

---

Electronic Theses and Dissertations, 2020-

---

2022

## Applications for Machine Learning on Readily Available Data from Virtual Reality Training Experiences

Alec Moore  
*University of Central Florida*

Find similar works at: <https://stars.library.ucf.edu/etd2020>  
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2020- by an authorized administrator of STARS. For more information, please contact [STARS@ucf.edu](mailto:STARS@ucf.edu).

---

### STARS Citation

Moore, Alec, "Applications for Machine Learning on Readily Available Data from Virtual Reality Training Experiences" (2022). *Electronic Theses and Dissertations, 2020-*. 1416.  
<https://stars.library.ucf.edu/etd2020/1416>

APPLICATIONS FOR MACHINE LEARNING ON READILY AVAILABLE DATA FROM  
VIRTUAL REALITY TRAINING EXPERIENCES

by

ALEC G. MOORE

M.S. Computer Science The University of Texas at Dallas, 2016

B.S. Computer Science The University of Texas at Dallas, 2014

B.S. Physics The University of Texas at Dallas, 2014

A dissertation submitted in partial fulfilment of the requirements  
for the degree of Doctor of Philosophy  
in the Department of Computer Science  
in the College of Engineering and Computer Science  
at the University of Central Florida  
Orlando, Florida

Fall Term  
2022

Major Professor: Ryan P. McMahan

© 2022 Alec G. Moore

## **ABSTRACT**

The purpose of the research presented in this dissertation is to improve virtual reality (VR) training systems by enhancing their understanding of users. While the field of intelligent tutoring systems (ITS) has seen value in this approach, much research into making use of biometrics to improve user understanding and subsequently training, relies on specialized hardware. Through the presented research, I show that with machine learning (ML), the VR system itself can serve as that specialized hardware for VR training systems.

I begin by discussing my explorations into using an ecologically valid, specialized training simulation as a testbed to predict knowledge acquisition by users unfamiliar with the task being trained. Then I look at predicting the cognitive and psychomotor outcomes retained after a one week period. Next I describe our work towards using ML models to predict the transfer of skills from a non-specialized VR assembly training environment to the real-world, based on recorded tracking data.

I continue by examining the identifiability of participants in the specialized training task, allowing us to better understand the associated privacy concerns and how the representation of the data can affect identifiability. By using the same tasks separated temporally by a week, we expand our understanding of the diminishing identifiability of user's movements. Finally, I make use of the assembly training environment to explore the feasibility of across-task identifiability, by making use of two different tasks with the same context.



To my light, my love, and my best friend, Kristina

## ACKNOWLEDGMENTS

I would like to thank Ryan McMahan for his expert guidance over the years. I've learned an incredible amount since he initially offered me a research role in his lab. The opportunity to learn from and work alongside Ryan has been not only been an invaluable experience, but has also proved to be some of the most fun I've ever had. This work would be impossible without him. I would also like to thank Dr. Nicholas Ruozzi for stepping up to provide his expertise when we began our explorations into Machine Learning. This work would not have been successful without him either. I would also like to thank my committee for providing the feedback and guidance needed to help me improve this work.

In addition to my mentors, I also owe a great deal to my lab partners from the FIVE Lab at The University of Texas at Dallas and the XRT Lab at University of Central Florida. I would like to specifically thank Dr. J. C. Eubanks. Working late into the night on the VR Robotic Operating Room together will forever be some of my fondest memories of my doctoral work. I also thank Tiffany D. Do, who graciously stepped up to run the FAB study once I moved away from Florida. Her support and feedback has been invaluable. I would also like to thank Austin Matthews as well. His energy and input has been incredibly helpful. Additionally, I am thankful for my colleagues in the UCF Knighthawk Audubon campus chapter, who provided the space needed to recharge.

Finally, I would like to thank my family for supporting me throughout my academic endeavours. My parents for their guidance and wisdom, my siblings for reminding me not to take things too seriously, and my fiancée, Kristina, who is the reason to wake up in the morning.

