, for example, an Air Force system causes an unexpected, undesirable emergent behavior in an Army system, each service is going to want the other to be the one to have to fix it. Further, do each of those programs execute their own emergence testing independently, or is there a centralized, joint organization responsible for testing emergence?

A centralized authority for emergence testing could reduce redundancy and help ensure extensibility.

The DoD should evaluate whether the current process for establishing interoperability requirements for deterministic, relatively static traditional systems is applicable to stochastic, behaviorally dynamic, and rapidly evolving AI&A technology. Current practice is to put the onus of interoperability on new systems being backwards compatible with a specified list of legacy systems. This is hard enough with traditional software development, but we predict it will become functionally impossible to develop systems that both learn their own desired behavior and cause no undesired emergence in legacy systems. In some cases, it will be easier to fix the legacy system than alter the new one, especially if the triggering behavior in the new system is critical to its own operation. DoD should assess whether more forward-looking interoperability requirements are necessary and feasible, and at the very least, establish a mechanism or policy for determining who is responsible for fixing emergent behavior.

1) Challenges of individual program responsibility

If DoD decides that individual programs are each responsible for conducting their own emergence testing, then solutions are needed to a number of problems. The authors are concerned about problems related to accelerating growth of resourcing requirements and redundancy, getting access to assets for all programs that need them, achieving sufficient competency and capacity for emergence testing in our workforce, exacerbating the extensibility and integration challenges above, and negotiating conflicting results.

If each program is responsible for testing emergence on their own, and especially if they need to develop their own simulations, the resources needed across the board to test emergence will become unmanageable. The test requirements for emergence are inherently reciprocal between systems. Emergence needs to be tested wherever System A and System B overlap. That overlap is the same regardless of whether A or B is executing the testing. If systems are responsible for their own testing, this results in what are essentially completely redundant tests. Furthermore, what is needed in the simulation to get adequately representative behavior from a system for a mission domain (e.g., Close Air Support, Mine Countermeasures) should be the same regardless of who is doing the testing. If systems also need to develop their own simulations, this balloons the effort required to VV&A all of the variations of the all the different simulations. If they are not doing this, and are using common simulations, and also need to test the same points, one must ask why emergence testing is left to the individual programs in the first place. This resourcing competition and redundancy is an even worse problem for asset management in live testing than for the simulations.

It is also unclear whether our workforce has the capacity to support high-quality AI M&S for all of the programs that need it. Both the AI and M&S skillsets are in high demand and low supply already, and the overlap between them will be even rarer. If the T&E community dilutes this capability across the entire family of AI&A programs, it raises concerns about whether any program will be able to execute emergence testing well. Secondarily, AI-enabled programs are likely to be Special Access Programs (SAP), and anyone involved in emergence testing would need clearances for all of the programs involved. Though this is a lower hurdle for an individual-program approach compared to other problems, it is further weight on the scales against it.

2) Benefits of Centralizing Test Responsibility

A centralized, joint organization responsible for testing emergent behavior along the lines of the Joint Interoperability Test Center (JITC) could mitigate many of these problems. A central organization limits the amount of overlapping effort, mitigating the problem of rampant growth and redundancy. Such an organization could create environments for common mission domains and more easily adopt standards to ensure compatibility and extensibility. Centralizing responsibility also makes it easier to concentrate our AI and M&S talent, and gather a group of people with the necessary clearances. It also provides advantages when it comes to coordinating systems that would participate in the JLE and using those data to validate simulations.

Creating a centralized focus for test responsibility does not require nor does it imply that it is the only place where emergence testing will occur. Programs will likely need to conduct more detailed combined testing, especially for high-fidelity LVC or live events with actual operators. However, it is worth at least considering a solution that could reduce redundancy by getting a 90 percent solution for 90 percent[70] of cases.

3) Getting to centralization

Centralized emergence testing would be a possible goal, but it will not be a day one capability. Though the problems with emergence testing *probably* are not on our five-year horizon, we predict the solutions to these problems will require significant lead time. There is little underlying theory or understanding on what makes emergent behavior more or less likely, and there are tools along the lines of the Range Adversarial Planning Tool (RAPT) or the Autonomous System Test Capability (ASTC) that need to be extended to help identify valuable test points during our low-resolution space exploration (Haugh et al., 2018). In the meantime, there are basic and applied research questions that working groups or organizations such as the JAIC can begin to address. Most immediate is the need for standards. We recommend that this be done through collaborations with current programs that attempt to co-develop the standards, rather than through a top-down imposition.

[70] Or whatever percentages are achievable and valuable

3. Human-System Interaction

RECOMMENDATION

Testers still need to emphasize human-system interaction for autonomous systems. Even in fully autonomous systems, a human will be involved in some part of their decision-making chain. To ensure responsible employment, these interactions must be fluid and minimize error, and warfighters must have appropriately calibrated trust of the system. The properties must be tested.

Despite the promise of independence in AI&A, humans will still interact with these systems, and the effectiveness of these interactions will partially determine system effectiveness. Evaluating this human-system interaction (HSI) also requires evaluating emergent properties, but in this case the methods are relatively well developed—if inconsistently applied—in DoD. These techniques provide a solid foundation for examining HSI in AI&A.

We recommend that the DoD prioritize HSI when writing requirements for AI&A (Defense Science Board, 2016). Historically, the emphasis of HSI has followed a cyclic pattern counter-phase to major safety disasters. When HSI quality is a priority, systems are easy to use, and consequently safer in demanding environments (Federal Aviation Administration, 2000), but it is an axiom in the field that usability is like oxygen: you do not notice it until it is gone. Creating systems with good HSI is not easy, but people take it for granted. As safety accidents become rarer, organizations become complacent and begin to scale back their budgets and workforce for HSI. This decreases the quality of HSI in new technologies, increases the risk of accidents, and can ultimately culminate in a string of disasters that renews the demand for HSI experts. A recent string of major disasters[71] rooted in poor HSI is renewing this demand now (Josephs, 2019; National Transportation Safety Board, 2019a). While it is not clear that this cycle can be realistically avoided, starting AI&A development with an emphasis on HSI may mitigate problems. The public has displayed strong distrust of AI&A, especially for military applications (Defense Science Board, 2016; US Department of Defense, 2019), and public response to

[71] E.g., the Pacific fleet destroyer crashes, the Boeing Max 8 accidents, and the Hawaii missile warning system.

accidents involving AI&A has been much stronger than reactions to human-equivalent incidents (Bernstein, 2018). The potential human cost of AI&A disasters should not be understated, but problems early in the history of AI&A could also set the field back by years due to public outcry. Emphasizing HSI from the start is one way to mitigate that potential.

Testers should identify the type of human control that exists: whether operators are in, on, initiate, or out-of the task loop, or whether the system is on the loop instead.

HSI will be relevant to all systems, but the level of human control and type of interaction will vary within and between systems. Tasks where the human chooses and performs actions with the aid of AI&A still have an operator-*in*-the-loop (OITL). Other systems may possess all types of autonomy and be capable of fully independent action on a task, yet their CONOPS calls for and the system is designed to have a human to monitor its behavior and potentially intervene—to have an operator-*on*-the-loop instead (OOTL).[72] In other cases, a human may order the system to perform a task but thereafter pay no attention to it—that system has an operator-*initiated*-loop (OIL).[73] Humans may have no interaction with the task at all—an operator *out-of*-the-loop (OOFTL) task. Finally, there will be (and already are) human-machine interactions where the relationship is system-on-the-loop (SOTL): the system monitors the task and takes over when certain conditions are met.[74]

As was the case when defining autonomy, testers should establish the type of operator-loop relationship at the task level. There might even be multiple types within a single system: one task might be manually controlled (OITL), another merely monitored (OOTL), and yet another performed fully independently by the system (OOFTL). Tasks are hierarchically organized, and the type of interaction may depend on the frame of reference. Testers should define systems' autonomous, operationally relevant tasks for that mission and assess the operator-loop relationship for those tasks.

Testers should measure, at minimum, usability, workload, and trust.

At a minimum, the HSI concepts of usability, workload, and trust will be relevant to most AI&A (e.g., Endsley, 2015). At its core, usability is the facility with which an operator can employ a system to accomplish an intent. This requires both that the system has the necessary capabilities (Utility) and that getting the system to execute the desired capability requires little effort (Ease of Use; Venkatesh & Bala, 2008). Workload is the personal resources (bio-mechanical, cognitive,

[72] A semi-autonomous system in DoD Directive 3000.09 terms.

[73] E.g., Fire-and-forget or OOTL systems with inattentive supervisors.

[74] E.g., Systems like the Automatic Ground Collision Avoidance System (Auto-GCAS)

temporal, etc.) demanded by a task relative to those available to an individual (Kahneman, 1973; Wickens, 2008). Workload and performance have a non-linear relationship, where very low and very high workloads result in poor performance (for different reasons), with a relatively stable plateau between steep drop-offs (Hancock & Warm, 1989). Finally, trust is a person's belief that something can be depended on in vulnerable situations (Wojton et al., 2020). Trust will have proximate causes (e.g., quality of system performance and the operator's understanding of the system) leading to distal effects (e.g., relying on the system in real situations). Testers should incorporate *at least* these three HSI concepts into their AI&A test plans.

However, the type of operator-loop relationship affects the details of HSI testing. The tools for measuring most HSI concepts will not likely change substantially in AI&A and are well explained in a variety of manuals (e.g., Gawron, 2008), and so we do not describe them here. However, depending on whether the task of concern has an OITL, OOTL, OIL, or SOTL relationship, how and where these tools are applied may change. In the follow sections, we describe these differences. Testers should see these differences as guidelines or topics for focus, rather than exclusive rules.

a. Usability

For OITL and OOTL systems, a key usability concern is how well the AI&A conveys information to the operator (Endsley, 2015; Endsley & Kaber, 1999). This conveyance cuts in two directions: providing key information while omitting extraneous bits. When the system performs perceptual tasks or acts as a decision-aid (has executive but no procedural autonomy), system effectiveness will be governed by how well the human can utilize the processing done by the AI&A. For OOTL systems, knowing whether to intervene requires knowledge of both system and environmental states. Testers should include both *what information is given* and *the manner in which it is provided* as facets of usability testing for these systems. OOTL systems should also undergo standard HSI testing for when tasks are reverted to human control.

For OITL systems with only procedural autonomy, the usability concern is how well the procedures selected by the system match user intent. For example, an operator may input a waypoint destination with the intent that the system take a certain route or action, but the automatic pathing does something different. This will lead to frustration and potentially ineffectiveness. Testing this kind of interaction will be most similar to the usability testing that already occurs with standard systems, and it will likely be obvious to operators when interfaces are poorly implemented.

For OIL systems, the initial goal-providing interaction is critical, and testers should examine the extent to which this process is fluid and error-free. We recommend that data should be harvested from, at minimum, three points: (1) the operator's perception of the Utility and Ease of Use of the order-giving process, (2) the probability that the system initiates a different task than intended, and (3) the operator's ability to understand what task the system is pursuing.

