Using cognitive agents to design dynamic task allocation systems

Christopher R. Fisher (christopher.fisher.27.ctr@us.af.mil)
Cubic Defense
Beavercreek, OH 45324 USA

Mary E. Frame (mary.frame@parallaxresearch.org)
Parallax Advanced Research
4035 Colonel Glenn Hwy, Beavercreek, OH 45431

Christopher Stevens (christopher.stevens.28@us.af.mil)
Air Force Research Laboratory
Wright Patterson, OH USA

Abstract

Although cognitive models are primarily used to formalize theories of cognition, they could be applied in artificial intelligence (AI) systems, such as autonomous managers (AMs) which optimize team performance through dynamic task allocation. Cognitive models can be incorporated into the AM's decision system to understand the implications of alternative task distributions. They can also be used as simulated agents to stress test AMs under a wide range of conditions. In a simulation study, we varied the cognitive model used in the AM's decision system and the cognitive model performing a task to explore the design space of AMs. We found a trade-off between optimality and robustness in which complex models performed the best when assumptions were met, but were not robust to violation of assumptions. These results highlight the importance of considering simple models when assumptions could be violated and showcase the utility of cognitive models in AI systems.

Keywords: task allocation; cognitive agents;

Introduction

Cognitive models have a multitude of uses ranging from formalizing theories of cognition and sharpening research questions (McClelland, 2009), to measuring individual differences in cognition among clinical and non-clinical populations (Riefer, Knapp, Batchelder, Bamber, & Manifold, 2002; Yechiam, Busemeyer, Stout, & Bechara, 2005). One of the promises of cognitive architectures—and perhaps cognitive models more generally—is the ability to scale up to complex tasks where the limits of theory and practical application can be pushed (Newell, 1990). Some complex tasks in which cognitive models have been applied include training teams involving synthetic teammates (McNeese, Demir, Cooke, & Myers, 2018) and driving (Salvucci, 2006).

One burgeoning area in which cognitive models could be informative is artificial intelligence (AI). Deep learning in particular is a remarkably flexible function learning algorithm. According to the Universal Approximation Theorem, a deep neural network containing a sufficient number of layers can approximate any continuous function with sufficient training data (Zhou, 2020). However, this flexibility comes at a high cost: copious amounts of data are required for training in order to compensate for the lack of predefined structure. In addition, deep learning and similar approaches have been criticized for being opaque, brittle, and vulnerable to sabotage (Nguyen, Yosinski, & Clune, 2015). One solution is to incorporate scientific models into AI systems in order to provide a structure that reduces demanding data requirements. Currently, there are efforts to integrate neural networks with differential equation models commonly used in physics, biology and pharmacology (Rackauckas et al., 2020) to achieve a better balance between flexibility and data requirements. Along similar lines, cognitive models could be integrated into AI systems in situations that require interacting with or reasoning about humans.

We argue that cognitive models can be integrated into autonomous managers (AMs) designed to optimize performance in team-based work environments. An AM monitors performance of a team and dynamically allocates tasks between workers in order to improve performance of the team. An autonomous task manager can draw upon several sources of information and methods to inform a task allocation decision, including performance history of workers, AI, and mathematical optimization. Cognitive models can be integrated into an AM in at least two ways. First, an AM could use a cognitive model to predict and understand the implications of alternative task allocations. Second, a wide range of cognitive models can perform a simulated task environment in order to investigate the robustness of the design space of AMs.

We performed a series of simulations to explore the design space of AMs that incorporate different cognitive models into the decision process. Our primary research goal was to identify trade-offs between optimality and robustness in the design space. For example, do some designs perform optimally when model assumptions are satisfied, but perform poorly when assumptions are violated? In our simulations, an AM dynamically allocates sub-tasks of a larger, more complex task to cognitive agents (i.e. instances of a cognitive model that perform the task). We varied both the type of cognitive agent that performed the task and the cognitive model that the AM used to inform task allocation decisions. The cognitive models/agents varied according to cognitive processing constraints and the relationship between workload and performance. In some cases, the cognitive agent and the model used in the AM were the same; in other cases, they differed.