# TABLE OF CONTENTS

LIST OF FIGURES . . . . .	xvii
LIST OF TABLES . . . . .	xx
CHAPTER 1: INTRODUCTION . . . . .	1
1.1 Overview . . . . .	1
1.2 Brief Background . . . . .	1
1.2.1 Virtual Reality . . . . .	1
1.2.2 Machine Learning . . . . .	2
1.2.3 Intelligent Tutoring Systems . . . . .	2
1.3 Motivation . . . . .	3
1.4 Problem Statement . . . . .	3
1.5 Research Questions and Methodologies . . . . .	4
1.5.1 Learning Outcomes . . . . .	4
1.5.1.1 RQ1: Can readily available VR tracking data be used to predict cognitive outcomes from a VR training scenario? . . . . .	4

1.5.1.2	RQ2: Can readily available VR tracking data be used to predict psychomotor outcomes from a VR training scenario? . . . . .	4
1.5.1.3	Methodologies . . . . .	5
1.5.2	Identifiability . . . . .	5
1.5.2.1	RQ3: How does identifiability of readily available VR tracking data change longitudinally? . . . . .	5
1.5.2.2	Methodology . . . . .	5
1.5.2.3	RQ4: How does identifiability of readily available VR tracking data change between tasks? . . . . .	6
1.5.2.4	Methodology . . . . .	6
1.6	Contributions . . . . .	6
1.6.1	Predicting Learning Outcomes from Readily Available Tracking Data in VR	7
1.6.2	Identifiability of VR Tracking Data . . . . .	7
CHAPTER 2: LITERATURE REVIEW . . . . .		8
2.1	Training Outcomes . . . . .	8
2.1.1	Predicting Cognitive Outcomes . . . . .	8
2.1.2	Predicting Cognitive Retention Outcomes . . . . .	10
2.1.3	Predicting Psychomotor Outcomes . . . . .	11

2.1.4	Predicting Affective Outcomes . . . . .	12
2.2	Identifiability and Obfuscation . . . . .	14
2.2.1	Machine Learning Biometrics for Authentication . . . . .	14
2.2.2	Machine Learning VR Tracking Data . . . . .	15
2.2.2.1	VR Tracking-based Authentication . . . . .	15
2.2.2.2	VR Tracking-based Identification . . . . .	16
2.2.3	Machine Learning Velocity for User Experiences . . . . .	17
2.3	Identifiability and Authentication Research by Inclusion of XR Tracking Features . . . . .	18
2.3.1	Identification and Authentication with Eye-tracking . . . . .	18
2.3.2	Identification and Authentication without Eye-tracking . . . . .	20
CHAPTER 3: EXTRACTING VELOCITY-BASED USER-TRACKING FEATURES TO PREDICT LEARNING GAINS IN A VIRTUAL REALITY TRAINING AP- PLICATION . . . . .		23
3.1	Introduction . . . . .	24
3.2	VR Training User Study . . . . .	27
3.2.1	Materials . . . . .	28
3.2.2	Procedure . . . . .	28
3.2.3	Participants . . . . .	29

3.2.4	Descriptive Statistics . . . . .	29
3.3	Machine Learning Experiment . . . . .	30
3.3.1	Data Pre-processing . . . . .	31
3.3.1.1	Low- and High-Learning Gains . . . . .	31
3.3.1.2	Training and Testing Data Sets . . . . .	31
3.3.1.3	Velocity-Based Input Features . . . . .	32
3.3.1.4	Data Segments . . . . .	33
3.3.1.5	Feature Representations . . . . .	33
3.3.2	Model Setup . . . . .	35
3.3.2.1	Hyperparameter Selection . . . . .	36
3.3.2.2	Evaluation Metrics . . . . .	38
3.3.3	Results . . . . .	38
3.3.3.1	Comparison of Feature Representations . . . . .	39
3.3.3.2	Noisy Data and Head Orientation . . . . .	40
3.4	Visual Inspection of Machine Learning Results . . . . .	41
3.5	Discussion . . . . .	44
3.5.1	Feasibility of Predicting Learning Gains . . . . .	45

3.5.2	Movements of Learners During VR Training . . . . .	46
3.5.3	Limitations . . . . .	46
3.6	Conclusion . . . . .	47
CHAPTER 4: EXPLORATION OF FEATURE REPRESENTATIONS FOR PREDICTING		
LEARNING AND RETENTION OUTCOMES IN A VIRTUAL REALITY		
TRAINING SCENARIO . . . . .		
		49
4.1	Introduction . . . . .	50
4.2	VR Learning and Retention User Study . . . . .	53
4.2.1	Materials . . . . .	55
4.2.2	Procedure . . . . .	56
4.2.3	Participants . . . . .	57
4.3	Machine Learning Experiments . . . . .	57
4.3.1	Learning and Retention Outcomes . . . . .	57
4.3.2	Experimental Design . . . . .	58
4.3.3	Input Features . . . . .	58
4.3.4	Input Feature Representation . . . . .	59
4.3.5	Hyperparameters . . . . .	59
4.3.6	Scoring Method . . . . .	60