Although humans are by definition not involved once SOTL AI&A takes over, this takeover can affect HSI for normal operations and should be examined. Calibrating these systems to intervene at the right moments without being too frequent is designing for HSI (Endsley, 2019), and testers should examine how well these thresholds are aligned with operational and safety needs.

b. Workload

The primary workload concern for OITL systems is whether workload is too high for the human. Once workload exceeds capacity, sub-tasks are shed to bring it back under control (e.g., Schulte & Donath, 2011). If those are important sub-tasks, performance will decline. One of the purposes of bringing autonomy to sub-tasks is to free up operator workload for more critical functions; however, though this is the intent, some have noted that incorporating autonomy can instead just shift workload from task performance to managing the AI&A (Endsley, 2015). In these systems, designers meant for the task to go from OITL to OIL or OOFTL, but instead it is just OOTL. Testing whether AI&A actually reduces operator workload will likely require A/B testing with and without the system to provide defensible evidence.

Meaningful human control relies on **appropriate workload** levels. If workload is too high, operators may shed oversight tasks, and if workload is too low, inattention, boredom, or complacency may prevent it instead. **Testers must assess workload under realistic conditions.**

While reducing workload during standard operations is still of interest and should be tested, it may be more important to preserve overall task performance during high-stress, task shedding situations. It may be the case that workload is not reduced during standard operations—the human is still monitoring the system when able—but that in high workload environments, the monitoring task is shed. Because the system autonomously performs its sub-task though, overall task performance may be preserved.[75] Testers should ensure that this performance preservation can be tested.

In OOTL systems, the operator tasking is explicitly about management, but the areas of concern become both overly high and overly low workloads. In these systems, the usual purpose of the overseer is to provide meaningful human control over the autonomous process (Deputy Secretary of Defense, 2012). At high workloads, operators may either shed some of the monitoring tasks or loop through them at an insufficient refresh rate; though there is technically a human

[75] In a non-AI&A situation, shedding the task means the task is not performed. AI&A performs that task even if it is not being monitored.

present, there is not meaningful human control over those systems. To explore the operator's true monitoring limits, testers should ensure that test conditions are not limited to relatively low-intensity scenarios.[76]

For OOTL systems, testers must also consider low workload problems. When workload is low, people become bored, disengaged, and/or complacent (Hancock & Warm, 1989). This in turn hurts performance when intervention is actually necessary, as they are not ready to take appropriate action (Endsley, 2015; Endsley & Kaber, 1999; National Transportation Safety Board, 2019b). Humans are notoriously bad at long-term vigilance tasks, especially when events are rare (Jerison & Pickett, 1964). In these cases again, though there is technically a human present, there is not meaningful human control. When a human monitor is part of the CONOPS from the start, testers should plan for an operational test that occurs over a long period with the goal of understanding operator complacency. At a minimum, we recommend that units who will be tested with these systems should have a long burn-in period before formal testing[77] so that the test is actually operationally realistic. In most cases, complacency does not develop overnight, and a short operational test with new operators is unlikely to reveal that behavior.

When human oversight is proposed as a post-hoc mitigation to system deficiencies, we recommend that the acquisition treat these suggestions with extreme skepticism. Human oversight *should not* be assumed to serve its desired purpose—the rarity and time-sensitivity of interventions are likely to dramatically affect success, and we recommend that the T&E community make it a policy that programs which need meaningful human control demonstrate that their CONOPS actually provides it.

c. Trust

For all AI&A, the goal of design and education should not be to foster trust, but to *appropriately calibrate* user trust (e.g., Defense Science Board, 2016; Endsley, 2015; Wojton et al., 2020). Both over- and under-trusting a system can lead to regrettable outcomes (Culley & Madhavan, 2013). Too much trust can endanger users who lean too heavily on the system or employ it in conditions where it performs poorly. Conversely, too little trust may lead users to abandon the system, defeating the point of acquiring it. Test events must be structured to evaluate the level of trust operators have in their systems (Director Operational Test & Evaluation, 2019), how much they *should* trust the systems, and to what extent operators are likely to rely on the systems once fielded.

Trust should be measured conditionally, not holistically. A common response to the question, "Do you trust the system?" is "It depends." Whether someone trusts a system depends on the nature of the task, consequences, and conditions (Lee & See, 2004). When trust is appropriately calibrated, the level of trust across combinations of these factors matches the system's capabilities

[76] This limit is often the case in current operational testing.

[77] BIRDs can allow data to be harvested from burn-in periods so that it is not wasted.

under those conditions. Trust is situational, and testers' evaluation of trust should be as well. We recommend trust be measured at the task and condition level—not at the system level—so that analysts can assess the situational calibration of operator trust.

The critical outcome of trust is reliance (Lee & See, 2004). Ultimately, we are concerned not about trust itself, but the behaviors trust is likely to produce. We want to know whether the operator would use the system when failure could lead to meaningful consequences; trust is an important predictor of those behaviors. We recommend that test plans be structured both to measure trust as well as assess what real operator reliance in the field would be.

> Too much or little trust can both be problematic. Ultimately, testers need to provide results that help operators **appropriately calibrate** their **trust** of the system so that they can make informed choices about when and where to rely on the system.

Using test events to evaluate reliance in the field may mean that system use must be optional in that test event. Current standard practice is to require, implicitly or explicitly, that the system under test be employed for the task given to the test players. While this allows planned test conditions to be evaluated, it also prevents assessment of whether those operators would have actually used the equipment in that situation. When that is the case, evaluators cannot examine whether trust and reliance are appropriately calibrated. This creates a tension between the need for structured test points and free-flowing organic system interactions. We recommend that test plans describe a mix of the typical quasi-experimental designs currently employed, as well as more observational designs. Readers should note that these observational, organic events are not just for trust, but can also serve to help evaluate other emergent effects, and with a BIRD, those observational events may fill in parts of the formal test matrix.

Some might argue that because warfighters are required to use certain equipment in specified ways, measuring trust is irrelevant because reliance is mandatory. Testers should be skeptical of this argument. The reality is that many pieces of fielded equipment are left behind, turned off, or ignored despite CONOPS that require otherwise.[78] Even if a system is physically integrated, cannot be shut off, and performs a task that the operator fundamentally cannot perform manually in parallel, trust will likely still be relevant to measure.

[78] Communicated with an expectation of non-attribution.

4. Agent-Agent Interaction: True Teaming

RECOMMENDATION

Testers should adapt existing methods for evaluating human teams for the T&E of human-machine teams. Though not all AI-human system relationships will truly involve teaming, systems that do will require a different approach to testing. The starting point for these evaluations should be the methods already created by the behavioral sciences and sports statisticians.

Although emergent behaviors can surface accidently, some systems will be designed to produce them. The real challenge of testing emergence will come from true teaming: systems which are intended to alter their decisions based on the actions of other agents pursuing a common goal. These agents might be biological, as in human-machine teaming (HMT), or they might be other AI&A systems. Though nature of the other teammates will change the details of test strategies, each option shares a set of common challenges for evaluating system effectiveness. In particular, teaming systems make it difficult to assign credit or blame for outcomes to individual agents versus emergent properties of the team.

Most AI&A systems are better described as tools than teammates.

However, many systems which currently claim to involve HMT might be better described as tools than as teammates. Teammates pursue common higher level goals—autonomous tools are assigned tasks in service of higher goals, but themselves are only pursuing the lower level goal. As with our definitions of autonomy, the goal of our definition of teammates is to identify the features of the category of systems that require different test methods.

Systems that are working toward **shared outcomes** with another agent and **alter their decisions** based on the actions of the other agent will add additional challenges beyond those of regular AI&A.

For the purposes of testing, we recommend true teammates be identified by three properties: each agent must (1) be able to influence each other's problem state; (2) be working toward a common higher-level goal, and (3) coordinate actions or decisions. Agents that influence each other's problem states but do not share goals only risk the type of accidental emergence described earlier in this section and require those test strategies. Two agents might work toward the same goal (e.g., win the war), but if they do not influence the other's problem state, there is no real potential for emergent behavior.[79] Finally, even with common goals and the ability to affect each other's problem states, if agents do not alter their behavior based on teammates' behaviors—i.e., if they do not coordinate—there is also no need for special test methods. Though coordination colloquially implies explicit shared planning, and though this is beneficial for effective teaming, it is not required to trigger the challenges of teaming. An agent representing the other agent's actions as part of its problem state is sufficient, even without conscious awareness of the other agent.[80]

When scoping what needs to be tested in teaming, testers should look at the unit appropriate to the functional level of a task. The most basic type of teaming will be between a dyad of two agents, but higher level goals might occur between teams of teams. For example, individuals on a team might coordinate with each other to clear a room, and the individual members would be the correct level of analysis for that teaming task, whereas inter-squad coordination might be the appropriate level of teaming to examine for clearing a building or securing a site.

Our definition of teaming does not preclude teammate specialization—in fact the division of labor is often essential to effective teaming (e.g., Murciano & Millán, 1996)—but teammates still have common goals. For example, a football guard has a specialized task, but is ultimately pursuing the same high-level goal as the other teammates. If circumstances demanded, they would alter their decisions or take over someone else's task to pursue that higher-level goal (e.g., picking up a fumbled ball and running, rather than blocking). Similarly, co-pilots might employ some division of labor while still pursuing the same higher-level goals and constraints.

However, designing a system to take over a person's division of labor tasks does not automatically make it a teammate—it still must be pursuing a common higher goal. Take as an

[79] And thus no need for special test methods

[80] For example, OpenAI Five demonstrated emergent coordination even though they were not explicitly trained to do so (OpenAI, 2018). Because the systems represented the other agents and their actions, and because they were trained extensively, they were able to learn coordination.

example a notional AI&A aircraft meant to act as a wingman for a human pilot. A human wingman might maintain formation with the lead, engage targets the lead specifies, and alert their partner if the lead is under threat. In performing those specialized tasks, the human wingman is working to accomplish the same larger goal as their teammate, and changes in circumstance could alter that tasking. However, one could design an AI&A aircraft that maintains formation, engages specifies targets, and alerts about threats, but does so without reference to the pursuit of higher level goals: that system would not be a true teammate. It would be no easy job to develop that system, but testing it would not involve anything that has not been described in the rest of this paper. In fact, there is little task-level difference between this notional system and a tethered glider that provides additional hardpoints for the pilot to use. The AI&A controlling flight is a much fancier, more flexible stochastic tether, but all that needs to be tested is how well the tether maintains the desired formation.[81] Having the human select targets for the system to engage is just one extra step beyond selecting targets with a fire-and-forget missile.[82] The threat alert is similar to (or may just be) a missile warning system. In this example, though the system has autonomy within its assigned tasks, the human pilot is *using* the system to pursue their mission level goals, not actually teaming with it. System effectiveness at those might alter human behavior, and this possibility should be tested, but this change would happen even with a completely 'dumb' system.[83] Other than maintaining formation, the system is not changing its decisions based on its partner's choices—it is just performing its tasks, and that is where testing can focus. If the system were a true teammate, the scope of testing would have to grow from just its effectiveness at its own assigned tasks to its effectiveness at all mission tasks and its ability to decide which tasks to pursue, and to do so across the context of its teammates' decisions.

