Development and Comparison of Operator Strategy Models

by

Haibei Zhu

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

_____
Mary (Missy) L. Cummings, Supervisor

_____
Miroslav Pajic

_____
Cynthia D. Rudin

_____
Martin A. Brooke

_____
Maria Gorlatova

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
Duke University

2021

ABSTRACT

Development and Comparison of Operator Strategy Models

by

Haibei Zhu

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

_____
Mary (Missy) L. Cummings, Supervisor

_____
Miroslav Pajic

_____
Cynthia D. Rudin

_____
Martin A. Brooke

_____
Maria Gorlatova

An abstract of a dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
Duke University

2021

Abstract

Human supervisory control (HSC), in which operators indirectly control autonomous systems by sending and receiving commands, is a commonly-used scheme for various human-automation interaction scenarios. While many studies have investigated how factors, such as different levels of autonomy and interface designs, affect operator performance in HSC scenarios, no previous research has quantitatively evaluated the impact of such factors on operator strategies. Thus, this research focuses on developing a quantitative metric to compare strategy models to determine whether changing specific factors in HSC scenarios would affect operator strategies.

Previous studies have shown that operator strategies can be represented by operator behavior patterns in conducting tasks and achieving goals. Given that hidden Markov models (HMMs) can represent operator strategies, researchers can investigate impacts from technology or process changes on operator strategies by comparing HMM strategy models. However, no quantitative and systematic HMM strategy model comparison metric has been proposed. To resolve this problem, this research uses the divergence distance measure to develop a mesh comparison metric to comprehensively compare strategy models and obtain quantitative model difference measures.

As a part of the comparison metric, the data quantity requirement for model development is determined using a large external dataset from a typical HSC video game. Strategy models were trained based on different data quantities and then compared to benchmark models developed from the whole dataset. Comparison results show that a minimum of 30 data sequences can represent the whole population and be effectively used to model operator strategies. Also, as another part of the metric, an observation alignment approach is proposed to compare strategy models developed from different HSC scenarios with non-equivalent training data elements.

Utilizing this comparison metric, researchers can quantitatively measure differences between strategy models. However, it is not clear how the magnitude of such comparison measures maps to meaningful degrees of difference in HSC scenarios. To address this issue, an initial baseline of strategy difference comparisons was established by comparing strategy models developed from human-subject experiment sessions. Then, a continuum of comparisons was generated to provide references for the magnitude of impacts from different factors on operator strategies. Thus, researchers can apply changes in HSC scenarios and evaluate the impacts from such changes on operator strategies by measuring differences between strategy models and referring to comparison baselines.

In summary, the contributions of this dissertation include 1) proposing an operator strategy model comparison metric to quantitatively measure differences between operator strategies modeled from HSC scenarios and 2) establishing strategy model comparison references across multiple HSC scenarios with varying settings.

Acknowledgements

It has been a wonderful journey for me during my Ph.D. study. I am tremendously grateful for all the support and help from many people along the way. I would like to express my sincere gratitude to all of them.

First and foremost, I would like to thank my advisor, Prof. Missy Cummings, for offering me such a great opportunity to join the Duke Humans and Autonomy Lab (HAL). I will be forever grateful for your continuous support and guidance throughout the long journey of my Ph.D. Thank you for all the insightful advice and encouragement you gave and your detailed editing of my documents. I have been incredibly fortunate to work with you, an excellent role model who inspired me to be a life-long learner.

I would like to thank my dissertation committee members, Prof. Miroslav Pajic, Prof. Cynthia Rudin, Prof. Martin Brooke, and Prof. Maria Gorlatova. Thank you for being willing to listen to my research progress and provide me great feedback. I would also like to thank Stacy Sabraw for proofreading my thesis and papers. To Sue Mathias, thank you for helping me rehearse my defense presentation. And to Honda Research Institute, thank you for offering me such a valuable internship opportunity.

I would like to thank all my colleagues from HAL and friends from Duke Robotics, including Dr. Alexander Stimpson, Dr. Michael Clamann, Dr. Lixiao Huang, Dr. Songpo Li, Dr. Vishwa Alaparthy, Dr. Victoria Nneji, Benjamin Bauchwitz, Anne French, Mahmoud Elfar, Xu Rong, Ziyao Wang, Guangshen Ma, Reza Lavaei, Yifan Zhu, and many others. It has been an incredible journey pursuing my Ph.D. with all of you. You made my graduate school experience such an enjoyable one.

Finally, I would like to thank my wife, Dr. Fan Wang, for always being there and putting my needs in front of hers. You have been with me through all the ups and downs I have encountered in this journey. You always encourage me and keep me motivated. I

would also like to thank my parents and other family members for their unconditional love and unending encouragement. I cannot express how grateful I am for my family's constant positive influence.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Human-automation interaction (HAI) studies how human operators interact with automation systems, such as robots and computer interfaces [1, 2, 3]. Specifically, HAI investigates the way humans control, receive information from, and are affected by automation systems [4]. With the development of automation technologies and artificial intelligence [5, 6, 7, 8, 9, 10, 11], humans now interact with automation systems with sophistication and on an increasing scale. While technology advances bring the benefits of increased efficiency and autonomy, achieving optimal outcomes with a highly complex system is not an easy task. Thus, an increasing number of studies and applications on human-automation interaction have emerged to address human-automation-related challenges and to improve the performance of human-in-the-loop systems [1, 2, 12, 13].

One of the most common human-automation interaction scenarios is driving a vehicle with advanced driver-assistance systems (ADAS). ADAS provides a group of functions in driving and parking for convenience and increased road safety [14, 15, 16]. In a scenario with low-level autonomous ADAS, the driver controls important control components, such as wheels and pedals, while ADAS performs auxiliary functions, such as cruise control and lane departure warning [17, 18, 19]. In a high-level autonomous ADAS scenario, a human operator is only asked to take control in case of an emergency [20, 21, 22]. As vehicle autonomy progresses, HAI studies on ADAS have also evolved quickly. While earlier work focused on the impact of traditional ADAS functionalities, such as collision warning on driver's attention [23], more recent work explores design and safety considerations of broader interactions between the drivers and the technology [24, 25, 26, 27].

Another common human-automation interaction scenario is air traffic control [28, 29, 30]. Increased air traffic in the last decades has created significant capacity and safety problems [31, 29, 32]. A series of solutions have been introduced to provide air traffic controllers with more accurate aircraft information and improve communications between pilots and controllers [33, 34, 35]. However, the introduction of such concepts brings a major paradigm shift in which autonomous technologies are allowed to perform safety-critical tasks [36, 37, 38]. Thus, a thorough investigation is required for autonomous system designs as well as human controllers' operations while interacting with such systems.

In human-automation interaction scenarios, a central role humans undertake is known as human supervisory control (HSC) [39, 40, 41]. HSC differs from the traditional interactions with tools of no intelligence, during which humans make all decisions and perform all sensing and control [42]. In HSC, humans remotely and indirectly interact with an autonomous system through receiving and sending commands to a control interface and complete high-level tasks through collaborative sensing and decision making [43]. The HSC control scheme is commonly viewed as a closed-loop process illustrated in Figure 1.1. In this loop, human operators first send control commands to the control interface of the system, which communicates to a computer, to initiate system interaction with real-world tasks. Feedback from the tasks will be collected through sensors and transferred back to the system. The operators will then receive such feedback via displays to infer the state of the system and make decisions regarding further control actions.

Human supervisory control has been widely used in many applications, including remote drone control, driver and ADAS interaction, and air traffic control [44, 45, 46]. However, many challenges and drawbacks of HSC have been observed over the years [39, 47]. Among the top challenges are information overload [48], decision biases [40], and tampering with the role of automation [49]. To address those challenges, numerous studies have emerged to investigate how humans interact with autonomous systems to improve human operators' performance while interacting with such systems [50, 51, 52]. Previous

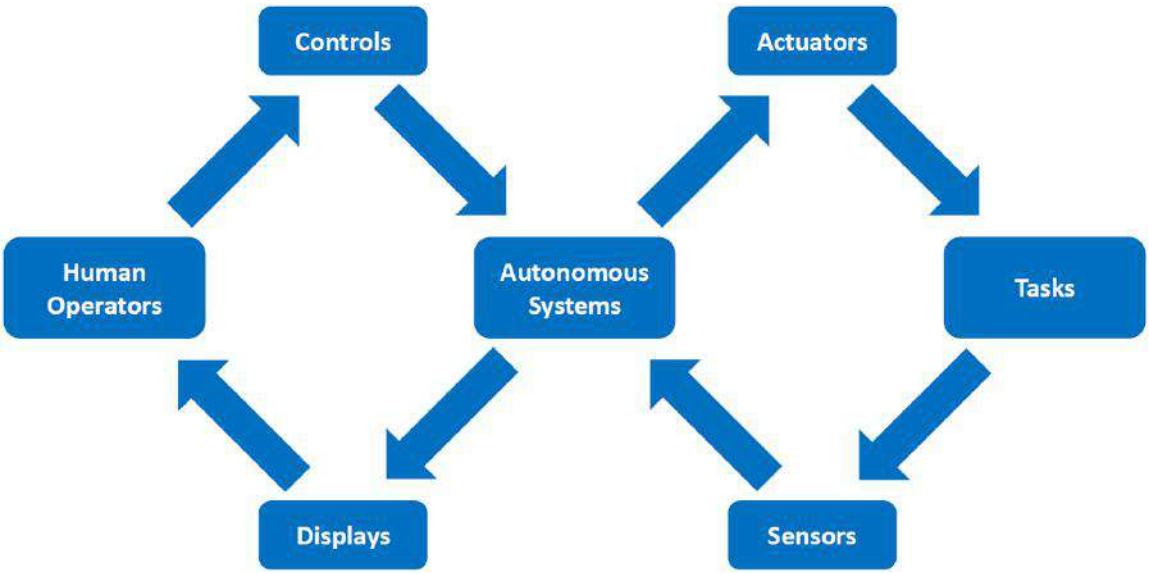


Figure 1.1: Human supervisory control loop.

literature commonly focused on one or a few factors, such as task load [53], autonomy level [54], and interface design [55], and proceed to investigate how those factors affect operators' performance as well as the risk associated with changing those factors.

One weakness we identified in the previous study is that they focused narrowly on how changes in HSC scenarios influence operators' performance but overlooked impacts from such changes on operators' underlying decision processes and behavior patterns that result in those performance changes. Thus, other than directly measuring operators' performance, we argue that it is also important to study how HSC scenario changes impact their decision processes and behavior patterns.

We considered operators' decision processes in determining actions and behavior patterns in conducting high-level tasks as operator strategies [56]. Both operator performance and strategies are important aspects in HSC scenarios [40]. In such scenarios, operators first make up their strategies by setting goals with different priorities, determining actions to achieve those goals, and allocating attention to different actions. After that, they perform various operations based on their strategies to complete tasks and achieve the final goal.

To effectively incorporate operator strategies into the evaluation of HSC scenarios, operator strategies need to be modeled, and the model should be measurable or quantitatively comparable. With quantitative measures of changes in operator strategies, researchers can evaluate the impact of changing factors on operator strategies. For example, if researchers need to examine whether a new function on a drone supervisory control interface can reduce inefficient decision making from its operators, researchers can compare operator strategy models before and after applying such a function, and measure the quantitative difference between strategies to estimate the impact of the new change.

However, no previous research has focused on quantitatively comparing operator strategies in HSC scenarios. This question, therefore, arises as to how to investigate and compare operator strategies to obtain meaningful strategy difference measures. Such a question leads to the objectives and research questions in this dissertation.

1.2 Research Questions

The gap in current human-automation interaction and human supervisory control research topics - comparisons of operator strategies in HSC scenarios with various settings - leads to the objectives of this research: 1) describe operator strategies by building operator strategy models, 2) develop a strategy comparison metric to quantitatively measure differences between strategy models, and 3) explain practical meanings of strategy difference measures by establishing comparison references. Thus, these research objectives lead to the following two research questions:

- How can operator strategies be quantitatively compared, or how are differences between operator strategies quantitatively measured, given strategy models?
- What are the practical meanings of the magnitudes of strategy model differences, or how would changes in HSC scenarios affect operator strategies at different levels?

Based on these research questions, the contributions of this research include 1) develop a systematic operator strategy comparison metric, which can quantitatively and comprehensively measure differences between operator strategies modeled from various HSC scenarios, and 2) establish a continuum of operator strategy comparisons to provide the references of strategy model difference magnitudes and practical meanings of operator strategy differences.

1.3 Thesis Organization

In order to address the research questions mentioned above, this thesis has been organized into the following chapters:

- Chapter 1, *Introduction*: This chapter first introduces the human-automation interaction and human supervisory control scenarios, and then presents the main gap in current research. Also, Chapter 1 presents the research questions with the contributions of this dissertation.

- Chapter 2, *Background*: This chapter provides more background information for the human supervisory control with related research topics. Also, this chapter presents existing methods of describing operator strategies in HSC scenarios, and specifically explains how to utilize hidden Markov models (HMMs) to model operator strategies.

- Chapter 3, *Operator Strategy Model Comparison Metric*: This chapter describes the operator strategy model comparison metric. It starts by illustrating the construction of divergence meshes, then presents two important components of the metric, the data quantity requirement and the observation alignment approach.

- Chapter 4, *Quantitative Operator Strategy Comparison References*: By utilizing the operator strategy model comparison metric, this chapter presents a continuum of

strategy model comparisons across multiple HSC scenarios with different settings to provide baselines and references of strategy model comparisons. It also explains the practical meanings of these references. Thus, researchers can refer to such baselines and references to evaluate the magnitudes of impacts from specific changes in HSC scenarios on operator strategies.

- Chapter 5, *Conclusion*: This chapter summarizes the two main contributions presented in Chapters 3 and 4. It also discusses the potential applications and the limitations in the comparison metric and comparison references. Finally, future research directions are also discussed.

Chapter 2

Background

This chapter surveys previous work on human supervisory control (HSC) and operator strategy modeling techniques in HSC scenarios. It starts with a review of human supervisory control theory and presents related studies in describing, modeling, and comparing operator strategies in HSC scenarios. The chapter later introduces hidden Markov models (HMM) as a promising model for operator strategies. The HMM strategy model development process is discussed in detail, followed by an example of such development for a simulation-based drone control HSC scenario.

2.1 Human Supervisory Control

With the introduction of an automation system that can perform sensing and computation automatically, such as a robot, the role of a human in a human-automation interaction scenario changes from that of a direct in-the-loop controller to a position of a supervisory controller. Human supervisory control (HSC) is a higher-level knowledge-based control scheme, in which human operators interact with the intermediary of the autonomous systems through planning, monitoring, and intervening to drive the end effector of the automation system towards the desired state [57, 39, 43]. An HSC scenario operates in a closed-loop fashion as illustrated in Figure 1.1. In such a loop, the operator sends control commands to a computer to remotely control real-world tasks and receive feedback collected by sensors and displayed on interfaces. Operators in HSC scenarios are mainly responsible for making higher-level decisions in conjunction with supportive autonomy.

HSC are widely used in transportation, energy, and defense realms [39, 58]. HSC studies usually target a particular application or a system to improve reliability, efficiency,

and reduce accidents for operators to control them. For example, Lind et al. proposed a Multilevel Flow Modeling (MFM) and applied the model for designing displays for supervisory control of industrial plants [59]. Farjadian et al. presented a supervisory control architecture for pilot-autopilot collaboration to ensure resilient tracking performance in the presence of anomalies [60], and many others [53, 54, 55]. Among the many HSC applications, drone or Unmanned Aerial Vehicle (UAV) control is an emerging scenario that attracts a lot of research attention. With a small footprint and easy deployment, drones have many potential applications in surveillance, research, and rescue [43, 61, 62, 63].

Many challenges and problems still exist in HSC scenarios [47], and the root causes trace back to both human operators and autonomous control systems. Some of the major challenges include decision biases [40], trust and reliability [64], roles of automation and multi-modal technologies [41]. Also, many factors can influence operator performances, including individual differences, interface designs, and varying levels of autonomy [40, 47, 65]. Thus, many studies have been conducted for solving such issues by investigating how various factors in HSC scenarios may affect operator performances [66, 58, 53].

Both operator performance and strategies are important aspects when evaluating an HSC scenario [40]. During interactions with complex systems, such as drones, human operator forms complex plans regarding work methodologies with the underlying autonomy. In this process, operators develop strategies in terms of which, when, and how resources should be utilized, in other words, they assign different priorities and allocate their attention to different tasks, and determine actions to interact with control systems before executing the task [63, 67, 68]. However, the importance of human strategy modeling has been largely overlooked in previous studies. Instead, this research focuses on operator strategies and studies how changes in HSC scenarios will affect operator strategies. Thus, it is important to understand and describe operator strategies as they illustrate fundamental behavior patterns how an operator conducts a certain task [56]. Unlike operators' performance, which can be directly measured by task success rates and time consumption,

operator strategies cannot be directly observed, so that modeling techniques are necessary for studying operators.

2.1.1 Techniques for Modeling Operators

To date, many operator modeling techniques have been proposed at various abstraction levels for the different subject of study [69, 70, 56, 71, 72, 73]. Summarized in Figure 2.1, operator modeling techniques can be categorized into low, middle, and high levels. Low-level models focus on estimating operator cognitive processes and describing operator cognitive structures. By using low-level models such as ACT-R [69] and GOMS [70], researchers can describe how information is defined and processed by operators and predict operator actions. However, these models cannot present operator behavior patterns and, thus, have limited access to operator strategies in conducting tasks.

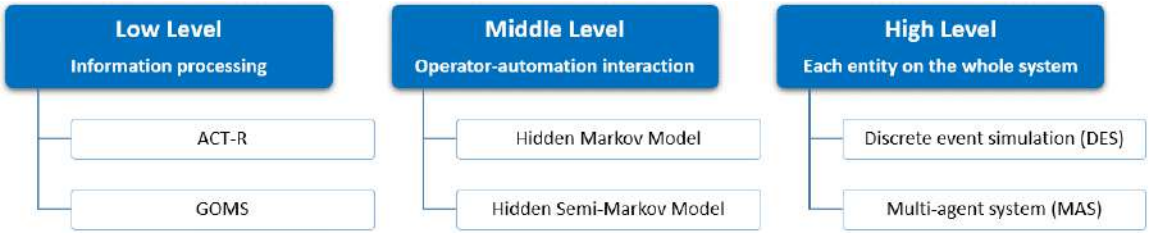


Figure 2.1: Different levels (architectures) of human operator models.

Middle-level models focus on the relations and cooperation between operators and autonomous systems. Models, such as the hidden Markov model (HMM) [56] and the hidden Semi-Markov model (HSMM) [71], can benefit from their structures to reveal both operator behavior patterns and operations. High-level models, including discrete event simulation (DES) [72] and agent-based modeling (ABM) [73], focus on assessing effects from each entity as well as subsystems in the whole system and investigating interactions between all individuals in large-scale applications. Thus, middle-level operator modeling techniques are utilized in this research to model operator strategies because they can

describe operator control strategies in interacting with autonomous systems.

As shown in Figure 2.1, both HMM and HSMM are middle-level models, which have been widely used for describing operator strategies and behavior patterns [56, 71, 61, 74, 75]. The HMM is a two-layer stochastic model that describes a Markov process with a higher layer of hidden system states and a lower layer of observable emissions from each hidden state [76]. The hidden semi Markov model (HSMM) has a similar structure, but the HSMM is based on a semi-Markov process instead of a Markov process. While state transitions in an HMM only depend on the geometric probability values, transitions in the HSMM depend on both transition probabilities and state durations. An HSMM is generally more complicated than an HMM developed from the same dataset, because the model training process for an HSMM requires a time component. Because of the high complexity and the time component in HSMM models, they can be utilized as predictive models for time-critical applications [77, 78].

However, because the HSMM has a more complex model structure than the HMM, even though they share the same number of hidden states, the HSMM requires more data for model training. More importantly, the time of state transitions is not an important aspect and is not considered in this research. Thus, the HMM is usually preferred as descriptive models for operator strategies in HSC scenarios [56, 79, 80, 62, 81]. In this case, the HMM observations represent the interactions between operators and autonomous systems. The HMM hidden states, which are weighted combinations of certain observations, and transition probabilities among these states illustrate operator strategies.

2.1.2 Operator Strategy Models

Operator strategies can be further described as the organization of operators' tactics used for performing tasks with different priorities that forms missions and goals [82]. To precisely and effectively describe operator strategies is the first major step in this research.

Understanding that this research focuses on quantitatively evaluating impacts from different changes in HSC scenarios on operator strategies, a requirement for operator strategy models is to have a mathematical representation so that differences between models can be measured quantitatively. Further details of measuring strategy model differences are illustrated in the next chapter.

The previous subsection introduces two common middle-level models, the HMM and the HSMM, for describing operator strategies. Other than these two models, researchers have also proposed other models for describing operator strategies and behavior patterns [82, 83, 84, 85, 86, 87, 88, 89]. Gindele et al. utilized a partially observable Markov decision process (POMDP) model to estimate driver strategies, especially the decision making process, in driving scenarios with advanced assistance systems [83]. Authors considered traffic environment elements, such as traffic participants and lanes, in their models to predict drivers' actions and future poses of traffic participants. However, environmental elements are not considered in our research, and we focus on diagnostic model structures instead of model predictive abilities in order to measure model differences. Thus, while the POMDP is an ideal model for estimating driver actions, it has limited capability to describe driving strategies.

Pentland et al. proposed to use a set of dynamic models, such as Kalman filters, sequenced by a Markov chain to represent human strategies [86, 90]. Authors first separated human strategies into two levels, small-scale and large-scale structures of human behaviors. Then they utilized Kalman filters to represent small-scale behaviors and a Markov chain to represent large-scale behaviors by connecting these Kalman filters. Similar to the HMM model structure, this framework also considered human strategies as a two-layer structure. However, the structure of this framework was predefined by the authors rather than trained using data-driven methods. Thus, the model framework proposed in this work can only describe operator strategies in specific scenarios and cannot be generalized to other HSC scenarios with different settings.

Another study conducted by Frias-Martinez et al. reviewed various soft computing techniques, such as fuzzy logic, neural networks, and genetic algorithms, for modeling human behavior patterns [87]. Soft computing is considered as computational techniques and models with tolerance of impression, uncertainty, and partial truth in authors' work. Authors presented several applications of using typical soft computing models. For example, fuzzy logic can be used to mimic the process of human decision-making and infer goals and plans, neural networks can be used in more complex scenarios with more human and system inputs. However, the structures of these models are not directly interpretable and explainable, and these models are usually used for prediction tasks. Thus, among such quantitative models for modeling human strategies, the HMM is selected as strategy models in this dissertation because of its high interpretability.

2.1.3 Operator Strategy Comparisons

Given techniques and methods for describing operator strategies, researchers can investigate strategy differences by comparing resulting strategy representations. However, comparison methods can be different based on corresponding strategy modeling techniques. While only limited previous work focused on studying differences in operator strategies and behavior patterns [91, 92, 93, 94], these efforts inspire us to develop an operator strategy comparison metric for HSC scenarios.

Previous research conducted by Haas et al. analyzed differences in individuals' strategies by studying external factors such as social, cultural, and institutional factors [91]. This work considered individuals' strategies as the combination of their behaviors, recognition of hazards, and decision-making processes. The authors interviewed participants with two occupations, including motorcyclists and mineworkers, with detailed questions about their interactions with risky environments. Then, the authors formalized participants' strategies by parsing their answers based on external factors and referring to codebooks. As a result,

the authors qualitatively described individuals' strategies in interacting with risky environments and compared them to infer how external factors affect strategies in different environments. While such a qualitative strategy comparison method can provide researchers preliminary insights about the impacts from various factors, it cannot quantitatively measure differences between strategies and evaluate such impacts.

Markkula et al. reviewed different drivers' strategies in on-road collision situations and qualitatively compared these strategies for validating those strategy models [92]. First, the authors pre-defined five categories for drivers' near-collision strategies. Then, they implemented and tested reviewed models on a driving scenario to investigate the effectiveness of such models and the similarity between them. The authors concluded that those models representing the pre-defined strategies were similar because they were capable of predicting drivers' actions during collision avoidance with similar success rates. While the pre-defined strategy categories may shed light on drivers' behavior states and characteristics in collision situations, these categories were subjectively defined and selective. Thus, objective strategy description and comparison methods are preferred to reduce subjectivity in investigating similarities among operator strategies.

Research conducted by Traulsen et al. studied how individuals' strategies update in evolutionary games [93] by comparing strategies across different phases of the game. In this work, the authors considered individuals' actions as strategies and used individuals' performance to evaluate strategies during the gaming process directly. Given the player performance comparison results, which represent player strategy comparisons, the authors found that the probability of player switching strategies increased as the payoffs from the game vary in the later phase of the game. This topic is closely related to presenting different strategies in evolutionary scenarios or different phases of interacting with autonomous systems. However, utilizing individuals' performance to evaluate and compare their strategies is too high-level. Such a method cannot provide researchers insights about why one strategy is better or more effective than another.

These previous studies demonstrate different methods of comparing individuals' strategies and behavior patterns formalized by various models and representations. While qualitative strategy description and comparison methods are useful for exploratory purposes such as examining factors, they cannot provide mathematical representations for operator strategies and numerically measure strategy differences. So, researchers cannot quantitatively evaluate impacts in HSC scenarios using qualitative methods. Similarly, subjective methods are not preferred because researchers' subjectivity can introduce individuals' biases, which can cause distorted strategy modeling and comparison results. Also, the strategy model properties should be considered in strategy comparisons since these properties describe operators' behavior states and decision processes. Therefore, a quantitative model comparison method is needed to measure differences among strategy models. Researchers can objectively investigate and evaluate how changes in HSC scenarios impact operator strategies with such quantitative measures. Further details of the model comparison method are illustrated in the next chapter.

2.2 Hidden Markov Model (HMM)

The previous section presents the basic concept of human supervisory control scenarios and existing techniques for modeling human operators at different levels. Also, the previous section justifies the use of hidden Markov models for modeling operator strategies in this research. This section starts with the introduction of the hidden Markov model structure and how we can use HMMs to model operator strategies. Then this section illustrates the HMM model development and selection process.

2.2.1 HMM Structures and Applications

The hidden Markov model is a stochastic model that describes a Markov process with some states and variables that are not observable [77, 95]. In a Markov model, all system variables, including system states and state transitions, are observable. However, in a hidden Markov model, system states are not directly observable, and the only observable variables are emissions and emission probabilities that are determined by hidden system states. Figure 2.2 illustrates a hidden Markov model with four observations, including O_1, O_2, O_3 and O_4, and three hidden states, S_1, S_2 and S_3. Each hidden state S can be considered as a weighted cluster of all observations, and all weights should equal 100%. HMM models are used to represent both higher-level human operator behavior states and lower-level operator interactions with human supervisory control systems in order to visualize the overall structure of operator strategies.

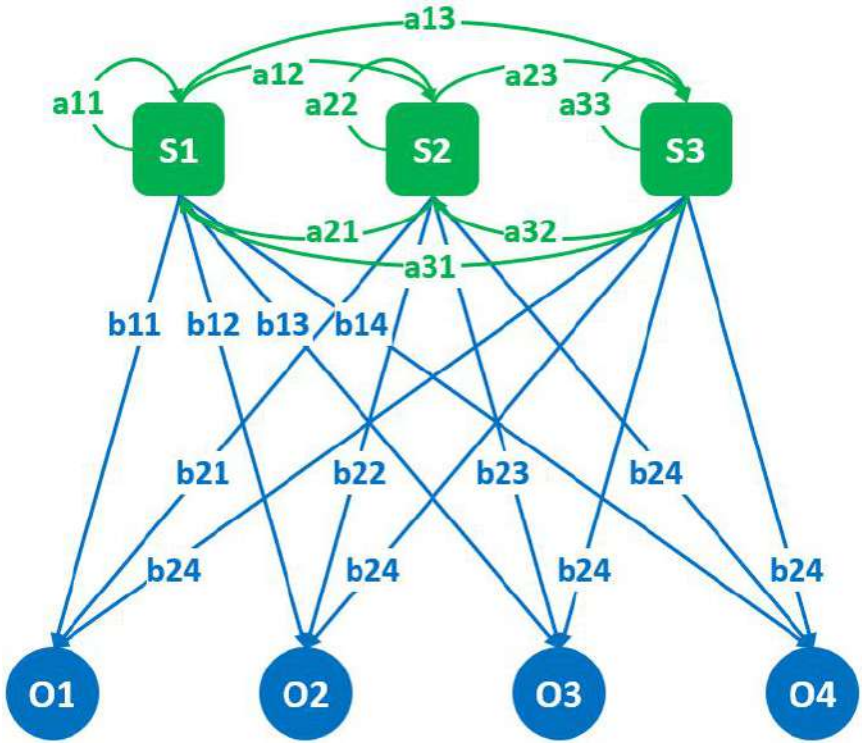


Figure 2.2: An example of a hidden Markov model structure.

The HMM structure shown in Figure 2.2 can be formally described as a tuple [77, 96], $\lambda = \{S, O, A, B\}$, in which, $S = \{S_1, S_2, \cdots, S_N\}$ represents N different hidden states, and $O = \{O_1, O_2, \cdots, O_M\}$ represents M different observation types. $A = \{a_{ij}\}$ is a $N \times N$ transition probability matrix, in which $a_{ij} = P\{S_j^{t+1} | S_i^t\}$, and $B = \{b_{ik}\}$ is a $N \times M$ emission matrix, in which $b_{ik} = P\{O_k | S_i\}$, $i, j \in [1, N]$, $k \in [1, M]$. The transition and emission matrices connect all hidden states and observations of an HMM. The tuple $\lambda = \{S, O, A, B\}$ can be illustrated as:

$$S = \{s_1, s_2, \cdots, s_N\}\ O = \{o_1, o_2, \cdots, o_M\}$$

$$A = \begin{bmatrix} a_1 & \cdots & a_{1N} \\ \cdots & \cdots & \cdots \\ a_N & \cdots & a_{NN} \end{bmatrix} B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1M} \\ b_{21} & b_{22} & \cdots & b_{2M} \\ \cdots & \cdots & \cdots & \cdots \\ b_{N1} & b_{N2} & \cdots & b_{NM} \end{bmatrix}$$

The HMM is widely used in the machine learning field, especially in speech recognition [76], image and video segmentation [97], biomedical image pattern recognition [98], and development of human operator behavior models [99, 79, 56]. Rabiner published a milestone study in HMM theory and applications, which illustrated the mathematical details of HMMs and provided many typical HMM structures [76]. He also investigated the HMM model developing process, including model training and model selection algorithms. Suzuki et al., developed a human driving behavior model using HMM to study driver collision avoidance behaviors [99]. In their work, they treated human driving behavior states as multiple simplified linear models, instead of specific cognitive representations, to collect experiment data more directly and to understand the model better from a mathematical perspective. The HMM is an appropriate quantitative model for this research because the HMM can efficiently capture information about higher-level human operator behavior states and transition probabilities among such states from lower-level observations to represent operator strategies in HSC scenarios.

2.2.2 HMM Model Training Process

As described in the previous subsection, researchers can utilize hidden Markov models to model operator strategies. As shown in Figure 2.3, the observable actions from operators can be represented by the observation layer in an HMM, and operator control strategies can be represented by the hidden states and transitions among them.

The first important factor that impacts HMM structure is the model training method. An HMM model can be trained via different methods, including supervised, smooth supervised, and unsupervised learning [79]. Comparing these training methods, the results showed that the unsupervised HMM model training provided the model with the highest model likelihood [56]. Understanding that a higher model likelihood value represents a model that is more likely to fit the data points, the unsupervised HMM model training is preferred in this research. The unsupervised model training is a data-driven method, which is also an expectation maximization (EM) algorithm, so that it can provide subjective strategy modeling results.

Given that the unsupervised learning approach is preferred, the multi-sequence Baum-Welch algorithm is utilized for training hidden Markov models [56]. Considering that such an HMM training process requires various hidden state numbers as initialization, the Bayesian information criterion (BIC), which is a widely-used model selection criterion, is utilized as the model selection for determining the number of hidden states for an HMM [100]. To avoid overfitting caused by adding parameters, the BIC penalizes the number of free parameters based on the model likelihood value. The calculation of BIC values can be represented as:

$$BIC = -2(logL) + numParam \times log(numObs) \tag{2.1}$$

Here $logL$ represents the log-likelihood value of models, *numParam* represents the number of free parameters, and *numObs* represents the number of observations in the training process. In model selection, HMM models with lower BIC values are generally preferred.

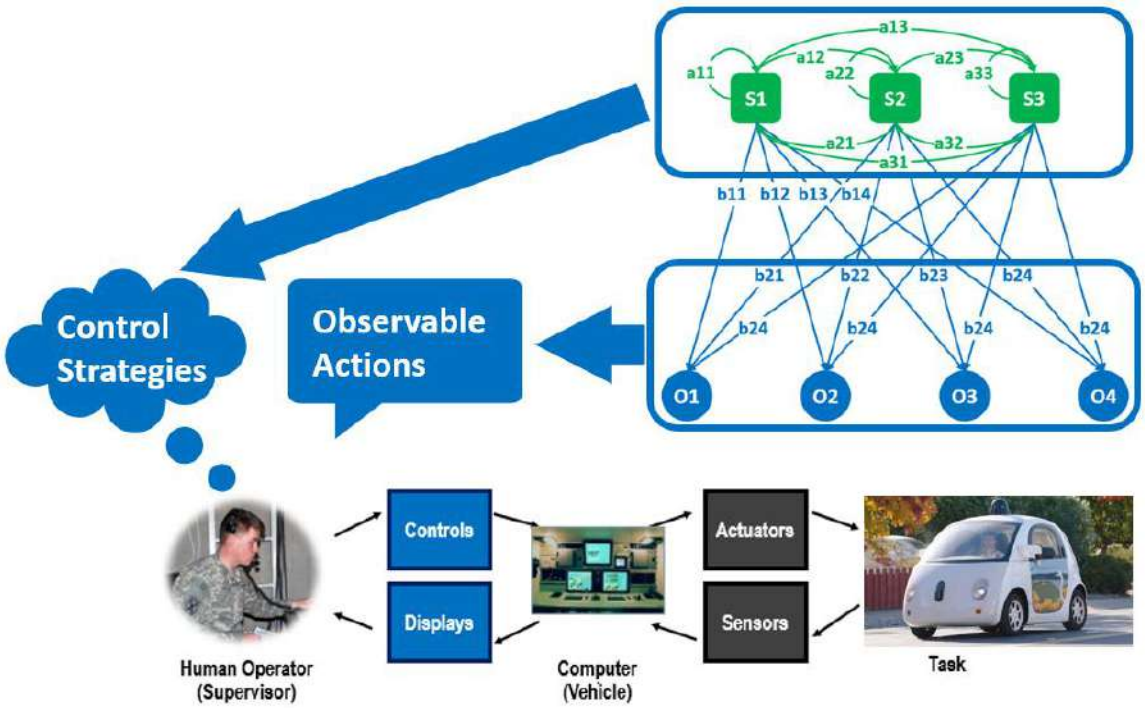


Figure 2.3: The connection between hidden Markov models and human supervisory control scenarios.