Overview

The remainder of the paper is organized as follows. First, we discuss the logic behind dynamic task allocation and the need for automated task allocation systems. Next, we describe the novel complex task environment used to test the effectiveness of AMs. Then, we describe the cognitive agents and AMs used in our simulation. Finally, we present and discuss the results of the simulation study. To preview a key result, we found a trade-off between optimality and robustness.

Dynamic Task Allocation

The performance of a team may vary according to numerous factors, including differences in skill specialization, sensitivity to workload, and temporal dynamics associated with fatigue. In some cases, team performance could be improved simply by allocating tasks to workers with the appropriate specialized skills. However, optimizing team performance is rarely this simple. For example, a person who skillfully performs two tasks in isolation may struggle to perform both together if she or he is sensitive to changes in workload level. Performance may also vary randomly from day to day due to unknown factors or could vary with fluctuations in task demands or fatigue. Dynamic task allocation is necessary in order to deal with uncertainty and to adapt to changes in performance across time. Collectively, these factors present a challenge for optimizing team performance.

Some work environments could benefit from an AM because it is difficult and costly for a human supervisor to manually monitor and allocate tasks. In addition, some research indicates that delegating work distribution decisions to workers can be disruptive due to the additional workload imposed by monitoring the performance oneself and others (Katidioti, Borst, van Vugt, & Taatgen, 2016; Won, Condon, Landon, Wang, & Hannon, 2011). One important research question that remains unanswered is how to design an AM that can adapt to dynamic situations and is robust to individual differences. For example, do more complex models provide a large performance gain compared to simpler models? Are more complex models less robust to violation of assumptions? We attempt to address these questions in the present research.

ISR-MATB

We developed a complex task environment called the Intelligence Surveillance and Reconnaissance Multi-Attribute Task Battery (ISR-MATB) to induce task demands similar to what is found in ISR operations. The ISR-MATB is a variation of the Multi-task Attribute Task Battery (Santiago-Espada, Myer, Latorella, & Comstock Jr, 2011) which was designed to emulate task demands in aviation. Whereas the MATB focuses on performing multiple tasks concurrently, the ISR-MATB focuses on goal switching, information search, interdependence between operational procedures, and synthesis of information into actionable decisions. The ISR-MATB uses variations of standard cognitive tasks to tap into each of these cognitive demands. Although each sub-task is relatively simple in isolation, they combine into a more complex whole due to inter-dependencies between sub-tasks. In what follows, we will describe each sub-task and then explain how they fit into an inter-dependent task flow.

Psychomotor Vigilance Task

The ISR-MATB uses a modified Psychomotor Vigilance Task (PVT) (Dinges & Powell, 1985) to emulate unpredictable changes in goals that may occur in ISR operations. The PVT is commonly used to measure fatigue and sustained attention. In a standard PVT, a millisecond counter appears after a uniformly distributed inter-stimulus interval of 2 to 10 seconds. Participants are instructed to respond as quickly as possible when the stimulus is presented. Upon responding, the millisecond counter stops and is displayed for 1 second as feedback. In the modified PVT, a target for the current trial is randomly selected from the stimulus set {grey Q, grey O, black Q, black O} and presented. As explained further below, the target is used to complete the visual and auditory search tasks.

Visual Search Task

The ISR-MATB uses the classic conjunctive visual search task (Treisman & Gelade, 1980) to emulate visually demanding search tasks in ISR, such as searching through a visually dense video feed for a target. The conjunctive search task requires participants to search for a target among an array of scattered distractors. Each stimulus has two dimensions: color and letter. A stimulus is considered a target if it matches on both dimensions (e.g. black Q). A distractor matches on one dimension but differs on the other (e.g. grey Q, black O). On half of the trials the target is present and on the other half of trials the target is absent.

Auditory Search Task

The ISR-MATB uses an auditory search task to emulate similar search tasks in ISR operations. For example, an operator might be required to search for keywords and phrases in communication channels and audio recordings where audio signals can be degraded or embedded in background noise. In the auditory search task, a participant is instructed to scan up to four radio channels containing background white noise for the search target (e.g. an audio recording of the words "black Q"). Difficulty is manipulated by changing the number of radio channels and the signal to noise ratio.