As the level of structure in the division of labor decreases, the difficulty of T&E increases. So far, we have described teammates with relatively well understood tasking. Unexpected emergent decision-making is a real possibility, but structuring the team is meant to make that the exception rather than the rule. This lets the focus of testing be on the assigned tasks while examining some of the edge-cases. When task assignments cannot be planned in advance, true teammates must effectively divide sub-tasks on the fly. This makes executive autonomy harder to assess. When task assignment is unstructured and/or occurs dynamically, the same outcome metric may apply to multiple agents. When outcome metrics are shared, statistical and interpretation issues arise.

Flexible retasking in response to teammate behaviors adds additional development and testing challenges.

[81] This is procedural autonomy.

[82] Most of which also have partial autonomy under our definitions and 3000.09.

[83] E.g., if a hammer had an oddly shaped handle, the human would use it differently than a normal hammer. That does not make the weird hammer more than a tool.

Teammates will compound the difficulty of evaluating whether executive decisions are correct. Even with a single agent, it will be difficult to analyze how goal choices influence effectiveness. The link between an early decision and an outcome is already indirect. Teammates introduce further degrees of freedom to these executive decisions, and consequently more difficulty when evaluating them. As just one example, testers might observe that an AI&A tool does not pursue a necessary task. This could be because the system made a bad decision; alternatively an AI&A teammate might not have pursued the task because they assumed their partner would perform that action. That is not necessarily a bad decision. Differentiating these options will require designed-in explainability and/or an incredible amount of post-hoc analysis.

It will be difficult but necessary to disentangle the extent to which individual team members contributed to the outcome of a task; techniques developed by sports statisticians and behavioral science can help. One would make very different recommendations if an HMT succeeded in spite of the system than if it only failed because of the human, for example. Low metric achievement by one team member cannot be directly interpreted as failure on that individual's part. A point guard in basketball may score few points directly, or a wingman may achieve few kills, but each may be the critical enabler of their other teammates. Alternatively, one teammate may be dead weight and carried to victory only by the excellence of their partner, or be such a drag that their mere presence leads to failure. A team's success or failure at the overall task cannot be interpreted as success or failure of individual members at teaming.

Sports statistics and behavioral sciences can provide a starting point for disentangling individual contributions to group outcomes.

The added variable of trust further compounds this problem. Trust is the glue and grease of an effective team: it holds them together and allows them to work together without friction. Not only is it especially critical to assess whether trust is appropriately calibrated in an HMT, as compared to standard autonomous systems, but the need to factor in trust increases the difficulty of evaluating system decisions. Teammates change their behavior depending on the behavior of the team; the decisions the human makes will be affected by their level of trust in the system. The human's decisions will in turn affect the decisions the system makes. This feedback-loop will make it even harder to evaluate executive decision effectiveness.

The challenge of interoperability is writ large when it comes to HMT: testing must demonstrate that systems can partner effectively not just with different systems but also different types of people. It is difficult enough getting deterministic systems to interact effectively even within a single service, let alone designing for joint interoperability. Getting effective human-human interactions across these populations is no easy feat either. When it comes to HMT, this challenge grows exponentially. Systems must not only deal with tradition-entrenched communication differences and cultures between services, they must navigate the idiosyncrasies of each member of the population of potential teammates. Complex cooperative behavior is relatively rare, and humanity's ability to do this is one of our key evolutionary advantages over other social organisms (Mafessoni & Lachmann, 2019). We have not yet cracked the design challenges to accomplish this in an artificial system. Yet even if we had a system that could successfully partner with different individuals, it would be another challenge to actually demonstrate that capability. Only a subset of individual differences or traits will influence the decisions the human makes within a task, but it is unlikely that we will know the mapping of these in advance. We will have to learn which traits are task relevant before we can even begin tackling the challenge of how to test whether the system can successfully navigate those traits.

We need to test whether systems are able to effectively team with partners varying across the range of physical, social, and emotional human traits in the operator population.

Teaming is strongly enhanced by operating off of a shared understanding of the problem state (Mathieu, Heffner, Goodwin, Salas, & Cannon-Bowers, 2000), and testing should examine to what extent agents have a common operating picture. If agents have different understandings of the current situation, they will make poorly coordinated decisions. The greater the problem state discrepancy, the greater the problem. Testers should examine the extent to which problem state representations match between agents at given time-points. When systems are designed such that the problem state is accessible (e.g., in a hybrid cognitive architecture), testers can retrieve it directly. To obtain it from humans, testers can consider using working memory probe techniques in some cases (e.g., Awh, Jonides, & Reuter-Lorenz, 1998; Endsley, 2000). However, testers should note that humans do not actually have access to all of their own problem states (Nisbett & Wilson, 1977). The decision criteria of which humans are consciously aware are not always aligned with what was actually used by the decision process (e.g., Festinger & Carlsmith, 1959). In these cases, psychologists treat the human as a black box and have developed a variety of experimentation techniques to make inferences about mental states.[84] Once these problem states are obtained from each agent, analysts can compare the states for differences. We recommend that state comparison be a weighted estimate, rather than a simple average. Some information dimensions are more critical to decision making that others, and analysis should account for this.

Understanding a teammate's current representation of the problem state can improve teaming effectiveness (e.g., Cooke, Gorman, Myers, & Duran, 2013; Falk & Johnson, 1977), and systems

[84] These techniques may not be universally feasible across all stages of contractor, developmental, and operational testing.

with this capability should have it tested. For example, if a teammate believes their partner is unaware of important information, they may attempt to communicate it. If they believe the partner already knows about it, communication is wasteful and should be avoided. Knowing other teammates' states also enables better predictions about their future decisions, allowing one's own decisions to be better coordinated with theirs. In this case, testers should compare the agent's representation of their partner's state with the partner's actual state. Eliciting these state descriptions can occur using similar working memory probes as above. Though not all systems will have this capability, humans will, and so part of all HMT testing should be the usability of the components that convey the system's current state to the human operator.

Creating and sharing common problem states for each teammate usually requires effective communication across both explicit and implicit channels (e.g., Easton & Martinoli, 2002; Espinosa, Lerch, & Kraut, 2002), and testers should evaluate this. Like trust, communication should be appropriately calibrated. However, what defines 'appropriate' changes as teammates gain experience with each other. In human teams, the pattern of how the amount of communication and its relationship to effectiveness changes over time has been studied (e.g., Butchibabu, Sparano-Huiban, Sonenberg, & Shah, 2016). For human-machine teams, the trajectory of this communication-effectiveness relationship with experience may be worth quantifying as well, especially if a down-selection between systems is planned. Systems that are able to adapt to their human partners (and vice-versa) faster are superior to slower ones, and this adaptation speed can help differentiate otherwise equivalent systems.

The world of sports statistics has developed some methods for evaluating individual teammate contributions to overall gestalt effectiveness (e.g., Duch, Waitzman, & Amaral, 2010), and these techniques may provide a starting point for evaluating AI&A teammates. However, these statistics are often reliant on well-defined intermediate outcomes, which while prevalent in sports, are currently ill-defined in military contexts.

For systems which involve true HMT, we recommend that testers employ a variant of matched-pairs designs. Here we provide a very brief overview of this concept. This topic is complex enough to require its own discussion and will be explored in detail as one of the future papers in this series. The design's goal is to evaluate how well this system is able to team with different members of its user population, who will vary along some number of individual traits, including but not limited to demographics, personality, and physical characteristics. This means that these traits must be an explicit part of the experimental design, and effort should be taken to prevent aliasing of these factor levels both by the trait level itself and the individual sampled to represent

Testers may need to consider human partner as an explicit factor for test design and replicate test points with different types of partners.

that trait level. First, testers should identify the traits that are likely to affect teaming.[85] This will require a mix of SME assessment and experimentation—for example, system designers might tell testers that how assertive the system thinks their partner is will affect its behavior, but testing might reveal that their height also affects the teaming task. Unlike with typical testing, where operators are captured through convenience sampling, testers will need to make an effort to procure partners who vary along the identified trait dimensions. This allows inference of how well this system will team across its user population. However, to make comparisons fair and interpretable, these different operator types cannot be confounded with the other operating conditions. Current test practice is to make little to no effort to control which operators are used in which test points with little to no replication across operators. This will not be acceptable for HMT AI&A. Operator traits, at least, must be treated as an explicit factor in test design. Most likely it will require testing different human-machine combinations (in pairs or larger groups as operationally relevant) on as close to the same scenario conditions as is feasible. Finally, more sophisticated analysis techniques will be required than those DoD typically employs. These techniques may have their roots in statistics from psychology (e.g., dyadic data analysis; Kenny, Kashy, & Cook, 2006), game theory (e.g., Shapley values; Shapley, 1953), or sports statistics, but it is likely that they will require further research to adapt to the specific needs of testing military AI&A teaming.

Fortunately for those tasked with developing test methods, but unfortunately for DoD, we predict effective true teaming is a far horizon for AI&A. The cognitive abilities[86] supporting cooperation and coordination are arguably one of humanity's greatest evolutionary advantages over other species (Decety, Norman, Berntson, & Cacioppo, 2012), yet even between humans teaming often fails. Bad teammates are a universal human experience, and good teaming relies on predicting teammates' behavior. This requires either sufficient representations of others' states and decision models to be able to predict what they will do, or truly massive exploration of the shared problem space so each member knows what the others will do. In most cases, the latter is not possible for HMT (especially military HMT), and DoD is not currently pursuing systems capable of the former.

If DoD wants to pursue true teaming, AI&A that have explicit state representations[87] and models—and that are also designed to be similar to human decision models—will make this easier (T. Miller et al., 2017). If one's decision model is similar to another's, simply knowing how the other represents the problem state allows one to use one's own model to simulate one's own decision in that situation, and therefore predict others' behavior (e.g., Mafessoni & Lachmann, 2019). The more similar models are, the easier and more accurate the predictions (Preston & de Waal, 2002). If systems are designed around these explicit representations (as we advocate with

[85] In the context of our broader framework, personality types and other traits are decision-relevant information dimensions that should be varied in test.

[86] For example, cognitive and affective empathy

[87] I.e., the problem state (the system's belief about the status of the current situation) is something that can be examined explicitly, for example a vector of quantified variables.

hybrid cognitive architectures), it is a much shorter step from making decisions for oneself to predicting those made by others. If they are designed to be similar to human models, these behavior predictions become more accurate for both the system predicting human teammates and human teammates predicting the system (e.g., Endsley, 2015; Zacharias, 2019a). These explicit representations not only make high-quality teaming more likely, they also make the systems easier to test by providing state representations on which the system's theory of mind can be tested.

G. Challenge #6: Exploitability

RECOMMENDATION

Testers should assess adversarial exploitation generational cycles. Evaluating the constantly evolving possibilities of cyber and tactical exploitation may require a cultural shift away from static, well-defined exploitation requirements. Testers should attempt to quantify how quickly adversaries can develop exploitations of our decision systems versus the speed at which we can re-counter them.

Evaluators must keep in mind the reasons why an estimate of effectiveness obtained in test might be wrong in real operations, and this final section focuses on an external cause: the enemy gets a vote. Whether in the form of cyberattack, physical damage, or behavioral exploitation, adversary actions will be able to diminish the performance of our autonomous systems (e.g., Zacharias, 2019b). When AI&A are discussed in policy, variations of the words 'resilient' and 'robust' almost always appear, but are rarely defined. In this section, we try to discuss some things testers might measure to evaluate AI&A robustness.