Another important factor in the model development process is the first order Markov assumption. Evaluating the first order Markov assumption of the memorylessness property is critical for applying HMM to develop operator strategy models. In human supervisory control scenarios, the first order Markov assumption may not hold that the current higher-level behavior state in operator strategy processes may only depend on the previous state, or depend on several previous states [101]. The Bayesian information criterion (BIC) can also be used for evaluating different Markov independent orders.

The difference in applying the first-order or higher-order Markov assumption in the development of HMMs for operator strategy models was studied in Boussemart's work by comparing different models with different Markov independent orders [102]. In this study, HMM models were developed based on certain human supervisory UAV control scenarios, and BIC values of HMM models were compared to select a preferred Markov order. As a

result, the HMM models with a second or third Markov independent order had significantly higher BIC values than HMM models with the first-order Markov property. This finding indicates that additional information captured in higher-order HMM models cannot balance out the significant increase in the model complexity. Thus, the first-order HMM models are preferred in the development of operator strategy models in human supervisory control scenarios.

Specifically focusing on the BIC method, BIC curves generated from HMM training processes may monotonically decrease or converge to certain values with the increase of the hidden state number. The model with the most states, which may have the lowest BIC value, may not be the most appropriate model since it cannot provide valuable implications about the clustered behavior states. To maintain the simplicity and interpretability of descriptive models, the number of rate states (NRS) was applied to assist the BIC with model selection by monitoring all rare states whose occurrence frequency was lower than a threshold, which is usually 5% [74].

2.3 An Example of Strategy Model Development

Given the introduction of human supervisory control in the first section and the utilization of hidden Markov models as operator strategy models in the second section, this section presents an example of the strategy model development process in an HSC scenario, which is a simulation-based human-subject experiment. Further details of this experiment are in Appendices A and B.

2.3.1 The HSC Scenario for Data Collection

The data used for developing strategy models was collected from a human-subject experiment conducted on the Security-Aware Research Environment for Supervisory Control of

Heterogeneous Unmanned Vehicles (RESCHU-SA) platform [103], which is an extension of the original RESCHU platform [104]. RESCHU-SA is a Java-based simulation platform for a single-operator with multi-UAV supervisory control scenarios. It provides the flexibility to design multi-tasking scenarios including both navigational and imagery analysis tasks. Moreover, this platform provides capability for simulating UAV GPS spoofing attacks, in which hacked UAVs deviate from the originally assigned paths and destinations, along with pop-up notifications that simulate autonomous GPS spoofing detection systems.

The primary objectives of operators using RESCHU-SA were to control multiple UAVs to 1) determine whether UAVs were under GPS spoofing attacks, 2) perform imagery analysis tasks of counting road intersections when UAVs reached targets, and 3) ensure that UAVs did not encounter hazard areas. While statistical analysis results of this experiment provide high-level understanding of the factors that impacted operator performance, we still need to further investigate the underlying nature of why such factors had certain effects on performance. Also, operator hacking detection strategies could not be inferred via statistical results. Thus, operator strategy models are needed for further describing operator behavior patterns and detection strategies in such UAV supervisory control scenarios.

2.3.2 General Strategy Model

The first step of the HMM training process for the general strategy model, which describes operator general strategies across experimental sessions, is observation selection. In RESCHU-SA, every key stroke and mouse action was recorded, along with the system status. In an HMM, the hidden higher-level behavior states are clusters of operator actions, so the interaction data should be aggregations of observations based on a pre-defined grammar. In this manner, there were 12 possible places for operators to click in RESCHU-SA, which yielded 12 observations, as presented in Table 2.1.

The multi-sequence Baum-Welch algorithm, an unsupervised model training method

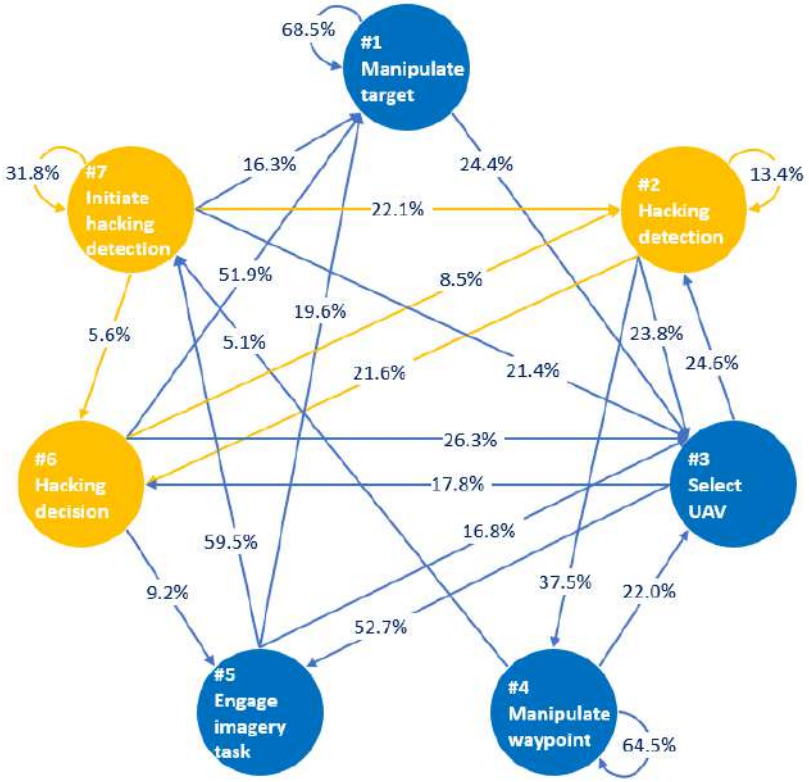
was used in model training [77]. HMM model training results were then selected using the Bayesian information criterion (BIC) [76, 100] and the number of rare states (NRS) method [74] to achieve both high model likelihood values and reasonable model structures.

Table 2.1: Observations of HMM models from RESCHU-SA experiment interface

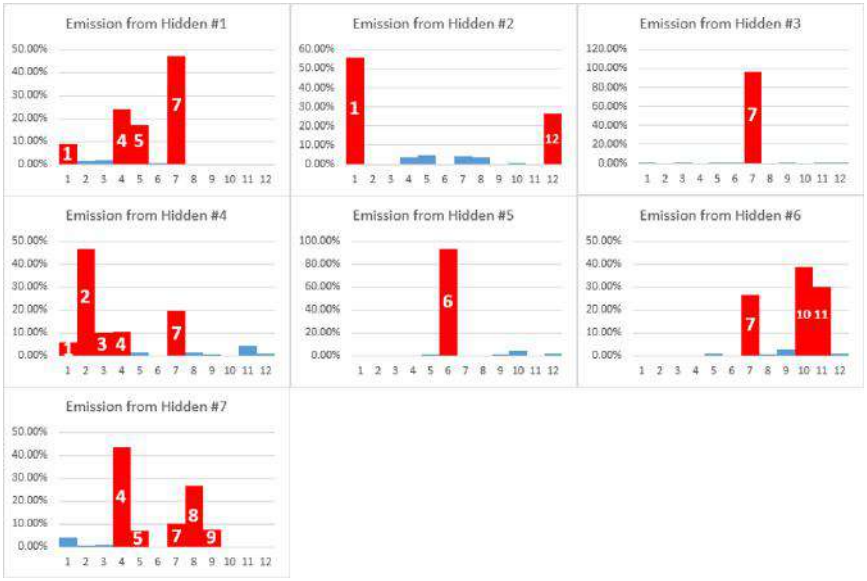
1 Add waypoint	2 Move waypoint	3 Delete waypoint	4 Move endpoint
5 Switch target	6 Engage task	7 Select UAV	8 Confirm info
9 Ignore info	10 UAV hacked	11 UAV not hacked	12 Adjust zoom

The general operator strategy HMM model was trained using observation sequences with 12 different observations as shown in Table 2.1. Based on the model selection process described in the previous section, the HMM model with 7 states had the lowest BIC value. Also considering that the 7-state model did not have any rare states and the HMM models with 8 or more states had at least one rare state, the general operator strategy model was determined to be a 7-state HMM model, as shown in Figure 2.4a. The interpretation for each hidden state was determined by the emission probabilities, shown as the histograms in Figure 2.4b.

The first state was interpreted as "Manipulate Target" because it was mainly a cluster of observations 4 (Move endpoint), 5 (Switch target), and 7 (Select UAV), which were directly related to UAV target manipulations. The second state was interpreted as "Hacking Detection" because this was the only state that had significant emission to observation 12 (Adjust zoom level), which indicated a typical operation of using cameras to compare against the map. The third state was interpreted as "Select UAV" because its only major emission was observation 7 (Select UAV). The fourth state was interpreted as "Manipulate Waypoint" because it was a cluster of observations 1 (Add waypoint), 2 (Move waypoint), 3 (Delete waypoint) and 7 (Select UAV), which were directly related to waypoint management. The fifth state was interpreted as "Engage Imagery Task" because its only major



(a)



(b)

Figure 2.4: The general operator strategy HMM model with emission histograms. (a) The 7-state hidden state layer. (b) Emission histograms for each hidden state.

emission was observation 6 (Engage task), indicating people were executing the intersection counting task. The sixth state was interpreted as "Hacking Decision" because it was the only state that had major emissions to observations 10 (Consider UAV hacked) and 11 (Consider UAV not hacked) which were decisions for hacking events. The seventh state was interpreted as "Initiate Hacking Detection" because it was the only state that had emissions to observations 8 (Confirm Notification) and 9 (Ignore Notification) which indicated the initiation of hacking detections.

The general operator strategy model represents the operator behavior states in navigating UAVs, conducting imagery searches, and dealing with potential hacking events. The first interesting fact shown in the model is that the UAV navigation (highlighted in blue) and hacking detection (highlighted in yellow) functional groups can be distinguished clearly based on the hidden state interpretations. The transitions between these two groups represent the probabilities of switching functional groups in operator behavior states.

Two major state transitions, which start from the "Initiate hacking detection" state and end at the "Hacking decision" state, in the general strategy model were highlighted and shown in Figure 2.5. These two transitions represent major behavior state flows in participant hacking detection strategies. The transition highlighted in yellow shows that some participants focused on detecting hacking events that they did not transit to other states in the navigation functional group. On the other side, participants who followed the purple transition switched between the two main functional groups during hacking detection tasks. These major transitions illustrate potentially different ways that participants detected hackings. In order to further investigate participants' strategies in hacking detection tasks, a hacking detection strategy model was developed on detection-related observation sequences and discussed in the next section.

A previous study on the original RESCHU platform, which only dealt with the navigation of the UAVs and did not have any hacking considerations [56], exhibited just four similar states to those blue states in the navigation functional group in Figure 2.4. This

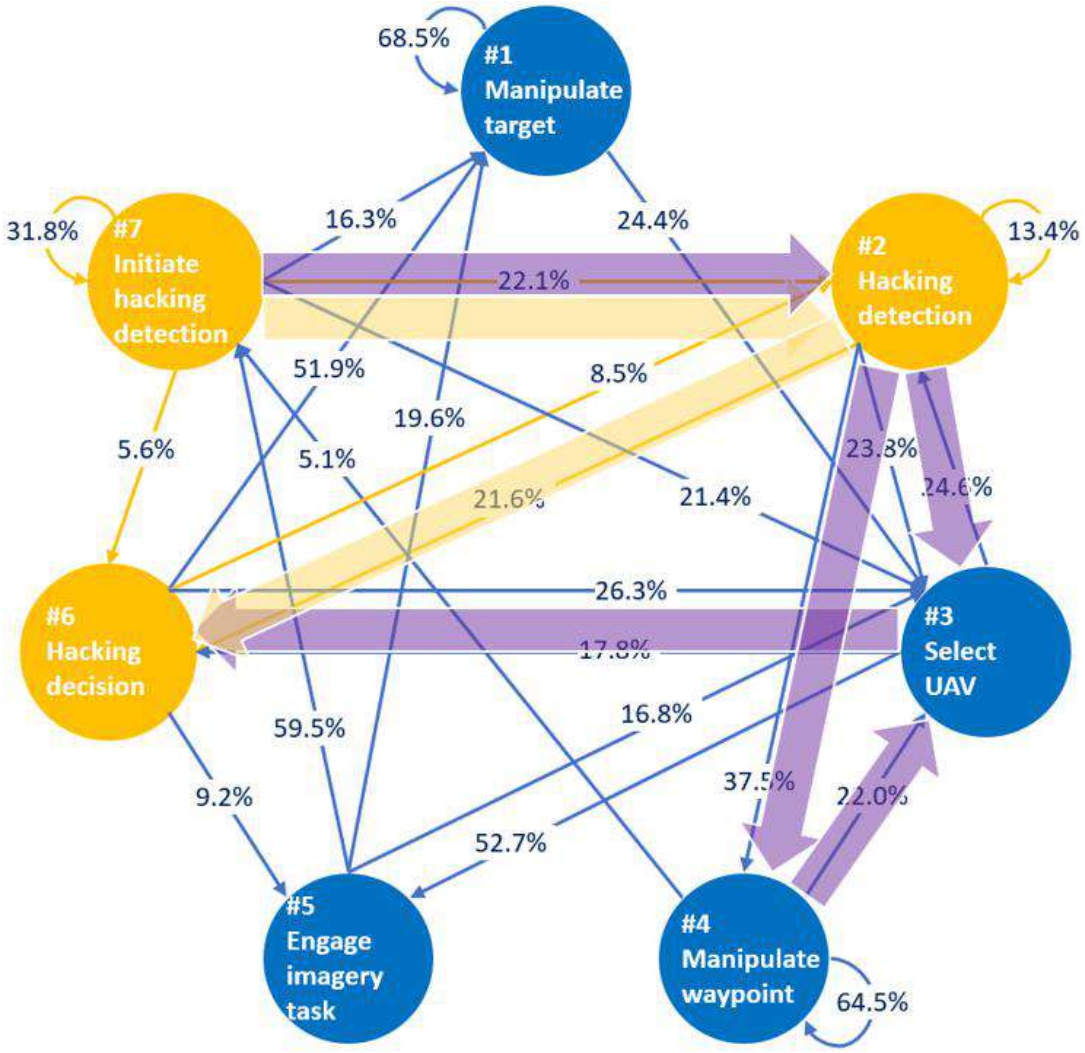


Figure 2.5: Major behavior state transitions in the general strategy model.

is an important finding since it means that the addition of a new set of tasks did not dramatically change the underlying states, rather the added functionality of hacking detection simply added more states. This suggests that at least in some supervisory control environments, functions may be modeled in a modular fashion, which would reduce the workload in adapting older models as new functions are added.

Also, the general RESCHU-SA model in Figure 2.4 shows some potential inefficiencies in operator strategies. In the navigation functional set of states, the first state of "Ma-

nipulate Target" and the fourth state of "Manipulate Waypoint" have high self-transition probabilities. These probabilities indicate that once operators entered these two behavior states, they tended to conduct repeated operations. Such repeated operations indicate potential inefficiencies that could be improved with future designs, such as assistant systems, for the UAV supervisory control interface.

Two hidden states, "Hacking Detection" and "Initiate Hacking Detection", in the hacking detection functional group also revealed potential problems with self-transitions. The time consumption in hacking detection was negatively correlated with the hacking detection success rate (Pearson=-0.375, p=0.001). Thus, the longer the person spent investigating a potential hacking event, the less likely a successful detection would occur. This result was surprising because as people gather more information, they should be able to increase their probability of successful detection. This result then led us to develop more detailed HMMs about operator hacking detection strategies in order to shed more light on this unexpected result. These more specific HMMs are detailed in the following section.

2.3.3 Hacking Detection Model

The HMM in Figure 2.4 provides an overall view into how operators approached the navigation and imagery tasks, while also dealing with hacking events. However, since this model does not provide enough details about how operator formed strategies for dealing with the hacking events, we elected to focus on those operator interactions from the beginning to the end of each hacking event. The resulting hacking detection model was trained based on 10 observations instead of the original 12 observations, as shown in Table 2.2. In the revised training dataset, original observations of "Confirm notification" and "Ignore notification" were combined into "Perceive hacking", and "Consider UAV hacked" and "Consider UAV not hacked" were combined into "Detection decision".

As shown in Figure 2.6a, the resulting hacking detection strategy model is a 6-state

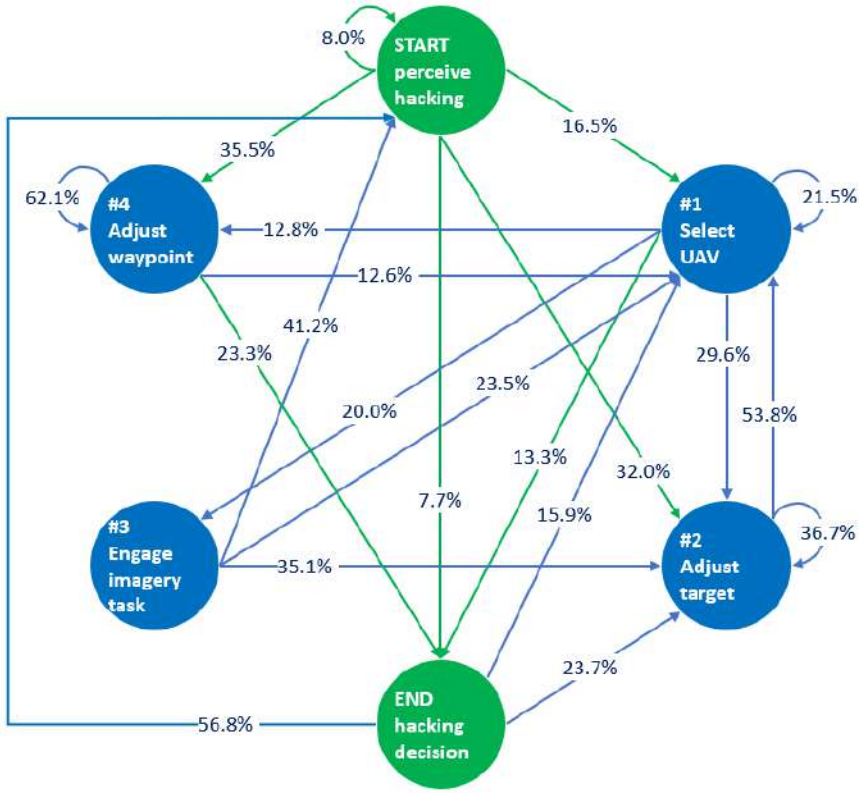
Table 2.2: Observations of the hacking detection strategy model

1 Add waypoint	2 Move waypoint	3 Delete waypoint	4 Move endpoint
5 Switch target	6 Engage task	7 Select UAV	8 Perceive hacking
9 Detect decision	10 Adjust zoom		

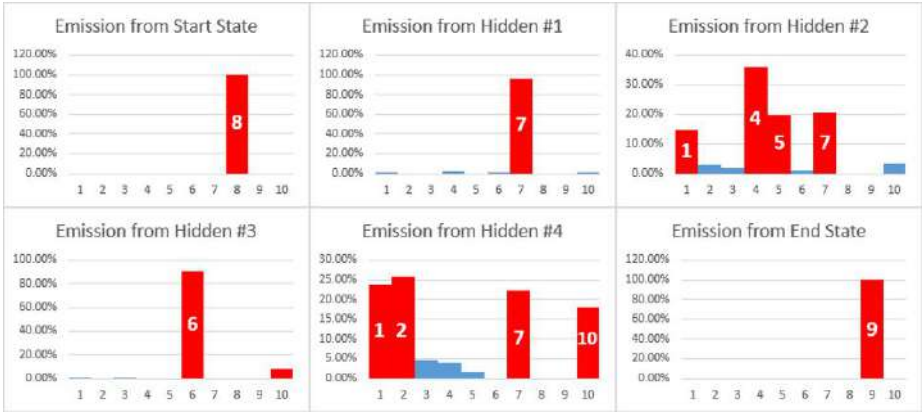
HMM based on a model selection process similar to the one used for the general operator behavior model. The interpretation for each hidden state was determined by the emission probabilities shown in Figure 2.6b. Although the observations were slightly different, the interpretation criteria were similar to the general behavior model. The six hidden states were interpreted as 1) the start state of Perceive Hacking; 2) Select UAV; 3) Adjust Target; 4) Engage Imagery Task; 5) Adjust Waypoint; and 6) the end state of Hacking Decision. The 56.8% transition from the END state to the START state represents overlapping hacking detections where operators finished a hacking detection and then on to immediately start another hacking event.

Two major behavior state transitions in the hacking detection HMM model are presented based on transition probabilities, as shown in Figure 2.7. Such transitions are considered as hacking detection operation flows because they began with the START state, in which operators perceived hacking events, to the END state, in which operators made detection decisions. The first major flow, indicated by blue arrows, has an intermediate state of "Adjust waypoint" between the start and the end state. The second major flow, shown by red arrows, has two intermediate states of "Adjust target" and "Select UAV". These two major operation flows suggest two dominant hacking detection strategies, termed "waypoint-oriented strategy" and "target-oriented strategy".

In the waypoint-oriented strategy, operators tended to manipulate UAV waypoints, including adding and moving waypoints, to detect hacking events. In this hacking detection strategy, operators typically either manipulated or introduced waypoints to investigate the



(a)



(b)

Figure 2.6: The hacking detection strategy HMM model with emission histograms. (a) The 6-state hidden state layer. (b) Emission histograms for each hidden state.

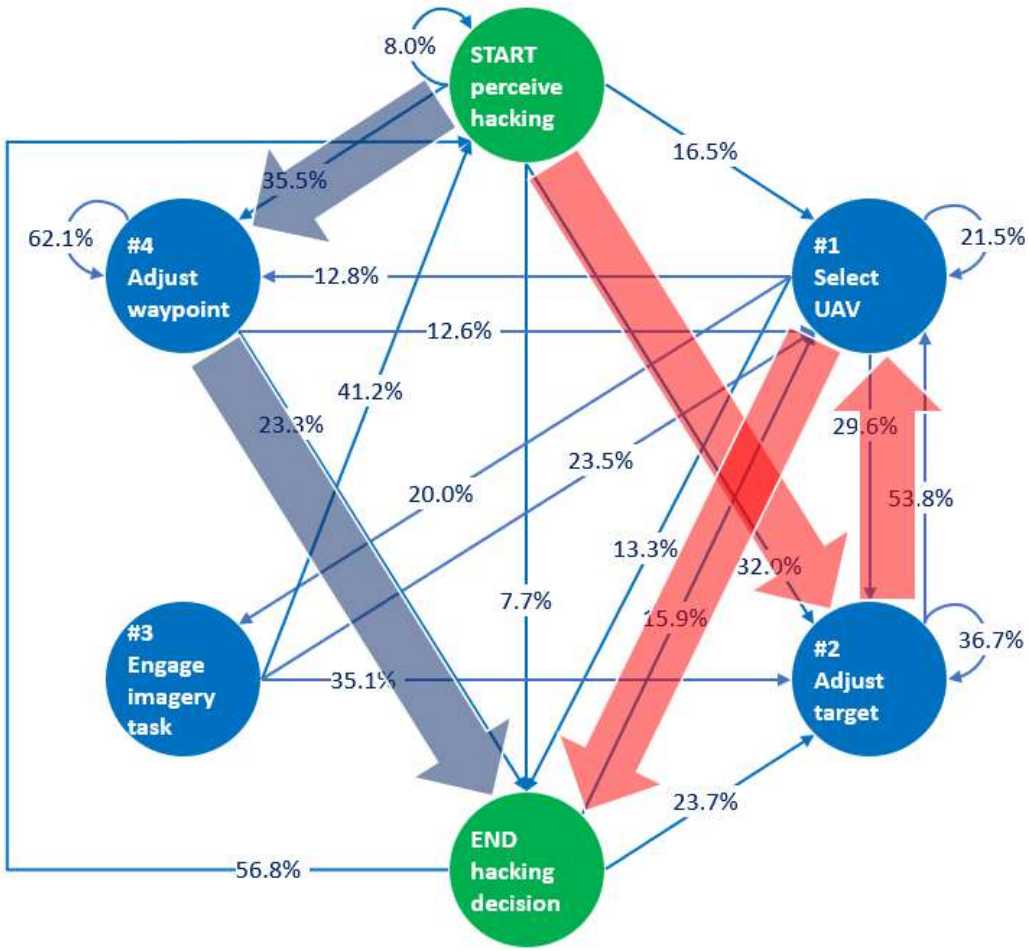


Figure 2.7: Major behavior state transitions in operator hacking detection strategies.

potential differences in the scene between the camera view and the surrounding map area. Operators who used this strategy usually fixated on comparing the effects of turning the UAV and the appearance of the ground in the camera feed to that expected while turning based on the map. This is considered as a dynamic strategy because motion was a key element in the determination of location.

In the target-oriented strategy, operators tended to directly switch UAV targets to detect hacking events. In this strategy, operators typically focused on the specific landmarks that the UAVs would fly over, such as unusual intersections or buildings. This is considered as a static strategy because operators would wait until the UAV reached a place of interest to

make a hacked or not hacked decision. Both strategies revealed inefficiencies, primarily through the self-transition probabilities. For example, in the waypoint-oriented strategy, 62% of people stayed in this state, repeatedly adding, moving, and deleting waypoints. Similarly, 37% of people repeatedly redirected vehicles to other targets, suggesting an inefficient target selection process. These inefficiencies could potentially be made better with advanced decision support.

The occurrence frequency and percentages of the waypoint- and target-oriented strategies for each participant was obtained by applying the hacking detection HMM model to each participant's data using the Viterbi algorithm [76]. Based on the occurrence percentage of the adjust waypoint and adjust target states, participants were classified into different hacking detection categories. As shown in the Table 2.3, participants were classified into four categories, including 1) waypoint strong dominant strategy; 2) waypoint weak dominant strategy; 3) target weak dominant strategy; and 4) target strong dominant strategy. The population of each strategy category was approximately one fourth of the total participant population. However, no strategy dominated in terms of performance.

Table 2.3: Participant classification based on different hacking detection strategies

Index	Strategy	Number	Percentage
1	Waypoint strong dominant	10	27.8%
2	Waypoint weak dominant	7	19.4%
3	Target weak dominant	11	30.6%
4	Target strong dominant	8	22.2%

Overall, the scenario strategy model, derived using RESCHU-SA-based experiments, shows 7 major human operator behavior states for supervision of UAVs that could be subject to hacking events. In this model, two functional groups emerged, including a hacking

detection group with three states and a UAV navigation group with four states. Also, the 6-state hacking detection strategy model allowed us to investigate operator hacking detection strategies in detail. Two major strategies can be observed from the model, including waypoint-oriented and target-oriented strategies. These diagnostic operator strategy models highlight that effective strategies can be inefficient. Further work can determine why people adopt different strategies and whether additional technology-based assistance can be used to improve operator strategies.

In summary, the operator strategy models in this section illustrate how operator strategies in conducting supervisory control tasks can be modeled through the use of HMMs. By utilizing HMM models, researchers can describe operator strategies in various HSC scenarios with different settings. In this case, researchers can evaluate if applying certain changes, such as adding a support system or modifying a control interface, will affect operator strategies by comparing resulting strategy models. Thus, the following chapters focus on the model comparison process and the practical meaning of model difference measures.

2.4 Chapter Summary

This chapter presents the background of this dissertation. This chapter starts with a section that introduces human supervisory control, which is a typical control scheme in the human-automation interaction. This section also reviews common techniques for modeling human operators. Specifically, hidden Markov models have been utilized in this research as operator strategy models. The second section of this chapter illustrates the structure of hidden Markov models and the model training and selection process. The last section in this chapter demonstrates an example of strategy model development from an HSC scenario of multiple drone control. In this example, a general strategy model and a hacking detection model are presented.

Chapter 3

Operator Strategy Model Comparison Metric

This chapter describes the details of the operator strategy model comparison metric, which is the first main contribution of this dissertation. The first section starts with a review of common methods of comparing HMM models and explains the divergence distance measure utilized in this dissertation in detail. Then, this section introduces the construction of a divergence mesh, which is an extension of the divergence distance measure and can provide more comprehensive model difference measures.

The subsequent sections in this chapter present two important parts of the model comparison metric. The first part describes the data quantity prerequisite for effectively modeling operator strategies and conducting strategy model comparisons. The second part details the observation alignment approach, which requires modification of data elements to compare strategy models with non-equivalent model observations.

3.1 Development of Divergence Meshes

This section presents the principle and calculation of the divergence distance measure, which is a commonly used HMM difference measurement method. This section also presents the development process of divergence meshes, which can provide more comprehensive comparison results for HMM models. Lastly, this section provides an example of the interpretation of comparison results using a divergence mesh.

3.1.1 HMM Model Difference Measures

Researchers compare HMM models to evaluate the underlying differences between the datasets that lead to such models. Other than qualitative model comparisons of state interpretations and model structures between HMM models, many quantitative HMM model difference measurement methods have been proposed for various applications, including the investigation of HMM development, HMM classification, and sensitivity tests on HMM model parameters [105, 106, 107].

Among these model comparison methods, two model difference measurement methods widely utilized to quantitatively investigate the similarities between HMM models include the divergence distance measure [108] and the co-emission probability distance measure [107]. Specifically, the divergence measure focuses more on the model fitting probability aspect of different HMM models applied to the same given data sequences, while the co-emission measure emphasizes the quantitative distance between HMM model data vectors in a high dimensional space.

The divergence distance method measures the difference between the model fitting probability of a pair of HMMs [108]. The representation of the divergence measure can be defined as:

$$D(\lambda_1 \| \lambda_2) = \frac{1}{num} |\log(P(O_{all}|\lambda_1)) - \log(P(O_{all}|\lambda_2))| \tag{3.1}$$

In the equation above, $D(\lambda_1 \| \lambda_2)$ is the divergence value calculated from the difference of model fitting likelihood on the dataset O_{all} between λ_1 and λ_2, which represent the first and second HMM model respectively. Specifically, O_{all} is the combination of both models' training datasets, and *num* is the total number of data points, or observations, in O_{all}.

Figure 3.1 presents the calculation process of a divergence value between two models developed from two different scenarios. Starting with data collection from HSC scenarios, HMM strategy models are trained based on collected data batches. Then an evaluation dataset is combined from training data batches of both models. The HMM models are

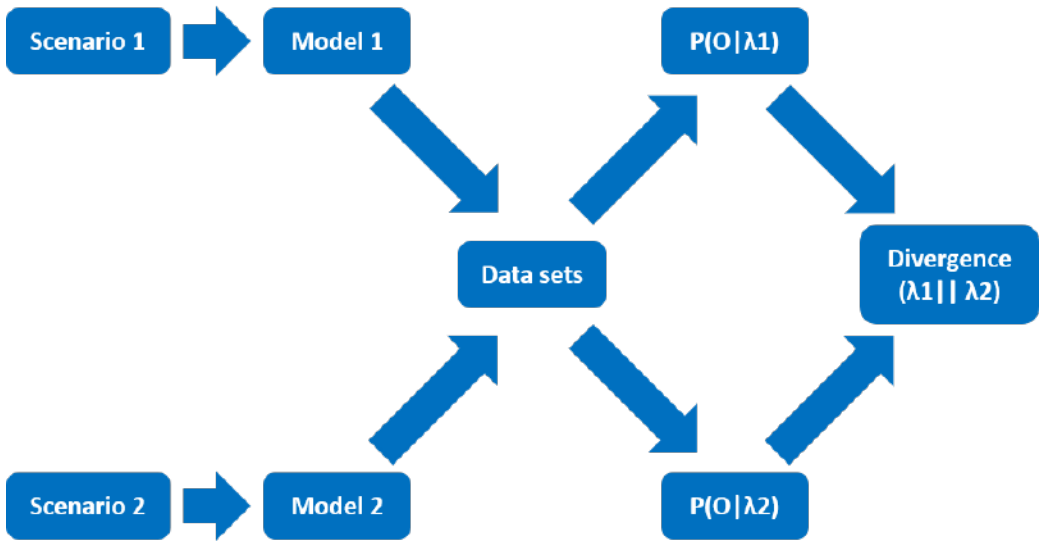


Figure 3.1: The divergence distance measure calculation process.

applied to the evaluation dataset to obtain model fitting probability values. Based on such model fitting probabilities, the divergence value can be calculated to quantitatively present the difference between these two models. The theoretical minimal divergence value is 0 based on the calculation shown in Equation (3.1). In this case, two HMM models share the same model fitting probability on a certain evaluation dataset, which means that the two models present the information of the underlying data patterns of the evaluation dataset at the same abstract level. Also, the upper bound of the divergence measure is unlimited so it can present model differences across a large value range.

The divergence measure has been utilized in many HMM-based applications to quantitatively measure differences between HMM models [109, 110, 111, 112, 113, 114]. HMM models have been widely used from the early stage of the speech recognition field [76], and Rabiner et al. used divergence measure to compare HMM recognition models to isolate digits and words in speech signals [109]. Yu et al. also used this method to distinguish data pattern models of HMMs [113]. The divergence measure can provide HMM model difference measures in a large scope, which starts from zero. Thus, researchers can evaluate the magnitude of HMM model differences in a wide range.

The second model difference measurement method is the co-emission probability distance measure, which focuses on the geometric distance between two HMM models. The geometric distance is considered as the angle of data vectors between two model data batches in a high dimensional space. The co-emission probability of two models also presents the generalizability of the models across given datasets [107]. The co-emission probability of two HMM models is defined as:

$$A(\lambda_1, \lambda_2) = \sum_{O_M \in O} P_{\lambda_1}(O_M) P_{\lambda_2}(O_M) \tag{3.2}$$

Similar to the divergence distance method, λ_1 represents the first HMM model and λ_2 represents the second HMM model. O_M is a sub-sequence of all observation sequences O, which is also a combined evaluation dataset. Thus, $P_\lambda(O_M)$ represents the probability of an HMM model fitting a given data sequence O_M. Then, the summation of the product of such model fitting probability values of sub-sequences from both models is considered as the co-emission probability of such two specific HMM strategy models on an evaluation dataset. The calculation process is illustrated in Figure 3.2.