Decision Task

In ISR operations, information from multiple sources must be synthesized into a decision to act or refrain from taking action. We capture this aspect of ISR operations with the a multiple-cue decision task inspired by similar tasks in the literature (Sieck & Yates, 2001). As summarized in Table 1, decisions are based on two binary cues: (1) whether the target state (i.e. present or absent) is the same or different in the visual and auditory search sub-tasks, and (2) whether confidence in the accuracy of the information is low or high. For example, if confidence is low and the target is present in the visual and auditory search tasks, the correct action is to refrain. The base rate of cue values is 50-50, meaning it is not possible to perform better than chance with incomplete information.

Table 1: A decision matrix of four rules based on whether the visual and auditory target states are the same (present in both or absent in both) or different and the confidence in the information accuracy.

Confidence	Target State	Correct Action
Low	Same	Refrain
Low	Different	Act
High	Same	Act
High	Different	Refrain

Task Flow

The ISR-MATB features an inter-dependent task flow in which information must be acquired and integrated into a decision to act or refrain. At the beginning of each trial, the target must be acquired in the PVT before other sub-tasks can be performed. Once the target has been acquired, it is used to perform the visual and auditory search tasks. Similarly, the visual and auditory search tasks must be completed in order to ascertain the the first binary cue for the decision task. The second binary cue is acquired by clicking on a designated button, which reveals whether confidence in the information is low or high. Once the cues are acquired and the correct rule is retrieved (see Table 1), the participant can decide the correct course of action.

Cognitive Agents

As illustrated in Figure 1, we developed five types of cognitive agents with performance profiles that differ as a function of workload. Although the cognitive agents are not based on high-fidelity cognitive models, they must operate with realistic cognitive constraints on performance. Importantly, this set of cognitive models provides a wide range of performance patterns against which the AMs can be stress tested.

Before proceeding, we note some common notation and characteristics across agents: Define $\theta_{s,j}$ as the accuracy of agent j on sub-task $s \in S = \{p, v, a, d\}$, which corresponds to the PVT, the visual search task, the auditory search task and the decision task, respectively. All cognitive agents guess on the visual and auditory search task if no response is provided to the PVT, which we denote as $y_p = 0$.

Constant

As the name implies, the accuracy of the Constant agent does not vary according to workload level. However, performance can differ between sub-tasks. See (Frame, Lopez, & Boydstun, 2019a) for a similar approach. Each parameter value $\theta_{s,j}$ is randomly sampled from the following distribution: $\text{Uniform}(.50, 1)$. Once a parameter is selected, it is fixed for the duration of the simulation, making the expected accuracy constant.

Random-Dynamic

As a stress test for the AM, we developed a Random-Dynamic agent which changes on randomly selected sub-tasks after a set of 30 trials have been completed. Initial parameter values $\theta_{s,j} \sim \text{Uniform}(.50, 1)$. After a block of 30 trials has been completed, a new value for each accuracy $\theta_{s,j}$ is re-sampled with probability $p_{\text{change}} = .20$. Otherwise, the accuracy parameter remains the same for the next block.

Capacity-Limited

Performance of the Capacity-Limited Agent decreases as a function of workload—defined here simply as the number of tasks assigned to the cognitive agent. The probability of a correct response on sub-task s is given by the following piecewise equation:

$$\theta_{s,j} = \begin{cases} \dfrac{1}{1 + e^{-(\beta_{0,j} + \beta_{1,j} \times w)}} & \\ .5 & \text{if } y_p = 0 \text{ and } s \in \{v, a\} \end{cases}$$

where $\beta_{0,j}$ is the intercept and $\beta_{1,j}$ the slope for agent j, and w is workload level. The slope $\beta_{1,j}$ represents the sensitivity of accuracy to changes in workload, such that negative values of $\beta_{1,j}$ lead to a decrease in accuracy with increasing levels of workload. The second piece of the equation above indicates the model guesses on the visual and auditory search tasks if no response on the PVT is provided within the response deadline. Parameter values are initialized such that $\beta_{0,j} \sim \text{Uniform}(0, 3)$ and $\beta_{1,j} \sim \text{Uniform}(-1, 0)$, subject to the constraint that $\theta_{s,j} \geq .5$ under maximum workload to ensure that performance cannot drop below chance levels.