1. Cybersecurity

AI&A will be the pinnacle of software-intensive systems, and bring with them cyber vulnerabilities both old and new (Sawers, 2019). In this section, we only address a small portion of these issues. Many of the issues in AI&A will be the same we encounter for any software-intensive system (Marshall, Rojas, Stokes, & Brinkman, 2018), and so we do not discuss those here. We also do not speculate about how to test cybersecurity for systems that do not yet exist. Doing so would require us to speculate first on what problems will exist—which will depend on how AI&A are instantiated both in their software and hardware (currently unknown). We would then have to speculate on how these unknown problems would be solved in order to speculate on how these unknown solutions should be tested. Instead, we focus on one novel component of AI&A cybersecurity that is likely to be an issue for most current systems: adversarial machine learning (AML).[88]

Adversarial Machine Learning (AML) represents a novel vulnerability for AI&A systems.

In ordinary ML, one trains the system by finding the changes to the intervening model that increase the probability that inputs are transformed into desired outputs: the inputs are held constant, and the model changes. In AML, the idea is somewhat reversed: one uses an existing model to find changes to inputs that maximally perturb the output (Karpathy, 2015). For example, in computer vision, one can find which pixels can be changed to alter how an image is categorized (Goodfellow et al., 2014; Goodfellow, Shlens, & Szegedy, 2015). In cybersecurity, one can use AML to train both offensive, protective, and identifying models.

A highly active area of AML in cybersecurity uses Generative Adversarial Networks (GANs; Goodfellow et al., 2014). GANs use deep learning neural networks as both the generative and discriminator models. At a base level, the generator will create an input that is passed to the discriminator alongside a stream of training data. The discriminator then classifies the inputs from the training data and the generator based on its algorithm. The generator then learns how its initial input was classified, adjusts its algorithm as necessary to optimize some goal, then re-engages the classification loop with the discriminator. This process can be iterated until the discriminator satisfies some goal, like classifying a panda as a gibbon (Goodfellow et al., 2014) or a 3D printed turtle as a gun (Athalye, Engstrom, Ilyas, & Kwok, 2018). These efforts can then be used to alter the original architecture in the discriminator.

Though some advocate for continuing a philosophy of day-zero assumed breach for AI&A cybersecurity, this philosophy may need to evolve. Under the old thinking, we must assume that our adversaries have access to our model and will devote sufficient processing power to finding an exploitative input that will have a worst-case scenario effect, and then they will deploy that input. Followed to its logical conclusion, assumed breach implies we should not field fully autonomous systems when things go kinetic, as the adversary will have already discovered how to defeat them. Though we do advocate that planners should consider what would happen if we lost all of our autonomous capability, we argue that developers and testers are better off trying to

[88] We do not imply this is the only new cyber challenge for current AI&A. Problems like data poisoning are possible in ML, and we concur with recommendations others have made such as doing VV&A of training data (Haugh et al., 2018). Unresolved test efficiency challenges like scalability remain, however.

minimize and mitigate day-zero probabilities through design choices reflective of adversarial training.

Similarly, we do not believe that it will be a worthwhile use of resources in mature systems to describe and then patch the AML attacks that are possible against the system's current version. This would be a never-ending battle. Models that are well-structured enough to produce coherent behavior will essentially always have *some* pattern of input that will lead to undesirable output or just simple disruption (Goodfellow et al., 2015). One should assume that if the adversary has the model and is willing to spend the resources, they will be able to use AML to achieve their desired effect. Updating one's network might make *that* attack no longer viable (though this is not even guaranteed; Karpathy, 2015), and it might theoretically make the network more difficult to disrupt, but patching will not immunize the network from perturbations as a whole (Laugros, Caplier, & Ospici, 2019). Furthermore, changing the network enough to stop an AML-created attack can also degrade the system's performance of its primary function (Qiu, Liu, Zhou, & Wu, 2019; Xie et al., 2019). Any patches to a decision network to defeat an AML exploitation would have to be regression tested to assess its core functioning, and this could end up being an endless and costly cycle that is unlikely to decrease overall vulnerability.

It is not possible to immunize against AML as a class of attack. Instead, the focus of T&E should be on the speed of mitigation and recovery.

Instead, we recommend that AML cybersecurity (and testing thereof) should be focused on attack detection, attack and defense shelf-life, and minimization of the probability that on any given day, the adversary's day-zero attack is ready. We should assume that eventually, the adversary will get a copy of a system's network and train an exploiting attack against it. Though we cannot make the network immune to AML as a whole, some networks are harder to train against than others. We recommend that robustness against AML be defined as the computational and/or data resources needed to train an adversarial model against our system.

a. Run Time Monitors

It will always take more resources to develop and certify a behaviorally functional system than it will to develop an adversarial attack capable of disrupting that network. There are a core set of functions that the decision network must maintain, and any changes to the network can threaten those functions. Regression testing will be necessary when core decision networks are changed. Adversarial attack networks do not suffer from this problem. In evolutionary terms, the generation time for adversary networks is faster, and so they will adapt faster. Like organisms need an immune system that can adapt at the same speed as viruses without hurting core genetic functioning, AI&A will need a system that can at least detect, and preferably mitigate or defeat adversarial attacks without altering the certified behavioral model.

We recommend that all AI&A systems have a middleware adversarial attack “immune system” (Lin, Shi, & Xue, 2019), and that the focus of cybersecurity testing against adversarial networks should be on this immune system. AML still trains models, and these are also vulnerable to having a model trained against them. The best way use AML is to have access to the model one wants to disrupt (Qiu et al., 2019). As we have access to our own decision models, we can train a system or systems to disrupt that model, for example using GANs. We can then in turn train an immune system to at minimum detect (Lin et al., 2019), and potentially mitigate or disrupt adversarial attacks from the first generation GANs we created. A system’s cognitive instrumentation can feed the data that enables this detection by a defending GAN (Haugh et al., 2018). As this immune network is also vulnerable to exploitation, and thus the competition is never done, we recommend that testers focus less on the existence or description of vulnerabilities and more on quality and timelines for the immune network. We recommend cyber testing examine at least a few key attributes: (1) the viability of AML attacks trained against old versions of a decision model to disrupt later versions, (2) the immune system’s ability to detect/mitigate/disrupt known adversarial attacks, and (3) the estimated generation times of our networks compared to adversary models. The cycle time is important, because in order to generate an AML attack, one needs either the network itself or massive quantities of input/output data from that network (Athalye et al., 2018; Ilyas, Engstrom, Athalye, & Lin, 2018), meaning it may take time for adversaries to acquire the decision and immune networks. It will also take resources to create an AML model that can both disrupt the decision network and escape the immune network. This essentially defines the adversary’s generation cycle. This should be compared to our generation cycle, which includes the rate at which decision network updates are released, the estimated viability of old AML attacks against new decision models, and our ability to update and certify new immune systems. Part of the continuum of cyber vulnerability to adversary attacks will be whether we have a faster generation time than our adversaries.

AML is also vulnerable to AML. We can theoretically use AML to detect AML attacks.

Testers can assess the **time and resources** required to **create** an **AML** attack on a system vs. the **time and resources** required for blue forces to **render that attack unviable** (i.e., the adversarial generation cycle time).

While this is not the only type of cyber testing that will be needed, it is one form that it could take. Cybersecurity is an area where we particularly invite the thoughts of the community on how to ensure reliability in AI&A in the face of adversarial action.

2. Anti-Tamper Mechanisms / Program Protection

Because of the importance of protecting our systems' decision software and the inability of AI&A to resist interrogation if captured, it will be important for systems to have anti-tamper mechanisms. T&E will need to demonstrate both that these anti-tamper mechanisms are effective and that they themselves are not vulnerable to exploitation. If an adversary captures the physical asset where our system's decision software is embedded, they have passed one of the first hurdles to employing AML or reverse engineering. Even if the software is itself somehow encrypted or protected from direct inspection, enough input-output queries can allow someone to recreate the decision system (Qiu et al., 2019). Systems will therefore need anti-tamper mechanisms to protect themselves from capture, and T&E must show that these mechanisms work. Because a system would likely be captured by near peer adversaries, these mechanisms would need to be tested under denied, degraded, and operationally realistic conditions. Furthermore, because these mechanisms provide a potential avenue for destroying or disabling our systems, the security of the anti-tamper mechanism must undergo exploitation testing of its own.

3. Traditional Adversarial Interference

Adversaries will also be able to disrupt these systems using traditional kinetic or electronic attacks, and testing must evaluate the robustness of decision making while subject to this kind of interference. In particular, we recommend that testing both systematically and realistically explore system effectiveness when sensors and effectors are degraded or destroyed. As discussed in the section on learning, crystalized networks may perform well under ideal conditions, but worse under degraded ones. Networks are sensitive not just to planned changes, but also to hardware losses during combat. Testers should ensure this type of evaluation is included in test plans.

4. Behavioral Exploitation

Testers should examine an adversary's ability to develop tactics that reliably defeat our AI&A. A system that reliably defeated the human agents we tested it against could still be behaviorally brittle (e.g., Defense Science Board, 2012; Zacharias, 2019a). The ability to predict adversary actions is a key component to defeating an opponent (Sun Tzu, c. 450 BCE). Whether based on hard-coded rules, reinforcement learning policies, or other mechanisms, systems make systematic decisions. When a counter-strategy has been discovered for an AI&A's policy, the more brittle the AI&A's decision-making, the more consistently the counter-strategy will defeat it.

Adversarial generation cycle times will also be relevant for tactical exploitation.

We recommend that testers attempt to quantify where AI&A decision-making lies on the continuum between brittleness and flexibility. One method might be to examine generation time, as we recommended for adversarial networks. Testers can quantify how long it takes an adversary to discover the system's policy and how long it takes to develop a counter policy. One could use the number of exposures, real elapsed time, resources, or any number of metrics to do so. This adversary generation time could be compared to the system's generation time. For early systems, this would probably be a measure of industrial agility. For example, in OpenAI Five's first game against professional Dota 2[89] players, it performed well early on, but within the space of a single game, the human players were able to identify its policy and develop a counter-strategy that consistently beat the AI (Vincent, 2018). The next year, a retrained, more sophisticated system beat the professional world champions 2-0 (Statt, 2019), but when released into the wild for a few days, some human opponents were eventually able to figure out its policy and begin winning consistently (Wiggers, 2019). We recommend that part of system development and testing be this kind of behavioral red teaming (Defense Science Board, 2016; Zacharias, 2019a), and these cycle times can be tracked to help measure system progress, much like we do now for system reliability growth. As systems become more advanced, this behavioral adaptation speed may even move into real time, with policy shifts and counter-strategy cycles occurring multiple times within a single engagement. Even if this becomes the case, the quantified cycle speed is still a relevant metric.

[89] Dota 2, is a multiplayer online video game pitting two teams of five players trying to destroy the other's base. Players, each of whom possesses unique skills and abilities, work as a team but also fight as individuals. See https://openai.com/blog/openai-five/ for a description of the AI challenges in Dota 2 relative to those in other games such as Go or chess.

4. Conclusions

AI-enabled or autonomous systems are not magic. The fundamentals of testing will not change: we still need to observe tasks, record outcomes, and do those things across a systematic set of conditions. What separates AI&A from our standard systems is that they make decisions on their own—they operate without an operator. To assess these systems, we have to assess their decision-making. Fortunately, the study of decision-making is not new. A variety of the behavioral sciences have developed methods to study this topic in humans. Though some adaptation may be necessary, the lessons learned by these other fields can help jump start T&E of AI&A within DoD.