4.4	Machine Learning Results . . . . .	60
4.4.1	Knowledge Acquisition Results . . . . .	61
4.4.2	Knowledge Retention Results . . . . .	63
4.4.3	Performance Retention Results . . . . .	63
4.5	Visual Inspection of Results . . . . .	66
4.6	Discussion . . . . .	67
4.6.1	Position-based versus Velocity-based Models . . . . .	67
4.6.2	Linear-and-Angular versus Linear-Only Models . . . . .	68
4.6.3	Cognitive versus Psychomotor Models . . . . .	69
4.6.4	Limitations . . . . .	70
4.7	Conclusion . . . . .	71
CHAPTER 5: PERSONAL IDENTIFIABILITY AND OBFUSCATION OF USER TRACK-		
ING DATA FROM VIRTUAL REALITY TRAINING SESSIONS . . . . .		72
5.1	Introduction . . . . .	73
5.2	VR Learning and Retention User Study . . . . .	77
5.2.1	Materials . . . . .	78
5.2.2	Procedure . . . . .	79



5.2.3	Participants . . . . .	80
5.3	Identifying Participants in the Same Session . . . . .	80
5.4	Identifying Participants in a Different Session . . . . .	83
5.5	Obfuscating Identities With Velocity-based Data . . . . .	85
5.6	Discussion . . . . .	88
5.6.1	Validation of Within-Session Identification . . . . .	88
5.6.2	Feasibility of Between-Session Identification . . . . .	90
5.6.3	Feasibility of Velocity-based Obfuscation . . . . .	91
5.6.4	Sharing Resources for Replication . . . . .	93
5.6.5	Limitations . . . . .	93
5.7	Conclusion . . . . .	94
CHAPTER 6: A SYSTEMATIC INVESTIGATION OF DEVICE COMBINATIONS AND SPATIAL REPRESENTATIONS FOR IDENTIFYING VIRTUAL REALITY USERS IN AN ASSEMBLY TRAINING ENVIRONMENT . . . . .		96
6.1	Introduction . . . . .	97
6.2	Full-scale Assembly Benchmark . . . . .	99
6.2.1	Experimental Design . . . . .	102
6.2.2	Procedure . . . . .	104

6.2.3	Materials . . . . .	106
6.2.4	Participants . . . . .	106
6.3	Machine Learning Experiment 1 . . . . .	106
6.4	Machine Learning Experiment 2 . . . . .	110
6.5	Machine Learning Experiment 3 . . . . .	113
6.6	Machine Learning Experiment 4 . . . . .	115
6.7	Discussion . . . . .	116
6.7.1	Tracking data from an unspecified task is likely not sufficient for identifying people against their will with these machine learning models . . . . .	116
6.7.2	More tracked devices yields better identification accuracy within the same task . . . . .	117
6.7.3	More tracked devices does not yield better identification accuracy across similar but slightly different tasks . . . . .	118
6.7.4	Using position and orientation generally yields higher identification accuracy	118
6.7.5	Identities can be partially obfuscated by encoding the velocities of tracked devices instead of positions . . . . .	119
6.7.6	Limitations . . . . .	119
6.8	Conclusion . . . . .	120

CHAPTER 7: INSIGHTS AND FUTURE WORK . . . . .	122
7.1 Introduction . . . . .	122
7.2 Insightful Conjectures . . . . .	122
7.2.1 The Appropriateness of Time-derivative Data . . . . .	122
7.2.2 Velocity is an Indicator of Cognitive Processing . . . . .	123
7.2.3 Position Encodes Physiology which is a Key Feature for Identifiability . . . . .	123
7.2.4 Task Categorization . . . . .	124
7.3 Future Work . . . . .	125
7.3.1 Predicting Real-world Transfer of Skills . . . . .	125
7.3.2 Obfuscation . . . . .	126
CHAPTER 8: CONCLUSION . . . . .	128
APPENDIX A: A FORMATIVE EVALUATION METHODOLOGY FOR VIRTUAL RE- ALITY TRAINING SIMULATIONS . . . . .	129
A.1 Introduction . . . . .	130
A.2 Formative Evaluation . . . . .	132
A.3 New Formative Evaluation Methodology . . . . .	133
A.4 Case Study: The Real-World Task . . . . .	136