Yerkes-Dodson

Accuracy for the Yerkes-Dodson agent follows a parabolic (e.g. inverse U-shaped) relationship with workload in which optimal performance is achieved with moderate levels of workload. Some evidence indicates that the relationship between arousal and performance might be parabolic under some circumstances (Yerkes & Dodson, 1908). Following this logic, if low levels of workload induce boredom or mind-wandering, and high levels of workload are overwhelming, then optimal performance for an agent might be achieved at a moderate level of workload. We make one small modification to the Capacity-Limited agent to incorporate this assumption:

$$\theta_{s,j} = \begin{cases} \dfrac{1}{1 + e^{-(\beta_{0,j} + \beta_{1,j} \times (w - 2.5)^2)}} & \\ .5 & \text{if } y_p = 0 \text{ and } s \in \{v, a\} \end{cases}$$

The primary difference is that w is replaced with $(w - 2.5)^2$. Subtracting 2.5 from w places the maximum at the midpoint between one and four tasks and the exponent of 2 produces the parabolic relationship. Parameter values are sampled from the same distributions used for the Capacity-Limited agent.

Fatigue-Dynamic

The performance of the Fatigue-Dynamic agent is based on a dynamical system composed of two opposing processes. See (Patterson, Lochtefeld, Larson, & Christensen-Salem, 2019) for a related model. One process represents the gradual depletion of cognitive resources due to fatigue which is modulated by the instantaneous level of workload. An opposing recovery process replenishes the cognitive resource during periods of sufficiently low workload. If the net effect of the opposing processes is zero, the system achieves a state of equilibrium in which no change occurs. We approximate this dynamical process with the following logistic difference equation:

$$v_t = v_{t-1} + \Delta \times (\beta_{1,j} \times w_{t-1} + \beta_{2,j}) \times (v_{t-1} - v_{\min}) \times \left(1 - \frac{v_{t-1} - v_{\min}}{v_{\max,j} - v_{\min}}\right)$$

where $v_{\min}$ and $v_{\max,j}$ are the lower and upper asymptotes of accuracy, respectively, t indexes the time step, $\Delta = 1$ (seconds) is the change in time per time step, $\beta_{1,j}$ the slope the fatigue decrement, w_{t-1} is the workload level at time step $t-1$, and $\beta_{2,j}$ is the slope for the recovery process. Accuracy is defined as:

$$\theta_{s,j} = \begin{cases} v_t & \\ .5 & \text{if } y_p = 0 \text{ and } s \in \{v, a\} \end{cases}$$

At the beginning of each simulation, parameters were initialized as follows: $v_{\max,j} \sim \text{Uniform}(.85, 1)$ $\beta_{1,j} \sim \text{Uniform}(-.0015, -.0005)$, and $\beta_{2,j} \sim \text{Uniform}(|\beta_{1,j}|, 2 \times |\beta_{1,j}|)$. The purpose of constraining $\beta_{2,j}$ in terms of $\beta_{1,j}$ is to ensure that the neither the fatigue nor the recovery process are dominant at all levels of workload. We fixed $v_{\min} = .5$ to ensure that performance cannot fall below chance. We set initial accuracy to $v_0 = .9 \times v_{\max,j}$ under the assumption that initial accuracy is near the maximum.

Autonomous Managers

We developed five autonomous managers (AM) that base task allocation decisions on different cognitive models [1]. Each AM has the following in common: First, each AM learns the performance profile of each cognitive agent from observed data. Second, each AM monitors ongoing performance and may change the task allocation after each block of 10 trials. Third, unless otherwise noted, sub-tasks are randomly allocated to cognitive agents on the first block to avoid bias in initial conditions.