However, AI-enabled or autonomous systems are not a homogenous category, and differences among these systems will drive choices about test strategies. In our framework, we identify a few broad characteristics that will affect how we test these systems. The attributes that separate decision-making systems from each other are (1) the kinds of decisions they make, (2) design choices regarding modular vs. monolithic architecture and symbolic vs. sub-symbolic processing, (3) the extent to which testers understand how the system makes its decisions, (4) the risk involved in those decisions, (5) how amendable to simulation they are, and (6) how these systems will interact with other decision-making agents. All of these factors have implications for how to test these systems.

We identified three types of decision—executive, perceptual, and procedural—that can affect what we test. Testers should evaluate whether the system makes decision types within a given task, not whether they make them at all as some kind of system property. Executive decisions are colloquially "should" decisions; more technically, systems with this kind of autonomy make decisions about their goals and constraints. Perceptual decisions are colloquially "what is" decisions—formally they define problem states. Procedural decisions are colloquially "how" decisions; formally they select the immediate next procedure[90] in pursuit of a goal. The biggest (though not the only) effect these different decision types have on testing is on the types of metrics that need to be collected.

Perhaps the most important attribute for testing, however, is the extent to which we understand what causally drives systems to make one decision over another. If we do not understand the system's decision model, we cannot make inferences about performance under conditions that have not been explicitly tested. Furthermore, we cannot validate that a simulation adequately represents reality if we do not understand which aspects of reality drive system decisions. Both of these have massive implications for the amount of testing that would be needed, to the extent that proceeding without a model of the system's decision-making will be impossible as a practical matter.

[90] Called an "operator" in the formal parlance of the problem space hypothesis

When systems make decisions that are hugely consequential under conditions that are difficult to simulate reliably, a new approach to testing will be necessary. These systems must be exposed to live, realistic conditions to credibly believe that the decisions observed in testing are what it will do when fielded, but it would not be safe to allow them to actually operate. In these cases, testers should certify low-risk capabilities, and then monitor them after they are fielded for those capabilities under human supervision. As they are exposed to realistic scenarios, have the systems evaluate what they *would* have decided regarding their riskier capabilities, and use these data to eventually certify further capabilities.

Finally, though AI&A notionally make their own decisions, they will do so in the context of other decision-making agents, human and artificial. AI&A systems must be tested in these interactive contexts to correctly understand their fielded behavior. The exact nature of these relationships (e.g., operator-on-the-loop, two artificial agents, or true teammates) will inform testers about which methods they need to use.

There are a several interdependent policy and design choices that will critically enable our test strategies. First, it is much easier to obtain and initially evaluate decision models for systems that are designed using at least hybrid symbolic and sub-symbolic approaches in a modular architecture. Second, systems need a built-in infrastructure for recording data (BIRD)—an end-to-end pipeline starting with internal cognitive instrumentation and extending to securely collate, transmit, and store these system internal data. The modules in a hybrid architecture can act as the hooks for the cognitive instrumentation.

This document is only Part One of a long endeavor. Though many might consider this document to be too long already, ironically it is neither comprehensive nor detailed enough for test execution. This framework is meant to identify *what* needs to be tested in these systems; it is not a how-to guide for test planning and execution. We have covered those topics to some degree, but more work remains to create a practitioner’s guidebook. In many cases, entire fields and complex procedures have been boiled down to a single sentence. A series of future documents will try to delve into these topics in sufficient detail for working-level testers to follow. Also, while we have discussed performance evaluation, we have not touched on the topic of test efficiency. This is the other half of our framework, and it is complex enough to require its own roadmap and product series. In sum, our work is far from done, but we hope this paper takes an important step forward in the quest to provide assurance for AI-enabled and autonomous systems.

5. Recommendations

- **Testers need to identify the features of autonomous systems that will (and will not) cause traditional test methods to misinform decision-makers about risk.** We need to identify when, why, and how testing will need to be different for AI-enabled systems. Overarching definitions of AI or autonomy often exclude some systems that would be difficult to test, and programs are not self-identifying as involving such risks. Other definitions suffer from disagreement over the meaning of words. In this paper, we define AI and autonomy as anything that makes decisions based on environmental information within the constraints of a specific task. We identify three types of decision—setting goals or constraints, defining the current situation, and choosing the next action—to help identify what does and does not change about testing. To avoid ambiguity, these definitions are grounded in a technical theory of decision-making.

- **Testers need more transparency in decision-making systems.** Transparency is important for end-users, but also for testers. Black-box systems prevent testers from making inferences about untested scenarios. Before we can confidently test system performance, we must understand how the system makes its decisions. This transparency can be built-in at the drawing board, or, as a less desirable option, the lack of transparency in design can be mitigated during early testing. We make recommendations for how to obtain, verify, and validate models of what causally drives system decision-making.

- **Testers need rights to system decision-making and learning processes and data generated by these systems.** In addition to benefits such as enabling modularity and reusability across systems, gaining ownership rights to the decision software is critical to testing. Proprietary concerns can cause an otherwise transparent system to be a black box to testers, as has already happened with several systems.

- **Common, modular cognitive architectures enable testing.** Many have discussed how modular cognitive architectures benefit system development, performance, and sustainment. Here we discuss how they facilitate efficient and effective T&E as well.

- **Research into and dissemination of methods for evaluating decision-making are needed.** These include metrics to quantify intermediate mission success, methods to qualitatively evaluate overall decision processes, novel calculations of classification accuracy for multi-categorical fuzzy groups, and ways to quantify a system's ability to learn.

- **Decision-making systems that have a built-in infrastructure for recording data (BIRD) become easier to certify.** We recommend a BIRD to enable testing, but it would serve many different needs. By having systems record data about themselves, by themselves, and by providing an infrastructural pipeline to securely collate, store, and disseminate these data, stakeholders can harvest data from a variety of previously inaccessible venues such as exercises and operational missions. These harvests can support many activities like T&E, operator and commander decision making, and post-fielding fleet-wide learning.

- **Testers can use a strategy of Graded Autonomy with Limited Capability Fielding for difficult-to-certify systems.** Some systems are too dangerous to test live, but too difficult to simulate credibly. These systems should be tested like we do with medical residents. Train all skills, and then certify and field their least risky capability for use under supervision. While acting in realistic situations in the field or exercises, have systems evaluate what they *would* have done with more risky capabilities. Use these data to spiral upward through risk and down through supervision levels as systems demonstrate safe competence.

- **Testers should characterize system flexibility as well as system performance.** Decision systems can achieve greater performance on a specific task by over-optimizing, which can create downstream costs and consequences when trying to upgrade, change, learn, or transfer to a related task. Testing should evaluate to what extent programs have made this tradeoff.

- **Testers need environments where different autonomous agents, including humans, can be tested together for emergent behavior.** When autonomous agents interact, you can get emergent behavior (EB). EB can be expected or unexpected, and it can be desirable or undesirable. Testers need to confirm that expected, desirable EB (such as teaming or synergy) functions correctly, while minimizing the probability of unexpected, undesirable EB. This must be tested under live as well as simulated environments. Centralizing test responsibility for EB can overcome a number of simulation challenges, while having a regular joint exercise would provide such a live test venue for validation while also helping troop readiness for existing and emerging technology employment.

- **Testers still need to emphasize human-system interaction for autonomous systems.** Even in fully autonomous systems, a human will be involved in some part of their decision-making chain, even if it is just issuing initial orders. These interactions must be fluid and minimize error to ensure responsible employment, and testers must evaluate this. Additionally, the acquisition community should assess whether warfighters will have *appropriately calibrated* trust of their systems.

- **Testers should adapt existing methods for evaluating human teams for the T&E of human-machine teams.** Though not all AI-human system relationships will truly involve teaming, systems that do will require a different approach to testing. The starting point for these evaluations should be the methods already created by the behavioral sciences and sports statisticians.

- **Testers should assess adversarial exploitation generational cycles.** Cyber and tactical exploitation is a never-ending, constantly evolving battle in learning systems. This may require a cultural shift away from testing against static, well-defined exploitation requirements. Testers should attempt to quantify how quickly adversaries can develop exploitations of our decision systems versus the speed at which we can re-counter them. Having a faster friendly than adversary cycle will likely be critical to meaningfully field these systems. At first this will be a test of industrial agility, though in time in may be a metric of systems' live behavioral flexibility.

References

"MCubed Program Requirements". (n.d.). Retrieved from https://mcubed.umich.edu/mcubed-program-requirements

Abel, D., Arumugam, D., Asadi, K., Jinnai, Y., Littman, M. L., & Wong, L. L. S. (2019). *State abstraction as compression in apprenticeship learning.* Paper presented at the Thirty-Third AAAI Conference on Artificial Intelligence.

Ahner, D. K., & Parson, C. R. (2016). *Workshop report: Test and evaluation of autonomous systems*. STAT Center of Excellence.

Ahner, D. K., Parson, C. R., Thompson, J. L., & Rowell, W. F. (2018). Overcoming the challenges in test and evaluation of autonomous robotic systems. *The ITEA Journal of Test and Evaluation, 39*, 86-94.

Altman, D. G., & Royston, P. (2006). The cost of dichotomising continuous variables. *BMJ, 332*(7549), 1080.

Arnold, T., & Scheutz, M. (2018). The "big red button" is too late: an alternative model for the ethical evaluation of AI systems. *Ethics and Information Technology, 20*(1), 59-69. doi:10.1007/s10676-018-9447-7

Ashby, F. G. (2000). A stochastic version of general recognition theory. *Journal of Mathematical Psychology, 44*(2), 310-329. doi:10.1006/jmps.1998.1249

Ashby, F. G., & Townsend, J. T. (1986). Varieties of perceptual independence. *Psychological Review, 93*(2), 154-179. doi:10.1037/0033-295X.93.2.154

Athalye, A., Engstrom, L., Ilyas, A., & Kwok, K. (2018). Synthesizing robust adversarial examples. *ArXiv e-prints*.

Awh, E., Jonides, J., & Reuter-Lorenz, P. A. (1998). Rehearsal in spatial working memory. *Journal of Experimental Psychology: Human Perception and Performance, 24*(3), 780-790. doi:10.1037/0096-1523.24.3.780

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., & Mordatch, I. (2019). Emergent tool use from multi-agent autocurricula. *ArXiv e-prints*.

Barlow, H. B. (1961). Possible principles underlying the transformations of sensory messages. In W. Rosenblith (Ed.), *Sensory Communication* (pp. 217-234). Cambridge, MA: MIT Press.

Bathaee, Y. (2018). The artificial intelligence black box and the failure of intent and causation. *Harvard Journal of Law & Technology, 31*(2), 890-938.

Bayes, T., & Price, R. (1763). An essay towards solving a problem in the doctrine of chance. By the late Rev. Mr. Bayes, communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. *Philosophical Transactions of the Royal Society of London, 53*, 370-418.