Using the co-emission probability, the similarity between two HMM models, λ_1 and λ_2, is defined as:

$$S(\lambda_1, \lambda_2) = A(\lambda_1, \lambda_2) / \sqrt{A(\lambda_1, \lambda_1) A(\lambda_2, \lambda_2)} \tag{3.3}$$

Here, $A(\lambda_1, \lambda_2)$ is the co-emission value between models λ_1 and λ_2. And $A(\lambda_1, \lambda_1)$, $A(\lambda_2, \lambda_2)$ are co-emission values within either model λ_1 or λ_2. Such a similarity measurement follows the calculation of cosine similarity, which is represented using a dot product and magnitudes of two vectors:

$$similarity = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2 \sum_{i=1}^{n} B_i^2}} \tag{3.4}$$

Based on the definition of the cosine similarity, a similarity value of 1 means two tested models share the exact same structure, and a similarity value of 0 indicates orthogonality or decorrelation between two models. Thus, the co-emission similarity measure, which

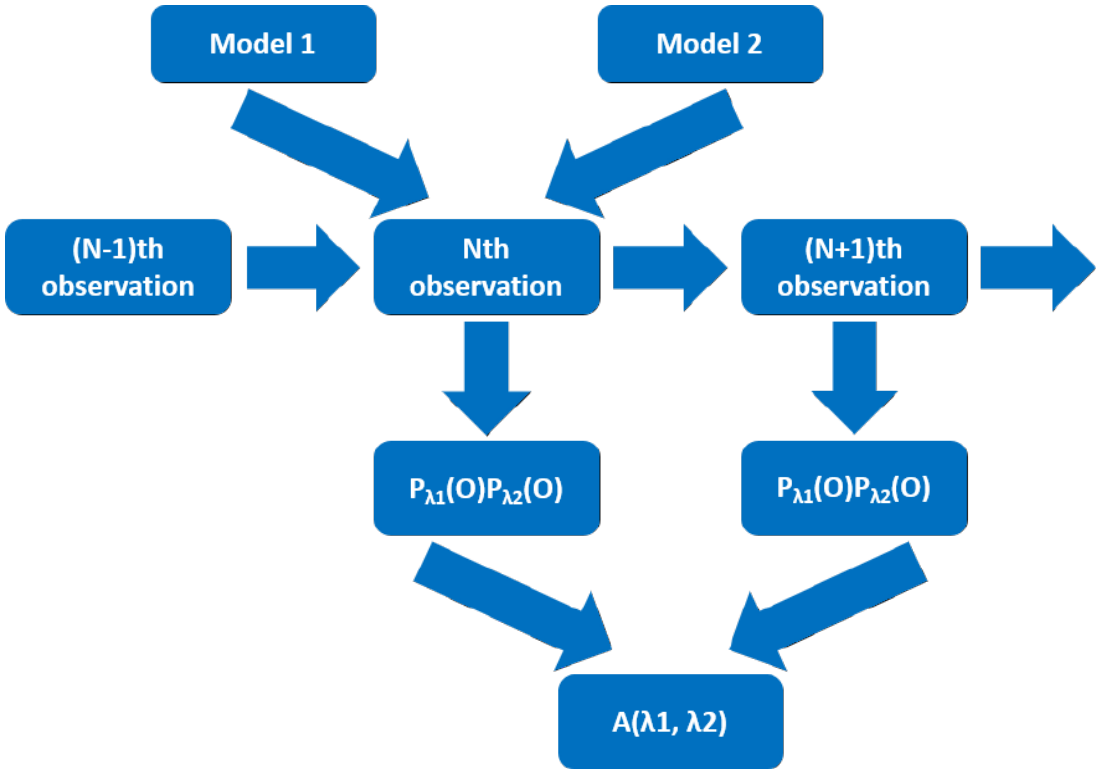


Figure 3.2: The co-emission probability measure calculation process.

presents the magnitude of how models are similar or different, generates a quantitative outcome range of 0 to 1.

The co-emission measure has also been used in some HMM model comparisons [115, 116, 117, 118]. Jagota et al. used HMMs to model bioinformatic data sequences and compare resulting HMM models with the co-emission measure to determine the differences between data sequences [115]. Another similar work conducted by Soding et al. also used the co-emission measure to compare HMM protein models to investigate the similarity between models and the homology of models [118]. Since the value range of co-emission measures is from 0 to 1, if the difference between models is not large, the co-emission measures will be squeezed into a small value range close to 1 according to the cosine property. In this case, it would be hard for researchers to evaluate the magnitude of model difference measures.

3.1.2 Development of Divergence Meshes

Given that the divergence distance measure provides a quantitative outcome value range that is larger than the co-emission probability measure, the divergence measure is used as the main HMM model comparison calculation method in this research. Further, this outcome range allow us to evaluate the magnitude of model differences in a wider range. As mentioned in the previous chapter, a model selection process is required in the model development to determine the optimal model structure to present operator behavior states and control strategies. The model selection result can be selective, and other model structures trained from the same data batch also contain information at different abstract levels. Thus, the approach of developing divergence meshes is proposed in this research to obtain comprehensive model comparison measures.

The main idea of the divergence mesh approach is to consider all possible model structures developed from data batches in model comparisons. For instance, model comparisons are conducted between datasets O_1 and O_2, which both contain M different observations. If two datasets have different numbers of observations, an observation alignment process, which is discussed in the following section, is required before developing HMM models. Based on the HMM model development process described in the previous chapter, $M-1$ models will be trained with 2 to M hidden states respectively on both datasets. If only the optimal model structure is considered in model comparison, then we will only have one divergence value representing the difference between these two models. If all model structures are considered, then we can have $(M-1)\times(M-1)$ comparisons and divergence values for plotting a divergence mesh.

A typical divergence mesh is shown in Figure 3.3. This divergence mesh is plotted based on model comparisons between two data batches, each of which contains 15 types of observations. Thus, the two horizontal axes, which are based on both data batches respectively, represent a different number of hidden states of resulting HMM models. Such mod-

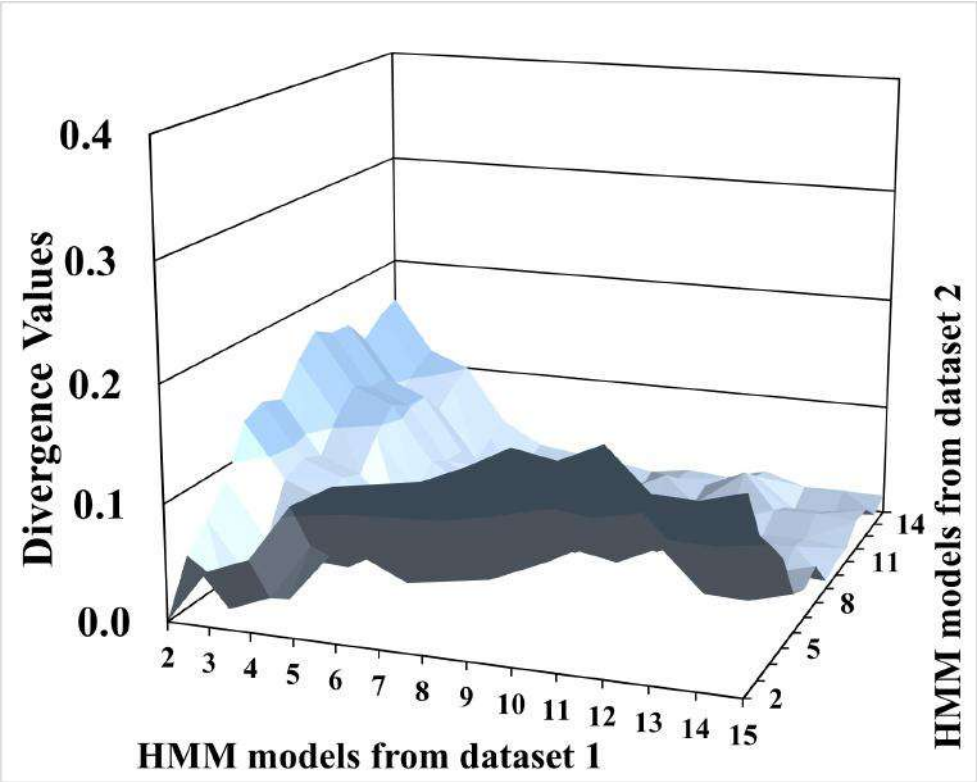


Figure 3.3: An example of a divergence mesh.

els contain operator strategy information at different abstract levels with different model structures. The vertical axis represents the magnitude of divergence values. In this case, a 3D divergence mesh can be plotted with $(15-1)\times(15-1)=196$ divergence values.

A divergence mesh can provide a comprehensive comparison of the underlying data patterns between two data batches by considering all possible resulting model structures rather than a single divergence value calculated from two optimal models. The average divergence value of a divergence mesh illustrates the general difference between two sets of models. The 3D shape of the mesh, including the changing trend of divergence values, shows how different model structures affect the divergence measure.

For the divergence mesh example shown in Figure 3.3, the average of all 196 divergence values is 0.0568, which can generally be considered as a low divergence value (the detailed quantitative model comparison baselines will be introduced in the next chapter).

An interesting fact about this mesh is that the comparisons on the diagonal show much lower divergence values than the comparisons on the side. This fact indicates that models from these two data batches with the same or similar number of hidden states would be quantitatively more similar than models with much different model structures. This result is expected because models with the same or similar model structure contain operator strategy information at a similar abstract level. In general, all divergence values on this mesh are lower than 0.2, which can be considered as low divergence values.

3.2 Data Quantity Requirement

This section describes the process of determining the minimal data quantity required by HMM operator strategy model development by utilizing the model comparison concept and generating divergence meshes. Strategy models developed from datasets with sufficient data quantities can avoid influences from individual variances and present the general operator strategies in HSC scenarios.

3.2.1 Analysis Approaches

Researchers and practitioners who design systems where humans team with autonomous systems often need to know the strategies that users of such systems develop. For safety-critical systems, designers need to know whether such strategies align with the system designer's intent and if resulting interaction strategies are safe and effective. While hidden Markov models (HMMs) have commonly been used to represent such strategy models, this modeling process can be affected by many factors including the size and number of individual datasets. For example, human-in-the-loop experimental data is notoriously hard to collect, so understanding the minimal information set that is needed for such models would guide such data collection efforts.

In this section, the impact of data quantity on modeling operator strategies in a human-autonomy teaming supervisory control scenario is investigated using HMM models. We focus on the divergence measure [108] and the divergence mesh approach mentioned in the previous section for model comparisons to determine the minimum number of observation sets needed. HMM models are used as strategy models in this section [77]. Recall from the previous chapter that the HMM structure can be represented as a tuple, $\lambda = \{S, O, A, B\}$, in which S represents hidden states, O represents observations, and A and B represent transition and emission probability matrices, respectively [96]. As mentioned in the previous chapter, HMM models are trained using the unsupervised multi-sequence Baum-Welch algorithm [77, 96].

For HMM-based applications in HSC scenarios, it is unclear how model quality is affected by data quantity. By data quantity, we mean the number of sets of observations, equivalent to the number of subjects needed for an HMM analysis. While more data is generally better in modeling, there is a cost-benefit tradeoff between obtaining enough data in HSC applications. It may be, for example, cost-prohibitive to obtain large datasets on air traffic controllers under controlled conditions to investigate a new air traffic control display. However, when using unsupervised learning models, insufficient data could lead to underfitting [119, 120]. This section attempts to quantify the minimum data quantity for HMMs to model operator strategies in HSC applications.

In order to determine a minimum sufficient data quantity for approximating the strategies of a population, some previous work has suggested the sample data size should be a certain proportion of the whole data quantity [121, 122]. Such sampling criteria require and depend on prior knowledge of the population, which is not typically available in HSC settings. Similarly, other work has also proposed criteria for determining the minimal data quantity based on statistical analysis [123, 124]. However, these methods focus on data size for qualitative approaches such as interviews and surveys, but not for unsupervised learning models like HMMs. Thus, the primary focus of this section is to determine the

minimum data quantity for effectively and precisely developing diagnostic models in a representative HSC scenario.

Many HMM model comparison metrics have been applied to HMM-based modeling scenarios to quantitatively measure the difference between HMM models [105, 125, 106]. In this section, we utilize the divergence measure metric to compare HMM strategy models, which measures the difference between the model fitting probability of a pair of HMMs [108]. The representation of the divergence measure can be found in Equation (3.1). Based on the divergence values of the comparisons between HMM strategy models developed from different training data quantities, we can infer the approximate minimum sets of data needed for modeling general operator strategies.

3.2.2 Data Generation

To develop strategy models, a large dataset of action sequences in an HSC application is needed. Given that databases of large HSC experiments are not publicly available like that of air traffic controllers, we utilized a public large dataset of StarCraft II gaming sessions in this research [126]. StarCraft II is a real-time strategy (RTS) game, which is similar to many real-world applications of supervisory control such as air traffic control and military planning scenarios. In this game, the player's goal is to defeat other players by conducting several high-level tasks, including gathering resources, developing buildings and technologies, and navigating units to battle. The StarCraft II game is an effective HSC proxy since players indirectly control all units via a keyboard and a mouse (a control interface) to conduct high-level tasks under time pressure. The order and transition of player commands, which constitute observable states, can illustrate players' in-game strategies.

StarCraft II replay files were accessed from Blizzard Entertainment's official website (https://starcraft2.com/en-us/). Replay files contain compressed information of games, including 1) game duration in seconds, 2) the name of the game map, 3) the role players

selected, 4) players' number of actions per minute (APM), 5) players' matchmaking rating (MMR), which is an official measure of players' skill-levels, and 6) players' game results in terms of wins or losses [126]. We leveraged the application programming interface (API) provided by Blizzard Entertainment (https://github.com/Blizzard/s2client-proto) to import replay files to the StarCraft II game and parse players' action sequences.

Because the entire database consists of over 500,000 replays with over 100 million data points, we filtered the replays to obtain a dataset that was similar in length and complexity (i.e., number of observations) to that generated by the widely used HSC research experimental platform of RESCHU-SA [127, 128, 62]. The collected StarCraft action sequences were based on specific criteria, including: 1) a single map (out of over 20 maps), called Triton LE, 2) a single role (out of three roles), called Terran, and 3) replays that lasted more than 300 seconds with more than 100 actions. In total, we parsed 2000 replays as the final dataset, in which each replay contains about 380 actions, on average.

Based on the action list embedded in the API, we converted raw actions into 15 primary observations, as shown in Table 3.1. We clustered raw actions with the same group name but different targets or objects into a single observation to reduce unnecessary model training complexity. For example, "Build_Armory", "Build_Assimilator", and "Build_Bunker" are all raw actions that represent building in-game constructions or units, so these were termed "Build". Also, "Train_Carrier", "Train_Drone", and "Train_Mothership" are raw actions that represent producing and generating in-game battle units, so these actions were termed "Train". Given these conversions, the final dataset contained 2000 parsed data sequences with the 15 observations shown in Table 3.1.

3.2.3 Model Development and Comparison Process

Once the 2000 StarCraft II game replays were selected, we then developed a series of models with differing numbers of sets of observations and compared them quantitatively to

Table 3.1: Observations in the StarCraft II game

Index	Observation	Description
1	Train	Generate or build units
2	Attack	Send units to attack enemies
3	Build	Make or build constructions
4	Effect	Maintain units, including repair and charge
5	Morph	The transformation of units
6	Cancel	Cancel current action
7	Unload	Unload from certain units or constructions
8	Lift	Select certain constructions and transport
9	Land	Settle certain constructions
10	Stop	Terminate auto progress
11	Burrow	Hide units underneath the surface
12	Harvest	Gather resources
13	Rally	Predefine destinations after generating units
14	Trainwarp	Generate special units
15	Hallucinate	Set hallucination effects on units

discover the stability and reliability of the HMM modeling technique for representing HSC strategies. As shown in Table 3.1, models will not have more than 15 hidden states because states represent weighted clusters of observations. The minimum number of hidden states for any model is two because transitions between at least two states are necessary to present transitions among players' behavioral states as a part of players' strategies.

In order to compare models built from different quantities of data, we parsed 2000 game replay data sequences into 9 different groups with different data quantities (5, 10, 20, 30,

50, 100, 200, 500, 1000 replays). Then for each data quantity, we randomly selected 30 batches of sequences for model development in order to capture variance in individual player strategies. Thus, for each 30-sequence data batch, we developed HMM models with all possible numbers of hidden states, from 2 to 15 states, utilizing the multi-sequence Baum-Welch algorithm [77]. For example, randomly selecting 30 batches of 100 replay sequences from the original 2000 sets of observations forms 14 clusters of HMMs (from 2 to 15 hidden states), each with 30 models. This overall model training procedure can be referred to as Algorithm 1. The dataset of 2000 replays is treated as a special data group. A set of HMM models with 2 to 15 hidden states was developed for these 2000 data sequences to represent the ground truth of HMM strategy models across all players.

Algorithm 1 Model Training among Data Groups with Different Data Quantities

procedure MODEL TRAINING PROCESS
  $Q = \{5, 10, 20, 30, 50, 100, 200, 500, 1000\}$
  for each $p \in Q$ **do**
    for $i \leftarrow 1$ to 30 **do**
      Randomly select the i_{th} data batch of p replays
      for $n \leftarrow 2$ to 15 **do**
        Train an HMM λ_{pin} with n states on the i_{th} data batch of p replays
        Record the model λ_{pin}
  return ▷ Finish model training

In order to compare the fidelity of each set of models for the different data quantities, we focus on model comparisons across 9 data groups (from 5 to 1000 replays). The overall comparison process is illustrated as Algorithm 2. Specifically, for each pair of resulting HMM models, we utilized their training data batches described in the previous subsection as an evaluation dataset to calculate divergence values to quantitatively present the difference between them [108].

Algorithm 2 Model Comparison among Data Groups with Different Data Quantities

1: **procedure** MODEL COMPARISON PROCESS
2: $Q = \{5, 10, 20, 30, 50, 100, 200, 500, 1000\}$
3: **for each** $p, q \in Q$ **do**
4: **for each** $i, j \in [1, 30]$ **do**
5: Combine datasets O_{pi} and O_{qj} to O_{pi+qj}
6: **for each** $n, m \in [2, 15]$ **do**
7: Calculate $D(\lambda_{pin} \| \lambda_{qjm})$ on O_{pi+qj}
8: Obtain an average divergence mesh of $D(\lambda_p \| \lambda_q)$
9: based on $D(\lambda_{pn} \| \lambda_{qm})$ by averaging over i, j
10: Record the mesh $D(\lambda_{pn} \| \lambda_{qm})$
11: **return** ▷ Finish model comparison

For divergence comparison between two HMM models, the minimum possible value is zero based on the divergence calculation shown in Equation (3.1). In this case, two models share the same model structure and produce the same fitting probability from the evaluation dataset. Although there is no upper limit of a divergence value, divergence values less than 0.1 are usually considered small and divergence values larger than 0.3 are usually considered large [81]. In this effort, small divergence values indicate a high similarity level between the model structures of two sets of HMM models and illustrates that operators' behavior patterns and strategies between two datasets are similar.

The data quantity level comparison is presented as the first "for" loop (line 3) in Algorithm 2 for every combination of two data groups with data quantities of p and q. When $p \neq q$, the comparison focuses on how different data quantities affect the difference of players' general strategies extracted respectively from two data groups. When $p = q$, the comparison focuses more on how model complexity affects the similarity between models developed from the same quantity of data.

The model structure level comparison is illustrated as the second and third "for" loop (lines 4 and 6) in Algorithm 2. As mentioned in the description of Algorithm 1, a set of 14 HMM models with hidden state numbers from 2 to 15 are trained respectively on each data batch. Thus, at this comparison level, $14 \times 14 = 196$ comparisons are conducted for a pair of data batches. For comparing two HMM models, we first combine their model training datasets into an evaluation dataset, which is the O_{all} in Equation (3.1) or the O_{pi+qj} in Algorithm 2. Then we obtain the number of data points, which is *num* in Equation (3.1), in the combined dataset and compute the model fitting probability, $P(O_{all}|\lambda_1)$ and $P(O_{all}|\lambda_2)$, for calculating the divergence value, $D(\lambda_1 \| \lambda_2)$.

Given that we have 30 data batches in each data quantity, divergence values between two specific model structures are averaged to represent the difference measure. While comparing data quantities of p and q (when $p \neq q$) with 30 batches respectively, we have $30 \times 30 = 900$ batch combinations in total, and for each combination, we have 196 divergence values, $D(\lambda_{pin} \| \lambda_{qjm}), n, m \in [2, 15]$. Thus, for a specific model structure combination n and m, the average, $D(\lambda_{pn} \| \lambda_{qm})$, of 465 divergence values $D(\lambda_{pin} \| \lambda_{qjm}), i, j \in [1, 30]$ is used for such a specific setting with variable of p, q, n, m. For a special case of $p = q$, the total number of batch combinations is $C_{30}^2 = 435$ to avoid comparisons between the same data batch and model structure. In other words, a restriction of $i \neq j$ will be considered for obtaining the average divergence values from $D(\lambda_{pin} \| \lambda_{qjm}), i, j \in [1, 30]$ when $p = q$.

Noting that the ground truth data group of 2000 replays only has one data batch, the average divergence values are calculated from $1 \times 30 = 30$ resulting values compared to other data groups with 30 batches. By averaging divergence values over batches, we can account for the variance in players' strategies. Once these values are calculated, we can plot a divergence mesh for a comparison of two data groups.

For example, Figure 3.4 illustrates the divergence mesh that can be plotted for all 196 comparisons between 1000 and 2000 replay data quantity groups. HMM models are trained with different numbers of hidden states for both data groups on each data batch, so

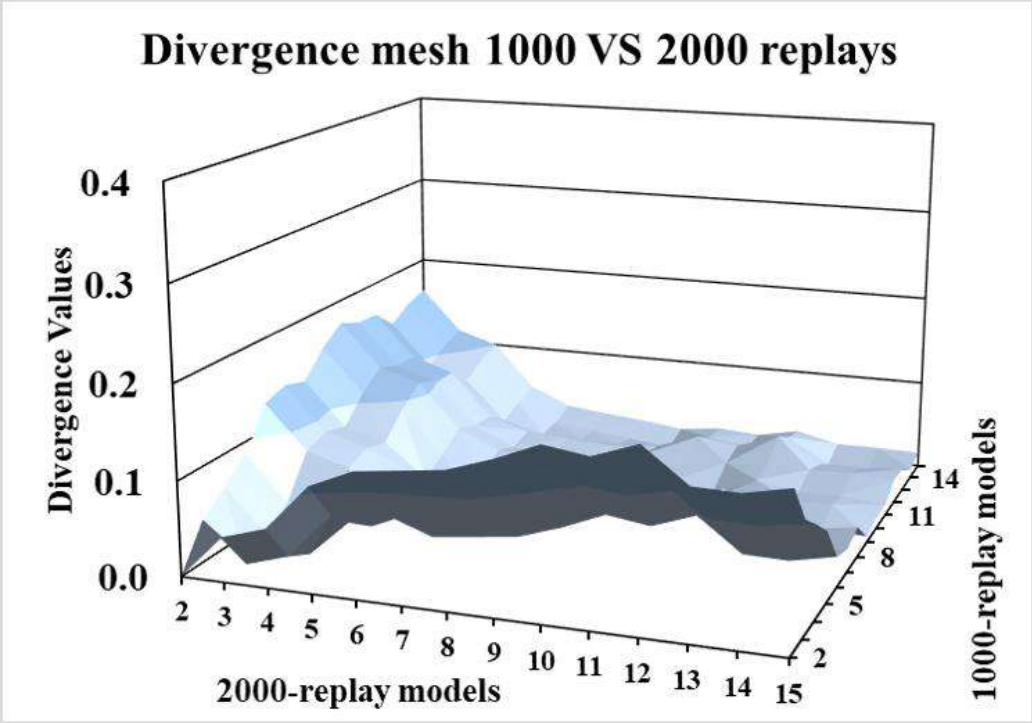


Figure 3.4: A divergence mesh example of the comparison between models developed from 1000 and 2000 replays. Each of the 196 points includes 30 comparisons.

the X and Y-axes represent the number of states in models from the two data groups respectively. The Z-axis, the vertical axis, represents the divergence values calculated from a combination of two models with certain model structures. For example, the first model, $\lambda_{1000,n=15}$, developed from a data batch of 1000 replays has a model structure of 15 hidden states, and the second model, $\lambda_{2000,n=2}$, developed from the ground truth data batch of 2000 replays, has a 2-state structure. Then, as shown in Figure 3.4, the divergence value between these two models, $D(\lambda_{1000,n=15} \| \lambda_{2000,n=2})$, is about 0.162. In total, a divergence mesh contains 196 such divergence values. Since the mesh is in three dimensions, the shape of the mesh or the changing trend of values based on hidden state numbers can also illustrate the relation between model structures.

Understanding that the minimum divergence value is 0 based on Equation (3.1), the smaller the divergence is, the higher the similarity between a pair of HMM models. Simi-

larly, the smaller the average distance between a mesh and the X-Y horizontal surface, the higher the similarity between the underlying players' behavior patterns and strategies from a pair of data batches. In the Figure 3.4 example, if models have more than 8 states, they share high similarity since the upper right corner of the mesh is close to a horizontal surface and has divergence values lower than 0.05. We propose that HMM models are similar and stable in these regions of low and consistent divergence metrics.

3.2.4 Analysis Results

Once divergence distance model comparisons were computed for all possible pairwise comparisons of HMM models created with 5, 10, 20, 30, 50, 100, 200, 500, 1000, and 2000 replays, we plotted them to investigate the variation tendencies across the meshes, as shown in Figure 3.5. From such variation trends, we can investigate the potential impact from the number of data sequences on the stability of the strategy modeling technique and also determine the minimum required number of data sequences.

As shown in Figure 3.5, for all numbers of data sequences, we compared corresponding models and plotted divergence meshes respectively. The upper-right half of Figure 3.5 contains divergence meshes, and given the diagonal symmetry, the lower-left half represents average values and boxplots of divergence values for the corresponding meshes.

Trends in a Single Mesh

A divergence mesh presents the comparisons between strategy models with different numbers of hidden states from two data batches. From the distance between different portions of a mesh and the horizontal X-Y surface, we can interpret how different model structures affect a model's ability to capture players' general strategies. To better understand such comparisons, three divergence meshes, shown in Figure 3.6, are selected from Figure 3.5 for further investigation.

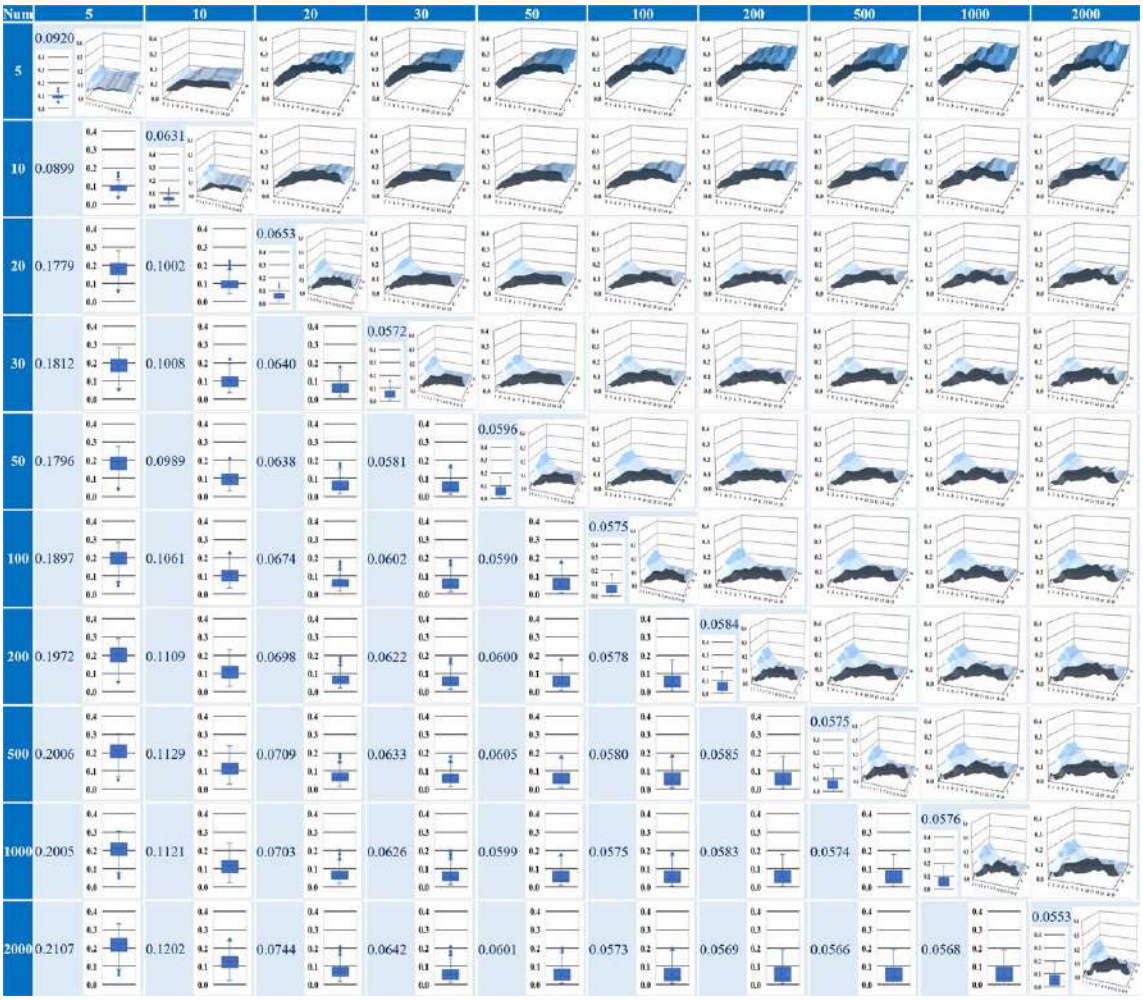
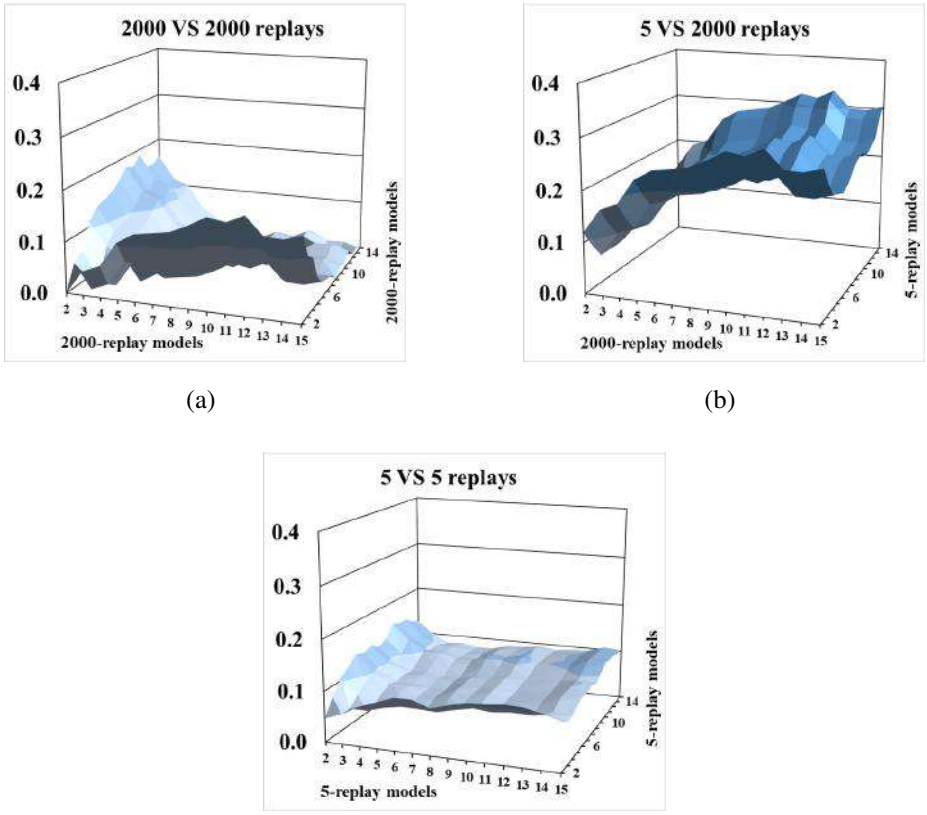


Figure 3.5: Divergence meshes with average values of all comparisons between different data groups with different quantities.

The divergence mesh in Figure 3.6a represents comparisons between models from 2-15 hidden states developed from the data group of 2000 replays. Since the 2000-replay data group represents ground truth, this divergence mesh is the benchmark comparison with an average divergence value of 0.0553, the lowest among all meshes. Such a low average value is expected because the 2000 replay group contains all data for all individuals.

One interesting observation is the diagonal-valley shape of Figure 3.6a. Such a shape indicates that if two models from the 2000-replay dataset share similar model structures



(a) (b)

(c)

Figure 3.6: Divergence meshes of 5 and 2000-replay datasets. (a) The 2000 and 2000-replay mesh. (b) The 5 and 2000-replay mesh. (a) The 5 and 5-replay mesh.

(number of hidden states), they share high similarity with divergence values under 0.01. However, as the comparison moves further away from the diagonal, the divergence between the two models increases. At the extreme, comparing a 2 hidden state model to a 15 state model yields a divergence metric of more than 0.15, meaning the models represent widely-differing player strategies. As noted, in the upper right region above 8 hidden states, the valley disappears. Such a change is verified by a t-test on the divergence values for blocks above and below the 8-state mark with a significance level of $\alpha = 0.05$ ($p < 0.001$). This suggests that regardless of the number of states, HMM models with 8 or more hidden states capture similar abstract strategy information. In other words, people who exhibit 8 or more

abstract clusters of behaviors adopt very similar strategies in executing their tasks.

Figure 3.6b illustrates comparisons between data groups with 5 (minimum) and 2000 (maximum) replays. This mesh is clearly different from Figure 3.6a with increasing values and no valley along the diagonal seen in the 2000 replay comparison (Figure 3.6a). The mesh in Figure 3.6b has an average divergence of 0.2107, the highest among all meshes shown in Figure 3.6. Such a high value indicates relatively low model similarity, suggesting that models developed from data batches with only 5 replays cannot accurately represent the player strategies seen in the 2000 replay set. This is not surprising given that 5 observations cannot capture the variability that a 2000 dataset can. Thus, the high divergence value of this mesh supports the fact that models developed from these two data groups are different.