Recent-Maximum

The Recent Maximum AM takes a data-driven approach, using the most recent block of trials as the best estimate of an performance of a cognitive agent. In other words, it makes no assumptions about the relationship between workload and accuracy. After each block, the AM allocates the sub-task to the cognitive agent whose most recent block of data for that sub-task is the highest. In order to promote exploration in early blocks, the AM initializes each agent's performance history with a high accuracy of .90 for each sub-task.

[1] We did not include an AM that uses a Fatigue-Dynamic agent as a model because parameters could not be reliably estimated with optimizers freely available in Java.

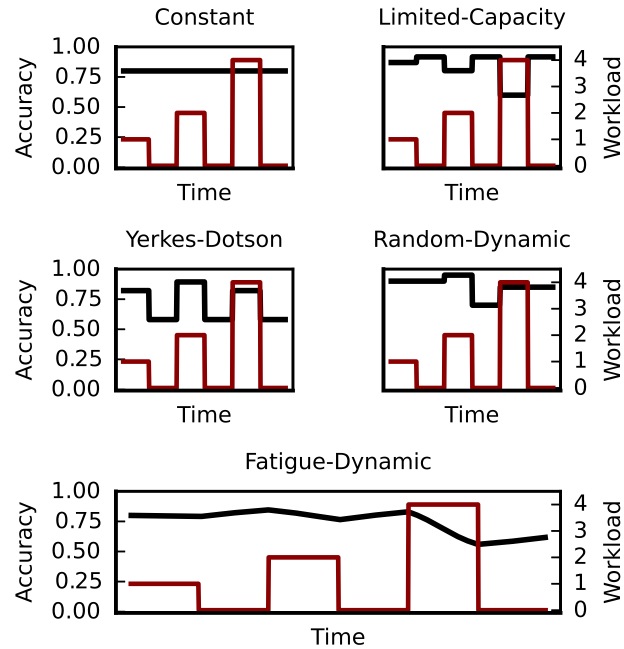

Figure 1: An illustration performance for each agent type as a function of workload. Black: agent performance on a single sub-task. Red: Agent workload.

Constant

The Constant AM assumes that the performance of each cognitive agent may differ by sub-task, but is otherwise constant across time and does not vary according to workload level. As such, the constant AM is similar to the Recent-Maximum AM, except it use all blocks of trials to estimate accuracy of each sub-task.

Capacity-Limited

The Capacity-Limited AM assumes that all cognitive agents are Capacity-Limited agents. After each block of trials, the Capacity-Limited AM estimates the parameters $\beta_{0,j}$ and $\beta_{1,j}$ from each agent's entire history of data. Using the maximum likelihood estimates, the Capacity-Limited AM iterates through all possible sub-task allocations and selects the allocation that maximizes expected accuracy on the decision task. Rather than allocating sub-tasks to agents randomly, the Capacity-Limited AM completes two exploration blocks to improve parameter estimation. During the first exploration block, one sub-task is allocated to the first agent, and three sub-tasks are allocated to the other agent. On the next exploration block, the sub-tasks are swapped between agents.

Yerkes-Dodson

The Yerkes-Dodson AM is identical to the Capacity-Limited AM, except it assumes the relationship between workload and accuracy is parabolic (e.g. inverted U-shape) rather than monotonically decreasing. After each block, the AM estimates the parameters for each cognitive agent. Using the best fitting parameter estimates, the AM selects the sub-task allocation that maximizes the accuracy of the decision task.

Random

As a point of reference, we include a Random AM, which randomly allocates sub-tasks to agents after each block. Thus, an AM is minimally successful if it performs better than the Random AM.

Simulation Design

We performed a set of simulations to assess the ability of different AMs to improve accuracy in the ISR-MATB by dynamically reallocating tasks to agents of different types. Each team consisted of two cognitive agents of the same type. We crossed cognitive agent type with each AM type to create a total of (cognitive agent type: 5) X (AM type: 5) = 25 simulation conditions.

All simulation conditions have several design parameters in common: First, the duration of each simulation was 60 minutes in simulated time. Second, each AM made a decision to reallocate the tasks among the cognitive agents after every block of 10 trials. Third, each simulation condition was repeated 500 times in order to approximate the expected accuracy for each AM.