Bernstein, L. (2018). How companies and the public are reacting to Uber's driverless car crash. *ABC 7 WJLA*. Retrieved from https://wjla.com/news/nation-world/how-companies-and-the-public-are-reacting-to-ubers-driverless-car-crash

Best, J. R., & Miller, P. H. (2010). A developmental perspective on executive function. *Child Development, 81*(6), 1641-1660. doi:10.1111/j.1467-8624.2010.01499.x

Box, G. E. P. (1976). Science and statistics. *Journal of the American Statistical Association, 71*(356), 791-799.

Box, G. E. P. (1979). Robustness in the strategy of scientific model building. In R. L. Launer & G. N. Wilkinson (Eds.), *Robustness in statistics*. New York: Academic Press.

Butchibabu, A., Sparano-Huiban, C., Sonenberg, L., & Shah, J. (2016). Implicit coordination strategies for effective team communication. *HUMAN FACTORS, 58*(4), 595-610. doi:10.1177/0018720816639712

Caseley, P. (2018). *Human-machine trust: Risk-based assurance and licensing of autonomous systems.* Paper presented at the SCI-313 Specialist Meeting Report.

Casola, L., & Ali, D. (2019). *Robust machine learning algorithms and systems for detection and mitigation of adversarial attacks and anomalies*. Washington, D.C.: The National Academies Press.

Chiesi, A. M. (2015). Network analysis. In J. D. Wright (Ed.), *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)* (pp. 518-523). Oxford: Elsevier.

Clarke, E. M., Klieber, W., Nováček, M., & Zuliani, P. (2012). Model checking and the state explosion problem. In B. Meyer & M. Nordio (Eds.), *Tools for Practical Software Verification: LASER, International Summer School 2011, Elba Island, Italy, Revised Tutorial Lectures* (pp. 1-30). Berlin, Heidelberg: Springer Berlin Heidelberg.

Cook, A. (2019). *Taming killer robots*. The JAG School Papers: Air University Press.

Cooke, N. J., Gorman, J. C., Myers, C. W., & Duran, J. L. (2013). Interactive team cognition. *Cogn Sci, 37*(2), 255-285. doi:10.1111/cogs.12009

Culley, K. E., & Madhavan, P. (2013). Trust in automation and automation designers: Implications for HCI and HMI. *Computers in Human Behavior, 29*(6), 2208-2210. doi:10.1016/j.chb.2013.04.032

Dastin, J. (2018). Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters*. Retrieved from https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

Decety, J., Norman, G. J., Berntson, G. G., & Cacioppo, J. T. (2012). A neurobehavioral evolutionary perspective on the mechanisms underlying empathy. *Progress in Neurobiology, 98*(1), 38-48. doi:https://doi.org/10.1016/j.pneurobio.2012.05.001

Defense Innovation Board. (2018). *DIB guide: Detecting agile BS*. Retrieved from https://media.defense.gov/2018/Oct/09/2002049591/-1/-1/0/DIB_DETECTING_AGILE_BS_2018.10.05.PDF.

Defense Science Board. (2012). *The Role of Autonomy in DoD Systems*. Washington, DC.

Defense Science Board. (2016). *Summer Study on Autonomy*. Washington, D.C.

Deonandan, I., Valerdi, R., Lane, J. A., & Macias, F. (2010). *Cost and risk considerations for test and evaluation of unmanned and autonomous systems of systems*. Paper presented at the 5th International Conference on System of Systems Engineering, Loughborough, UK.

Autonomy in Weapons Systems, 3000.09 C.F.R. (2012).

Director Operational Test & Evaluation. (2009). *Using design of experiments for operational test and evaluation*.

Director Operational Test & Evaluation. (2010). *Guidlines for operational test and evaluation of information and business systems*.

Director Operational Test & Evaluation. (2019). *Guidance for testing and evaluating human-system interaction*.

Duch, J., Waitzman, J. S., & Amaral, L. A. N. (2010). Quantifying the performance of individual players in a team activity. *PLOS ONE, 5*(6), e10937. doi:10.1371/journal.pone.0010937

Dunbar, R. I. M., & Shultz, S. (2007). Evolution in the social brain. *Science, 317*(5843), 1344-1347. doi:10.1126/science.1145463

Durst, P. J., & Gray, W. (2014). *Levels of autonomy and autonomous system performance assessment for intelligent unmanned systems*. (ERDC/GSL SR-14-1). US Army Engineer Research and Development Center (ERDC).

Easton, K. I., & Martinoli, A. (2002). *Efficiency and optimization of explicit and implicit communication schemes in collaborative robotics experiments.* Paper presented at the International Conference on Intelligent Robots and Systems, EPFL, Lasuanne, Switzerland.

Endsley, M. R. (2000). Direct measurement of situational awareness: Validity and use of the SAGAT. In M. R. Endsley & D. J. Garland (Eds.), *Situational awareness analysis and measurement.* Mahwah, New Jersey: Lawrence Erlbaum Associates.

Endsley, M. R. (2015). *Autonomous horizons: System autonomy in the Air Force – A path to the future*. Maxwell AFB, AL: Air University Press.

Endsley, M. R. (2019). Paper presented at the 63rd International Annual of the Human Factors & Ergonomics Society, Seattle, Washington.

Endsley, M. R., & Kaber, D. B. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics, 42*(3), 462-492. doi:10.1080/001401399185595

Erhan, D., Bengio, Y., Courville, A., & Vincent, P. (2009). *Visualizing higher-layer features of a deep network*. Retrieved from Montreal, QC, Canada:

Espinosa, A., Lerch, J., & Kraut, R. (2002). Explicit vs. implicit coordination mechanisms and task dependencies: One size does not fit all. In E. Salas, S. M. Fiore, & J. A. Cannon-Bowers (Eds.), *Team cognition: Process and performance at the inter- and intra-individual level*. Washington, DC: American Psychological Association.

Executive Order No. 13859. (2019). *Executive Order 13859: Maintaining American Leadership in Artificial Intelligence*. United States: Office of the Federal Register.

Falk, D. R., & Johnson, D. W. (1977). The effects of perspective-taking and egocentrism on problem solving in heterogeneous and homogeneous groups. *The Journal of Social Psychology, 102*(1), 63-72. doi:10.1080/00224545.1977.9713241

Federal Aviation Administration. (2000). *Human factors engineering and safety principles & practices*.

Federal Aviation Administration. (2018). *Flight test guide for certification of transport category aircraft*. (Advisory circular 25-7D).

Ferreira, S., Faezipour, M., & Corley, H. W. (2013). *Defining and addressing the risk of undesirable emergent properties*. Paper presented at the 2013 IEEE International Systems Conference (SysCon), Orlando, FL.

Festinger, L., & Carlsmith, J. M. (1959). Cognitive consequences of forced compliance. *Journal of Abnormal and Social Psychology, 58*, 203-210.

Fisher, R. A. (1935). *The design of experiments*. Oxford, England: Oliver & Boyd.

Fletcher, G. J. O. (1986). Psychology and common sense. *IEEE Engineering Management Review, 14*(4), 30-40. doi:10.1109/emr.1986.4306241

Friston, K. J. (2011). Functional and effective connectivity: A review. *Brain Connect, 1*(1), 13-36. doi:10.1089/brain.2011.0008

Funahashi, K. I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks, 2*(3), 183-192.

Gawron, V. J. (2008). *Human performance, workload, and situational awareness measures handbook* (2nd ed.): CRC Press.

Giampapa, J. A. (2013). Test and evaluation of autonomous multi-robot systems. Pittsburgh, PA: Software Engineering Institute.

Gil, Y., & Selman, B. (2019). A 20-year community roadmap for artificial intelligence research in the US. *Computing Community Consortium (CCC) and Association for the Advancement of Artificial Intelligence (AAAI).*

Goerger, S. R. (2004). *Validating human behavioral models for combat simulations using techniques for the evaluation of human performance*. Paper presented at the SCSC '03.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). *Generative adversarial nets*. Paper presented at the Advances in Neural Information Processing Systems 27 (NIPS 2014).

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *ArXiv e-prints*.

Goodhart, C. (1981). Problems of monetary management: The U.K. experience. In A. S. Courakis (Ed.), *Papers in Monetary Economics*. Totowa, New Jersey: Barnes & Noble Books.

Google. (2019). *AI Explainability Whitepaper*. Retrieved from https://storage.googleapis.com/cloud-ai-whitepapers/AI%20Explainability%20Whitepaper.pdf

Gray, C. (2015, May 19, 2015). When the only winning move is not to play. Retrieved from https://conradthegray.com/blog/when-the-only-winning-move-is-not-to-play/

Green, D. M., & Swets, J. A. (1966). *Signal detection theory and psychophysics*. Oxford, England: John Wiley.

Greer, K. (2013). A metric for modelling and measuring complex behavioural systems. *IOSR Journal of Engineering (IOSRJEN), 3*(11).

Gunning, D. (2017). Explainable Artificial Intelligence program update: DARPA.

Gunning, D. (2018). Machine Common Sense Concept Paper. *arXiv e-prints*. Retrieved from https://ui.adsabs.harvard.edu/abs/2018arXiv181007528G

Gunning, D. (2019). *Explainable Artificial Intelligence*. Paper presented at the Board of Human-Systems Integration: Explainable AI Frontiers: Human-Systems Integration Challenges and Opportunities, Washington, D.C.

Gunning, D., & Aha, D. W. (2019). DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Magazine, 40*(2), 44-58. doi:https://doi.org/10.1609/aimag.v40i2.2850

Halpern, S. D., & Detsky, A. S. (2014). Graded autonomy in medical education — Managing things that go bump in the night. *The New England Journal of Medicine, 370*(12), 1086-1089.

Hancock, P. A., & Warm, J. S. (1989). A dynamic model of stress and sustained attention. *Human performance in extreme environments : the journal of the Society for Human Performance in Extreme Environments, 31*(5), 519-537. doi:10.1177/001872088903100503

Harikumar, J., & Chan, P. (2019). *Developing knowledge and understanding for autonomous systems for analysis and assessment events and campaigns*. (ARL-TR-8649).

Haugh, B. A., Sparrow, D. A., & Tate, D. M. (2018). *The status of test, evaluation, verification, and validation (TEV&V) of autonomous systems*. Retrieved from Alexandria, VA:

Hazan, T., Papandreou, G., & Tarlow, D. (2016). Introduction. In T. Hazan, G. Papandreou, & D. Tarlow (Eds.), *Perturbations, Optimization, and Statistics* (pp. 1-4). Cambridge, Massachusetts: The MIT Press.

Helle, P., Schamai, W., & Strobel, C. (2016). *Testing of autonomous systems: Challenges and current state-of-the-art*. Paper presented at the 26th Annual INCOSE International Symposium (IS 2016), Edinburg, Scotland, UK.

Hernández-Orallo, J. (2016). Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement. *Artificial Intelligence Review, 48*(3), 397-447. doi:10.1007/s10462-016-9505-7

Hess, J. T., & Valerdi, R. (2010). *Test and evaluation of a SoS using a prescriptive and adaptive testing framework.* Paper presented at the 2010 5th International Conference on System of Systems Engineering, Loughborough, UK.

Heyes, C. (2012). New thinking: The evolution of human cognition. *Philos Trans R Soc Lond B Biol Sci, 367*(1599), 2091-2096. doi:10.1098/rstb.2012.0111

Horowitz, M., & Scharre, P. (2015). *Meaningful human control in weapon systems: A primer*: Center for a New American Security.