In addition to the high divergence shown in Figure 3.6b, the tilted shape of the mesh shows that comparisons with 2000-replay models with 6 or fewer states have smaller divergence values than models with 8 or more states. Given that 5-replay models cannot precisely capture players' strategies, this fact indicates that 2000-replay models with oversimplified model structures cannot present players' strategies well.

Figure 3.6c illustrates the divergence mesh comparison of 5-replay models, with relatively low divergence values at an average of 0.0920. The relative flat shape of the mesh in Figure 3.6c indicates that models generated from low numbers of observations share high similarity regardless of hidden state numbers. However, when compared to the flat region in Figure 3.6a, the 2000 replay comparison models above 8 hidden states have an average divergence of 0.0180, which is 80.4% lower than 0.0920. Thus, while models generated from 5 replays may be similar to one another, they are not as similar as models generated from 2000 replays. The question then becomes how many replays are needed to approximate the mesh in Figure 3.6a. This answer then forms the basis of understanding the minimum number of data sequences needed to generate effective strategy models.

Mesh Pairwise Comparisons

The average value of all divergence points on a mesh can quantitatively illustrate the difference between players' strategies and action patterns captured from the models developed from two data groups with different data quantities. Thus, differences between average mesh divergence values can indicate how much data is needed to effectively train HMM strategy models. As shown in the lower-left half of Figure 3.5, each boxplot of divergence values represents model comparisons between a pair of data groups.

Not surprisingly, the divergence values on the diagonal decrease from 5 replay comparisons to the ground truth data at 2000 comparisons. These numbers should be the lowest since the models with the same numbers of states are compared to one another. The off-diagonal comparisons should grow with increasing difference between data quantities. For instance, the divergence mesh average of 5 vs. 5 replays is 0.0920, with the average value increasing to 0.1897 for the comparison of 5 vs. 100 replays, and then to 0.2107 for 5 vs. 2000 replays. This more than 200% average increase suggests that when the data quantities are small, resulting HMM models are generally different from models developed from data groups with large data quantities. However, such a large increasing trend is not obvious in data groups of 30 or more replays. The mesh average of 30 vs. 30 replays is 0.0572, and the average of 30 vs. 2000 replays is 0.0642, an increase of less than 12%. Similarly, for datasets with 50, 100, or more replays, their off-diagonal values are much smaller.

To statistically determine the difference between divergence value distributions in divergence meshes, a series of pairwise comparisons was conducted between all other data quantities vs. 2000-replay meshes, as well as the 2000 vs. 2000-replay mesh. Understanding that the 2000 vs. 2000-replay mesh is a comparison within the same data batch, the values on the diagonal of this mesh are all zeros because those values represent divergence measures between the same model structures. Thus, the diagonal values were not considered in pairwise comparisons to avoid skewing comparison results.

Table 3.2: Divergence mesh paired Wilcoxon Sign-Rank Test results

paired comparison between 2000 vs. 2000-replay mesh and	z-score	p-value
5 vs. 2000-replay mesh	11.2954	$< 1.0 \times 10^{-9}$
10 vs. 2000-replay mesh	8.4107	$< 1.0 \times 10^{-9}$
20 vs. 2000-replay mesh	4.7604	1.93×10^{-6}
30 vs. 2000-replay mesh	2.7736	0.0055
50 vs. 2000-replay mesh	2.1554	0.0311
100 vs. 2000-replay mesh	0.9442	0.3451
200 vs. 2000-replay mesh	1.3053	0.1918
500 vs. 2000-replay mesh	1.3236	0.1856
1000 vs. 2000-replay mesh	0.9765	0.3288

The paired Wilcoxon Sign-Rank Test results in Table 3.2 show that comparisons among the 2000-replay data group and data groups with small data quantities, including 5, 10, and 20 replays, models are significantly different between the benchmark comparison of 2000 vs. 2000 replays (family-wise significance level of $\alpha = 0.05/9 \approx 0.0055$). However, for data groups with 30 and larger data quantities, their comparisons with the 2000-replay group are not significantly different. This result indicates that HMM strategy models developed from data batches with 30 or more replays (data sequences) are not statistically different from the benchmark models trained from 2000 replays. Thus, for this dataset, 30 sequences can be considered as a sufficient data amount for effectively training HMM models to present operators' strategies.

Analysis Conclusion

HMMs can be useful abstractions of supervisory control operator behaviors, giving insight into the plans and strategies that people form when working with autonomous systems in order to meet shared goals. However, the number of operators studied in the building of such models could affect the reliability and generalizability of any resulting HMM models. To determine how undersampling of operators could affect outcomes and the minimum number of observations needed to form stable HMMs, we parsed a large dataset of StarCraft II game replays to develop and compare HMM models based on 15 observations.

Using divergence values that measure probabilistic similarities between two models, we systematically reduced a 2000 player dataset to demonstrate how models built from fewer data sequences compared to those built from all the data. Based on these model comparison results, we concluded that 30 data sequences produced models that were not statistically different from models built with 2000 data sequences. Understanding such a data quantity threshold can reduce the experimental burden and the costs of collecting human-in-the-loop data. While such knowledge can be used prospectively, it can also be used retrospectively to evaluate such a decision. Furthermore, models of current data can be compared against this data to determine degrees of similarity.

In addition to examining the role of data quantity in the development of HMMs for humans teaming with supervisory control systems, we also demonstrated how meshes of divergence values from the comparison of two models can indicate model stability. This effort further illustrated that meshes with low and consistent divergence values indicate model stability, where changes in the number of hidden states do not cause significant changes in divergence values. Also, given that the StarCraft II game has several well-defined tasks, one possible limitation of this work is the generalizability of the result to other HSC scenarios and other diagnostic models. Thus, further studies are necessary to investigate these issues.

3.3 Observation Alignment Approach

This section introduces another important part of the strategy model comparison metric, the observation alignment approach. HSC scenarios with different interfaces or high-level tasks will generate different interactions between operators and control systems. Such interactions are considered as the HMM observations in the strategy model development process. However, when researchers try to compare models between two datasets with different numbers of observations, non-equivalent observations can cause mathematical issues in the model comparison process. Similarly, if researchers try to investigate the impacts of certain changes before and after a scenario on operator strategies, they face the same problem of non-equivalent observations caused by potential observation changes in comparing strategy models. Thus, we propose an observation alignment approach, which aligns observations to the same number by reducing observation types or clustering certain observations, for operator strategy model comparisons with non-equivalent observations.

3.3.1 Basic Concept and Analysis Approaches

The fundamental concept of observation alignment is to match observations, or data types, from two datasets by using the same indices to represent observations. The detailed process of aligning observations depends on the similarity between the HSC scenarios that generate the two datasets. If two scenarios share a similar interface, then the two resulting datasets may share some observations. In this case, the dataset with a larger number of observations should reduce or cluster observations that are unique in this dataset to match the observations in the other dataset. If two scenarios have different interfaces, then the resulting datasets may have different observations. Then, we propose that researchers align observations based on the occurrence percentage rankings. In this section, we demonstrate an example of the observation alignment process for model comparisons between two human-subject experiments conducted on two interfaces.

The first experiment was conducted on the Research Environment for Supervisory Control of Heterogeneous Unmanned Vehicles (RESCHU) platform [104, 127], which is shown in Figure 3.7. RESCHU is a simulation-based platform that allows a single operator to control multiple UAVs in a supervisory control scenario. It includes both UAV navigational and imagery analysis tasks where operators focus on the map area when navigating UAVs and shift their attention to individual vehicle cameras when a UAV reaches a target [104]. Further experimental details are presented in Appendix C.

The second experiment was conducted on the RESCHU-SA interface as shown in Figure 3.8. A summary of the RESCHU-SA experiments is introduced in Section 2.3.1, and further details are presented in Appendices A and B. The RESCHU-SA interface provides three main tasks, including the two tasks in the RESCHU interface and an additional UAV hacking detection task. Thus, these two interfaces share observations generated from the navigational and imagery tasks, and the RESCHU-SA interface has observations for the hacking detection task.

As shown in Table 3.3, the RESCHU interface generates 7 observations, which are shared with the RESCHU-SA interface, and the RESCHU-SA interface generates additional 3 observations. Specifically, observations 1, 2, and 3 are directly related to the UAV navigational task, and observations 4, 5, and 6 are related to the imagery surveillance task. Observations 8 to 10 are related to the UAV hacking detection task, in which operators need to determine if UAVs are hacked and navigated to unexpected destinations.

Since these two interfaces generate different numbers of observations, we need to align observations to compare HMM models developed from them. The goal of this observation alignment approach is to reduce observations from the RESCHU-SA interface in order to match the observations from the RESCHU interface. In other words, this method modifies the observation selection criteria for RESCHU-SA datasets. Since the hacking detection task is a unique task in the RESCHU-SA interface, observations related to this task are clustered to reduce the total observation number.

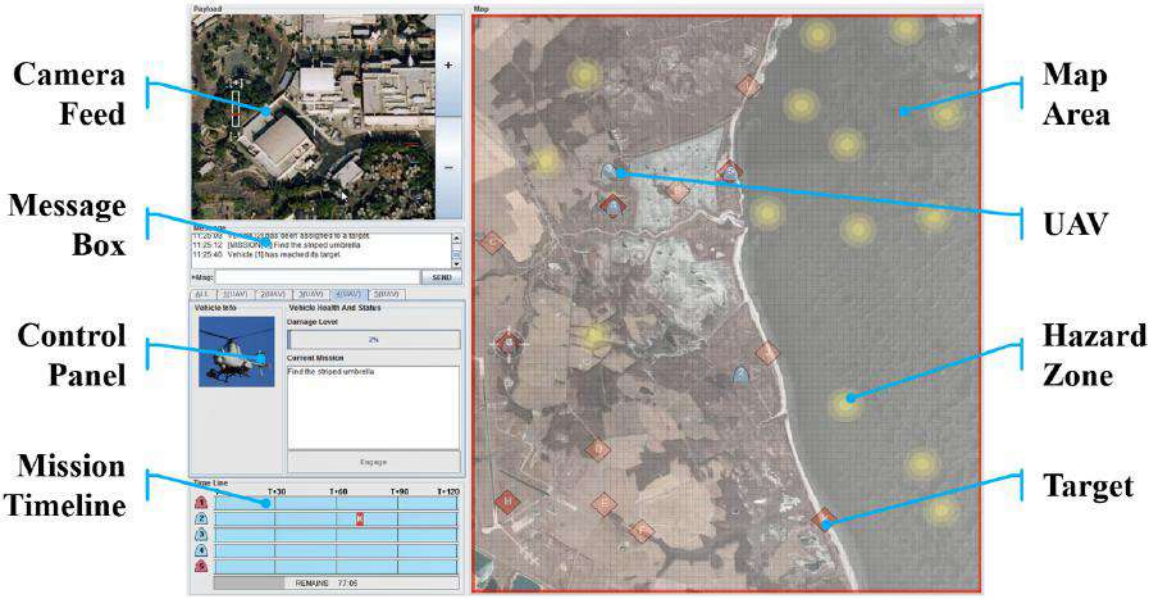


Figure 3.7: The RESCHU interface.

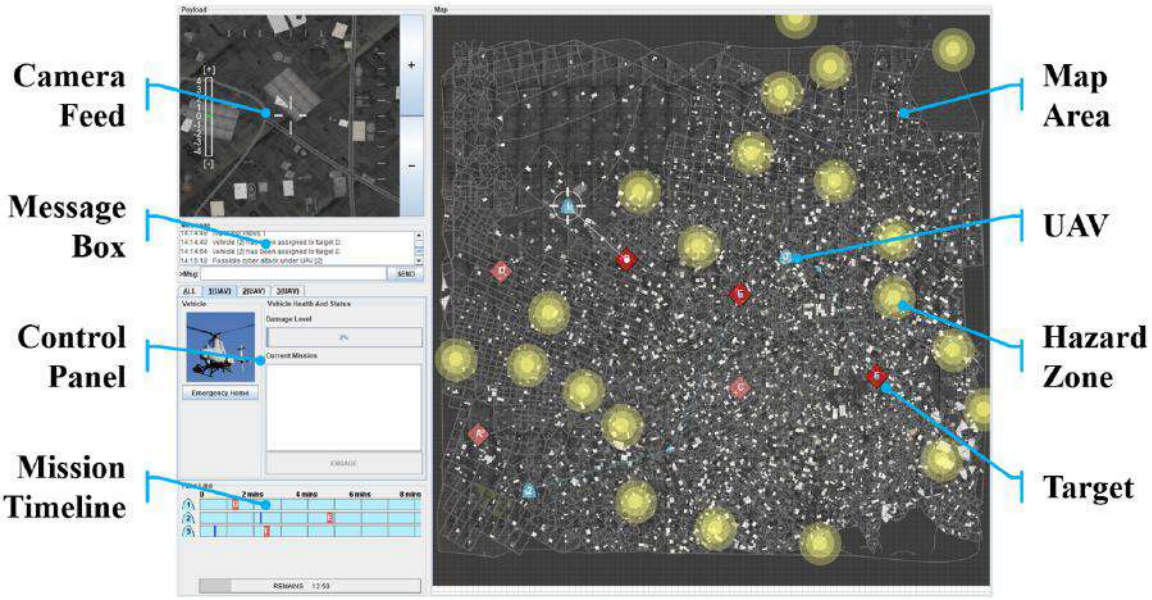


Figure 3.8: The RESCHU-SA interface.

Table 3.3: Observations from the RESCHU and RESCHU-SA experiment platforms

Shared observations in both interfaces		
1 Add waypoint	2 Move waypoint	3 Delete waypoint
4 Move endpoint	5 Switch target	6 Engage task
7 Monitor UAV		
Unique observations in the RESCHU-SA interface		
8 Perceive hacking	9 Detection decision	10 Adjust zoom level

Based on the HMM model notation mentioned in Section 2.2, assume that model λ_1 is developed from RESCHU experiments with $M_1 = 7$ observations and model λ_2 is developed from RESCHU-SA experiments with $M_2 = 10$ observations. Thus, the first model has fewer observations than the second model. To align such observations, data points in the λ_2 training dataset, which is O_{λ_2}, need to be re-screened to match the M_1 observations.

3.3.2 The Viterbi Propagation with Observation Reduction

With this observation reduction approach, the number of observations in λ_1 and λ_2 are aligned to M_1 such that the dataset O_{λ_2} is reformulated to M_1 types of observations as O_{λ_1}. Thus, the emission matrices of λ_1 and λ_2 share the same number of columns, M_1. Applying λ_1 and λ_2 to a one-dimensional sequence $O_{seq1} = (o_1, o_2, \cdots, o_t)$, the Viterbi propagation in Equation (3.1) can be updated for both models.

$$V_{\lambda_1:t,s_t} = \max_{s_t \in S_1}(b_{\lambda_1:s_t \to o_t} \cdot a_{\lambda_1:s_{t-1} \to s_t} \cdot V_{\lambda_1:t-1,s_{t-1}}) \tag{3.5}$$

$$V_{\lambda_2:t,s_t} = \max_{s_t \in S_2}(b_{\lambda_2:s_t \to o_t} \cdot a_{\lambda_2:s_{t-1} \to s_t} \cdot V_{\lambda_2:t-1,s_{t-1}}) \tag{3.6}$$

Given that both models contain M_1 types of observation, they share the same expectations of emission probabilities.

In an HMM model, a higher number of hidden states could lead to a lower expectation of state transition probabilities and a highly differentiated model structure with a higher model complexity. However, a high hidden state number could cause high corresponding emission probabilities. Thus, regardless of the number of hidden states, different model structures would only have limited influence on the product of $a_{s_{t-1} \to s_t} \cdot V_{t-1,s_{t-1}}$ if models share the same number and type of observations.

Therefore, the difference in model likelihood values between both HMM models in Equations (3.5) and (3.6) would only be affected by the underlying patterns in the dataset, rather than by the Viterbi algorithm propagation process. In this case, the divergence measure value, which is directly calculated from $\log(P(O_{all}|\lambda_1))$ and $\log(P(O_{all}|\lambda_2))$, will reflect the quantitative measure of the similarity level between the two models more precisely. Thus, the observation reduction method preserves information from the original observation at a different abstract level by re-selecting observations. While this method will not significantly affect the resulting model development and comparison processes based on the theoretical analyses, a sensitivity test is needed for justification.

3.3.3 The Observation Reduction Sensitivity Test

To understand the reliability of such an observation reduction approach, a sensitivity test was conducted to evaluate how collapsing observations impacted the overall divergence distance metric by utilizing the strategy model comparison metric and developing divergence meshes. Considering that collapsing observations may cause loss of information, it is necessary to ensure that collapsing certain observations will not significantly change the divergence metrics.

Data sequences collected from the RESCHU-SA experiments were categorized into two data batches based on the order of experimental sessions. The sensitivity test was conducted on these two data batches. Given that the hacking detection task was uniquely

embedded in the platform, we combined hacking detection-related observations based on different levels of abstraction. In order to understand the impact of collapsing the data from 10 to 7 observations, the revised data selection criteria for 7, 8, and 9 observations are shown in Table 3.4.

Table 3.4: Observation reduction criteria for the sensitivity test

<table>
<tr><th></th><th>7 observations</th><th>8 observations</th><th>9 observations</th><th>10 observations</th></tr>
<tr><td>1</td><td>Add waypoint</td><td rowspan="6">Same as 7 observations</td><td rowspan="6">Same as 7 observations</td><td rowspan="6">Same as 7 observations</td></tr>
<tr><td>2</td><td>Move waypoint</td></tr>
<tr><td>3</td><td>Delete waypoint</td></tr>
<tr><td>4</td><td>Move endpoint</td></tr>
<tr><td>5</td><td>Switch target</td></tr>
<tr><td>6</td><td>Engage task</td></tr>
<tr><td>7</td><td>Hacking detection</td><td>Monitor UAV</td><td>Monitor UAV</td><td>Monitor UAV</td></tr>
<tr><td>8</td><td>-</td><td>Hacking detection</td><td>Hacking detection</td><td>Perceive hacking</td></tr>
<tr><td>9</td><td>-</td><td>-</td><td>Adjust zoom level</td><td>Detection decision</td></tr>
<tr><td>10</td><td>-</td><td>-</td><td>-</td><td>Adjust zoom level</td></tr>
</table>

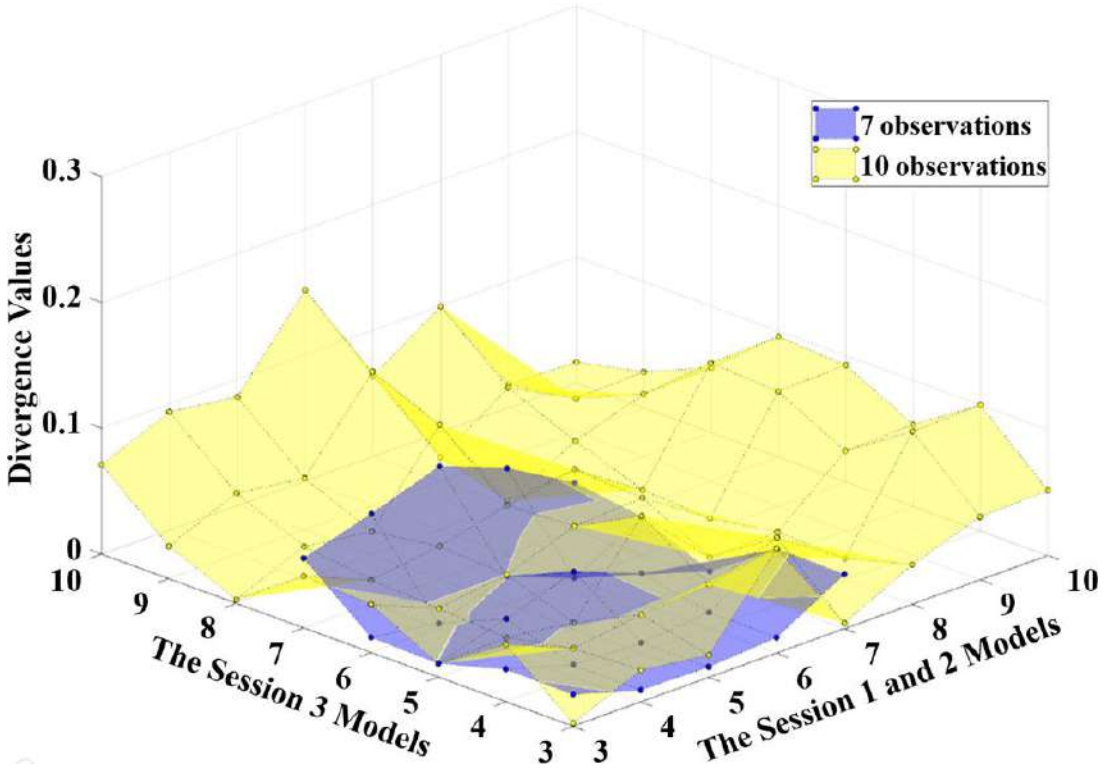
With the revised observation-reduced criteria shown in Table 3.4, HMM strategy models with all possible numbers of hidden states were retrained on the realigned datasets using the same methods discussed in the model development section. Then, visualizations of divergence measures between the HMM models were created following the divergence mesh

development process. Such divergence meshes were composed of divergence values from all possible combinations of model comparisons based on the different number of hidden states, or model structures.

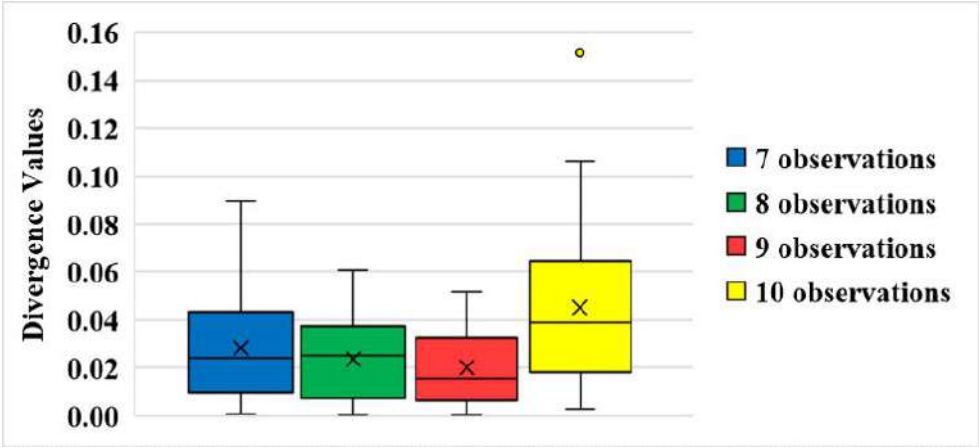
As shown in Figure 3.9a, two divergence meshes represent the 7 and 10 observation model comparisons from the two RESCHU-SA data batches with corresponding observation reduction criteria. The 8 and 9 states were omitted for clarity but shared the same space. Quantitatively, the ranges of the average divergence values of these four meshes are all within $0.02 - 0.05$ as shown in Figure 3.9b. Mann-Whitney tests with a family-wise significance level of ~ 0.008 $(0.05/6)$ were conducted on the divergence values between the two meshes. The statistical results show that the distribution of divergence values of both 8 and 9 observation meshes are significantly lower than the divergence distribution in the 10 observation mesh ($p < 0.001$ for both comparisons). However, the value distribution in the 7 observation mesh is not significantly different from the 10 observation mesh $(p = 0.017 > 0.008)$.

Thus, although the observation reduction criteria of 8 and 9 observations may change the underlying patterns in datasets and affect model comparisons with the original dataset of 10 observations, the reduction criteria of 7 observations only introduce a limited influence on the divergence measures. In this case, all hacking detection related observations could be collapsed to a single observation. In other words, the observation selection criterion for RESCHU-SA experiment models can follow the 7-observation rule instead of the original 10 observations without significant loss of information.

Thus, this section presents the observation alignment approach for comparing operator strategy models developed from similar HSC scenarios with non-equivalent observations by reducing observation types and re-parsing data into a higher level of abstraction. A sensitivity test was conducted based on a dataset collected from the RESCHU-SA experiment to show the potential impact of reducing observations related to a specific task.



(a)



(b)

Figure 3.9: The observation reduction sensitivity test. (a) Divergence value meshes based on different observation reduction criteria. (b) Boxplots of the divergence values.

3.4 Chapter Summary

This chapter presents the development of the operator strategy model comparison metric, which is the first main contribution of this dissertation. The purpose of building such a comparison metric is to comprehensively and quantitatively measure the difference between HMM-based operator strategies extracted from HSC scenarios. By utilizing this comparison metric, researchers can quantitatively measure the magnitude of potential impacts from scenario changes on operator strategies in order to evaluate such changes. Specifically, a data quantity prerequisite and an observation alignment approach are described in this chapter as two important parts of the comparison metric.

Chapter 4

Quantitative Operator Strategy Comparison References

The previous chapter presents the operator strategy model comparison metric based on the divergence measure with two prerequisites of the data quantity requirement and the observation alignment approach. This comparison metric provides researchers quantitative measures of differences between strategy models developed from various HSC scenarios with different settings. However, the practical meaning of the magnitudes of quantitative model difference measures is still not clear to researchers. For example, it is not known whether the average divergence distance value represents a significant, or a negligible, impact on operator strategies in HSC settings. Thus, comparison references are needed as baselines for researchers.

This chapter provides quantitative strategy comparison references by developing and comparing operator strategy models from multiple HSC human-subject experiments with certain factor changes, such as different operator groups, similar control interfaces with additional tasks, and different scenarios. With such resulting references, researchers can quantitatively evaluate whether specific changes in their HSC scenarios can significantly affect operator strategies and the possible magnitude of the impacts. This chapter first presents the development of the strategy model comparison baselines, which represent general operator individual strategy variances across HSC scenarios. This chapter also presents a continuum of strategy model comparisons to generate a continuous reference metric covering different changes in HSC scenarios for researchers.

4.1 Comparisons 1 and 2

This section presents the strategy model comparison baseline. This reference is developed between two data batches collected from two human-subject experiments respectively with the same participants and the same control interface with repeated settings. Thus, this reference is expected to present the least difference between operator strategies since the same operators executing the same tasks align with the human notation of sameness. Then this baseline will be compared to increasingly different scenarios.

4.1.1 Data Generation and Experiment Sessions

In order to establish comparison references, model training data for such a continuum of strategy model comparisons was collected from human-subject experimental sessions where participants controlled multiple drones to conduct high-level tasks using two different interfaces [127, 128, 62, 129]. Experimental sessions were conducted on the RESCHU interface [104], shown in Figure 3.7, and the RESCHU-SA interface [62], shown in Figure 3.8. Their interfaces are introduced in Section 3.3, and other detailed descriptions about both interfaces and experimental sessions are illustrated in Appendices A, B, and C. Specifically, four experimental sessions were conducted using the RESCHU-SA interface. The visualization shown in Figure 4.1 illustrates all strategy model comparisons related to RESCHU and RESCHU-SA experimental sessions. Four comparisons between strategy models were conducted to quantitatively measure the differences in operator strategies across these experimental sessions as listed in Table 4.1.

In addition to these four comparisons, an extra model comparison was also conducted to explore operator strategy differences with different interfaces and tasks. Such a comparison was between the StarCraft II game mentioned in Chapter 3 and the RESCHU-SA interface. Thus, as listed in Table 4.1, five strategy model comparisons are included in the continuum in this chapter.

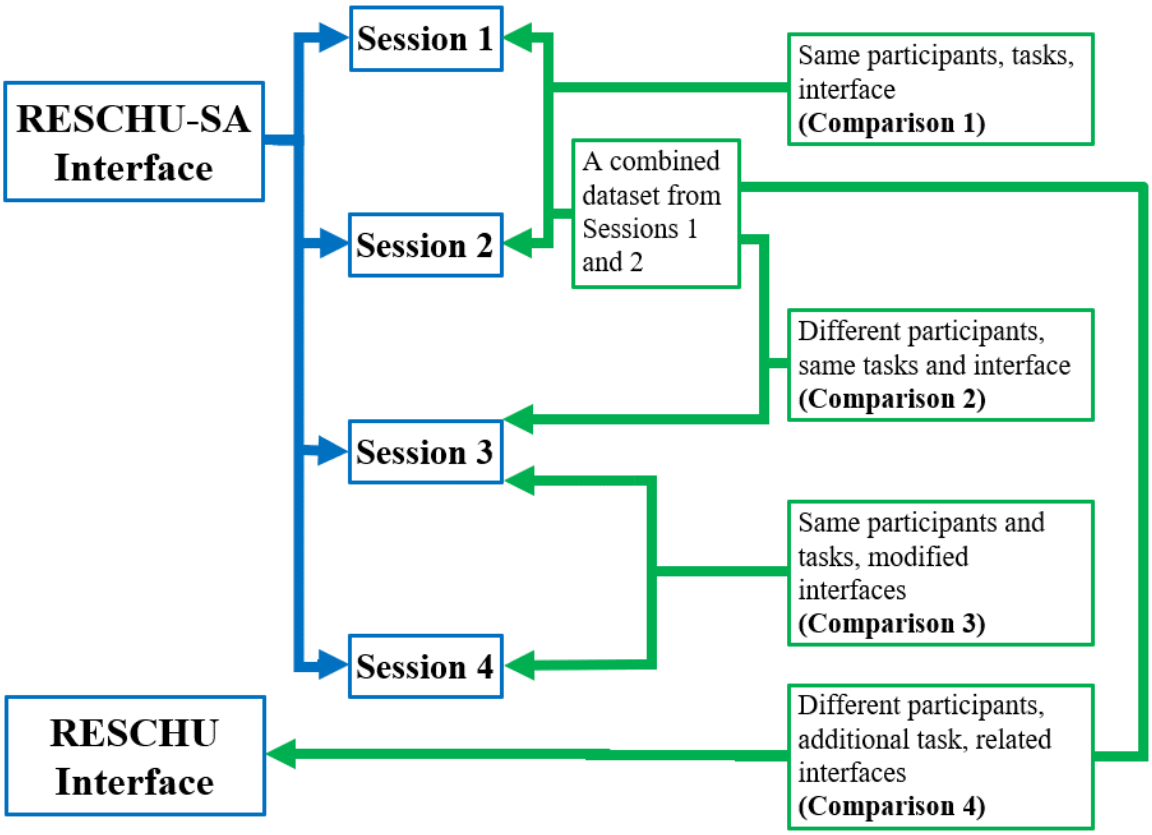


Figure 4.1: Comparisons between RESCHU and RESCHU-SA experimental sessions.

In the first comparison between Sessions 1 and 2 within the RESCHU-SA interface, strategy models were compared with the same interface, tasks, and participants. Thus, the expectation of the divergence measure of this comparison would be the least. This comparison is considered as the baseline for no significant difference in strategy model comparisons. This baseline also presents the strategy variety introduced by participants' individual variance.

For the second comparison among Sessions 1, 2 and 3, we compared the same interface and same task, but with different groups of operators. Our expectation was that this comparison would only yield a slightly increased divergence measure because of the potential variance brought by different operators. The third comparison was between Sessions 3 and 4, which shared the same participants and tasks, but with slightly modified interfaces.

Table 4.1: Comparisons between experimental sessions and interfaces

Comparison	Scenario settings	Corresponding sessions
1	Same participants, tasks, interface	Session 1 vs. Session 2
2	Different participants, same tasks and interface	Sessions 1+2 vs. Session 3
3	Same participants and tasks, modified interfaces	Session 3 vs. Session 4
4	Different participants, additional task, related interfaces	RESCHU vs. RESCHU-SA
5	Different participants, task, interfaces	StarCraft II vs. RESCHU-SA

Session 4 provided a decision support system, which was not embedded in other sessions, to assist participants in UAV hacking detection tasks. The expectation of this comparison was to have a higher divergence measure than the baseline and the second comparison.

The fourth comparison, which was between the RESCHU and RESCHU-SA interface, was expected to present a larger divergence measure than the third comparison. Because the RESCHU-SA interface is a modified version of the RESCHU interface, and the RESCHU-SA interface provides an additional task of UAV hacking detection. The fifth comparison, which was between the StarCraft II game and the RESCHU-SA interface, was expected to have the highest divergence measure among these comparisons because the participant groups, interfaces, and tasks were all different such that the resulting strategy models were expected to be significantly different.

The visualization of the hypothesis of quantitative strategy model comparisons with increasingly different scenarios is shown in Figure 4.2, in which the horizontal axis repre-

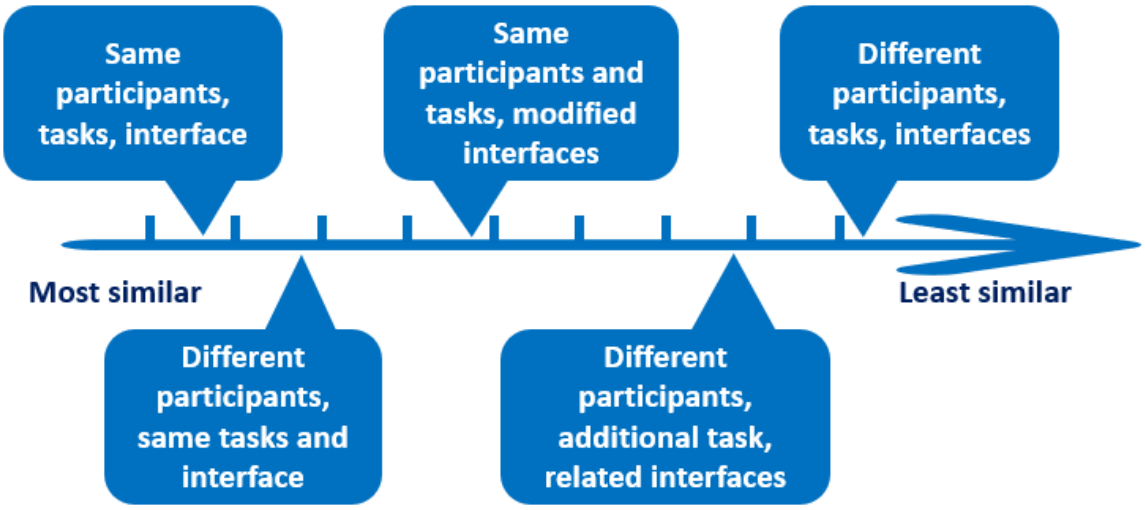


Figure 4.2: The model comparison hypotheses based on increasingly different scenarios.

sents the relative magnitude of model comparisons. These hypotheses are verified in this chapter with quantitative magnitude measures.