A.5	Case Study: The VR Simulation . . . . .	136
A.6	Case Study: The Rigorous User Study . . . . .	141
A.6.1	Independent Variable . . . . .	141
A.6.2	Dependent Variables . . . . .	141
A.6.2.1	Subtask Metrics . . . . .	141
A.6.2.2	Questionnaires . . . . .	141
A.6.2.3	Knowledge Posttest . . . . .	142
A.6.3	Materials . . . . .	142
A.6.4	Procedure . . . . .	142
A.6.5	Participants . . . . .	143
A.7	Case Study: The Subtask Analyses and Results . . . . .	143
A.7.1	Subtask Analyses of Completion Times . . . . .	143
A.7.2	Subtask Analyses of Errors . . . . .	144
A.7.3	Questionnaire Results . . . . .	146
A.7.4	Knowledge Posttest Results . . . . .	147
A.8	Case Study: The Interaction Inspection . . . . .	149
A.8.1	Subtask #2 Inspection: Consult surgeon . . . . .	149

A.8.2	Subtask #7 Inspection: Hold instrument carriage . . . . .	149
A.8.3	Subtask #9 Inspection: Rotate wrench . . . . .	150
A.8.4	Subtask #12 Inspection: Remove instrument . . . . .	151
A.8.5	Subtask #17 Inspection: Use cannula lever . . . . .	151
A.9	Discussion . . . . .	152
A.9.1	A Useful Formative Evaluation Methodology . . . . .	152
A.9.2	Recommendations for Subtask Analyses . . . . .	152
A.9.3	Limitations . . . . .	153
A.10	Conclusion . . . . .	154
A.11	Acknowledgments . . . . .	154
APPENDIX B:	ALL RESULTS FROM SYSTEMATIC INVESTIGATION OF DEVICE COMBINATIONS AND SPATIAL REPRESENTATIONS FOR IDENTIFY- ING VIRTUAL REALITY USERS IN AN ASSEMBLY TRAINING ENVI- RONMENT . . . . .	155
B.1	Random Forest Results . . . . .	156
B.2	Gradient Boosting Machine Results . . . . .	160
B.3	K Nearest Neighbors Results . . . . .	164
APPENDIX C:	IRB APPROVAL . . . . .	168

## LIST OF FIGURES

Figure 3.1: The VR training simulation used in our user study. . . . .	27
Figure 3.2: Visualizations of a pair of corresponding low- and high-learning gain segments. The underlying VR tasks involved: 1) Requesting an instrument release kit (IRK) via a floating dialog menu, 2) Receiving the IRK from a virtual agent, 3) Using the IRK on the instrument arm, 4) Leaving the IRK on the back table, and 5) Removing the instrument. The blue-red color gradient of the connected HMD-controller lines indicates the oldest (blue) and newest (red) frames in which a linear velocity exceeded 1 m/s. . . . .	43
Figure 4.1: A first-person perspective of the VR training application used for our research.	55
Figure 4.2: Visualizations of motion data from a low-performance participant and a high-performance participant, based on all three metrics, for the same set of actions.	65
Figure 5.1: A first-person perspective of the VR training application. . . . .	79
Figure 5.2: Classification accuracy using position feature vectors, averaged over 20 Monte-Carlo cross-validations within session data for each participant. Error bars indicate range of values. . . . .	83
Figure 5.3: Classification accuracy using position feature vectors, averaged over 10 sub-sessions for each participant, across sessions. Error bars indicate range of values. . . . .	85

Figure 5.4: Correct identifications for each additional second of position-based features for the first 10 minutes of data per participant. . . . .	86
Figure 5.5: Classification accuracy using velocity feature vectors, averaged over 20 Monte- Carlo cross-validations for each participant. Error bars indicate range of values.	88
Figure 5.6: Correct identifications for each additional second of velocity-based features for the first 10 minutes of data per participant. . . . .	89
Figure 6.1: Two of the types of connectors in the toy set and their corresponding models.	100
Figure 6.2: Attaching a tube in the Full-scale Assembly Benchmark. From top left, grab- bing a pipe, attaching it to the work in progress, inserting a screw, and turning the screw with the key. . . . .	101
Figure 6.3: The completed structures. Build A is shown on the left, and build B on the right. Screws do not appear as an artifact of the way the screenshots were captured, but were visible in the application. . . . .	102
Figure 6.4: Both builds made use of the same pieces, in the same locations. They con- sisted of two of each color pipe (red, green, blue, yellow), two elbow connec- tors, three three-way connectors, and 12 screws. Also visible in the middle is the metallic key used to turn the screws to secure connections. . . . .	103
Figure 6.5: A still of the participant’s perspective of the VR onboarding task. . . . .	104
Figure 6.6: The full participant flow through the experiment. . . . .	105