Howes, A., Lewis, R. L., & Vera, A. (2009). Rational adaptation under task and processing constraints: Implications for testing theories of cognition and action. *Psychological Review, 116*(4), 717-751.

Huttenlocher, P. R. (1979). Synaptic density in the human prefrontal cortex: Developmental changes and effects of aging. *Brain Research, 163*(2), 195-205. doi:10.1016/0006-8993(79)90349-4

Ilachinski, A. (2017). *AI, robots, and swarms issues: Questions and recommended studies*. Retrieved from Arlington, VA:

Ilyas, A., Engstrom, L., Athalye, A., & Lin, J. (2018). Black-box adversarial attacks with limited queries and information. *ArXiv e-prints*.

Jafferjee, A. (2019). Machine learning for data cleaning and unification. Retrieved from https://towardsdatascience.com/machine-learning-for-data-cleaning-and-unification-b3213bbd18e

Jerison, H. J., & Pickett, R. M. (1964). Vigilance: The importance of the elicited observing rate. *Science, 143*(3609), 970-971. doi:10.1126/science.143.3609.970

Johnson, G. (1984). Eurisko, the computer with a mind of its own. *the APF Reporter*. Retrieved from https://aliciapatterson.org/stories/eurisko-computer-mind-its-own

Joint Chiefs of Staff. (2018). *Joint Publication 3-0*.

Josephs, L. (2019). Under fire for Boeing 737 Max crashes, FAA chief vows to examine how humans interact with automated aircraft systems. Retrieved from https://www.cnbc.com/2019/11/12/faa-chief-vows-to-examine-how-humans-interact-with-aircraft-systems.html

Kahneman, D. (1973). *Attention and effort*. Englewood Cliffs, New Jersey: Prentice Hall.

Kandel, E. R., Schwartz, J. H., Jessell, T. M., Siegelbaum, S. A., & Hudspeth, A. J. (2012). *Principles of neural science* (5th ed.): McGraw-Hill.

Karpathy, A. (2015). Breaking linear classifiers on ImageNet. Retrieved from http://karpathy.github.io/2015/03/30/breaking-convnets/

Kenny, D. A., Kashy, D. A., & Cook, W. L. (2006). *Dyadic data analysis*. New York, New York: The Guilford Press.

Kheradpisheh, S. R., Ghodrati, M., Ganjtabesh, M., & Masquelier, T. (2016). Deep networks can resemble human feed-foward vision in invariant object recognition. *Sci Rep, 6*, 32672. doi:10.1038/srep32672

Kosiorek, A. R., Sabour, S., Teh, Y. W., & Hinton, G. E. (2019). *Stacked capsule autoencoders*. Paper presented at the 33rd Conference on Neural Information Processing Systems, Vancouver, Canada.

Krichmar, J. L. (2012). Design principles for biologically inspired cognitive robotics. *Biologically Inspired Cognitive Architectures, 1*, 73-81. doi:10.1016/j.bica.2012.04.003

Krichmar, J. L., Severa, W., Khan, M. S., & Olds, J. L. (2019). Making BREAD: Biomimetic strategies for artificial intelligence now and in the future. *Front Neurosci, 13*, 666. doi:10.3389/fnins.2019.00666

Kurup, U., & Lebiere, C. (2012). What can cognitive architectures do for robotics? *Biologically Inspired Cognitive Architectures, 2*, 88-99. doi:10.1016/j.bica.2012.07.004

Kwashnak, P. (2019). *Autonomous Systems Test Capability (ASTC) overview*. Paper presented at the Workshop on Test and Evaluation of Artificial Intelligence Enabled Systems, Aberdeen Proving Grounds, Maryland.

Laird, J. E. (2012). *The SOAR cognitive architecture*. Cambridge, Massachusetts: The MIT Press.

Laugros, A., Caplier, A., & Ospici, M. (2019). *Are Adversarial Robustness and Common Perturbation Robustness Independant Attributes?* Paper presented at the Proceedings of the IEEE International Conference on Computer Vision Workshops.

Laverghetta, T. J., Leathrum, J. F., & Gonda, N. (2018). *Integrating virtual and augmented reality based testing into the development of autonomous vehicles*. Paper presented at the MODSIM World 2018, Norfolk, VA.

Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *HUMAN FACTORS, 46*(1), 50-80. doi:10.1518/hfes.46.1.50_30392

Lin, Z., Shi, Y., & Xue, Z. (2019). *IDSGAN: Generative adversarial networks for attack generation against instrusion detection*. arXiv.

Linkenhoker, B. A., & Knudsen, E. I. (2002). Incremental training increases the plasticity of the auditory space map in adult barn owls. *Nature, 419*(6904), 293-296.

Lowrance, C., Herman, H., Schneider, J., & Kasemer, J. (2019). *Automatic Target Recognition (ATR) from Mobile Cooperative and Autonomous Sensors (MCAS)*. Paper presented at the Workshop on Test and Evaluation of Artificial Intelligence Enabled Systems, Aberdeen Proving Grounds, Maryland.

Luna, S., Lopes, A., Tao, H. Y. S., Zapata, F., & Pineda, R. (2013). Integration, verification, validation, test, and evaluation (IVVT&E) framework for system of systems (SoS). *Procedia Computer Science, 20*, 298-305. doi:10.1016/j.procs.2013.09.276

Macias, F. (2008). The Test and Evaluation of Unmanned and Autonomous Systems. *ITEA Journal, 29*, 388-395.

Mafessoni, F., & Lachmann, M. (2019). The complexity of understanding others as the evolutionary origin of empathy and emotional contagion. *Scientific Reports, 9*(1), 5794. doi:10.1038/s41598-019-41835-5

Marcum, J. I. (1947). *A statistical theory of target detection by pulsed radar*. Retrieved from Santa Monica, California:

Marcus, G. (2018). Deep learning: A critical appraisal. *ArXiv e-prints*. Retrieved from https://arxiv.org/ftp/arxiv/papers/1801/1801.00631.pdf

Marshall, A., Rojas, R., Stokes, J., & Brinkman, D. (2018). Securing the future of artificial intelligence and machine learning at microsoft. Retrieved from https://docs.microsoft.com/en-us/security/securing-artificial-intelligence-machine-learning

Mathieu, J. E., Heffner, T. S., Goodwin, G. F., Salas, E., & Cannon-Bowers, J. A. (2000). The influence of shared mental models on team process and performance. *Journal of Applied Psychology, 85*(2), 273-283. doi:10.1037/0021-9010.85.2.273

McLean, A. L., Bertram, J. R., Hoke, J. A., Rediger, S. S., & Skarphol, J. C. (2016). *LVC-enabled testbed for autonomous system testing*. Rockwell Collins. Retrieved from https://insights.rockwellcollins.com/2016/10/31/lvc-enabled-testbed-for-autonomous-system-testing/

Meyes, R., Lu, M., Puiseau, C., & Meisen, T. (2019). Ablation studies in artificial neural networks. *ArXiv e-prints*.

Micskei, Z., Szatmári, Z., Oláh, J., & Majzik, I. (2012). *A Concept for Testing Robustness and Safety of the Context-Aware Behaviour of Autonomous Systems.* Paper presented at the KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications.

Miller, H. (2019). *Report on test infrastructure for emerging technology*.

Miller, T., Howe, P., & Sonenberg, L. (2017). Explainable AI: Beware of inmates running the asylum Or: How I learnt to stop worrying and love the social and behavioral sciences. *ArXiv e-prints*. Retrieved from https://arxiv.org/abs/1712.00547v2

Montgomery, D. C. (2019). *The design and analysis of experiments* (10th ed.): Wiley.

Mueller, S. T., Hoffman, R. R., Clancey, W., Emrey, A., & Klein, G. (2019). *Explanation in human-AI systems: A literature meta-review synopsis of key ideas and publications and bibliography for explainable AI*.

Murciano, A., & Millán, J. d. R. (1996). Learning signaling behaviors and specialization in cooperative agents. *Adaptive Behavior, 5*(1), 5-28. doi:10.1177/105971239600500102

Narla, A., Kuprel, B., Sarin, K., Novoa, R., & Ko, J. (2018). Automated classification of skin lesions: From pixels to practice. *Journal of Investigative Dermatology, 138*(10), 2108-2110. doi:https://doi.org/10.1016/j.jid.2018.06.175

National Transportation Safety Board. (2019a). *Collision between US Navy Destroyer John S McCain and Tanker Alnic MC Singapore Strait, 5 Miles Northeast of Horsburgh Lighthouse August 21, 2017*. (NTSB/MAR-19/01 or PB2019-100970).

National Transportation Safety Board. (2019b). *Collision between vehicle controlled by developmental automated driving system and pedestrian, Tempe, Arizona, March 18, 2018*. (NTSB/HAR-19/03 or PB2019-101402).

Neema, S. (2019). *Assured autonomy*. Paper presented at the SafeAI 2019, Honolulu, Hawaii.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, Massachusetts: Harvard University Press.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, N.J.: Prentice Hall.

Nisbett, R. E., & Wilson, T. D. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological Review, 84*(3), 231-259.

Nowak, W., & Guthke, A. (2016). Entropy-based experimental design for optimal model discrimination in the geosciences. *Entropy, 18*, 409.

O'Connell, M. E. (2015). *Integration of multidimensional signal detection theory with fuzzy signal detection theory.* (Doctor of Philosophy), University of Central Florida, Orlando, Florida. Retrieved from http://purl.fcla.edu/fcla/etd/CFE0005983 (CFE0005983)

O'Connor, T., & Wong, H. Y. (2012). Emergent properties. *Stanford Encyclopedia of Philosophy.* Spring 2012. Retrieved from https://plato.stanford.edu/archives/spr2012/entries/properties-emergent/

OpenAI. (2018). OpenAI Five. Retrieved from https://openai.com/blog/openai-five/

Overholt, J., & Kearns, K. (2013). *AFRL autonomy*. (88ABW-2013-3169).

Owens, D. (2020). *Introduction to AI in the Army context.* Paper presented at the Artificial Intelligence (AI) Working Group focusing on Human Systems Integration (HSI), Aberdeen, MD.

Parasuraman, R., Masalonis, A. J., & Hancock, P. A. (2000). Fuzzy signal detection theory: Basic postulates and formulas for analyzing human and machine performance. *HUMAN FACTORS, 42*(4), 636–659.

Peterson, W., Birdsall, T., & Fox, W. (1954). The theory of signal detectability. *Transactions of the IRE Professional Group on Information Theory, 4*(4), 171-212.

Pinelis, J. (2019). *Challenges in test and evaluation of AI: DoD's Project Maven*. Paper presented at the DATAWorks 2019, Springfield, Virginia.

Polk, T., & Newell, A. (1995). Deduction as verbal reasoning. *Psychological Review, 102*(3), 533-566.

Porter, D. J., Pinelis, Y. K., Bieber, C. M., Wojton, H. M., McAnally, M. O., & Freeman, L. J. (2018). *Operational testing of systems with autonomy*. Retrieved from

Preston, S. D., & de Waal, F. B. M. (2002). Empathy: Its ultimate and proximate bases. *Behavioral and Brain Sciences, 25*(1), 1-71.