4.1.2 Model Comparison Results

HMM strategy models were developed based on observations, which were parsed from operator actions, collected during experiment sessions as listed in Table 2.2. As mentioned in the background section, HMM models were trained using the unsupervised multi-sequence Baum-Welch algorithm [77], which is a common expectation-maximization (EM) algorithm. Specifically, to increase the confidence of the training results, more than 100 randomly generated initializations were used in the HMM model training process for each specific model structure with a certain number of hidden states, and the resulting model with the highest data fitting likelihood was selected.

To quantitatively determine the baseline, or the "zero" benchmark point, of strategy model differences, we focused on comparing the two experimental sessions conducted on the RESCHU-SA interface with the same participants as highlighted in the red rectangle in Figure 4.3. Specifically, the 10 RESCHU-SA interaction observations, shown in Table 2.2,

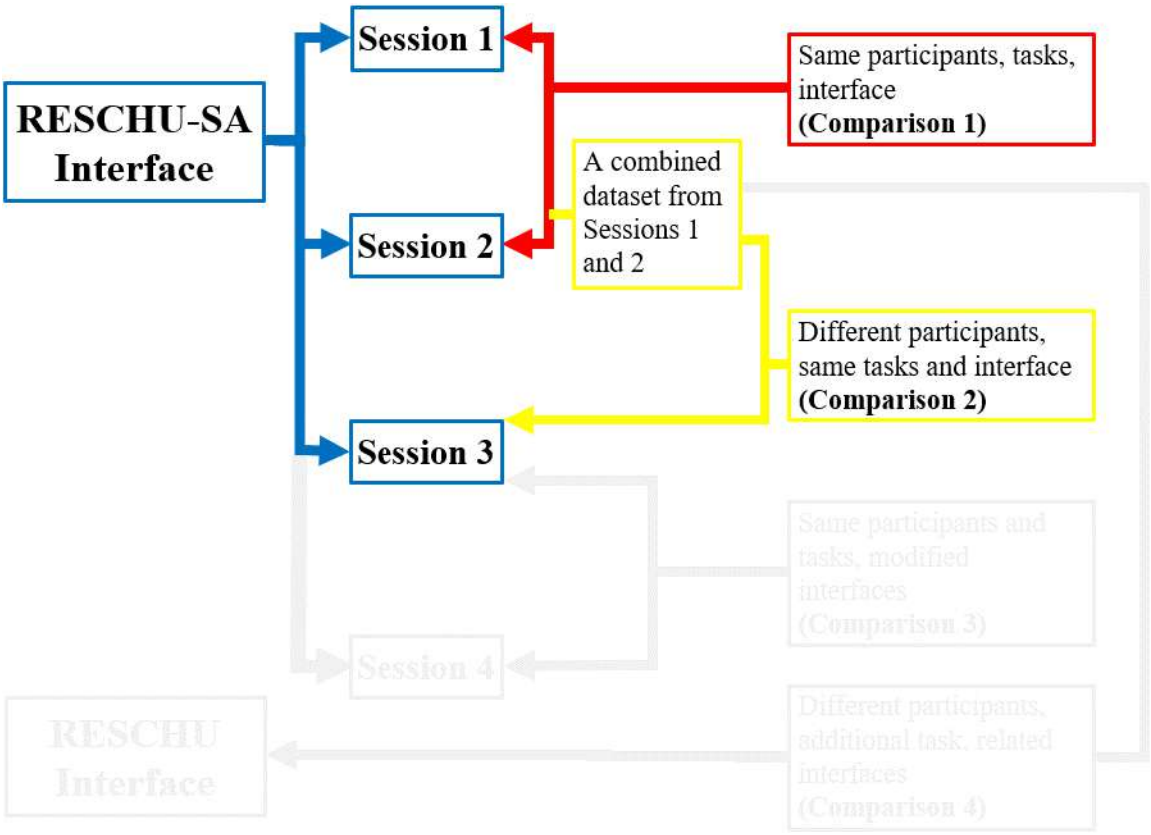


Figure 4.3: The baseline comparison and the comparison with different participant groups.

were used for this comparison and divergence meshes were plotted. Understanding that the RESCHU-SA interface provides three primary tasks of the UAV navigational task, the imagery analysis task, and the UAV hacking detection task, we consider that the minimum hidden state number in model comparisons is 3. Also, given that hidden states represent abstract cognitive groupings, the maximum number of hidden states should not be greater than the number of observation types, which is 10.

Similar to the baseline comparison, we compared strategy models developed from Sessions 1, 2, and 3 using the strategy model comparison metric to evaluate the potential impact of different participant groups on operator strategies. As highlighted in the yellow rectangle in Figure 4.3, the first set of strategy models were developed from the combined dataset of Sessions 1 and 2, and the second set of models were developed from Session 3.

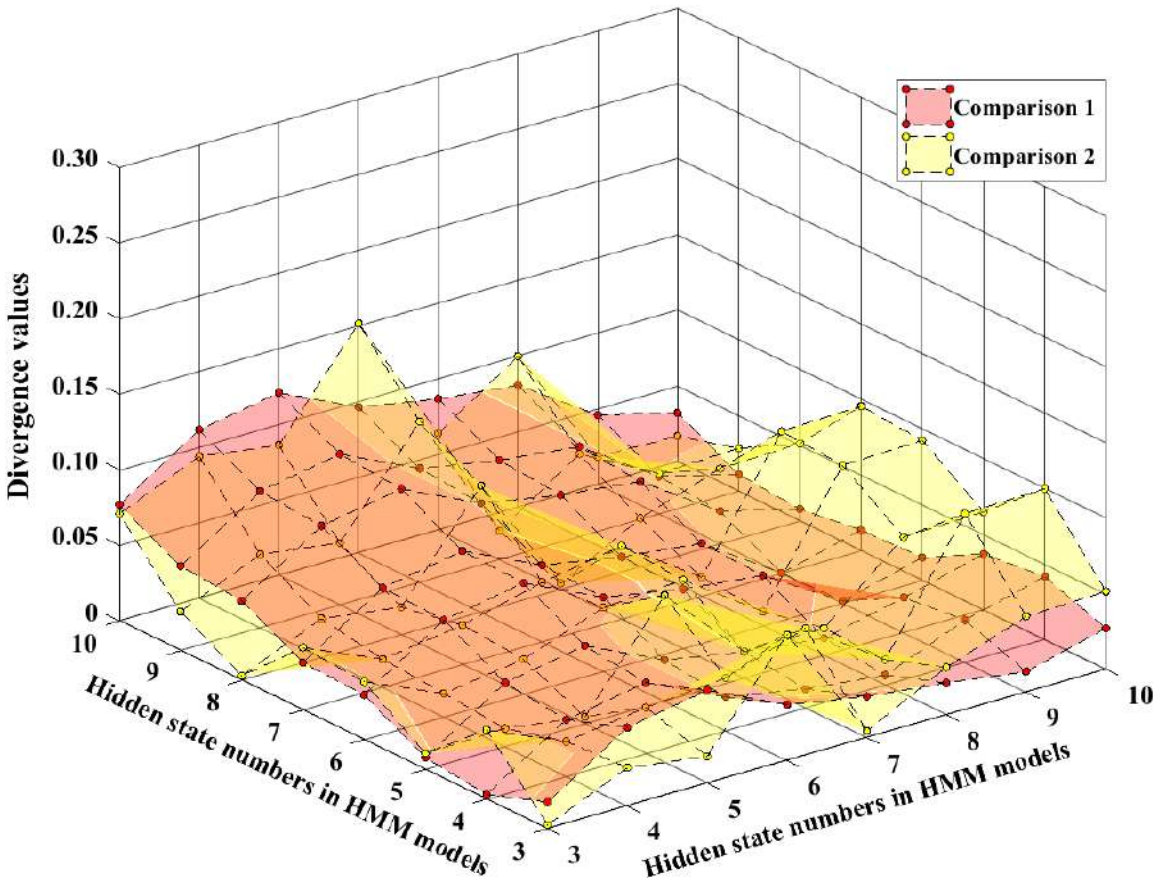


Figure 4.4: Divergence value meshes for the baseline comparison and the comparison with different participant groups.

While Sessions 1 and 2 shared the same group of participants, Session 3 had another group of participants. Thus, the comparison between strategy models developed from these two data batches illustrate the magnitude of the potential impact of different participants on operator strategies.

As shown in Figure 4.4, we plotted both meshes for these two comparisons, which shared an average divergence value of 0.045 with SD (standard deviation) = 0.029 and SD = 0.033, respectively. Thus, the baseline for strategy model comparisons without any changes in the HSC scenario can be considered as an average divergence value of 0.045. The two meshes in Figure 4.4 are interlaced and cannot be distinguished clearly. A non-

parametric Mann-Whitney test with a significance level of $\alpha = 0.05$ was conducted and showed no significant difference between these two divergence value distributions ($p = 0.768$). Given that both experiments had more than 35 participants, the high similarity between these two comparisons indicates that different participant groups introduce limited variability in participant overall strategies between experimental sessions with the same interface and tasks. These results align with the experimental data in that there was no significant difference in overall participant performance. Thus, this divergence mesh result establishes the baseline that in terms of human-in-the-loop strategies in an HSC scenario, what it means to be similar or have no impact from factor changes can be roughly measured at 0.045 using the strategy comparison metric.

4.2 Comparison 3

This section presents a strategy model comparison between Sessions 3 and 4 with the same participants and tasks, but slightly different interfaces. While both Sessions 3 and 4 are conducted on the RESCHU-SA interface, Session 4 provides a decision support system for participants to assist them with the UAV hacking detection task. This comparison is expected to have a larger divergence measure than the baseline. Also, this comparison result can provide insights into the effectiveness of such a support system.

4.2.1 Scenarios with Different Interfaces and an Assistant Tool

The visualization of the comparison between Sessions 3 and 4 is shown in Figure 4.5. Session 3 repeated the experimental settings in Sessions 1 and 2, but with a different group of participants. Session 4 shared the same participants with Session 3, and Session 4 also repeated the three major tasks provided by the RESCHU-SA interface. Further experimental details of Sessions 3 and 4 can be found in Appendix B.

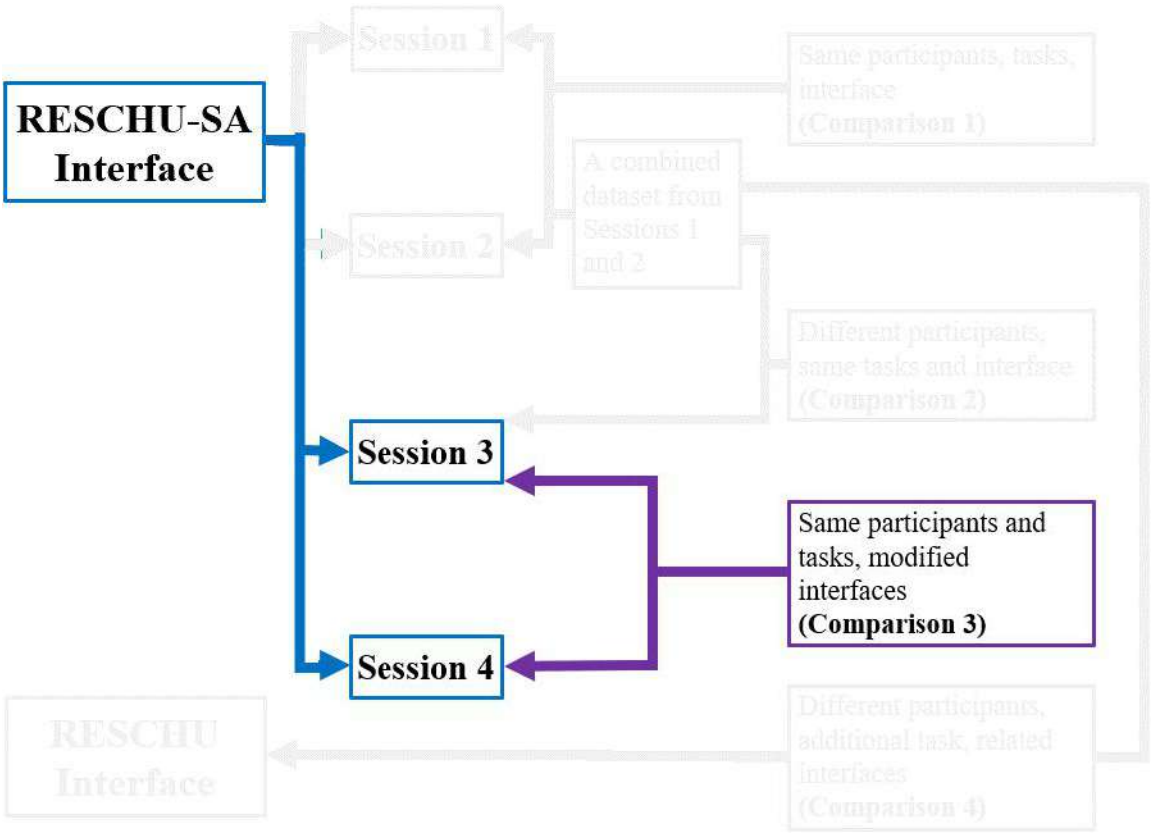
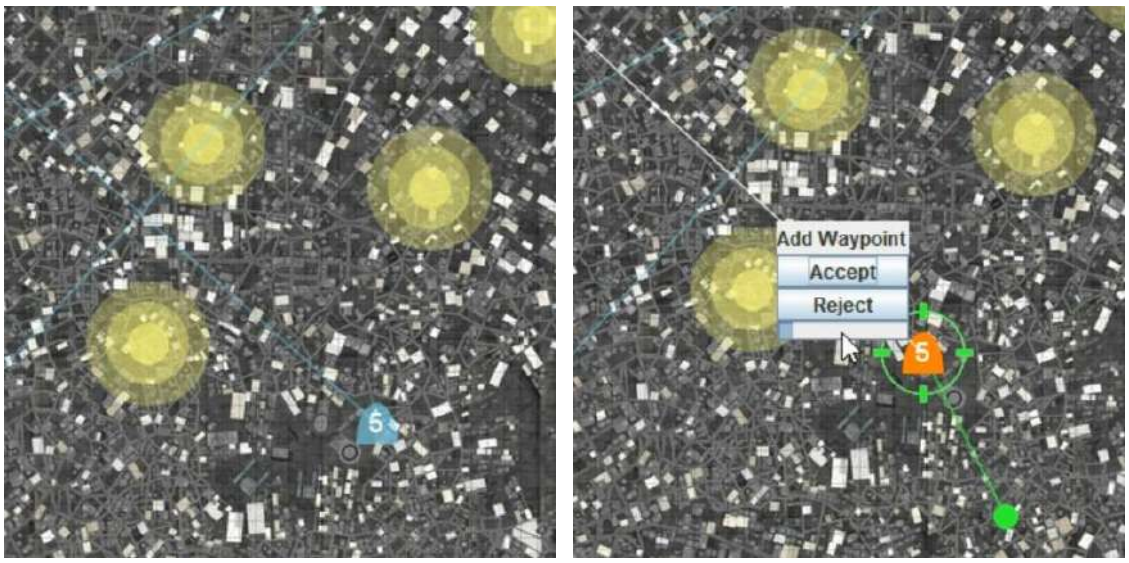


Figure 4.5: Model comparisons between different interfaces with an assistant tool.

It is worth noting that in Session 4, the RESCHU-SA interface was embedded with a decision support system, which could provide suggestions and simplify the procedure of detecting potential UAV hacking events for participants. This decision support system was developed based on participants' hacking detection performance and hacking event locations in Sessions 1 and 2 since all these experimental sessions shared the same simulated map as shown in Figure 3.8. The difference on the map area when participants received UAV hacking notifications is illustrated in Figure 4.6.

In Sessions 1, 2, and 3, when hacking notifications occurred and participants acknowledged the notification, the map area remained the same. Participants were required to select notified UAVs and determine potential hackings by themselves. Figure 4.6a illustrates an example of a notified UAV without any extra support. In Session 4, when the



(a) (b)

Figure 4.6: RESCHU-SA interfaces with/without the hacking detection support system. (a) The map area without the support system. (b) The map area with the support system.

interface prompted a hacking notification, a pop-up window emerged simultaneously with suggestions of adding waypoints or changing assigned endpoints for the notified UAV. An example of the suggestion window is shown in Figure 4.6b. Participants can accept such suggestions to detect potential hackings or reject suggestions if they tend to follow their detection strategies. If they accept a waypoint-oriented suggestion as in Figure 4.6b, a new waypoint, as highlighted in green, will be automatically added to the trajectory of the notified UAV, and the UAV will start to approach the new waypoint immediately. As shown in Table 4.2, a new observation, "Suggestion decision", was parsed and collected from Session 4 to represent the action of accepting or rejecting suggestions from the decision support system.

In order to compare strategy models developed from Sessions 3 and 4, observations from these two sessions need to be aligned. Thus, the "Suggestion decision" observation in the Session 4 dataset needs to be clustered with another observation. Considering that to

accept or reject suggestions is also a part of the initialization process of hacking events, the "Suggestion decision" observation is clustered with "Perceive hacking" as one observation. Thus, after reparsing observations in Session 4, datasets from these two sessions share the same 10 observations shown in Tables 2.2 and 4.2.

Table 4.2: HMM observations from Sessions 3 and 4 in the RESCHU-SA interface

Shared observations in Session 3 and 4			
1 Add waypoint	2 Move waypoint	3 Delete waypoint	4 Move endpoint
5 Switch target	6 Engage task	7 Monitor UAV	8 Perceive hacking
9 Detection decision		10 Adjust zoom	
Unique observations in Session 4			
11 Suggestion decision			

4.2.2 Model Comparison Results

HMM models were developed on the re-parsed datasets and compared based on the strategy model comparison metric. The strategy model comparison results are shown in the divergence mesh in Figure 4.7. The average divergence value of this divergence mesh is 0.068 with a standard deviation of 0.041. The divergence value distribution of this divergence mesh was compared to the baseline mesh and the second comparison mesh mentioned in the previous section via non-parametric Mann-Whitney tests with $\alpha = 0.05$. Test results showed that the divergence distribution of this mesh was significantly different from the other two meshes $(p < 0.001)$.

Compared to the average divergence (0.045) of the baseline, the average of this comparison is $0.068 - 0.045 = 0.023$ higher. Thus, utilizing an assistant tool, the decision support

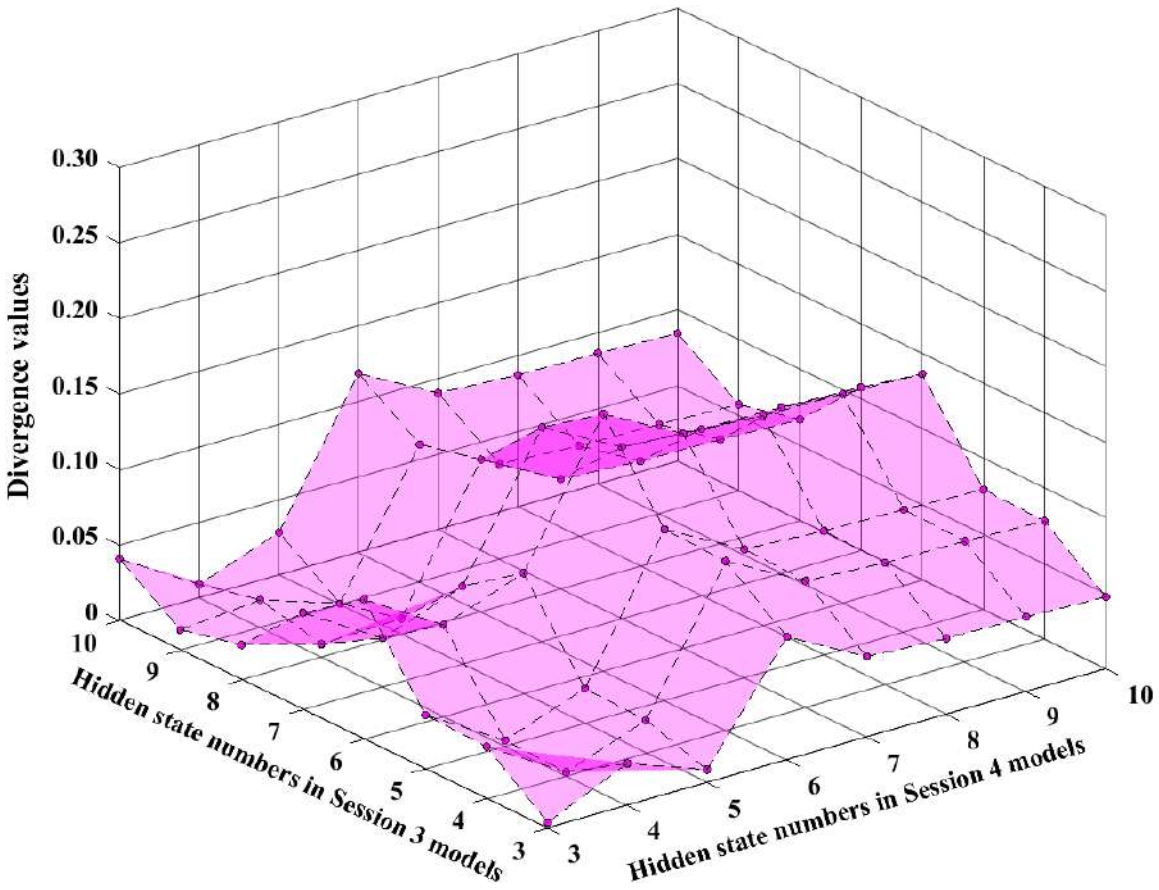


Figure 4.7: Divergence meshes for the model comparison between Sessions 3 and 4.

system provided in Session 4, in the RESCHU-SA interface will introduce changes in operator strategies. Such strategy changes can be quantitatively measured by an extra 0.023 divergence increase above the comparison baseline. Since the divergence value distribution of the second comparison mesh was not significantly different from the baseline mesh, different operator groups did not introduce changes to operator strategies. Thus, we can infer that applying an assistant tool in HSC scenarios will introduce a clear impact with a larger divergence metric measure on operator strategies than just changing operator groups.

This comparison result also provides a measure of the effectiveness of the decision support system. Researchers can evaluate the usage of the assistant system from such a measure. Based on the usage analysis of the decision support system, participants accepted

65.4% of all suggestions for hacking detections and rejected the other suggestions. Thus, the intended design of this assistant tool did not work as desired. This fact is also reflected in the limited increase of the average divergence measure in this divergence mesh. While the assistant tool affected operator strategies, its impact was measured with a limited and relatively small divergence value increase of 0.023 from the baseline. Based on the detailed analyses with developing hacking detection strategy models mentioned in Appendix B, participants were revealed to insist on their preliminary detection strategies even with supports from the assistant tool. In summary, this section demonstrates that strategy model comparisons can provide researchers insights about the effectiveness of intended interface designs, or applying supportive functions, by quantitatively measuring potential impacts on operator strategies.

4.3 Comparison 4

This section presents the fourth comparison with different interfaces and an additional task. This comparison provides reference for researchers on how an additional task in HSC scenarios potentially affects operator strategies. The observation alignment approach was utilized for comparing models developed with non-equivalent observations.

4.3.1 Scenarios with Different Interfaces and an Additional Task

The comparison presented in this section is between the RESCHU and RESCHU-SA interface, as shown in Figure 4.8. Specifically, data from Sessions 1 and 2 were used as the RESCHU-SA dataset because Sessions 1 and 2 shared the same experimental settings. Table 3.3 shows the observations for HMM strategy model development from the RESCHU and RESCHU-SA interfaces. While these two interfaces share 7 observations, the RESCHU-SA interface contains 3 other observations related to the additional task of

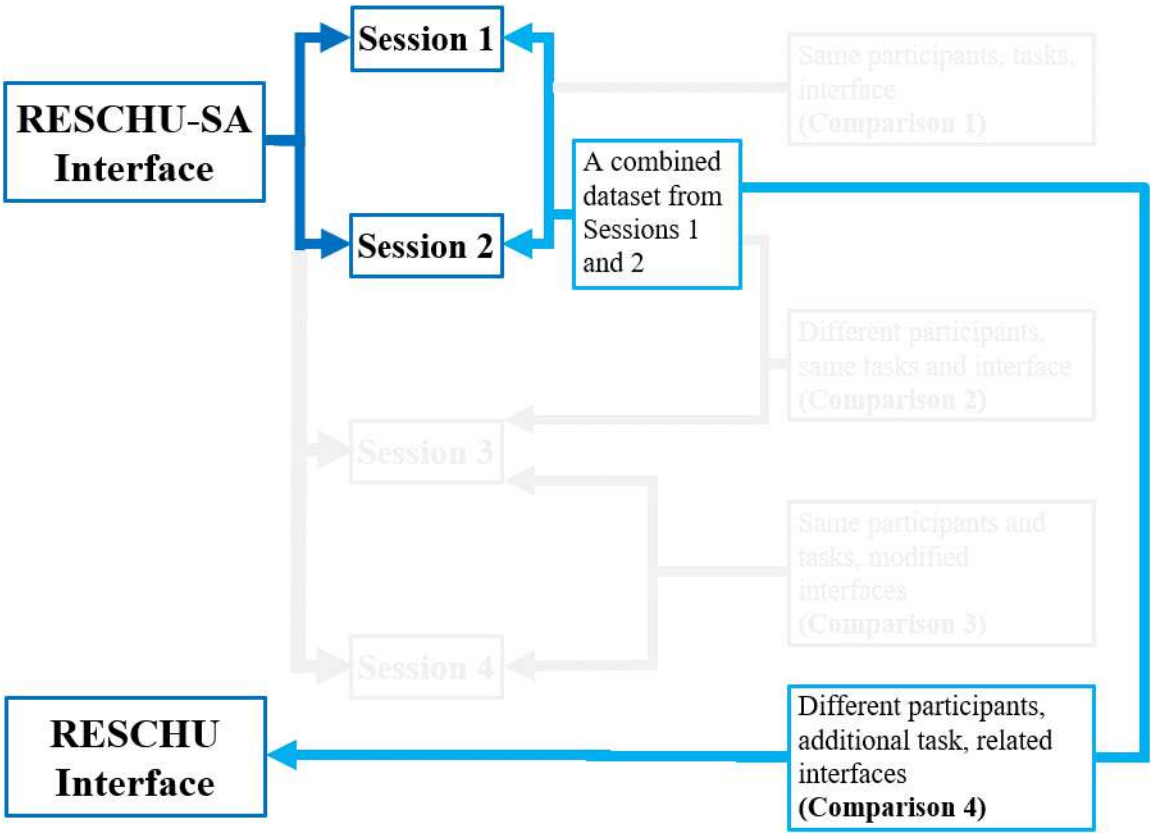


Figure 4.8: Model comparisons between different interfaces with an additional task.

the UAV hacking detection. Thus, we need to align observations for these two interfaces in order to conduct strategy model comparisons. Based on the observation reduction sensitivity test mentioned in Section 3.3, we can compare HMM models between these two interfaces using the re-parsed 7 observations shown in Table 4.3.

4.3.2 Model Comparison Results

The model comparison results are shown in the divergence mesh in Figure 4.9, which illustrates the differences in strategy models developed between the RESCHU and RESCHU-SA interfaces. The divergence mesh has a value range of $0.40 - 0.75$, which indicates a relatively large difference compared to the baseline comparison. The average divergence

Table 4.3: Revised HMM observations from RESCHU and RESCHU-SA interfaces

1 Add waypoint	2 Move waypoint	3 Delete waypoint
4 Move endpoint	5 Switch target	6 Engage task
RESCHU	7 Monitor UAV	
RESCHU-SA	7 Hacking detection	

value of this mesh is 0.563 ($SD = 0.078$). Comparing to the value ranges of the two divergence meshes mentioned in Section 4.1, we can infer that the divergence distance metric captures differences between the two interfaces. Because the RESCHU-SA interface contains an additional primary task of UAV hacking detection, which is not provided in the original RESCHU platform, participants using RESCHU-SA had clearly different strategies and behavioral patterns compared to those using RESCHU.

It is worth noting that the meshes of Comparisons 1 and 2 are relatively flat. This flatness indicates model stability in capturing operator general strategies with 3 or more hidden states when two scenarios share the same interface and primary tasks. At the same time, the Comparison 4 mesh is relatively uneven across comparisons with all possible hidden states. Thus, comparisons of HMM strategy models with different interfaces may be less stable, which could be a result of the non-equivalent observation manipulations. Future studies can investigate how such manipulations affect the quantitative measures.

In this analysis, the most similar comparison is between the same operators using the same interface for the same tasks, Comparison 1. Comparison 1 is also considered as the strategy model comparison baseline. Comparisons 4 looks at different interfaces and tasks with different operators. The difference between these means can be measured at 0.518. Understanding that different groups of operators only introduce limited variance to the divergence metrics, such an mean difference can be considered as a quantitative similarity

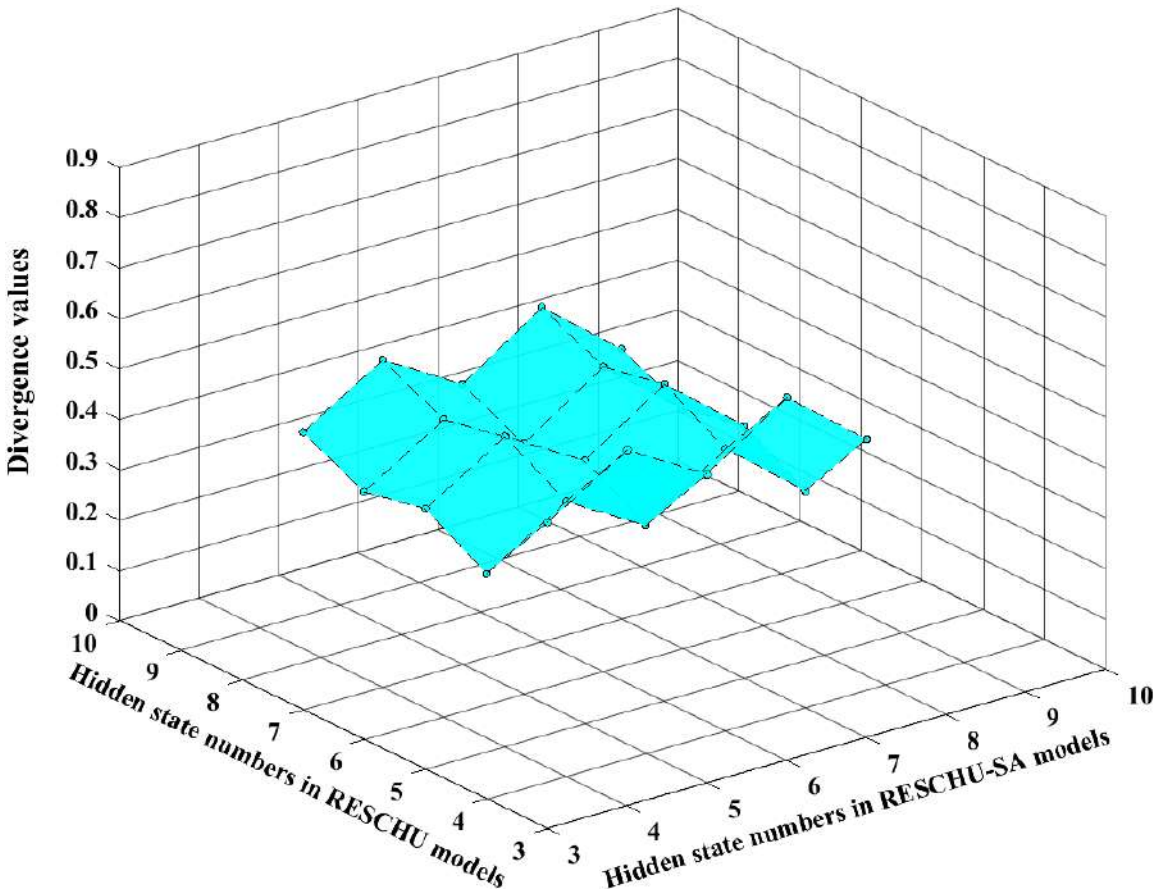


Figure 4.9: The divergence mesh for model comparisons between the RESCHU and RESCHU-SA interfaces.

metric for including an additional primary task in an HSC scenario. However, it remains to be seen whether this relative difference holds between other interfaces with different tasks and how it can be leveraged in various applications.

In summary, this section presents a model comparison between the RESCHU and RESCHU-SA interfaces with different participants and an additional task. This comparison establishes a quantitative reference for researchers that an additional task in HSC scenarios could introduce about a 0.518 increase above the baseline in the divergence measure.

4.4 Comparison 5

This chapter presents a strategy model comparison between two HSC scenarios with different interfaces and tasks. The first scenario is the RESCHU-SA experimental platform, which is mentioned in the previous sections and chapters. The second scenario is the StarCraft II computer game, which is mentioned in Section 2.2. The similarities between these two scenarios are that both can be considered as common HSC scenarios, and they share a similar number of high-level tasks. Thus, this section provides a comparison reference for researchers on how operator strategies will be modeled differently given two HSC scenarios with the same number of tasks but different interfaces and tasks.

4.4.1 Scenarios with Different Interfaces and Tasks

Similar to the strategy model comparisons in previous sections, data collected from Sessions 1 and 2 in the RESCHU-SA interface were used to develop operator strategy models. All data parsed from the StarCraft II game was also utilized, and this dataset contains only a single game map and all players selected a specific gaming character to represent a determined HSC scenario. The overall visualization of model comparisons is shown in Figure 4.10. Similar to the previous section, datasets from Sessions 1 and 2 are combined since they share the same participants, interface and tasks. Then, the strategy model comparison was conducted between the StarCraft II and the RESCHU-SA interface.