Figure 7.1: The participant flow as experienced through the experiment to support real-world transfer of skills . . . . .	125
Figure A.1: Overview of our formative evaluation methodology. . . . .	134
Figure A.2: A frame of the video used to capture subtask metrics. . . . .	137
Figure A.3: One of the floating windows used for communicating with the virtual surgeon and non-sterile staff member. . . . .	139
Figure A.4: Example of using the simple virtual hand technique to rotate the grip release wrench. . . . .	140
Figure A.5: We required walking and looking to read error messages. . . . .	140



## LIST OF TABLES

Table 2.1: An overview of prior work that have used machine learning to predict learning outcomes. . . . .	8
Table 2.2: An overview of related work that have predicted learning and retention outcomes. . . . .	13
Table 2.3: A comparisons of the input features used by related works. Each letter under a given paper indicates a condition that was evaluated in that paper consisting of the data for each row in which the letter appears. The chart also shows what kinds of classifiers each work explored as well as the number of participants, and the highest accuracy attained by their best-performing classifier among the explored representations. . . . .	19
Table 3.1: Value range of hyperparameters considered. . . . .	37
Table 3.2: Best hyperparameter for each model. . . . .	38
Table 3.3: Experimental results on <i>Train</i> split using different attributes combination and feature extraction technique. . . . .	39
Table 3.4: Experimental results on <i>Test</i> split using different attributes combination and feature extraction technique. . . . .	39
Table 4.1: The subtasks and their required interactions involved in the VR training simulation. . . . .	54

Table 4.2: Evaluation metrics and categorizations of participants. . . . .	58
Table 4.3: Value range of hyperparameters considered. . . . .	60
Table 4.4: Knowledge Acquisition prediction results. For each of the six sets of input features, the accuracy of the low-performance and high-performance predictions and the MCC are shown for both the training and testing data. Accuracy and MCC are also shown for each set of input features across the entire dataset (i.e., both the training and testing data). . . . .	62
Table 4.5: Knowledge Retention prediction results. For each of the six sets of input features, the accuracy of the low-performance and high-performance predictions and the MCC are shown for both the training and testing data. Accuracy and MCC are also shown for each set of input features across the entire dataset (i.e., both the training and testing data). . . . .	62
Table 4.6: Performance Retention prediction results. For each of the six sets of input features, the accuracy of the low-performance and high-performance predictions and the MCC are shown for both the training and testing data. Accuracy and MCC are also shown for each set of input features across the entire dataset (i.e., both the training and testing data). . . . .	64
Table 4.7: Results of Mann-Whitney U tests comparing the distribution differences of low and high performances among participants with or without prior VR experiences. . . . .	67
Table 5.1: The subtasks and their associated required interactions involved in our VR training simulation. . . . .	78

Table 6.1: The order of steps for builds A and B. The letters R, G, B, and Y represent red, green, blue, and yellow pipes respectively, T and L represent Three-way and Elbow joints, respectively. . . . .	103
Table 6.2: An overview of the conditions compared in this chapter. Explored conditions include the 0th and 1st order time derivative, within- and between-session predictions, the inclusion and exclusion of individual trackers, the type of data used from those trackers, and the model used. . . . .	107
Table 6.3: Within-session identification accuracy for Random Forest, with position and orientation data, trained and evaluated with data from session A. . . . .	108
Table 6.4: Within-session identification accuracy for Random Forest, with position and orientation data, trained and evaluated with data from session B. . . . .	108
Table 6.5: Between-session identification accuracy for Random Forest, with position and orientation data, trained on A and evaluated on B. . . . .	111
Table 6.6: Between-session identification accuracy for Random Forest, with position and orientation data, trained on B and evaluated on A. . . . .	112
Table 6.7: Within-session identification accuracy for Random Forest, with velocity and angular velocity data, trained and evaluated on A. . . . .	114
Table 6.8: Within-session identification accuracy for Random Forest, with velocity and angular velocity data, trained and evaluated on B. . . . .	114
Table 6.9: Between-session identification accuracy for Random Forest, with velocity and angular velocity data, trained on data from A and evaluated with data from session B. . . . .	115