Pretz, J. E., Naples, A. J., & Sternberg, R. J. (2003). Recognizing, defining, and representing problems. In J. E. Davidson & R. J. Sternberg (Eds.), *The Psychology of Problem Solving*: Cambridge University Press.

Qiu, S., Liu, Q., Zhou, S., & Wu, C. (2019). Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences, 9*(5), 909.

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *ArXiv e-prints*. Retrieved from https://arxiv.org/pdf/1811.12808.pdf

Robbins, C., & Steffen, M. R. (2018). The future of autonomous ground and surface systems testing. *The ITEA Journal of Test and Evaluation, 39*, 82-85.

Roelofs, R. (2019). *Measuring generalization and overfitting in machine learning.* (Doctor of Philosophy), University of California, Berkeley, Berkeley, CA. Retrieved from https://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-102.pdf (Technical Report No. UCB/EECS-2019-102)

Roske, V. P., Kohlberg, I., & Wagner, R. (2012). *Autonomous systems: Challenges to test and evaluation.* Paper presented at the National Defense Industrial Association Test & Evaluation Conference.

Russell, S. J., & Norvig, P. (2009). *Artificial intelligence: A modern approach* (3 ed.): Prentice Hall.

Santoni de Sio, F., & van den Hoven, J. (2018). Meaningful human control over autonomous systems: A philosophical account. *Frontiers in Robotics and AI, 5*. doi:10.3389/frobt.2018.00015

Sawers, P. (2019). Artificial stupidity: 'Move slow and fix things' could be the new mantra AI needs. *VentureBeat*. Retrieved from https://venturebeat.com/2019/10/05/artificial-stupidity-move-slow-and-fix-things-could-be-the-mantra-ai-needs/

Scharre, P., & Horowitz, M. C. (2015). *An introduction to autonomy in weapon systems*. Retrieved from

Schulte, A., & Donath, D. (2011, 2011//). *Measuring self-adaptive UAV operators' load-shedding strategies under high worload.* Paper presented at the Engineering Psychology and Cognitive Ergonomics, Berlin, Heidelberg.

Shapley, L. S. (1953). A value for n-person games. In H. W. Kuhn & A. W. Tucker (Eds.), *Contributions to the Theory of Games* (pp. 307-317): Princeton University Press.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., . . . Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), 484-489. doi:10.1038/nature16961

Simen, P., & Polk, T. (2010). A symbolic/subsymbolic interface protocol for cognitive modeling. *Log J IGPL, 18*(5), 705-761. doi:10.1093/jigpal/jzp046

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. *ArXiv e-prints*. Retrieved from https://arxiv.org/pdf/1312.6034.pdf

Simpson, B. (2019).

Singh, S., Lewis, R. L., Barto, A. G., & Sorg, J. (2010). Instrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development, 2*(2), 70-82. doi:10.1109/tamd.2010.2051031

Smith, J. D., Zakrzewski, A. C., Johnson, J. M., Valleau, J. C., & Church, B. A. (2016). Categorization: The view from animal cognition. *Behav Sci (Basel), 6*(2). doi:10.3390/bs6020012

Soni, D. (2019). Should you use machine learning? *Medium*. Retrieved from https://medium.com/better-programming/should-you-use-machine-learning-73a7746f7280

Sparrow, D. A., Tate, D. M., Biddle, J. C., Kaminski, N. J., & Madhavan, P. (2018). *Assessing the quality of decision-making by autonomous systems*. Retrieved from

Stanton, N. A. (2006). Hierarchical task analysis: developments, applications, and extensions. *Appl Ergon, 37*(1), 55-79. doi:10.1016/j.apergo.2005.06.003

Statt, N. (2019). OpenAI’s Dota 2 AI steamrolls world champion e-sports team with back-to-back victories. *The Verge*. Retrieved from https://www.theverge.com/2019/4/13/18309459/openai-five-dota-2-finals-ai-bot-competition-og-e-sports-the-international-champion

Steinberg, M. (2019). Explainable AI / Human Factors panel.

Sternberg, R. J. (1996). Costs of expertise. In *The road to excellence: The acquisition of expert performance in the arts and sciences, sports, and games.* (pp. 347-354). Hillsdale, NJ, US: Lawrence Erlbaum Associates, Inc.

Strathern, M. (1997). 'Improving ratings': Audit in the British University system. *European Review, 5*(3), 305-321. doi:10.1002/(SICI)1234-981X(199707)5:33.0.CO;2-4

Sun Tzu. (c. 450 BCE). The art of war. In.

Tanner, W. P., & Swets, J. A. (1954). A decision-making theory of visual detection. *Psychological Review, 61*(6), 401-409. doi:https://doi.org/10.1037/h0058700

Tate, D., & Sparrow, D. (2015). Lesson 2: How do autonomous capabilities affect T&E? *Automous Systems Test & Evaluation - CLE 002*: Defense Acquisition University.

Templeton, B. (2019). Tesla's "Shadow" testing offers a useful advantage on the biggest problem in robocars. *Forbes*. Retrieved from https://www.forbes.com/sites/bradtempleton/2019/04/29/teslas-shadow-testing-offers-a-useful-advantage-on-the-biggest-problem-in-robocars/#2349d1943c06

Thuloweit, K. (2019). Emerging Technologies CTF conducts first autonomous flight test. Retrieved from https://www.af.mil/News/Article-Display/Article/1778358/emerging-technologies-ctf-conducts-first-autonomous-flight-test/

Trent, S. (2019). *The Joint Artificial Intelligence Center: Transforming the DoD with human-centered technology*. Paper presented at the 63rd International Annual of the Human Factors & Ergonomics Society, Seattle, Washington.

US Department of Defense. (2011). *Unmanned Systems Integrated Roadmap FY2011-2036*.

US Department of Defense. (2015). *Department of Defense Instruction (DoDI) 5000.02: Operation of the defense acquisition system*.

US Department of Defense. (2019). *Summary of the 2018 Department of Defense Artificial Intelligence Strategy: Harnessing AI to Advance Our Security and Prosperity*.

Venkatesh, V., & Bala, H. (2008). Technology Acceptance Model 3 and a research agenda on interventions. *Decision Sciences, 39*(2), 273-315.

Vincent, J. (2018). OpenAI's DOTA 2 defeat is still a win for artificial intelligence. *The Verge*. Retrieved from https://www.theverge.com/2018/8/28/17787610/openai-dota-2-bots-ai-lost-international-reinforcement-learning

Visnevski, N. A., & Castillo-Effen, M. (2010). *Evolutionary computing for mission-based test and evaluation of unmanned autonomous systems*. Paper presented at the IEEE Aerospace Conference.

Wegener, J., & Bühler, O. (2004). *Evaluation of Different Fitness Functions for the Evolutionary Testing of an Autonomous Parking System.* Paper presented at the Genetic and Evolutionary Computation Conference.

Wickens, C. D. (2008). Multiple resources and mental workload. *HUMAN FACTORS, 50*(3), 449-455. doi:10.1518/001872008X288394

Wiggers, K. (2019). OpenAI's Dota 2 bot defeated 99.4% of players in public matches. *VentureBeat*. Retrieved from https://venturebeat.com/2019/04/22/openais-dota-2-bot-defeated-99-4-of-players-in-public-matches/

Wojton, H. M., Avery, K. M., Freeman, L. J., Parry, S. H., Whittier, G. S., Johnson, T. H., & Flack, A. C. (2019). *Handbook on statistical design & analysis techniques for modeling & simulation validation*. Retrieved from https://www.ida.org/-/media/feature/publications/h/ha/handbook-on-statistical-design-and-analysis/d-10455.ashx

Wojton, H. M., Porter, D. J., & Lane, S. T. (2020). Initial validation of the Trust of Automated Systems Test (TOAST). *Social Psychology*.

Wood, R. T., Upadhyaya, B. R., & Floyd, D. C. (2017). An autonomous control framework for advanced reactors. *Nuclear Engineering and Technology, 49*(5), 896-904. doi:10.1016/j.net.2017.07.001

Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A., & Le, Q. V. (2019). Adversarial examples improve image recognition. *arXiv preprint arXiv:1911.09665*.

Zacharias, G. L. (2019a). *Autonomous horizons: The way forward*. Maxwell AFB, Alabama: Air University Press.

Zacharias, G. L. (2019b). *Emerging technologies: Test and evaluation implications*. Paper presented at the DATAWorks 2019, Springfield, Virginia.

Zhou, Z., & Sun, L. (2019). Metamorphic testing of driverless cars. *Communications of the ACM, 63*(3), 61-67.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 05-2020	**2. REPORT TYPE** IDA Publication	**3. DATES COVERED** *(From - To)*

4. TITLE ANDSUBTITLE

Trustworthy Autonomy: A Roadmap to Assurance -- Part 1 System Effectiveness

5a. CONTRACT NUMBER
Separate Contract

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Daniel J. Porter (OED); Michael O. McAnally (N/A); Chad M. Bieber (N/A); Heather M. Wojton (OED);

5d. PROJECT NUMBER
C9082

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, Virginia 22311-1882

8. PERFORMING ORGANIZATION REPORT NUMBER

P-10768-NS
H 2019-000369

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, Virginia 22311-1882

10. SPONSOR/MONITOR'S ACRONYM(S)
IDA

11. SPONSOR/MONITOR'S REPORT NUMBER

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release. Distribution Unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Department of Defense (DoD) has invested significant effort over the past decade considering the role of artificial intelligence and autonomy in national security (e.g., Defense Science Board, 2012, 2016; Deputy Secretary of Defense, 2012; Endsley, 2015; Executive Order No. 13859, 2019; US Department of Defense, 2011, 2019; Zacharias, 2019a). However, these efforts were broadly scoped and only partially touched on how the DoD will certify the safety and performance of these systems. More recent work has done this big-picture thinking for the test and evaluation (T&E) community (e.g., Ahner & Parson, 2016; Haugh, Sparrow, & Tate, 2018; Porter et al., 2018; Sparrow, Tate, Biddle, Kaminski, & Madhavan, 2018; Zacharias, 2019b). In parallel, individual programs have been generating their own working-level solutions for their own particular use-cases and challenges. The framework proposed in the current work bridges the gap between the big picture policy recommendations already made and individual program needs. It is meant to serve as a roadmap framework that the T&E community can follow in order to provide evidence that artificial intelligence (AI)-enabled and autonomous systems function as intended. At times we echo broad policy recommendations made by others as they will also enable T&E activities. In other places we make more specific recommendations relating to test planning and analysis.

In this document, we present part one of our two-part roadmap. We discuss the challenges and possible solutions to assessing system effectiveness. A future part two will deal with test efficiency, simulation, and infrastructure.

Due to the scope of this project, even the main body of this document only provides a survey of the challenges and our proposed solutions. However, this roadmap serves as an outline to a future series of technical papers covering these topics in detail for working-level testers and analysts.

15. SUBJECT TERMS
Artificial Intelligence (AI);autonomous systems; autonomy; Machine Learning (ML); Military AI; T&E; Test & Evaluation; Test Methods; Test Strategy; test, evaluation, verification, and validation (TEV&V); TEVV

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Rebecca Medlin (OED)
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	Unlimited	119	**19b. TELEPHONE NUMBER** *(include area code)* (703) 845-6731