Observations from the RESCHU-SA interface are shown in Table 2.2 and observations from the StarCraft II game are shown in Table 3.1. Given that these two scenarios with different interfaces do not share any observation and that the StarCraft II game generates 15 observations, observation alignment is needed to compare strategy models across these two scenarios. The observation alignment process between RESCHU and RESCHU-SA in the previous section is straightforward such that we only need to reduce observations in the RESCHU-SA interface by reparsing hacking detection-related observations. However,

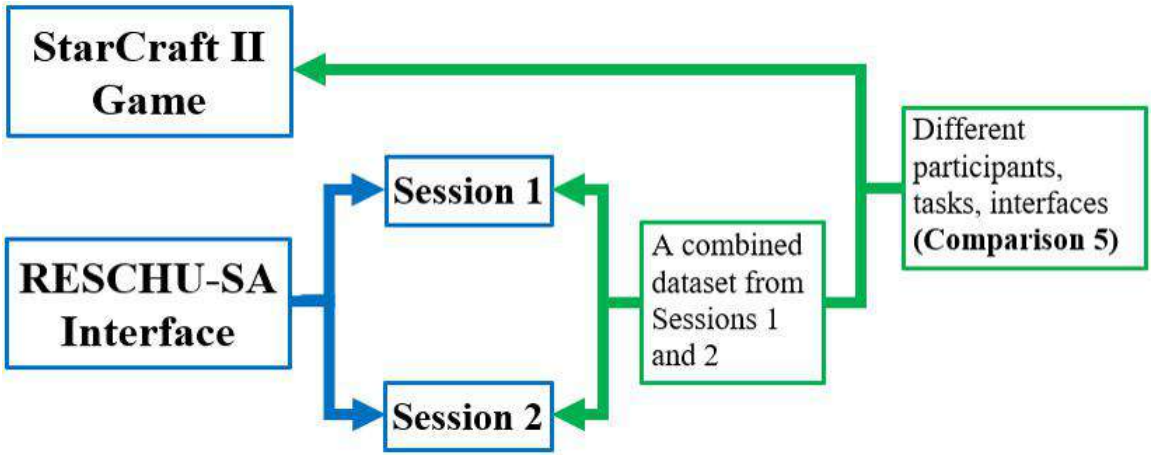


Figure 4.10: Model comparisons between the RESCHU-SA experiment sessions and the StarCraft II game.

observations from the StarCraft II game and the RESCHU-SA interface need to be aligned via a slightly complex approach.

The observation alignment process for this strategy model comparison contains two main steps: 1) reduce StarCraft II observations to the same number as the RESCHU-SA observations by clustering some observations, and 2) align observations based on the occurrence frequency rankings. Understanding that StarCraft II game generates 15 observations, the first step is to reduce the observation number to 10, which is equal to the number in RESCHU-SA. Some observations with a similar meaning or those that represent similar actions are clustered as a single observation, such as "Cancel" and "Stop" being combined as "Terminate". Other observations, which usually occur sequentially and represent a single task, are clustered as a new observation, such as "Lift" and "Land", which represent when players manage and relocate constructions, are clustered as "Lift/land". The revised StarCraft II observations are listed in Table 4.4.

After reducing StarCraft II observations from 15 to 10, the next step is to align observations between these two scenarios. The main goal of aligning observations is to have the same mathematical representations, or indices, to present two sets of observations from

Table 4.4: Revised StarCraft II observations

1 Effect	2 Lift/land	3 Harvest	4 Attack
5 Terminate	6 Morph	7 Train	8 Load/unload
9 Rally	10 Build		

two datasets in the divergence measure calculation. For these two datasets, we utilized the observation occurrence frequency rankings in each dataset to align observations. We consider that observations with the same frequency ranking represent a similar significance in a scenario. The occurrence frequency-based observation alignment result is shown as Figure 4.11. First, we determined the occurrence frequency ranking in the RESCHU-SA interface and obtained the pairing information between the observation indices and rankings. Then we obtained the rankings from the StarCraft II game and reordered them to the RESCHU-SA observation rankings. Thus, we can use a single set of indices to represent observations from both interfaces to develop divergence meshes.

4.4.2 Model Comparison Results

Observations are re-parsed and operator strategy models are retrained for the StarCraft II dataset. Then we developed a divergence mesh as shown in Figure 4.12. This mesh represents the comparison between StarCraft II models and RESCHU-SA models developed from Sessions 1 and 2. The average divergence value of this mesh is 0.654 (SD = 0.081), which is higher than the comparison with an additional task shown in the previous section. This divergence meshes shows a consistent and relatively flat shape with slight variance. This fact shows that regardless of model structures, the difference between the underlying patterns in these two datasets are consistent and stable. In other words, the difference between operator strategies from these two scenarios overwhelms the difference in strategies

Index	RESCHU-SA Obs	Freq	Rank	StarCraft Obs	Freq
1	Add waypoint	12.33%	4	Effect	9.13%
2	Move waypoint	9.18%	6	Lift/land	2.04%
3	Delete waypoint	2.38%	10	Harvest	0.36%
4	Move endpoint	14.44%	2	Attack	23.60%
5	Switch target	7.71%	7	Terminate	1.87%
6	Engage task	11.92%	5	Morph	6.28%
7	Monitor UAV	17.13%	1	Train	41.47%
8	Perceive hacking	5.92%	8	Load/unload	0.96%
9	Detection decision	5.79%	9	Rally	0.39%
10	Adjust zoom level	13.20%	3	Build	13.14%

Figure 4.11: Observation alignment based on occurrence percentage rankings.

captured by different model structures with different abstract levels.

In this section, we established a strategy model comparison reference for HSC scenarios with different interfaces and tasks. This comparison reference is measured at 0.654 using the strategy model comparison metric. This measure has the highest magnitude among all measures in this continuum of strategy model comparisons we conducted. Such a result is expected because different HSC scenarios with different interfaces and tasks are considered as the largest changes comparing to other scenario changes. This quantitative reference also illustrates relative magnitude differences to other comparison references.

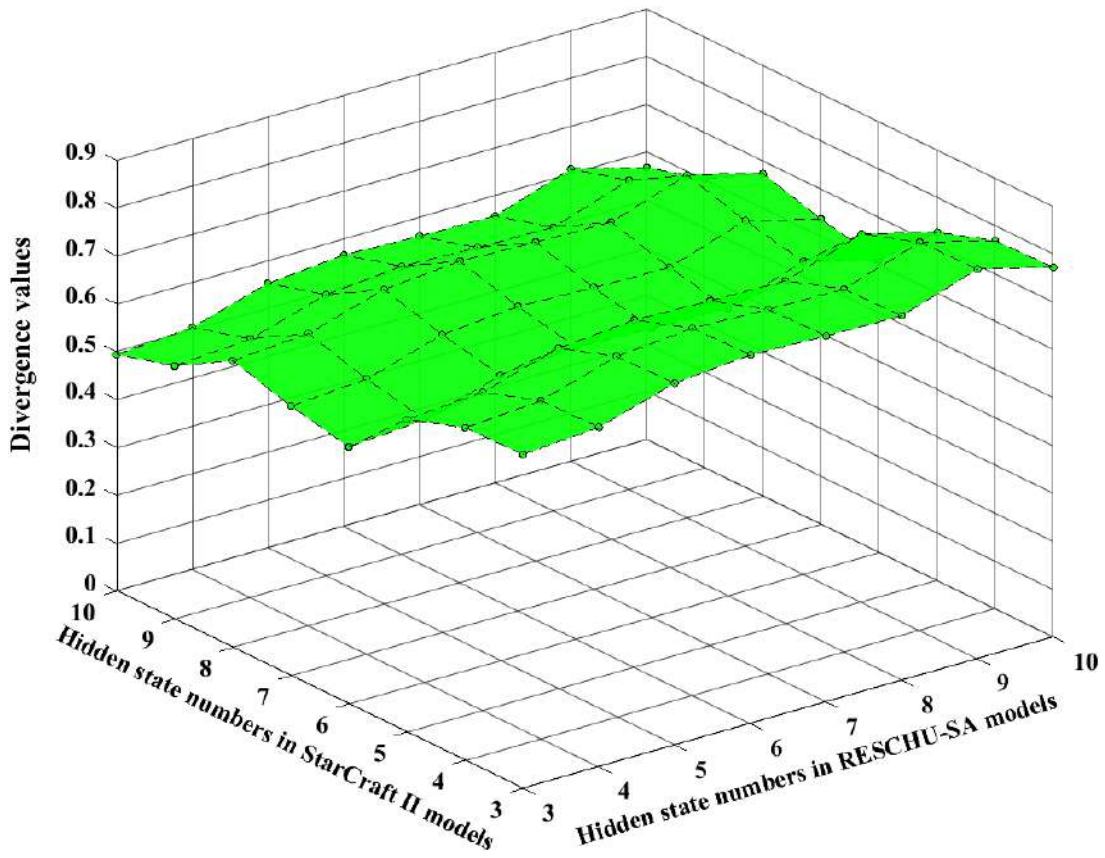


Figure 4.12: The divergence mesh for model comparisons between the StarCraft II and RESCHU-SA interfaces.

4.5 Chapter Summary

This chapter presents a continuum of five strategy model comparisons, including 1) a strategy model comparison baseline, which is measured at 0.045 and presents the benchmark of comparisons across HSC scenarios with the same settings, 2) a model comparison reference with different participants, sharing the same divergence average with the baseline and showing that changing operators while keeping other settings will not significantly affect operator strategies, 3) a model comparison between modified interfaces with an assistant tool, having a divergence average of 0.068 and showing that an additional decision support system will change operator strategies to a limited degree, 4) a comparison reference between related interfaces with an additional task, demonstrating that an additional task can result in an increase of 0.518 on divergence measures from the baseline, and 5) a comparison reference between HSC scenarios with different interfaces and tasks, showing that the difference between operator strategies modeled from two different HSC scenarios can be measured at a divergence value range of 0.654. The average divergence values of these comparison meshes were visualized in Figure 4.13, illustrating the magnitudes of these average divergences and quantitative distances among them.

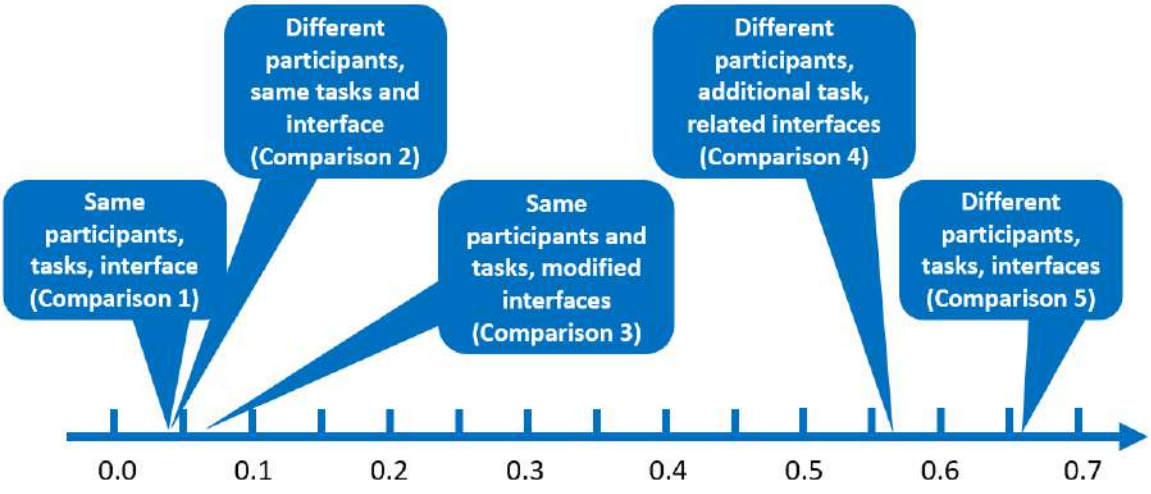


Figure 4.13: A continuum of strategy model comparisons.

Given the hypothesis of strategy model comparisons with increasingly different scenarios in Figure 4.2 in Section 4.1.1, model comparison results in this chapter verified this comparison ranking hypothesis. As the baseline, the comparison with the same participants, interface, and tasks presents the least divergence measure, representing the highest similarity. While different participant groups were expected to slightly affect operator strategies, model comparison results showed changing participants did not affect strategy model outcomes.

An assistant tool embedded in a slightly modified interface was expected to introduce certain changes in operator strategies, and model comparison results verified this impact. Also, the factors of additional tasks and different interfaces brought significant impact to operator strategies as expected. Furthermore, model comparison results illustrated the magnitudes of strategy difference measures for these factors. This continuum of strategy model comparisons established a preliminary reference framework for illustrating the magnitude of impacts on operator strategies. Future studies can extend this reference framework by exploring influences from other HSC factors.

With such references, researchers can quantitatively evaluate the impacts on operator strategies from applying certain changes, such as intended designs and additional tasks, in HSC scenarios. Also, the development of such a continuum of references demonstrates the general process of evaluating operator strategy changes across various HSC scenarios.

Chapter 5

Conclusion

As the human-automation interaction and human supervisory control-related applications develop rapidly, it is important to investigate operator strategies and measure how various changes in HSC scenarios may affect operator strategies. With measures of potential impacts from such changes, researchers can evaluate the magnitude of impacts and determine the effectiveness of specific designs in control interfaces or training programs for operators. This research first proposes an operator strategy model comparison metric and then establishes a set of strategy comparisons as a continuum of quantitative references.

5.1 Dissertation Summary

This dissertation started with the motivation for this research in Chapter 1, which was in the human-automation interaction field and specifically focused on human supervisory control scenarios. While many studies have been conducted to investigate how different control settings or different training programs may affect operator performance in conducting tasks, no previous research has quantitatively evaluated potential impacts from such scenario changes on operator strategies. Thus, the main gaps between previous studies and this research were 1) the methods of quantitatively comparing operator strategies to measure the differences between strategies and 2) the practical interpretations of such measures and how such measures could benefit HSC scenarios. Such gaps led to two main questions of this research: 1) how can operator strategies be quantitatively compared, and 2) what are the practical meanings of such strategy difference measures.

Chapter 2 first presented the background of human supervisory control with existing techniques for modeling human operators at different levels. Based on the results of pre-

vious studies, this research utilized the hidden Markov model to model operator strategies. Then this chapter illustrated the detailed structures of hidden Markov models with the strategy model development process. Also, this chapter provided an example of how to develop operator strategy models for a remote drone control HSC scenario from a simulation-based human-subject experiment.

Using HMMs as operator strategy models, researchers can investigate operator strategies. Then the next step was to compare strategy models to obtain the model difference measures. An operator strategy model comparison metric was proposed in Chapter 3. This comparison metric utilized the divergence distance measure to quantitatively calculate the difference between HMM strategy models. In this research, the divergence measure was extended to a divergence mesh method to provide a systematic and comprehensive difference measure between strategy models. By developing strategy models on existing datasets and comparing models using divergence meshes, two requirements were proposed as prerequisites for the comparison metric. Such prerequisites included 1) the data quantity requirement – 30 or more operator data sequences are required and also likely sufficient for effectively modeling operator strategies in HSC scenarios and 2) the observation alignment approach – if strategy models contain non-equivalent observations, then researchers need to realign observations in training datasets by reparsing observations and retraining models to conduct model comparisons.

Given the quantitative measures of strategy model differences, it was still unclear how the magnitude of such comparison measures map to meaningful degrees of difference in HSC scenarios. Thus, Chapter 4 presented a continuum of strategy model comparisons to establish baselines and references of strategy model comparisons to quantitatively illustrate the impacts from factor changes in HSC scenarios on operator strategies. This continuum of comparisons included 1) a general operator strategy model comparison baseline, 2) a comparison reference with different groups of operators, 3) a comparison between modified interfaces with an assistant tool, 4) a comparison reference between related interfaces

with an additional task, and 5) a comparison across two HSC scenarios with different interfaces and tasks. This continuum provided references for researchers to measure impacts on operator strategies while applying changes in HSC scenarios by utilizing the strategy model comparison metric and evaluating such impacts by referring to such references.

5.2 Contributions

The objectives of this research were to 1) describe operator strategies by developing strategy models, 2) develop a quantitative and systematic strategy model comparison metric to measure differences between operator strategies, and 3) explain practical meanings of strategy differences by establishing strategy model comparison baselines and references. Given such research objectives, two major contributions were presented in this research, including 1) developing a strategy model comparison metric, presented in Chapter 3, and 2) establishing strategy model comparison references, presented in Chapter 4.

5.2.1 The Development of Strategy Model Comparison Metric

In order to fill the gap in investigating impacts on operator strategies, this research proposed a strategy model comparison metric for comparing strategy models and a means to quantitatively measure strategy differences. The fundamental calculation in the comparison metric was the divergence distance measure, which calculated the difference between two HMM models by evaluating the model fitting probabilities on an evaluation dataset. As mentioned in Chapter 3, the divergence measure was extended to the divergence mesh by considering all possible strategy model structures to provide subjective and comprehensive model comparison measures. Also, from divergence meshes, researchers could investigate how different model structures, which capture operator strategy information in different abstract levels, could affect difference measure results. Two important components of the

metric were also proposed, which were that researchers need to have 30 or more operator data sequences to effectively model and compare operator strategies and researchers need to align observations for strategy model comparisons with non-equivalent observations. Thus, as the first step of investigating impacts on operator strategies, the strategy model comparison metric with its two components was considered as the first major contribution in this research.

The procedure in the comparison metric is shown in Figure 5.1. If researchers need to evaluate if certain changes in their HSC scenarios will affect operator strategies and cause potential issues, they can utilize this comparison metric and follow this procedure. The first step is to develop operator strategy models from HSC scenarios. Researchers need to ensure that they have collected 30 or more data sequences to effectively model operator strategies. If researchers fail to collect such an amount of data, the resulting models would be biased because of individual variance and cannot represent the general operator strategies across a scenario.

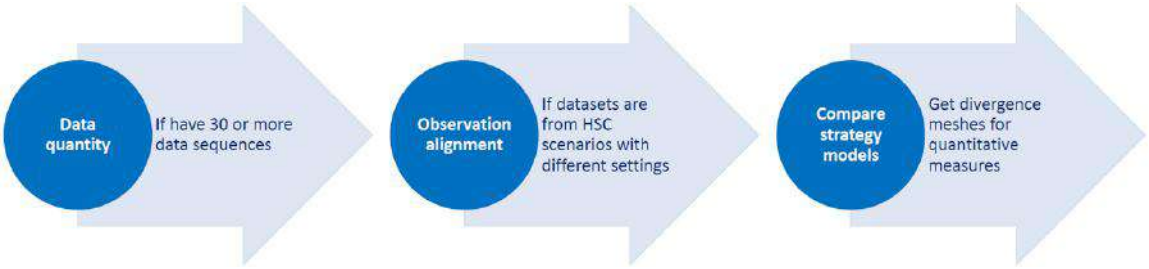


Figure 5.1: The flow of the strategy model comparison metric.

The second step for comparing models, if comparing strategy models from two datasets with non-equivalent observations, is to apply the observation alignment approach. For example, a certain change in an HSC scenario may generate new interactions between operators and control systems, and such new interactions will be parsed as new observations in the strategy model training process. In this case, these new observations will cause calculation failures in model comparisons using the strategy model comparisons metric

because these observations will cause Viterbi algorithm failures in obtaining the model fitting probability values on models developed from the other dataset. Thus, researchers need to align observations and retrain strategy models for the two datasets for comparisons with non-equivalent data types by reducing, re-parsing, or re-ranking observations.

Following these two steps, which are two important components of the comparison metric, researchers can develop divergence meshes to quantitatively measure operator strategy differences. Understanding that this metric requires all possible HMM model structures to construct divergence meshes, researchers can compare models developed from the same data batches and investigate the topology of outcome meshes to determine the optimal model structure for describing operator strategies. Moreover, the method of comparing models from the same dataset can also benefit researchers in determining the optimal training data parsing criterion.

5.2.2 The Establishment of Strategy Model Comparison References

Researchers need baselines or references to understand how such quantitative measures would map to practical meanings in HSC scenarios and evaluate the magnitudes of measures. Such needs led to the second major contribution of this research for which a continuum of strategy model comparisons were conducted to provide quantitative references for model comparisons among HSC scenarios with various settings.

Five quantitative references established in this contribution include 1) the strategy model comparison baseline, 2) the comparison reference with the same interface but different operators, 3) the comparison between modified interfaces with an assistant tool, 4) the comparison reference between related interfaces with an additional task, and 5) the comparison reference across HSC scenarios with different interfaces and tasks. As mentioned in Chapter 4, the RESCHU and RESCHU-SA experiment interfaces were utilized for data collection, and their datasets were used for strategy model development to establish such

baselines and references. Also, the StarCraft II gaming dataset mentioned in Chapter 3 was utilized in the fifth reference.

The first comparison of the baseline was developed based on two datasets collected from experimental sessions with the same settings. This baseline presents the benchmark point at which, without any factor changes, operators are expected to present the same general strategies, and such a similarity can be measured at 0.045 using the strategy model comparison metric. This baseline also illustrates the measure of operator individual strategy variances. The second comparison with different operators presents the same average divergence measure as in the first comparison. This fact shows that different groups of operators do not introduce a significant variance in resulting strategy models if researchers have a sufficient quantity of data.

The third comparison with a modified interface and an assistant tool was measured at a divergence average of 0.068, which is 51.1% larger than the baseline. While this measure is larger than the baseline, it can be considered as a relatively small divergence that such an assistant tool introduced limited impacts on operator strategies. The fourth comparison with a related interface and an additional task was measured at 0.563, which is over 12 times larger than the baseline. We can infer that the impact from an additional task is more significant than from an assistant tool. However, further studies are necessary to explain such a proportionality. Also, the fifth comparison with different interfaces and tasks was measured at 0.654. This measure shows the impact of changing HSC scenarios is larger than the impact brought by an additional task. This result is expected because an additional task can be considered as a partially changed scenario. Based on these quantitative measures, we can infer the ranking of impacts introduced by various changes in HSC scenarios. The relation between such changes and the proportionality and additivity of these measures is further discussed in the following sections.

With the comparison baselines, researchers can determine if specific changes in HSC scenarios will impact operator strategies by referring to the comparison baseline. All com-

parison references provide researchers a ranking of impacts from different factors based on the comparison metric measures to show the order of the significance levels of such factors. These references provide a continuum of quantitative measures that researchers can evaluate the magnitude and significance of how certain changes affect operator strategies. Furthermore, the measures of the impacts from different factors in HSC scenarios can provide researchers implications about the relations between such factors.

5.3 Generalizability and Limitations

The first main effort of this research is to develop a comprehensive operator strategy model comparison metric, which provides researchers with a procedure to train and compare HSC strategy models in order to obtain quantitative difference measures between models. The expected input for this metric is observation sequences parsed from interactions, such as mouse clicks or keyboard strokes, between operators and control interfaces. The expected output of this metric is divergence meshes, which represent differences between model fitting probabilities on evaluation datasets across all possible model structures using the divergence measure. Specifically, the average values of divergence meshes are usually used as the main indicator of model differences.

There are few limitations in using this strategy comparison metric. The first limitation is that this metric focuses on using hidden Markov models, which cannot take into account some complex factors, such as temporal and environment factors. Since this comparison metric focuses on measuring the difference between model fitting probabilities, it is able to process other Markov models, such as hidden semi-Markov models (HSMMs) and partially observable Markov decision process (POMDP), that also generates model data fitting likelihoods. However, further studies are needed to ensure that other models can be used with this comparison metric.

Another limitation is that the comparison process focuses only on the overall model

fitting probabilities. The divergence measure is the fundamental method for obtaining quantitative model difference measures, and the Viterbi algorithm is used in divergence measures. The outcome of the Viterbi calculation is a single model fitting likelihood value. While such a number illustrates the overall model data fitting probability on a specific dataset, researchers cannot obtain information about which part of the data fits the model well. Thus, future studies can gain access to the model fitting probabilities of specific data portions and investigate details of strategy differences.

The second main effort of this research is to establish a continuum of strategy model comparisons as baselines and references for researchers to evaluate the magnitude of potential impacts from various factor changes on operator strategies. These references demonstrate quantitative difference measures compared to baselines to which an additional task will bring an extra 0.518 divergence measure to the comparison metric. Also, these references provide a ranking for potential impacts from different changes in HSC scenarios and such a ranking matches the human notion of similarity.

However, this continuum of references also has a limitation in verifying and demonstrating the additivity of the comparison metric. While ranking and quantifying are two important concepts in general comparisons, additivity is also important for providing further insights into the relations between various changes in HSC scenarios. For instance, the impact of adding an additional task of UAV hacking detection to the RESCHU interface, as the RESCHU-SA interface, will cause an increase of 0.518 in divergence measures. However, it is still unclear that if two additional tasks are added, operator strategies will change with a quantitative measure of the twice of 0.518.

5.4 Future Work

Given the limitations mentioned in the previous section, five main areas of future work have been identified: 1) utilize the strategy comparison metric in other HSC scenarios, 2)

explore the usage of divergence mesh shapes in establishing comparison references, 3) use the divergence mesh to assess strategy model selections, 4) extend the current continuum of comparison references for other common changes in HSC scenarios, and 5) investigate the additivity property of the comparison metric and resulting references.

As the human-automation field and human supervisory control related applications have developed rapidly, the need to study various HSC scenarios and to investigate operator strategies has also increased significantly. Thus, the first main future work area is to extend the usage of this strategy comparison metric to other HSC scenarios to establish comparison references. This work will require more data collected from human-subject experiments with different settings. Also, understanding that divergence mesh shapes generated from the comparison metric can present how different strategy model structures affect operator strategies, future research can explore the potential usage of such mesh shapes in the metric to provide more comparison details. Divergence meshes can also provide insights about selecting the optimal diagnostic strategy model structures by investigating the topology of mesh structures.

Given the increasing complexity of HSC scenarios, many other factors may affect operator strategies. While the current references presented in this research only consider a limited number of factors in HSC scenarios, future studies can extend the current continuum of references by considering other common factors. Similarly, this work also requires further experiments to collect operator action data sequences. With more comparison references, future studies can explore the relations between such reference measures, or impacts from HSC scenario changes, especially if these measures are quantitatively addable. The future results can shed light on the underlying principle of operator strategies and the relation between operator workload and strategies.

Appendix A

Appendix – The 1st RESCHU-SA Experiment

This appendix describes the experimental motivation, procedure, and results of the first RESCHU-SA experiment. In summary, the first RESCHU-SA experiment was conducted on the RESCHU-SA interface, which was a simulation-based experimental platform. Each participant experienced two sessions, which are Sessions 1 and 2 mentioned in previous chapters and shown in Figure 4.1.

A.1 Motivation and Background

Unmanned aerial vehicles (UAVs) have significantly increased commercial market and extensive applications in both civilian and military realms [130]. Many of these UAVs rely on the Global Positioning System (GPS) for navigation. However, this reliance leaves UAVs vulnerable to malicious attacks targeting GPS signals. One common attack is GPS spoofing, in which attackers deceive GPS receivers to override the navigation systems and redirect UAVs to unexpected destinations [131, 132]. Thus, detecting GPS spoofing attacks with a high success rate is important for UAV control systems. We propose a human-autonomy collaborative approach of human geo-location in that humans can aid in the detection of possible GPS spoofing attacks on UAVs. This approach was evaluated via an experiment, which was designed and conducted using the RESCHU-SA platform.

A common UAV control scheme is human supervisory control, in which a human operator monitors the multi-UAV system, intermittently navigating UAVs, and conducting other higher-level tasks [39]. In this experiment, we assume that human operators are responsible for the higher-level decision, and autonomous systems are in charge of lower-level UAV control and navigation operations [43]. UAVs typically rely on an embedded navigation

system known as GPS, which provides accurate position, velocity, and time information for GPS receivers in most areas on Earth. GPS receivers can report their locations to UAV control interfaces to provide location views for operators. However, GPS receivers are vulnerable to GPS spoofing attacks, in which GPS spoofers generate counterfeit signals to attack GPS receivers by manipulating the target position, velocity and time [131, 132].

Many researchers have presented autonomous GPS spoofing detection methods [133, 134, 135, 136, 137, 138]. However, false alarms and detection mistakes still exist while applying autonomous detection techniques [139, 140]. Thus, supplementary detection methods are needed. In the common design of military UAVs, a UAV is usually equipped with both a GPS navigation system and a payload camera, whose signal is independent of the UAV GPS signal [141]. Thus, the UAV payload camera view could be used as an independent reference for the detection of GPS spoofing attacks.

In order to utilize a UAV payload camera to detect UAV GPS attacks, interpreting the UAV's real-time location through the camera view and comparing this to a certain landmark or position estimate from a map could be the central mechanism for making such an assessment. While autonomous localization techniques may have limited performance [142], human vision has advantages in such complex search and surveillance tasks [143, 144, 145]. Based on human visual advantages, a human operator can potentially aid in UAV localization and thus detect potential UAV GPS spoofing attacks.

Based on the assumption that UAV cameras can show the true surrounding scene of UAVs, we propose that human operators can act as supplementary sensors and assist autonomous systems to detect UAV hacking attacks through comparative geo-location between the camera and map position estimates. In human geo-location, the operator can compare the non-tempered video feed coming from the UAV to the potentially falsified GPS location. This allows the operator to detect inconsistencies between these two sensing feeds. If the operator thinks the location interpreted from the camera view does not match the location shown on the map, then the UAV is most likely hacked via GPS spoofing.

A.2 Experiment Settings

An experiment was designed utilizing a modified version of the RESCHU experiment platform [104], known as Security-Aware RESCHU (RESCHU-SA) [103]. RESCHU-SA is a Java-based single operator with multi-UAV supervisory control simulation platform, which provides the capability to design multitasking scenarios that include both navigational and imagery search tasks. Moreover, the platform allows for simulating GPS spoofing attacks, in which hacked UAVs deviate from their originally assigned path and target to other unexpected destinations, along with warning notifications that simulate autonomous GPS spoofing detection systems.

The interface of the RESCHU-SA platform is shown in Figure 3.8. The interface features five main components: the payload camera view, message box, control panel, mission timeline, and map area. The primary objectives for operators in RESCHU-SA are to control multiple UAVs to 1) perform reconnaissance imagery tasks of counting road intersections when UAVs reach assigned targets, 2) ensure UAVs do not encounter hazard areas, and 3) determine whether UAVs are under GPS spoofing attacks.

For this experiment, GPS spoofing attack events followed a pre-defined schedule, unknown to the participants. When triggered internally, the hacked UAV changed its heading by a random angle within 30 to 60, or 300 to 330 degrees, which was larger than the human direction discrimination threshold. A hacking notification appeared 10 to 20 seconds after the attack event, simulating an external agency detection of a possible GPS spoofing attack. However, as in real systems, the notification could be a false alarm. In fact, about half of all notifications in this experiment were false alarms in the pre-defined schedule of each test session. Once the operator received notification from the system that a certain UAV was under possible cyber-attack, the operator could then investigate the UAV by checking the UAV's camera view and matching it against the position of the UAV on the map. The operator was expected to make a decision before the hacked UAV either exceeded the

map boundary or the experiment ended. If the operator decided the UAV was hacked, the operator could override the hacked UAV and send it home.

When UAVs that were not hacked reached a target, the operator engaged in an imagery task of counting the road intersections from the UAV's camera view at a pre-specified zoom level. This side task represents the primary purpose of such a mission, which is typically information gathering. While engaging in a counting task, the operator was required to enter an answer before the counting task was finished. The counting task allowed us to assess participants' performance based on the number of attempted tasks and the task correctness percentage. The path planner for the UAVs was intentionally suboptimal that the planner did not necessarily pick the most efficient assignment of UAVs to targets. In addition, UAVs would possibly encounter hazard areas that appeared and disappeared randomly. The suboptimal planner and the dynamic nature of hazard areas allowed experimenters to assess how much spare attention participants could devote to optimize the navigation and target assignment.

Thirty-six participants took part in this experiment, including 22 males and 14 females. Age ranged from 19 to 34 years with an average of 25.2 and a standard deviation of 3.8 years. Among the participants, 18 had little gaming experience, 6 had monthly gaming experience, 5 played video game several times a week, another 5 had weekly gaming experience, and only 2 had daily gaming experience.

The experiment procedure consisted of four main sections. The first section was a self-paced tutorial session. The second section was an 18-min practice session to allow participants to get more familiar with the user interface. The third section included the test sessions with two scenarios of different task loads, which were counterbalanced in terms of order of presentation. Specifically, for a high task load scenario, operators controlled six UAVs with nine different targets and nine hacking events, and in each low task load scenario, operators controlled three UAVs with six different targets and six hacking events. The fourth section was the debriefing session.

In both scenarios, the number of hazard areas, which generated and disappeared randomly, was constantly twenty-one. Each test scenario lasted 18 minutes, and each participant completed both high and low sessions. Each participant's performance scores were calculated based on the total vehicle damage, the correct percentage of imagery counting tasks, and the correct percentage of hacking identifications.

A.3 Experiment Results

We used a multivariate repeated-measures ANOVA model and Pearson correlation with a significance level of 0.05 to analyze data. In data analysis, independent variables included task load, which task load was experienced first, gender, and video game experience as a covariate. Task load was a within factor variable. Dependent variables included the percentage of correct hacking detections, the aggregated damage sustained by vehicles over a test session, and the overall correct percentage intersection counts per test session. These variables represent the primary objectives of performing the counting tasks, keeping vehicles out of the damaging areas, and successfully detecting hacking events.

Table A.1: The confusion matrix of hacking detection decisions in different notifications.

	Real hacking notification	False alarm notification
Consider UAV hacked	224	40
Consider UAV not hacked	63	207

An important question was whether human operators could successfully detect the UAV hacking events. A successful detection was indicated by a correct decision for a specific hacking event, including overriding the UAV and sending it home if the UAV was hacked or recognizing the notification was a false alarm. Among all hacking events in both test

sessions for each participant, 7 (4 in high task load and 3 in low task load) were pre-defined as false alarms, which meant the threshold for incorrect hacking notifications was 47%. As shown in Table A.1, out of all real hacking notifications across all participants, the overall success rate was 78%, and for the false alarms, the success rate was 84%. Thus, operators were slightly better at detecting false alarms than identifying real hacking notifications.

When looking at each individual's performance per test session, results showed that 23 out of total 72 experiment sessions (32%) resulted in 100% of successful hack identifications in a single test session, with another 24 (33%) above 80% successful attack identification. Thus, 65% of total experiment sessions exhibited 80% correct hacking detection or better without having any prior formal training. Also, for the three performance scores of vehicle damage, the correct percentage intersection counts, and correct percentage of hacking events, the only variable affected by task load was vehicle damage $(F(1,31) = 32.93), p < 0.001)$. Participants with less workload suffered less damage as they had more time to optimize their paths and avoid hostile areas.