Performance Evaluation

Our analysis focused on decision accuracy because that is the most important criterion in ISR operations. We normalized accuracy as a percentage of maximum possible performance for each simulation using the following formula:

$$a_{\text{norm}} = \frac{a_{\text{AM}} - a_{\text{min}}}{a_{\text{max}} - a_{\text{min}}}$$

a_{min} and, a_{max} are the minimum and maximum possible expected accuracy, and a_{AM} is the expected accuracy of the AM's allocation. Using a normalized accuracy metric has several benefits. First, it adjusts for differences in the range of possible performance, which varies according to the type of cognitive agent as well as the difference in performance between the cognitive agents in a team. Second, it allows one to identify whether further improvement is possible.

Results

The results of the simulation are summarized in Figure 2. In most cases, AMs performed better than chance (i.e. the Random AM). One exception to this finding is that AMs performed similar to chance for Fatigue-Dynamic agents. Another finding was that the greatest performance was achieved when the model of the AM matched the cognitive agent. For example, the best AM for Yerkes-Dodson agents was the Yerkes-Dodson AM (see first sub-plot in Figure 2). However, the advantage of using a model that matches the agent was not consistently large. Furthermore, complex models tended to be less robust when their assumptions were violated. For example, the Capacity-Limited and Yerkes-Dodson AMs performed poorly for Constant and Random-Dynamic agents. By contrast, the Constant AM, which uses a simple model of cognitive agent performance, was more robust across different types of cognitive agents. Although the Constant AM did not always achieve the best performance, it performed moderately well even when its internal model did not match the cognitive agents.

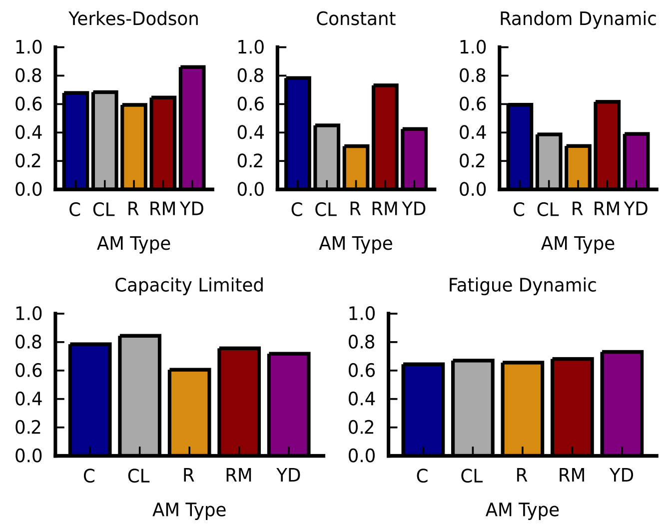

Figure 2: Sub-plots display % of maximum accuracy for each AM (colored bars) for a given agent type labeled in the subplot title. AM abbreviations: C: Constant, CL: Capacity-Limited, R: Random, RM: Recent-Maximum, YD: Yerkes-Dodson

Discussion

The goal of the present research was to explore how to integrate cognitive models into AMs to improve work productivity. AMs are designed to monitor performance of teams and dynamically allocate tasks to workers to improve performance. Cognitive models are an ideal candidate for augmenting the decision module of AMs because they can be used to predict the performance implications of alternative task distributions. Furthermore, the data requirements for most cognitive models are less onerous compared to deep neural networks and similar AI.

In our simulation study, we examined how AMs performed across a wide variety of conditions on a relatively complex ISR-themed task. We varied the type of cognitive agents that performed tasks and the internal model of the cognitive agent that the AM used to make task allocation decisions. One key finding is that AMs based on simple models were more robust compared to those based on more complex models. The relationship between complexity and robustness is due, in part, to a well-known statistical phenomenon called variance-bias trade-off (Brighton & Gigerenzer, 2015). Models with more parameters—an indicator of complexity—produce more error variance due to over-fitting. In fact, an AM with a sufficiently complex internal model may perform at chance levels even if the internal model matches the cognitive agent performing the task. In addition, more complex models might be more brittle due to the increased number of assumptions that could be wrong.