Table 6.10: Between-session identification accuracy for Random Forest, with velocity and angular velocity data, trained on data from B and evaluated with data from session A. . . . . 116

Table A.1: The subtasks for our VR simulation and their associated interactions and SME completion times (in seconds). . . . . 138

Table A.2: The percentage of participants that successfully completed each subtask, based on the four proposed time thresholds. Asterisks indicate subpar subtasks. . . . 145

Table A.3: The mean number of errors and percentage of participants that successfully (without errors) completed each subtask. Asterisks indicate subpar subtasks. . 146

Table A.4: Descriptive statistics for standardized questionnaires. . . . . 147

Table A.5: Descriptive statistics for scenario fidelity questions. . . . . 147

Table A.6: Comparison of success on the posttest, time, and errors. . . . . 148

# CHAPTER 1: INTRODUCTION

## 1.1 Overview

This chapter begins by providing a background of research in virtual reality, intelligent tutoring systems, and machine learning techniques. While not comprehensive, the intent of this section is to provide a foundational background for the reader, and more detailed and relevant works to the research presented are discussed in chapter 2. After providing some background information, this chapter then moves into discussing the motivation, methodologies and contributions presented herein.

## 1.2 Brief Background

### *1.2.1 Virtual Reality*

Virtual reality (VR) has storied history stemming from Ivan Sutherland's "Ultimate Display" [112]. Since then, VR still falls short of the complete replacement of the senses originally envisioned by Sutherland, but in the last decade cheaper hardware has reduced the price of VR systems to the point that they are now relatively affordable for individuals. Outside of personal gaming, VR has also seen a rise in popularity among organizations that see value in its use for training and education, particularly medical and military organizations, where real-world in situ training can be dangerous or costly. For such uses, VR offers some unique benefits that make the cost of developing training applications worthwhile over traditional non-immersive trainers, video tutorials, or text guides.

After conducting a survey of interaction design philosophies, Bowman, Kruijff, LaViola, and

Poupyrev [11] found that natural interactions should be used when the replication of physical actions is important, and Saposnik et al. [104] determined that these kinds of interactions can be useful for training gross motor skills, which are afforded by readily available VR systems. While VR can afford benefits for psychomotor learning (concepts related to the perception, action, and motor activity of educational objectives [110]), it should be noted that for other tasks, such as those that focus on cognition, natural interactions can be an impediment to performance [11, 76].

### *1.2.2 Machine Learning*

Similar to VR, machine learning (ML) has existed as a field since the 1960's, and has recently seen a great increase in use. While cheaper and improved hardware has also been a source of improved utility, the greater availability of large datasets has resulted in the success of models conceptualized even in the early days of ML [41]. In broad terms, ML is the process of allowing a machine to learn how to do a given task by exposing it to representative data of that task and allowing it to improve its success as measured by a given heuristic [82]. This broad category of algorithms can build models that are successful at several tasks including classification, regression, and generation of data.

### *1.2.3 Intelligent Tutoring Systems*

The field of intelligent tutoring systems (ITS) seeks to improve learning by modeling “*what* is being taught, *who* is being taught, and *how* to teach [them],” [108]. This is motivated on the consensus that individualized teaching is the most effective form of education for most domains [93]. While the ITS field has the word “intelligent” in its name, many of the works in the field do not necessarily make use of artificial intelligence or ML concepts, instead relying on crafted models by domain experts. Some ITSs that do make use of ML, such as those researched by Won

et al. [123], and Schneider and Blikstein [106], use streams of data beyond the typical event and error tracking, such as posture and eye-gaze data, improving the system's understanding of the user's mental state and learning.

### 1.3 Motivation

As noted, the success of ML models is broadly contingent on the availability of large quantities of data. Simultaneously, recent developments in VR hardware that enable better and more detailed interactions also, by necessity, provide better and more detailed tracking data. At the time of writing, consumer VR systems can track the positions and orientations of the head-mounted display and controllers with high accuracy at 90Hz [52]. Several HMDs also afford eye-tracking, and some controllers offer more detailed hand-pose information than binary button presses. As this VR tracking data improves in quality and more streams of data are made available, its possibility for use in developing ML models to improve tutoring systems increases.