One result showed a significant negative correlation between the time expended in hacking detections and correct detections $(Pearson = -0.375, p = 0.001)$, which meant that participants who took longer to detect the hacking events had a lower success rate in hacking identifications. This suggests that early detection was better from the operator standpoint. The covariate of the video game experience had a significant effect on participants' correct hacking detections $(F(1,31) = 4.652, p = 0.039)$. This means that the more video game experience, the higher the chance of a correct hacking detection. Another result showed that participants' task inputs were effective in that the more time they navigated the UAVs, the less time UAVs intersected with hostile areas $(Pearson = -0.345, p = 0.003)$. We also found that time expended in the imagery task was negatively correlated with the percentage of correct hacking detection $(Pearson = -0.275, p = 0.019)$. This result was expected as participants who spent more time in counting tasks were less likely to detect hacking events.

While using human geo-location in UAV hacking detections, operators will compare the non-tempered UAV camera video feed to the potentially falsified GPS location to detect inconsistencies between them. After receiving a hacking notification, operators can purposely navigate the notified UAV to some specific areas that can potentially provide more inconsistencies to increase the confidence of making a correct decision to a hacking event. Thus, analyzing the map usage in hacking detections will benefit the future design of autonomous decision-supporting tools for hacking identification.

The resulting heat map represents the frequency distribution of areas of participant interest during hacking detections and is shown in Figure A.1. Different colors represent varying frequency of operations, including adding waypoints and switching targets for UAVs, on a specific point. The warmer the color, the more participants interacted with a specific point, for example, red represents 5 or 6 operations. Understanding that the density of targets on the lower left quadrant of the map is slightly higher than other regions, this quadrant is more attractive to operators since operators can navigate UAVs between targets to get engaged to more imagery tasks in a shorter time range.

Landmarks used in hacking detections are classified into three categories, including special road patterns, geographic feature transition, and special buildings, like shown in Table A.2. Geographic feature transitions are defined as the transition between land and sea areas, on which operators can clearly observe the sudden change of geographic patterns. Special buildings are defined as distinctive shapes with contrastive colors that are used to represent a single building or a group of buildings on the map. As the percentage of total special road patterns and special buildings are approximate the same, special road patterns are more attractive to operators.

The frequency of different landmarks used in different detection decisions was examined. In correct hacking detections with both real hacking and false alarm notifications, the percentage of operations based on special road patterns is slightly over 60%, which is higher than the percentage in incorrect hacking detection with real hacking notification

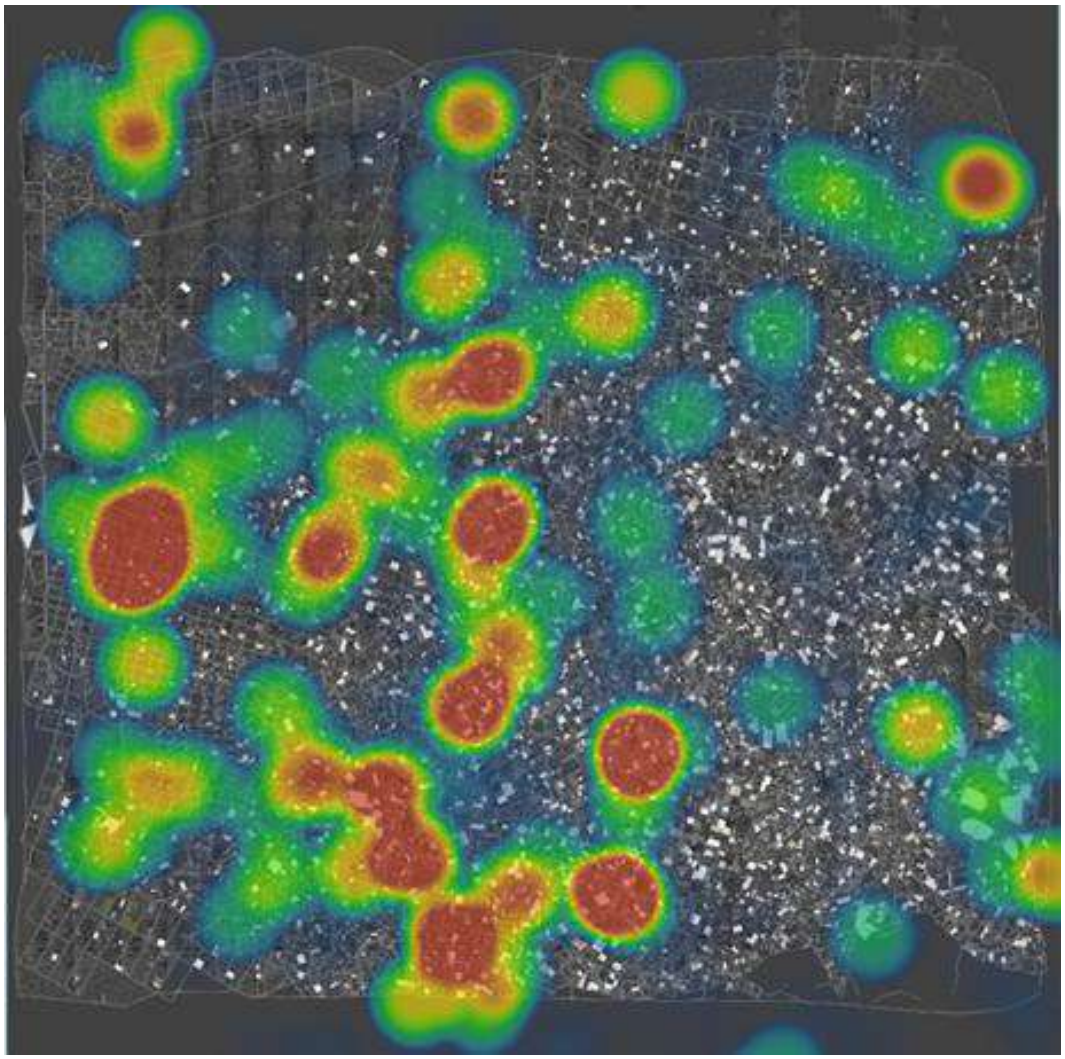


Figure A.1: The heat map of reference points in UAV hacking detection.

(45.5%) and false alarm notification (56.0%). Another interesting fact is that special road patterns lead to the highest success rate of 86.1% in hacking detections, while geographic feature transition lead to 79.6% and special buildings lead to 80.7%. These results provide insight for how a future advanced map-based hacking detection support tool for human operators could be designed.

In this experiment, we analyzed if a human operator could serve as a supplementary sensor in supervisory UAV control systems by successfully detecting UAV hacking events.

Table A.2: The frequency of different types of landmarks used in hacking detections.

	Special road patterns	Geographic feature transition	Special buildings
Occurrence frequency	380	152	109
Occurrence percentage	59.3%	23.7%	17.0%

Experiment results supported this hypothesis in that 65% of total experiment sessions reached over 80% hacking detection correctness. The experiment results indicated that some factors affected operators' performance and operations. Given that the operator's video game experience significantly affected the success rate in hacking detections, future personnel selection for supervisory control systems with human visual tasks could focus more on the experience in similar applications or more training. Also, the result of the negative correlation between the time expended and the success rate in hacking detection provides implications of increasing hacking detection correctness by guiding better search strategies and earlier detections.

The map analysis showed the heat map of participants' preferences for hacking detection. This analysis also provided some insights on a more efficient way to utilize different landmarks. Lastly, all these results established a baseline of performance of applying human geo-location in UAV hacking detection. Future studies will focus on how higher-level automation or advanced decision support tools could be utilized to assist human operators in improving the success rate of hacking identifications.

Appendix B

Appendix – The 2nd RESCHU-SA Experiment

This appendix describes the second RESCHU-SA experiment, which is Sessions 3 and 4 in Figure 4.1. Similar to the first RESCHU-SA experiment, this experiment was also conducted on the RESCHU-SA interface as shown in Figure 3.8. This experiment was designed to examine if extra hacking detection supports, such as a decision support system and an additional training, could increase the success rate of UAV hacking detection and reduce inefficiencies in participants' detection strategies.

B.1 Experiment Settings

The RESCHU-SA experiment platform [104] was also utilized in this experiment to study human operators' performance and strategies in controlling multiple UAVs. Like the first RESCHU-SA experiment, participants in this experiment experienced a similar experimental procedure with two experiment sessions, as shown in Sessions 3 and 4 in Figure 4.1. In experiment sessions, participants focused on three major tasks, including the UAV navigation task, the imagery analysis task, and the UAV hacking detection task. With a special focus on the hacking detection task, we tried to answer the research question that if extra hacking detection supports could benefit human operators in increasing hacking detection success rate and reducing potential inefficiencies in detection strategies.

The experiment settings is shown in Table B.1. Forty-five participants were recruited and randomly assigned to three different experimental groups. All participants experienced two experiment sessions with a constant task load of managing five UAVs with eight targets and eight hacking events, including four real hackings and four false alarms. No additional support was provided in the first experiment session. However, different supports were

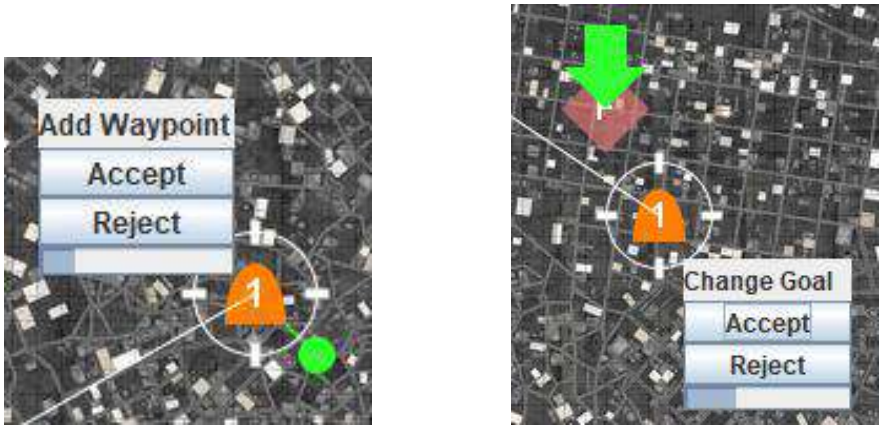
provided in the second session based on different experimental groups shown in Table B.1. Specifically, participants' hacking detection actions were collected, and their detection strategies were inferred in the first session. Participants' strategies were categorized into either waypoint-oriented or target-oriented strategies, which were determined in the first RESCHU-SA experiment and shown in Figure 2.7 and Table 2.3. Then in the second session, participants received extra supports with different focuses based on the category of their hacking detection strategies.

Table B.1: Different extra supports based on different experimental groups.

	Group 1	Group 2	Group 3
Participants	15	15	15
The first session	No extra support	No extra support	No extra support
The second session	Decision support system only	Detection strategy training only	Both

The decision support system provides either suggested locations for adding waypoints or suggested targets for switching assigned targets based on participants' strategy categories. Once a hacking notification pop-up window appears on the interface, a suggestion dialog box also appears simultaneously, providing a waypoint or target suggestion with two options of accepting or rejecting the suggestion. An example of two types of suggestion boxes is shown in Figure B.1. The waypoint-oriented system provides a suggested waypoint, which is highlighted in green in Figure B.1a. If participants accept the suggestion, a waypoint will be automatically added for the notified UAV, and the UAV will change its direction to the newly added waypoint. The target-oriented system provides a suggested target, which is indicated by a green arrow in Figure B.1b. If participants accept the suggestion, the target of the notified UAV will be switched to the suggested target, and the

UAV will approach the newly assigned target. The suggestions from the decision support system are determined based on 1) the distance between the notified UAV and heat map hot spots shown in Figure A.1, 2) detection success rates of surrounding hot spots, and 3) the number of UAVs waiting to engage to imagery tasks. Also, operators could manipulate suggested waypoints or targets after accepting the suggestion.



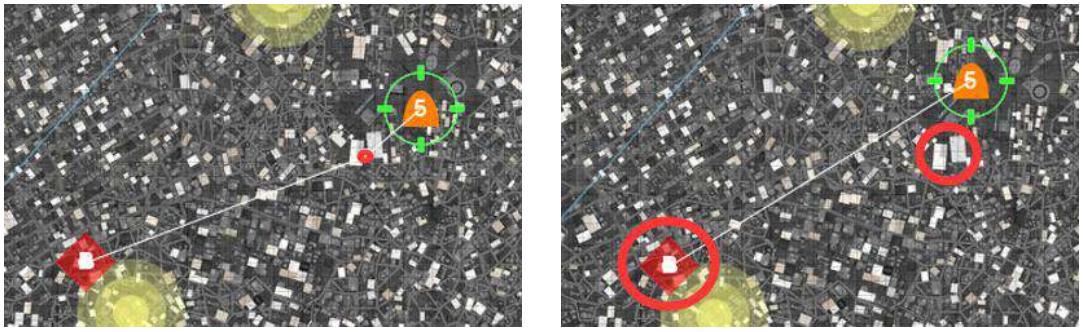
(a) (b)

Figure B.1: Examples of the decision support system. (a) The waypoint-oriented suggestion. (b) The target-oriented suggestion.

Another extra support, the additional hacking detection strategy training, was provided between the first and the second experiment session. Participants received this training with emphases on either waypoint- or target-oriented hacking detection strategies for the second session. Participants received different types of training based on their detection behavior patterns inferred in the first session.

As an example, if a participant revealed to have a proactive waypoint-oriented detection strategy in the first session, then before the second scenario, the participant would be trained by the waypoint strategy. Such a strategy requires operators to proactively change UAVs' trajectory after receiving hacking notifications by manipulating UAVs' waypoints to navigate notified UAVs to obvious landmarks, which is highlighted with a red dot in Figure B.2a as an example. If a participant tended to have a reactive target-oriented detec-

tion strategy in the first session, the participant would be trained by the target strategy. In this strategy, participants are required to switch the assigned target for the notified UAV to change the UAV's moving direction to certain obvious landmarks as highlighted with a red circle in Figure B.2b. In general, the main purpose of providing strategy training is to guide operators to specific strategies to simplify their behavior patterns in hacking detections to save their cognitive resources.



(a) (b)

Figure B.2: Examples of the detection strategy training. (a) The waypoint-oriented training. (b) The target-oriented training.

Both the decision support system and the detection strategy training constructed a feedback loop on the experiment sessions. Firstly, operator strategy models were developed from the first RESCHU-SA experiment, and operation flows were observed from the strategy models. Potential inefficiencies were also observed in the first experiment strategy models that participants revealed to conduct repeated actions. Then, such operation flows were utilized in the second experiment to assist operators in hacking detections by reducing repeated actions and enhancing strategy adherence. The decision support system assists operators in reducing operators' extra efforts by providing suggestions of human geo-location tasks for hacking detections and automatically applying such suggestions. And the detection strategy training assists operators in guiding them to a specific strategy observed from the first-stage models in hacking detection.

B.2 Experiment Results

The hypotheses of this experiment include 1) both types of additional supports in the second experiment session can improve operators' success rate in hacking detection compared to the first session, 2) the decision support system can reduce the inefficiency of manipulating UAVs' trajectory in hacking detection actions, and 3) the detection strategy training can enhance the adherence of operators' detection strategies to known strategies.

The distribution of the strategies utilized in additional supports was uneven in that most participants received hacking detection supports with the proactive waypoint-oriented strategy instead of the reactive strategy. Considering additional supports embedded with detection strategies that operators tended to have in the first session, most participants revealed to have the proactive strategy. In this case, different strategies in additional supports were not considered in statistical analyses. To investigate the first hypothesis, a repeated-measure multi-variate ANOVA model was developed to explain the potential effects of the additional supports on operators' performance in hacking detections.

The statistical analysis results from the ANOVA model showed that additional supports provided in the second experiment session increased participants' UAV hacking detection success rate ($F(1,43) = 6.003, p = 0.019$). The boxplots of the success rates based on different experimental groups and sessions are shown in Figure B.3. For all these three experimental groups, the medium of participants' detection success rates in the second session were higher than those in the first session. Especially in the second experimental group, the medium success rate was increased from 75% to 100% by receiving the detection strategy training.

Participant strategy models, as shown in Figure B.4, were developed to describe their hacking detection strategies and verify experiment hypotheses. Understanding that all participants experienced the same experimental settings in the first experiment session, one strategy model was developed across all groups for the first session as shown in Figure

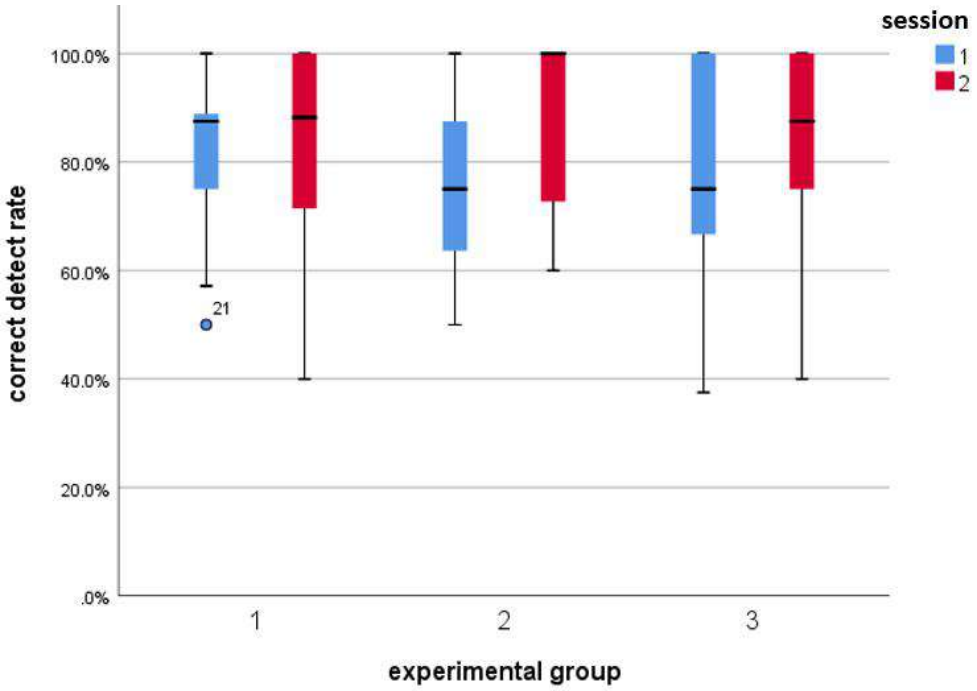


Figure B.3: Boxplots of UAV hacking detection success rates by different experimental groups and sessions.

B.4a. For the second experiment session, three strategy models were developed for the three experimental groups, respectively, as shown in Figures B.4b, B.4c, and B.4d.

In the second hypothesis, the decision support system provided in Groups 1 and 3 was expected to reduce the repeated operations of manipulating UAVs' trajectory in hacking detections while applying human geo-location. Compared to the hacking detection model developed from the first session, the Group 1 model and Group 3 model were expected to significantly lower the self-transitions in UAV trajectory manipulation-related states. In fact, participants rejected 34.6% suggestions in Groups 1 and 3, and they further modified the suggested points after accepting the suggestions due to their preference. This can be observed from the Group 1 model in Figure B.4b that the self-transition on the "adjust way-point" state is 44.8%, and the "adjust target" state has a self-transition of 83.1%. Similar high self-transitions also appear in the Group 3 model in Figure B.4d.

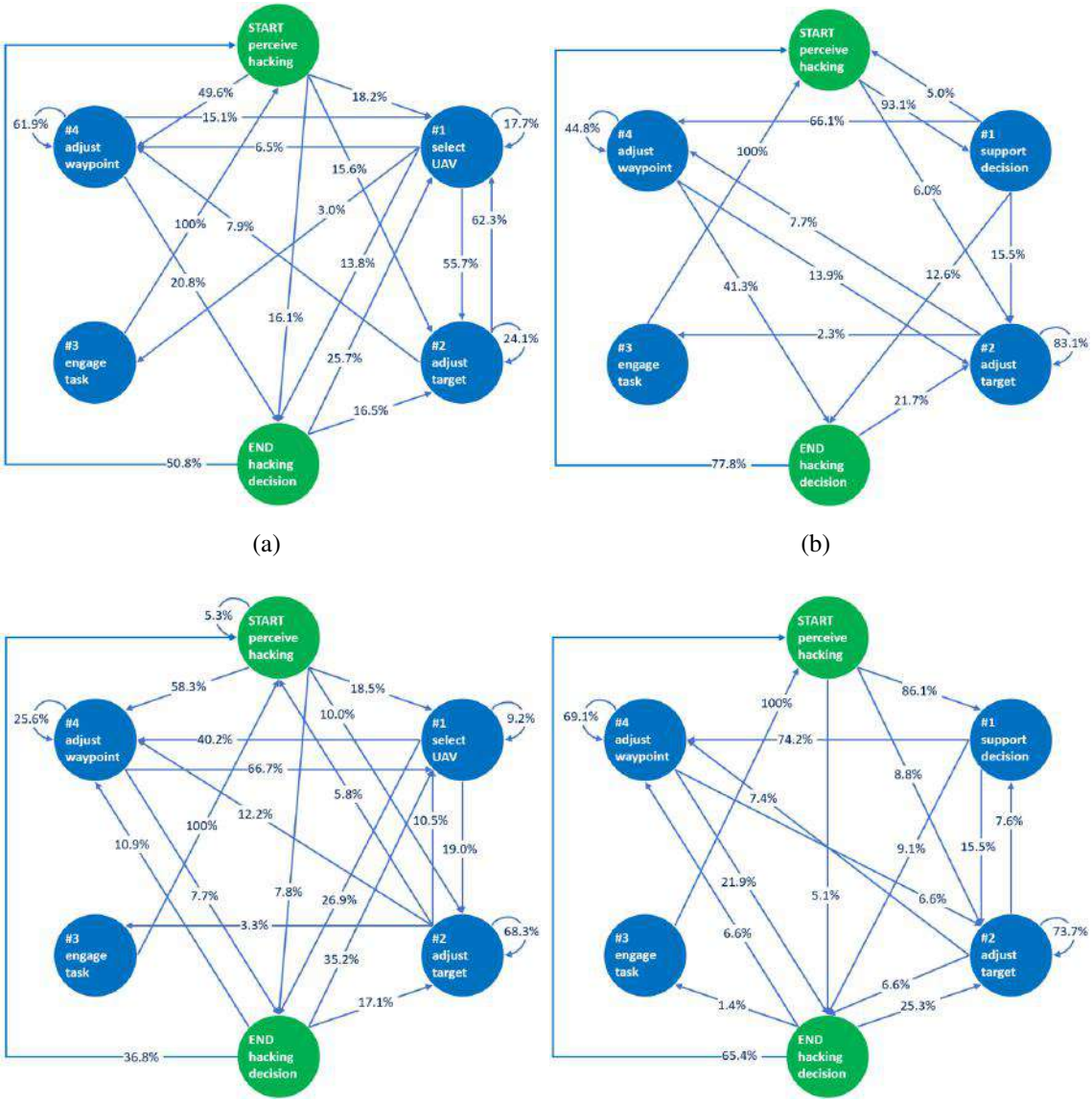


Figure B.4: Hacking detection models. (a) The 1st session model. (b) The 2nd session group 1 model. (c) The 2nd session group 2 model. (d) The 2nd session group 3 model.

Based on the self-transitions shown in hacking detection models, we can conclude that the decision support system did not reduce the inefficient operations in hacking detections. Given that the decision support system increased participants' detection success rate but failed to reduce inefficient operations, further studies of the system usage among participants in Groups 1 and 3 are needed. Also, the relation between different suggestion acceptance levels and participants' detection success rates needs further investigations.

In the third hypothesis, the detection strategy training was expected to change operators' strategy adherence that operators were expected to adhere to a single operational flow in hacking detection. In the Group 2 model in Figure B.4c, the transition from the "perceive hacking" state to the "adjust waypoint" state is 58.3%, which is larger than the same transition of 49.6% in the first session model in Figure B.4a. Also, the transition from "perceive hacking" to "support decision" reaches 86.1%, and the adjacent transition from "support decision" to "adjust waypoint" is 74.2% in the Group 3 model. Such transitions illustrate that participants adhered to the waypoint-oriented hacking detection strategy as expected with the strategy training.

Based on the state transitions shown in the models, we can conclude that the detection strategy training generally enhanced the adherence of participants' strategies to the known waypoint-oriented strategy. As future studies, the detailed assessment of operators' strategy adherence is necessary to quantitatively investigate the levels of adhering to known strategies. Then, the hacking detection performance comparison among operators with different levels of adherence can further evaluate the effectiveness of the additional strategy training process.

Appendix C

Appendix – The RESCHU Experiment

This appendix describes the RESCHU experiment, which was conducted on the RESCHU interface shown in Figure 3.7. The RESCHU interface provides two major tasks for participants, including the UAV navigation task and imagery analysis task. Compared to the RESCHU-SA interface, the RESCHU interface is not embedded with the UAV hacking detection task. This experiment was designed to evaluate the effect of increasing autonomy and different training methods on UAV supervisory control.

C.1 Experiment Settings

With significantly increasing commercial market of unmanned aerial vehicles (UAVs), extensive UAV applications have been designed and provided for both civilian and military realms [130]. Understanding that human operators play an important role in a common UAV control scheme of human supervisory control (HSC) [43], many factors can potentially affect human operators' performance in UAV control scenarios, including UAV navigation and high-level surveillance tasks. Considering higher-level automation was introduced generally to decrease human efforts, different levels of autonomy were investigated in this experiment that whether a higher level of autonomy would benefit operators' performance in conducting high-level UAV tasks. In addition, knowing that a more detailed training process may result in better performance, different training processes were investigated in this experiment that how different training processes may affect operators' performance in UAV control scenarios.

Different types and levels of automation have been introduced to most kinds of human-machine systems by developing and utilizing computer hardware and software applications

to reduce human efforts [146, 147]. Different levels of automation have also been introduced to human supervisory UAV control systems to assist human operators in conducting high-level tasks [133, 134, 135]. The presence of the technique of automatic target recognition (ATR) is an example of utilizing higher-level automation in UAV surveillance tasks. With the assistance of ATR, operators could spend less effort in searching and localizing tasks meanwhile performing a high success rate. However, operators may not trust higher-level automation in supervisory UAV controls since the automation may fail in assisting operators in such tasks.

In addition to different LOA, the distribution of autonomous functions and the construction of collaboration should also be investigated. The SRKE (skill, rule, knowledge, expertise) model provides useful implications for developing automated systems by illustrating different information-processing behaviors [148, 149, 150, 151]. The SRKE model divides cognitive tasks into four levels. Skill-based behaviors are highly practiced and automatic responses to stimuli. Rule-based behaviors utilize memorized procedures based on experience or instruction in response to familiar situations. Knowledge- and expertise-based behaviors function in novel situations where conceptual understanding of the environment is required. Thus, different training levels may affect human operators' performance in conducting tasks in supervisory UAV control scenarios.

In order to investigate the effects from factors mentioned in previous sections, an experiment was designed and conducted utilizing a modified version of the Research Environment for Supervisory Control of Heterogeneous Unmanned Vehicles (RESCHU) experiment platform [104]. The interface of the RESCHU platform is shown in Figure 3.7. The interface features four main components, including the payload camera view, control panel, mission timeline, and map area.

Thirty participants took part in this experiment, including 15 Duke undergraduate students and 15 army personnel. The main difference between the two groups of subjects was the level of experience of using military-related equipment. Because of the limited number

of participants, age and gender were not investigated in this experiment. Based on the first RESCHU-SA experiment results that operators' video game experience may affect their performance [128], participants' gaming experience was collected.

The primary objectives for operators in this experiment are to control multiple UAVs: 1) perform imagery tasks of searching specific targets when UAVs reach assigned targets, 2) ensure UAVs do not encounter hazard areas. The experiment procedure consisted of three main sections. The first section was a training section, in which subjects received different types of training. The second section was a practice session to allow participants to get familiar with the experiment interface. The third section was data collection with two different scenarios.

Specifically, in the training section, two types of training processes were provided for different groups of participants. The first type was the rule-based training, in which subjects would go over the self-paced tutorial training slides and ask questions to experimenters. In the rule-based training, participants would learn the experiment objectives and basic operations on the experiment interface. The second training type was a combination of skill- and rule-based training, in which subjects received a specific practice of aiming targets on screen followed with the training tutorial slide. The aiming target practice was considered as skill-based training because it would potentially decrease participants' reaction time and increasing the success rate of searching targets via camera video feed.

In the data collection section, which was also the experiment section, each subject had two different scenarios with different imagery searching tasks. The number of UAVs, targets, and hazard areas for different scenarios remained the same. Thus, the taskload of different scenarios was similar. However, the images used in searching tasks were different in that these images were assigned to each scenario based on a specific sequence. The main purpose of setting different scenarios was to investigate the order effect that whether subjects would perform better in the second scenario.

The assistance of the automation target recognition (ATR) in imagery searching tasks

was also introduced in experiment sessions as increasing autonomy. The assistance of the ATR was presented as where the automation believed the target was located. With the assistance of the ATR, the target would appear in the range of the original camera video feed without participants moving or zooming the view. In this case, participants with the assistance of ATR could easily locate where the targets were in imagery searching tasks. However, ATR was designed to have only a 70% probability of successfully assisting in imagery tasks, so ATR might work properly or fail. For participants without the assistance of ATR would be responsible for controlling the camera view to search the target by panning and zooming the camera view.

C.2 Experiment Results

Based on the experiment design mentioned in the previous section, participants were assigned to different groups to received different experimental treatments, as shown in Table C.1.

Table C.1: Different participant groups with different experiment treatments.

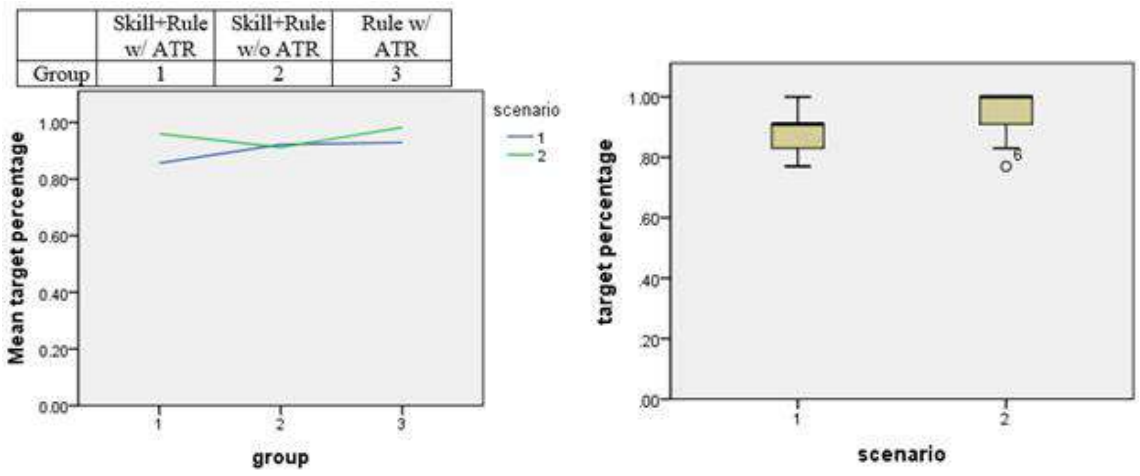
	Group 1	Group 2	Group 3
	Skill- and rule-based training with ATR	Skill- and rule-based training without ATR	Rule-based training only with ATR
Army personnel (participants)	5	5	5
Duke students (participants)	5	5	5

Thus, independent experiment variables can be summarized as 1) scenarios, 2) participant occupation, 3) participant group. And dependent variables can be summarized as 1) the number of hazard areas incurred, 2) the average time expended in hazard areas, 3) the success rate of conducting imagery searching tasks, 4) the average time expended in imagery searching tasks, 5) the average waiting time for UAVs to engage to imagery tasks, 6) the success rate of imagery tasks with or without the assistance of ATR. A covariate is 1) participants' video game experience.

A multivariate repeated-measure ANOVA model (rm-MANOVA) with a significance level of 0.05 was used for statistical analyses. Multicollinearity can be avoided in that three independent variables were designed by experimenters, and these variables were crossed to each other. Also, participants were assigned to different experimental groups evenly and randomly. Thus, participants' video game experience was independent of three major independent variables.

A significant interaction between scenario and different participant groups on the success rate of imagery tasks was presented ($F(0.95;2,23) = 5.644, p = 0.010, power = 0.810$). As shown in Figure C.1, participants in Groups 1 and 3 (both groups had the assistance of ATR) had a higher success rate in imagery searching tasks in the second scenario than the first scenario. However, participants in Group 2 without the assistance of ATR had similar correctness in imagery tasks in both scenarios. The scenario, as a within-subject factor, had an overall significant effect on the task correctness ($F(0.95;1,23) = 8.752, p = 0.007, power = 0.809$). Considering the main difference between Group 2 and Groups 1 and 3 was the presence of ATR, this result suggests a training effect occurred in experimental groups with the assistance of ATR.

In Groups 1 and 3, participants had the first experiment session as a knowledge-based training to get more familiar with utilizing ATR in searching tasks so that their success rates of searching targets were increased in the second scenario. However, in Group 2, participants without the assistance of ATR did not present the training effect that their searching



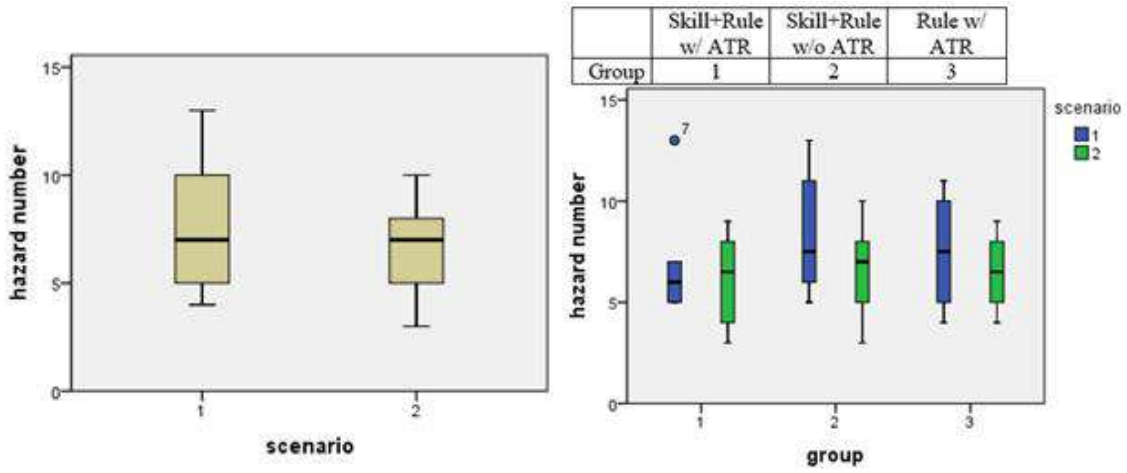
(a) (b)

Figure C.1: The interaction between scenario and participant groups on the imagery task success rate. (a) Line plots based on scenarios and groups. (b) Corresponding boxplots.

tasks success rate was similar in both scenarios. This fact indicates that conducting such imagery searching tasks was difficult to practice without any autonomy assistance.