One direction for future research is to investigate the use of cognitive architectures, such as ACT-R (Anderson et al., 2004). In the present research, we used simpler cognitive models because they are tractable and generate a wide variety of distinct performance profiles. However, cognitive architectures provide the opportunity to explore additional interventions, such as providing feedback to strengthen declarative memory, or prescribing more effective strategies for task completion.

Conclusion

Cognitive models have a wide range of applications. The present research demonstrates how cognitive models can be incorporated into technologies and the design process to improve task performance.

Acknowledgments

The opinions expressed herein are solely those of the authors and do not necessarily represent the opinions of the United States Government, the U.S. Department of Defense, the U.S. Air Force, or any of their subsidiaries or employees. The contents have been reviewed and deemed Distribution A. Approved for public release. Case number: AFRL-2021-1002.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, *111*(4), 1036.

Brighton, H., & Gigerenzer, G. (2015). The bias bias. *Journal of Business Research*, *68*(8), 1772–1784.

Dinges, D. F., & Powell, J. W. (1985). Microcomputer analyses of performance on a portable, simple visual rt task during sustained operations. *Behavior research methods, instruments, & computers*, *17*(6), 652–655.

Frame, M., Lopez, J., & Boydstun, A. (2019a). *Dynamically managing task allocation between humans and machines in surveillance operations* (Tech. Rep.). Wright State Research Institute Beavercreek United States.

Katidioti, I., Borst, J. P., van Vugt, M. K., & Taatgen, N. A. (2016). Interrupt me: External interruptions are less disruptive than self-interruptions. *Computers in Human Behavior*, *63*, 906–915.

McClelland, J. L. (2009). The place of modeling in cognitive science. *Topics in Cognitive Science*, *1*(1), 11–38.

McNeese, N. J., Demir, M., Cooke, N. J., & Myers, C. (2018). Teaming with a synthetic teammate: Insights into human-autonomy teaming. *Human factors*, *60*(2), 262–273.

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

Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 427–436).

Patterson, R. E., Lochtefeld, D., Larson, K. G., & Christensen-Salem, A. (2019). Computational modeling of the effects of sleep deprivation on the vigilance decrement. *Human factors*, *61*(7), 1099–1111.

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., ... Edelman, A. (2020). Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*.

Riefer, D. M., Knapp, B. R., Batchelder, W. H., Bamber, D., & Manifold, V. (2002). Cognitive psychometrics: Assessing storage and retrieval deficits in special populations with multinomial processing tree models. *Psychological Assessment*, *14*(2), 184.

Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human factors*, *48*(2), 362–380.

Santiago-Espada, Y., Myer, R. R., Latorella, K. A., & Comstock Jr, J. R. (2011). The multi-attribute task battery ii (matb-ii) software for human performance and workload research: A user's guide.

Sieck, W. R., & Yates, J. F. (2001). Overconfidence Effects in Category Learning: A Comparison of Connectionist and Exemplar Memory Models. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *27*(4), 1003–1021. doi: 1O.1O37//O278-7393.27.4.1OO3

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, *12*(1), 97–136.

Won, J. C., Condon, G. R., Landon, B. R., Wang, A. R., & Hannon, D. J. (2011). Assessing team workload and situational awareness in an intelligence, surveillance, and reconnaissance (isr) simulation exercise. In *2011 ieee international multi-disciplinary conference on cognitive methods in situation awareness and decision support (cogsima)* (pp. 163–167).

Yechiam, E., Busemeyer, J. R., Stout, J. C., & Bechara, A. (2005). Using cognitive models to map relations between neuropsychological disorders and human decision-making deficits. *Psychological science*, *16*(12), 973–978.

Yerkes, R. M., & Dodson, J. D. (1908). The relation of strength of stimulus to rapidity of habit-formation. *Punishment: Issues and experiments*, 27–41.

Zhou, D.-X. (2020). Universality of deep convolutional neural networks. *Applied and computational harmonic analysis*, *48*(2), 787–794.