Another potential training effect was indicated via the slight decrease in the number of hazard area incursions in the second scenario. As shown in Figure C.2, different scenarios affected the number of hazard area incursions ($F(0.95;1,23) = 7.424, p = 0.012, power = 0.742$). Although the power value is slightly less than 0.8, this result still provides some implications that the more experience of interacting with RESCHU interface participants had, the less likely they would navigate UAVs to encounter hazard areas. Also, since each experiment session only lasted 15 minutes, the time duration of a single experiment session maybe not enough for participants to practice. Thus, the training effect on UAV navigation was not significant.

Interestingly, participants' video game experience as a covariate did not have any significant interaction with other independent variables and did not significantly affect any dependent variables. Thus, this result indicates that participants' video game experience would not benefit their performance in visual searching tasks. Additionally, participant



(a) (b)

Figure C.2: The training effect on the number of hazard area incursions. (a) Boxplots based on scenarios. (b) Corresponding detailed boxplots based on groups.

type as an independent variable did not interact with other independent variables and did not have a significant effect on all dependent variables. This result shows that experience in using military equipment would not benefit army personnel's performance in both visual searching tasks and UAV navigation since both army personnel and Duke students had a similar overall performance.

Participants assigned to different groups received different training processes and different autonomy assistance in imagery searching tasks. Data analyses show that participant group as an independent variable had significant effect on the average time expended in imagery searching tasks ($F(0.95;2,23) = 10.901, p < 0.001, power = 0.981$), and the average imagery task waiting time ($F(0.95;2,23) = 7.539, p = 0.003, power = 0.912$).

Focusing on the average time expended in imagery tasks, participants in Group 2 spent significantly more time than Groups 1 and 3. A Bonferroni pairwise comparison with a family-wise error rate of $0.05/3 = 0.017$ was conducted. The results show a moderate but not significant difference in the average time consumption between Groups 1 and 2 ($p = 0.027$), and a significant difference between Groups 2 and 3 ($p < 0.001$). This result

states that participants who received the assistance of ATR in imagery tasks (Groups 1 and 3) spent less time than participants without the assistance of ATR.

Similarly, the presence of ATR also significantly affected the average task waiting time. The Bonferroni pairwise comparison results showed that participants in Group 1 spent significantly less time in waiting than Group 2 ($p = 0.006$), and participants in Group 3 also spent less time than Group 2 ($p = 0.015$). This result can be explained that without the assistance of ATR, participants spent significantly more time in searching tasks. Consequently, UAVs in their sessions waited significantly longer to be engaged to imagery tasks.

Understanding that the ATR only provided a 70% success rate in imagery searching tasks, participants who received the assistance of ATR might trust or doubt whether the ATR provided reasonable camera view ranges for searching targets. Also, considering the potential knowledge-based training effect on the success rate of imagery tasks on participant Groups 1 and 3, participants' success rate was investigated based on scenarios and whether ATR functioned or failed. Another repeated-measure ANOVA model with a significance level of 0.05 was used to investigate the correctness of imagery tasks in both ATR success and failure situations. The scenario was considered as a within-subject variable, whether ATR correctly functioned was considered as a between-subject variable, and the only independent variable was the success rate of imagery searching tasks.

A significant interaction between scenario and ATR function situation was presented ($F(0.95;1,38) = 49.429, p < 0.001, power > 0.999$). When the ATR correctly functioned in assisting participants with searching tasks, most participants reached a 100% success rate in both scenarios. When the ATR failed to provide appropriate suggestions, participants had a significantly better success rate in Scenario 2 than Scenario 1. This interesting fact also suggests a potential knowledge-based training effect on conducting imagery tasks. Interestingly, this training effect only occurred when ATR was introduced in experiment sessions even participants were told that ATR might not function properly.

In summary, the experiment results provide insights into our initial questions with im-

plications for future studies and applications. In summary, participants had an overall 93.0% success rate in the imagery searching task through all experiment sessions. A potential knowledge-based training effect on the success rate of imagery tasks was indicated by the significant difference between scenarios and participant groups. Participants who received the assistance of ATR in imagery tasks had significantly higher correctness in the second scenario. Thus, given the assistance of ATR, more practice of conducting imagery tasks would potentially increase operators' success rate in such searching tasks.

Understanding that ATR had a 70% probability of functioning correctly, participants reached an overall 98.0% success rate when ATR correctly functioned in both scenarios. The main difference in the success rate indicated by the training effect was caused by the imagery tasks with ATR failing to function correctly. Participants had an overall 62.0% success rate in the first scenario when ATR failed. However, their success rate increased to 97.0% in the second scenario with ATR functioned incorrectly. This significant improvement indicated that both the assistance of ATR and practice are necessary for improving performance in imagery searching tasks.

Since a significant difference in the average imagery task time consumption and the average task waiting time was presented based on the presence of ATR, the assistance of ATR is necessary to reduce time consumption in imagery searching tasks. Thus, understanding that time is critical in conducting such high-level UAV tasks, the introduction of a higher level of automation is necessary for time-critical supervisory UAV control scenarios.

Bibliography

[1] P. A. Hancock, R. J. Jagacinski, R. Parasuraman, C. D. Wickens, G. F. Wilson, and D. B. Kaber, "Human-automation interaction research: Past, present, and future," *Ergonomics in Design*, vol. 21, no. 2, pp. 9–14, 2013.

[2] C. P. Janssen, S. F. Donker, D. P. Brumby, and A. L. Kun, "History and future of human-automation interaction," *International Journal of Human-Computer Studies*, vol. 131, pp. 99–107, 2019.

[3] A. Degani and M. Heymann, "Formal verification of human-automation interaction," *Human Factors*, vol. 44, no. 1, pp. 28–43, 2002.

[4] T. B. Sheridan, *Humans and Automation: System Design and Research Issues*. Human Factors and Ergonomics Society, 2002.

[5] T. Wang, Y. Tao, and H. Liu, "Current researches and future development trend of intelligent robot: A review," *International Journal of Automation and Computing*, vol. 15, no. 5, pp. 525–546, 2018.

[6] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.

[7] J. Z. Gul, M. Sajid, M. M. Rehman, G. U. Siddiqui, I. Shah, K. Kim, J. Lee, and K. H. Choi, "3d printing for soft robotics–a review," *Science and Technology of Advanced Materials*, vol. 19, no. 1, pp. 243–262, 2018.

[8] L. Royakkers and R. van Est, "A literature review on new robotics: Automation from love to war," *International Journal of Social Robotics*, vol. 7, no. 5, pp. 549–570, 2015.

[9] J. Torresen, "A review of future and ethical perspectives of robotics and ai," *Frontiers in Robotics and AI*, vol. 4, p. 75, 2018.

[10] W. Wang and K. Siau, "Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: A review and research agenda," *Journal of Database Management*, vol. 30, no. 1, pp. 61–79, 2019.

[11] R. R. Murphy, *Introduction to AI Robotics*. MIT Press, 2019.

[12] M. L. Bolton, E. J. Bass, and R. I. Siminiceanu, "Using formal verification to evaluate human-automation interaction: A review," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 3, pp. 488–503, 2013.

[13] D. B. Kaber, "Issues in human-automation interaction modeling: Presumptive aspects of frameworks of types and levels of automation," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 7–24, 2018.

[14] C. Guo, J. Meguro, Y. Kojima, and T. Naito, "Cadas: A multimodal advanced driver assistance system for normal urban streets based on road context understanding," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 228–235, IEEE, 2013.

[15] R. Yasrab, N. Gu, and X. Zhang, "An encoder-decoder based convolution neural network (CNN) for future advanced driver assistance system (ADAS)," *Applied Sciences*, vol. 7, no. 4, p. 312, 2017.

[16] O. Sawade and I. Radusch, "Survey and classification of cooperative automated driver assistance systems," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pp. 1–5, IEEE, 2015.

[17] J. Liu, Y. Su, M. Ko, and P. Yu, "Development of a vision-based driver assistance system with lane departure warning and forward collision warning functions," in *2008 Digital Image Computing: Techniques and Applications*, pp. 480–485, IEEE, 2008.

[18] V. Gaikwad and S. Lokhande, "Lane departure identification for advanced driver assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 910–918, 2014.

[19] A. M. Kumar and P. Simon, "Review of lane detection and tracking algorithms in advanced driver assistance system," *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 7, no. 4, pp. 65–78, 2015.

[20] C. Braunagel, W. Rosenstiel, and E. Kasneci, "Ready for take-over? a new driver assistance system for an automated classification of driver take-over readiness," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 4, pp. 10–22, 2017.

[21] T. Sugawara, H. Altmannshofer, and S. Kakegawa, "Applications of road edge information for advanced driver assistance systems and autonomous driving," in *Advanced Microsystems for Automotive Applications*, pp. 71–86, Springer, 2018.

[22] M. Gerstmair, M. Gschwandtner, R. Findenig, O. Lang, A. Melzer, and M. Huemer, "Miniaturized advanced driver assistance systems: A low-cost educational platform for advanced driver assistance systems and autonomous driving," *IEEE Signal Processing Magazine*, vol. 38, no. 3, pp. 105–114, 2021.

[23] N. Ovcharova, M. Fausten, and F. Gauterin, "Effectiveness of forward collision warnings for different driver attention states," in *2012 IEEE Intelligent Vehicles Symposium*, pp. 944–949, IEEE, 2012.

[24] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, 2009.

[25] A. Shaout, D. Colella, and S. Awad, "Advanced driver assistance systems-past, present and future," in *2011 Seventh International Computer Engineering Conference (ICENCO'2011)*, pp. 72–82, IEEE, 2011.

[26] M. Li, H. Cao, X. Song, Y. Huang, J. Wang, and Z. Huang, "Shared control driver assistance system based on driving intention and situation assessment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4982–4994, 2018.

[27] M. Zahabi, A. M. A. Razak, A. E. Shortz, R. K. Mehta, and M. Manser, "Evaluating advanced driver-assistance system trainings using driver performance, attention allocation, and neural efficiency measures," *Applied Ergonomics*, vol. 84, p. 103036, 2020.

[28] N. R. Council *et al.*, *The Future of Air Traffic Control: Human Operators and Automation*. National Academies Press, 1998.

[29] F. T. Durso and C. A. Manning, "Air traffic control," *Reviews of Human Factors and Ergonomics*, vol. 4, no. 1, pp. 195–244, 2008.

[30] A. R. Isaac and B. Ruitenberg, *Air Traffic Control: Human Performance Factors*. Routledge, 2017.

[31] S. Kahne and I. Frolow, "Air traffic management: Evolution with technology," *IEEE Control Systems Magazine*, vol. 16, no. 4, pp. 12–21, 1996.

[32] M. S. Nolan, *Fundamentals of Air Traffic Control*. Cengage Learning, 2011.

[33] N. R. Council, *Flight to the Future: Human Factors in Air Traffic Control*. National Academies Press, 1997.

[34] H. Erzberger and R. A. Paielli, "Concept for next generation air traffic control system," *Air Traffic Control Quarterly*, vol. 10, no. 4, pp. 355–378, 2002.

[35] V. D. Hopkin, *Human Factors in Air Traffic Control*. CRC Press, 2017.

[36] J. M. Hoc and M. P. Lemoine, "Cognitive evaluation of human-human and human-machine cooperation modes in air traffic control," *The International Journal of Aviation Psychology*, vol. 8, no. 1, pp. 1–32, 1998.

[37] S. Debernard, B. Guiost, T. Poulain, I. Crévits, D. Annebicque, and P. Millot, "Integrating human factors in the design of intelligent systems: an example in air traffic control," *International Journal of Intelligent Systems Technologies and Applications*, vol. 7, no. 2, pp. 205–226, 2009.

[38] H. Gürlük, "Concept of an adaptive augmented vision based assistance system for air traffic control towers," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pp. 1–10, IEEE, 2016.

[39] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, MA: MIT Press, 1992.

[40] C. D. Wickens, J. G. Hollands, S. Banbury, and R. Parasuraman, *Engineering Psychology and Human Performance*. Hove, U.K.: Psychology Press, 2015.

[41] C. D. Wickens, S. E. Gordon, and Y. Liu, *An Introduction to Human Factors Engineering*. New York, NY: Longman, 1998.

[42] T. B. Sheridan and R. Parasuraman, "Human-automation interaction," *Reviews of Human Factors and Ergonomics*, vol. 1, no. 1, pp. 89–129, 2005.

[43] M. L. Cummings, S. Bruni, S. Mercier, and P. Mitchell, "Automation architecture for single operator, multiple UAV command and control," *The International Command and Control Journal*, vol. 1, no. 2, pp. 1–24, 2007.

[44] T. B. Sheridan, "Human supervisory control of aircraft, rail and highway vehicles," *Transactions of the Institute of Measurement and Control*, vol. 21, no. 4-5, pp. 191–201, 1999.

[45] T. B. Sheridan, "Adaptive automation, level of automation, allocation authority, supervisory control, and adaptive control: Distinctions and modes of adaptation," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 4, pp. 662–667, 2011.

[46] A. S. Clare, J. C. Ryan, K. F. Jackson, and M. L. Cummings, "Innovative systems for human supervisory control of unmanned vehicles," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 56, (Los Angeles, CA), pp. 531–535, SAGE Publications, 2012.

[47] M. L. Cummings, S. Bruni, and P. J. Mitchell, "Human supervisory control challenges in network-centric operations," *Reviews of Human Factors and Ergonomics*, vol. 6, no. 1, pp. 34–78, 2010.

[48] J. S. Walker, *Three Mile Island: A Nuclear Crisis in Historical Perspective*, vol. 41. University of California Press, 2004.

[49] F. Lambert, "Understanding the fatal tesla accident on autopilot and the NHTSA probe," *Electrek*, vol. 1, July 2016.

[50] T. Sheridan, "Human supervisory control of robot systems," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 808–812, IEEE, 1986.

[51] T. B. Sheridan, "Human-robot interaction: Status and challenges," *Human Factors*, vol. 58, no. 4, pp. 525–532, 2016.

[52] J. R. Peters, V. Srivastava, G. S. Taylor, A. Surana, M. P. Eckstein, and F. Bullo, "Human supervisory control of robotic teams: Integrating cognitive modeling with engineering design," *IEEE Control Systems Magazine*, vol. 35, no. 6, pp. 57–80, 2015.

[53] S. R. Dixon, C. D. Wickens, and D. Chang, "Mission control of multiple unmanned aerial vehicles: A workload analysis," *Human Factors*, vol. 47, no. 3, pp. 479–487, 2005.

[54] C. D. Wickens, H. Li, A. Santamaria, A. Sebok, and N. B. Sarter, "Stages and levels of automation: An integrated meta-analysis," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 54, (Los Angeles, CA), pp. 389–393, Sage Publications, 2010.

[55] C. A. Miller and R. Parasuraman, "Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control," *Human Factors*, vol. 49, no. 1, pp. 57–75, 2007.

[56] Y. Boussemart, M. L. Cummings, J. L. Fargeas, and N. Roy, "Supervised vs unsupervised learning for operator state modeling in unmanned vehicle settings," *Journal of Aerospace Computing, Information, and Communication*, vol. 8, no. 3, pp. 71–85, 2011.

[57] T. B. Sheridan, "Considerations in modeling the human supervisory controller," *IFAC Proceedings Volumes*, vol. 8, no. 1, pp. 223–228, 1975.

[58] J. Y. Chen, M. J. Barnes, and M. Harper-Sciarini, "Supervisory control of multiple robots: Human-performance issues and user-interface design," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 4, pp. 435–454, 2010.

[59] M. Lind, "Plant modelling for human supervisory control," *Transactions of the Institute of Measurement and Control*, vol. 21, no. 4-5, pp. 171–180, 1999.

[60] A. B. Farjadian, B. Thomsen, A. M. Annaswamy, and D. D. Woods, "Resilient flight control: An architecture for human supervision of automation," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 1, pp. 29–42, 2020.

[61] V. Rodriguez-Fernandez, A. Gonzalez-Pardo, and D. Camacho, "Modelling behaviour in UAV operations using higher order double chain Markov models," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 28–37, 2017.

[62] H. Zhu, M. L. Cummings, M. Elfar, Z. Wang, and M. Pajic, "Operator strategy model development in UAV hacking detection," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 6, pp. 540–549, 2019.

[63] M. Cummings and P. Mitchell, "Operator scheduling strategies in supervisory control of multiple UAVs," *Aerospace Science and Technology*, vol. 11, no. 4, pp. 339–348, 2007.

[64] P. de Vries, C. Midden, and D. Bouwhuis, "The effects of errors on system trust, self-confidence, and the allocation of control in route planning," *International Journal of Human-Computer Studies*, vol. 58, no. 6, pp. 719–735, 2003.

[65] J. Y. Chen and M. J. Barnes, "Human–agent teaming for multirobot control: A review of human factors issues," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 1, pp. 13–29, 2014.

[66] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 30, no. 3, pp. 286–297, 2000.

[67] A. S. Clare, J. C. Macbeth, and M. L. Cummings, "Mixed-initiative strategies for real-time scheduling of multiple unmanned vehicles," in *2012 American Control Conference (ACC)*, pp. 676–682, IEEE, 2012.

[68] A. S. Clare, P. C. Maere, and M. L. Cummings, "Assessing operator strategies for real-time replanning of multiple unmanned vehicles," *Intelligent Decision Technologies*, vol. 6, no. 3, pp. 221–231, 2012.

[69] A. Newell, *Unified Theories of Cognition*. Harvard University Press, 1994.

[70] S. K. Card, *The Psychology of Human-Computer Interaction*. CRC Press, 2018.

[71] T. V. Duong, D. Q. Phung, H. H. Bui, and S. Venkatesh, "Human behavior recognition with generic exponential family duration modeling in the hidden semi-Markov model," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 202–207, IEEE, 2006.

[72] S. Robinson, *Simulation: the Practice of Model Development and Use*, vol. 50. Wiley Chichester, 2004.

[73] V. Grimm and S. F. Railsback, *Individual-Based Modeling and Ecology*. Princeton, NJ: Princeton University Press, 2013.

[74] V. Rodríguez-Fernández, A. Gonzalez-Pardo, and D. Camacho, "Finding behavioral patterns of UAV operators using multichannel hidden Markov models," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, IEEE, 2016.

[75] X. Meng, K. K. Lee, and Y. Xu, "Human driving behavior recognition based on hidden Markov models," in *2006 IEEE International Conference on Robotics and Biomimetics*, pp. 274–279, IEEE, 2006.

[76] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[77] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[78] S. Yu, "Hidden semi-Markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010.

[79] Y. Boussemart, J. Las Fargeas, M. L. Cummings, and N. Roy, "Comparing learning techniques for hidden Markov models of human supervisory control behavior," in *AIAA Infotech*, p. 1842, 2009.

[80] J. Zhou, H. Zhu, M. Kim, and M. L. Cummings, "The impact of different levels of autonomy and training on operators' drone control strategies," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 8, no. 4, pp. 1–15, 2019.

[81] H. Zhu, R. Xu, and M. L. Cummings, "Quantitative operator strategy comparisons across human supervisory control scenarios," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10968–10974, IEEE, 2020.

[82] L. D. Segal and C. D. Wickens, "Taskillan ii: Pilot strategies for workload management," in *Proceedings of the Human Factors Society Annual Meeting*, vol. 34, (Los Angeles, CA), pp. 66–70, SAGE Publications, 1990.

[83] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 69–79, 2015.

[84] I. Dagli, M. Brost, and G. Breuel, "Action recognition and prediction for driver assistance systems using dynamic belief networks," in *International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World*, pp. 179–194, Springer, 2002.

[85] T. Gindele, S. Brechtel, and R. Dillmann, "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 1625–1631, IEEE, 2010.

[86] A. Pentland and A. Liu, "Modeling and prediction of human behavior," *Neural Computation*, vol. 11, no. 1, pp. 229–242, 1999.

[87] E. Frias-Martinez, G. Magoulas, S. Chen, and R. Macredie, "Modeling human behavior in user-adaptive systems: Recent advances using soft computing techniques," *Expert Systems with Applications*, vol. 29, no. 2, pp. 320–329, 2005.

[88] M. Pantic, A. Pentland, A. Nijholt, and T. S. Huang, "Human computing and machine understanding of human behavior: A survey," in *Artifical Intelligence for Human Computing*, pp. 47–71, Springer, 2007.

[89] M. L. Bolton and E. J. Bass, "Generating erroneous human behavior from strategic knowledge in task models and evaluating its impact on system safety with model checking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 6, pp. 1314–1327, 2013.

[90] A. Pentland and A. Liu, "Toward augmented control systems," in *Proceedings of the Intelligent Vehicles Symposium*, pp. 350–355, IEEE, 1995.

[91] E. J. Haas and M. Mattson, "A qualitative comparison of susceptibility and behavior in recreational and occupational risk environments: Implications for promoting health and safety," *Journal of Health Communication*, vol. 21, no. 6, pp. 705–713, 2016.

[92] G. Markkula, O. Benderius, K. Wolff, and M. Wahde, "A review of near-collision driver behavior models," *Human Factors*, vol. 54, no. 6, pp. 1117–1143, 2012.

[93] A. Traulsen, D. Semmann, R. D. Sommerfeld, H.-J. Krambeck, and M. Milinski, "Human strategy updating in evolutionary games," *Proceedings of the National Academy of Sciences*, vol. 107, no. 7, pp. 2962–2966, 2010.

[94] F. Lieder, D. Plunkett, J. B. Hamrick, S. J. Russell, N. Hay, and T. Griffiths, "Algorithm selection by rational metareasoning as a model of human strategy selection," *Advances in Neural Information Processing Systems*, vol. 27, pp. 2870–2878, 2014.

[95] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology," *Bulletin of the American Mathematical Society*, vol. 73, no. 3, pp. 360–363, 1967.

[96] L. R. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 3, pp. 4–16, Jan. 1986.

[97] J. S. Boreczky and L. D. Wilcox, "A hidden Markov model framework for video segmentation using audio and image features," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98*, vol. 6, pp. 3741–3744, IEEE, 1998.

[98] H. Choi and R. G. Baraniuk, "Multiscale image segmentation using wavelet-domain hidden Markov models," *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1309–1321, 2001.

[99] T. Suzuki, S. Sekizawa, S. Inagaki, S. Hayakawa, N. Tsuchida, T. Tsuda, and H. Fujinami, "Modeling and recognition of human driving behavior based on stochastic switched arx model," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 5095–5100, IEEE, 2005.

[100] G. Schwarz *et al.*, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[101] A. J. Stimpson, L. S. Buinhas, S. Bezek, Y. Boussemart, and M. L. Cummings, "A model-based measure to assess operator adherence to procedures," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 56, (Los Angeles, CA), pp. 2452–2456, SAGE Publications, 2012.

[102] Y. Boussemart and M. L. Cummings, "Predictive models of human supervisory control behavioral patterns using hidden semi-Markov models," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 7, pp. 1252–1262, 2011.

[103] M. Elfar, H. Zhu, A. Raghunathan, Y. Y. Tay, J. Wubbenhorst, M. Cummings, and M. Pajic, "Platform for security-aware design of human-on-the-loop cyber-physical systems," in *Proceedings of the 8th International Conference on Cyber-Physical Systems*, pp. 93–93, ACM, 2017.

[104] B. Donmez, C. Nehme, and M. L. Cummings, "Modeling workload impact in multiple unmanned vehicle supervisory control," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, pp. 1180–1190, Nov. 2010.

[105] S. M. E. Sahraeian and B. Yoon, "A novel low-complexity HMM similarity measure," *IEEE Signal Processing Letters*, vol. 18, no. 2, pp. 87–90, 2011.

[106] C. Bahlmann and H. Burkhardt, "Measuring HMM similarity with the bayes probability of error and its application to online handwriting recognition," in *Proceedings of Sixth International Conference on Document Analysis and Recognition (ICDAR)*, pp. 406–411, IEEE, 2001.

[107] R. B. Lyngso, C. N. Pedersen, and H. Nielsen, "Metrics and similarity measures for hidden Markov models," in *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-99)*, pp. 178–186, 1999.

[108] B. Juang and L. R. Rabiner, "A probabilistic distance measure for hidden Markov models," *AT&T Technical Journal*, vol. 64, pp. 391–408, Feb. 1985.

[109] L. R. Rabiner, B. Juang, S. E. Levinson, and M. M. Sondhi, "Recognition of isolated digits using hidden Markov models with continuous mixture densities," *AT&T Technical Journal*, vol. 64, no. 6, pp. 1211–1234, 1985.

[110] M. Hwang and X. Huang, "Shared-distribution hidden Markov models for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 4, pp. 414–420, 1993.

[111] J. A. Whittaker and M. G. Thomason, "A Markov chain model for statistical software testing," *IEEE Transactions on Software Engineering*, vol. 20, no. 10, pp. 812–824, 1994.

[112] M. N. Do, "Fast approximation of kullback-leibler distance for dependence trees and hidden Markov models," *IEEE Signal Processing Letters*, vol. 10, no. 4, pp. 115–118, 2003.

[113] C. Yu, M. Deng, and S. Yau, "DNA sequence comparison by a novel probabilistic method," *Information Sciences*, vol. 181, no. 8, pp. 1484–1492, 2011.

[114] E. Coviello, A. B. Chan, and G. R. Lanckriet, "Clustering hidden Markov models with variational HEM," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 697–747, 2014.

[115] A. Jagota, R. B. Lyngsø, and C. N. Pedersen, "Comparing a hidden Markov model and a stochastic context-free grammar," in *International Workshop on Algorithms in Bioinformatics*, pp. 69–84, Springer, 2001.

[116] R. B. Lyngsø and C. N. Pedersen, "Complexity of comparing hidden Markov models," in *International Symposium on Algorithms and Computation*, pp. 416–428, Springer, 2001.

[117] R. B. Lyngsø and C. N. Pedersen, "The consensus string problem and the complexity of comparing hidden Markov models," *Journal of Computer and System Sciences*, vol. 65, no. 3, pp. 545–569, 2002.

[118] J. Söding, "Protein homology detection by HMM–HMM comparison," *Bioinformatics*, vol. 21, no. 7, pp. 951–960, 2005.

[119] M. H. Kutner, C. J. Nachtsheim, J. Neter, and W. Li, *Applied Linear Statistical Models*, vol. 5. McGraw-Hill Irwin New York, 2005.

[120] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science and Business Media, 2009.

[121] M. Bar-Hillel, "The role of sample size in sample evaluation," *Organizational Behavior and Human Performance*, vol. 24, no. 2, pp. 245–257, 1979.

[122] A. D. Well, A. Pollatsek, and S. J. Boyce, "Understanding the effects of sample size on the variability of the mean," *Organizational Behavior and Human Decision Processes*, vol. 47, no. 2, pp. 289–312, 1990.

[123] M. Mason, "Sample size and saturation in PhD studies using qualitative interviews," in *Forum: Qualitative Social Research*, vol. 11, 2010.

[124] J. J. Francis, M. Johnston, C. Robertson, L. Glidewell, V. Entwistle, M. P. Eccles, and J. M. Grimshaw, "What is an adequate sample size? operationalising data saturation for theory-based interview studies," *Psychology and Health*, vol. 25, no. 10, pp. 1229–1245, 2010.

[125] Y. Qi, J. W. Paisley, and L. Carin, "Music analysis using hidden Markov mixture models," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5209–5224, 2007.

[126] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, *et al.*, "Starcraft ii: A new challenge for reinforcement learning," *arXiv preprint arXiv:1708.04782*, 2017.

[127] M. Cummings, L. Huang, H. Zhu, D. Finkelstein, and R. Wei, "The impact of increasing autonomy on training requirements in a UAV supervisory control task," *Journal of Cognitive Engineering and Decision Making*, vol. 13, no. 4, pp. 295–309, 2019.

[128] H. Zhu, M. Elfar, M. Pajic, Z. Wang, and M. L. Cummings, "Human augmentation of UAV cyber-attack detection," in *International Conference on Augmented Cognition*, pp. 154–167, Springer, 2018.

[129] H. Zhu and M. L. Cummings, "The stability of human supervisory control operator behavioral models using hidden Markov models," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6971–6978, IEEE, 2019.

[130] G. Pajares, "Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs)," *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 4, pp. 281–330, 2015.

[131] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner Jr, "Assessing the spoofing threat: Development of a portable GPS civilian spoofer," in *Proceedings of the ION GNSS International Technical Meeting of the Satellite Division*, vol. 55, p. 56, 2008.

[132] D. P. Shepard, T. E. Humphreys, and A. A. Fansler, "Evaluation of the vulnerability of phasor measurement units to GPS spoofing attacks," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 3-4, pp. 146–153, 2012.

[133] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[134] K. D. Wesson, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "An evaluation of the vestigial signal defense for civil GPS anti-spoofing," in *Proceedings of the ION GNSS Meeting*, 2011.

[135] A. Broumandan, A. Jafarnia-Jahromi, V. Dehghanian, J. Nielsen, and G. Lachapelle, "GNSS spoofing detection in handheld receivers based on signal spatial correlation," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pp. 479–487, IEEE, Apr. 2012.

[136] K. Wesson, M. Rothlisberger, and T. Humphreys, "Practical cryptographic civil GPS signal authentication," *Journal of the Institute of Navigation*, vol. 59, no. 3, pp. 177–193, 2012.

[137] M. L. Psiaki, B. W. O'Hanlon, J. A. Bhatti, D. P. Shepard, and T. E. Humphreys, "GPS spoofing detection via dual-receiver correlation of military signals," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2250–2267, 2013.

[138] M. Pajic, J. Weimer, N. Bezzo, O. Sokolsky, G. J. Pappas, and I. Lee, "Design and implementation of attack-resilient cyberphysical systems: With a focus on attack-resilient state estimators," *IEEE Control Systems*, vol. 37, no. 2, pp. 66–81, 2017.

[139] K. D. Wesson, B. L. Evans, and T. E. Humphreys, "A combined symmetric difference and power monitoring GNSS anti-spoofing technique," in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 217–220, IEEE, 2013.

[140] T. E. Humphreys, "Detection strategy for cryptographic GNSS anti-spoofing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1073–1090, 2013.

[141] J. Sun, B. Li, Y. Jiang, and C. Wen, "A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes," *Sensors*, vol. 16, no. 11, p. 1778, 2016.

[142] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 294–307, 2005.

[143] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[144] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[145] B. De Bruyn and G. A. Orban, "Human velocity and direction discrimination measured with random dot patterns," *Vision Research*, vol. 28, no. 12, pp. 1323–1335, 1988.

[146] M. R. Endsley, "Level of automation effects on performance, situation awareness and workload in a dynamic control task," *Ergonomics*, vol. 42, no. 3, pp. 462–492, 1999.

[147] D. B. Kaber and M. R. Endsley, "The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task," *Theoretical Issues in Ergonomics Science*, vol. 5, no. 2, pp. 113–153, 2004.

[148] J. Rasmussen, "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 3, pp. 257–266, 1983.

[149] M. Cummings, "Informing autonomous system design through the lens of skill-, rule-, and knowledge-based behaviors," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 58–61, 2018.

[150] A. I. Glendon, S. Clarke, and E. McKenna, *Human Safety and Risk Management*. CRC Press, 2016.

[151] S. Drivalou and N. Marmaras, "Supporting skill-, rule-, and knowledge-based behaviour through an ecological interface: An industry-scale application," *International Journal of Industrial Ergonomics*, vol. 39, no. 6, pp. 947–965, 2009.

Biography

Haibei Zhu received his B.S. (Magna Cum Laude) degree in Electrical Engineering (2015) at Rensselaer Polytechnic Institute, NY, USA. He is graduating with his Ph.D. in Computer Engineering at Duke University, NC, USA. His research interests include human-computer interaction, machine learning and data mining.

His peer-reviewed articles include:

- Zhu, H., Xu, R., & Cummings, M. L. (2020, October). Quantitative Operator Strategy Comparisons across Human Supervisory Control Scenarios, In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2020) (pp.10968-10974). IEEE.
- Zhu, H., & Cummings, M. L. (2019, November). The Stability of Human Supervisory Control Operator Behavioral Models Using Hidden Markov Models. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019) (pp. 6971-6978). IEEE.
- Zhu, H., Cummings, M. L., Elfar, M., Wang, Z., & Pajic, M. (2019). Operator Strategy Model Development in UAV Hacking Detection. IEEE Transactions on Human-Machine Systems, 49(6), 540-549.
- Zhu, H., Elfar, M., Pajic, M., Wang, Z., & Cummings, M. L. (2018, July). Human Augmentation of UAV Cyber-Attack Detection. In International Conference on Augmented Cognition (pp. 154-167). Springer, Cham.
- Elfar, M., Zhu, H., Cummings, M. L., & Pajic, M. (2019, May). Security-Aware Synthesis of Human-UAV Protocols. In 2019 International Conference on Robotics and Automation (ICRA) (pp. 8011-8017). IEEE.
- Zhou, J., Zhu, H., Kim, M., & Cummings, M. L. (2019). The Impact of Different Levels of Autonomy and Training on Operators' Drone Control Strategies. ACM Transactions on Human-Robot Interaction (THRI), 8(4), 1-15.
- Cummings, M., Huang, L., Zhu, H., Finkelstein, D., & Wei, R. (2019). The Impact of Increasing Autonomy on Training Requirements in a UAV Supervisory Control Task. Journal of Cognitive Engineering and Decision Making, 13(4), 295-309.
- Elfar, M., Zhu, H., Raghunathan, A., Tay, Y. Y., Wubbenhorst, J., Cummings, M. L., & Pajic, M. (2017, April). Platform for Security-Aware Design of Human-on-the-Loop Cyber-Physical Systems. In Proceedings of the 8th International Conference on Cyber-Physical Systems (pp. 93-93).

- Zhu, H., Misu, T., Martin, S., Wu, X., & Akash, K. Improving Driver Situation Awareness Prediction using Human Visual Sensory and Memory Mechanism, submitted to 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2021) in 2021.02, accepted in 2021.06
- Zhu, H., & Cummings, M. L. The Impact of Data Quantity on Modeling Operator Strategies. submitted to IEEE Transactions on Human-Machine Systems in 2021.03