### 1.4 Problem Statement

Because of the improving availability of readily available tracking data, understanding what applications for machine learning are feasible is important to provide a basis for which researchers and practitioners can implement systems that leverage such models. Additionally, investigation into how identifiable this data is is important so that consumers and platforms can recognize privacy implications of the collection of such data and appropriately weigh the benefits of the potential uses.

## 1.5 Research Questions and Methodologies

The goal of this research is to investigate the feasibility of developing and applying ML models to readily available tracking data in VR training scenarios. Ultimately we believe that such an approach is an opportunity to improve learning outcomes by allowing for prediction-based scaffolding, but we are also interested in understanding how identifiable this data is. To this end, we have identified 5 main research questions.

### 1.5.1 Learning Outcomes

*1.5.1.1 RQ1: Can readily available VR tracking data be used to predict cognitive outcomes from a VR training scenario?*

**H1:** Previous work has indicated that using data related to the posture of the body can be useful for predicting learning gains [106, 123] thus our hypothesis was that the readily available tracking data which is correlated to the user's posture would be useful for developing ML models for predicting knowledge acquisition in a VR training scenario. In the same way that one can tell if someone is paying attention by their posture and movements, we anticipated that a model could interpret something similar from the positions and orientations of the controllers and HMD, which are derived from posture and movements.

*1.5.1.2 RQ2: Can readily available VR tracking data be used to predict psychomotor outcomes from a VR training scenario?*

**H2:** We anticipate that tracking data will be able to predict psychomotor outcomes. This anticipation is partially driven by the same reasoning as H1, as well as increasing evidence that machine



learning on movements elicited by learners can be leveraged for automated assessment of skills for tasks like laparoscopic or robotic surgeries [17].

### *1.5.1.3 Methodologies*

Chapters 3, and 4 make use of a robotic operating room VR training scenario, that was designed using the formative evaluation methodology described in Appendix A. These studies investigate cognitive and psychomotor outcomes. Our expectation was that a machine learning model could be trained to recognize movements associated with attention and engagement during training, and would be able to predict these learning outcomes for participants.

## *1.5.2 Identifiability*

### *1.5.2.1 RQ3: How does identifiability of readily available VR tracking data change longitudinally?*

**H3** We anticipated that identifiability would decrease due to variations in physiological, emotional, and mental state. Prior work such as that by Ferber et al. [38] has shown within-day movement variability to be lower than between-day movement variability, and we would expect similar effects to introduce difficulty in identifying participants based on their movements.

### *1.5.2.2 Methodology*

We examined the identifiability of the data collected from participants in Chapter 5. By training models to perform within-session identification and across-session identification for sessions spaced one week apart, our expectation was that the identifiability would decrease across-session,

but still remain well above random chance.

*1.5.2.3 RQ4: How does identifiability of readily available VR tracking data change between tasks?*

**H4:** We anticipated that the identifiability will decrease only slightly, as the model should be able to learn anatomically and physiologically derived motions apart from the task specifics. This hypothesis was guided in part by our investigation into H4, in which we found that our models performed well at identifying participants when they were performing the same task, suggesting the models had to learn features unique to individuals, rather than unique to the virtual environment [87].

*1.5.2.4 Methodology*

In Chapter 6, we make use of an ordered assembly task VR training scenario. The design of which allowed us to train users with two different building tasks using the same platform, models, and interaction cues, so we can investigate the identifiability between tasks.

## 1.6 Contributions

These research contributions fit into two main categories. We have determined a number of facets of predicting different learning outcomes from readily available VR tracking data. We have also investigated the identifiability of this tracking data across both time and task, through a lens of privacy, motivated by the potential usefulness of this data.

### *1.6.1 Predicting Learning Outcomes from Readily Available Tracking Data in VR*

1. We demonstrate that predicting Learning Gains from HMD and controller tracking data is feasible. (Ch. 3)
2. We show that it is also possible to predict knowledge retention as well as performance retention. (Ch. 4)
3. We determined that models trained with velocity-based feature representations outperformed those trained on positional data for knowledge acquisition, knowledge retention, and performance retention. (Ch. 4)
4. We found indication that tracking data can be better used to predict psychomotor-based outcomes than cognitive-based outcomes. (Ch. 4)

### *1.6.2 Identifiability of VR Tracking Data*

1. We show that identifying users through their tracking data with high degrees of accuracy within a VR training scenario is feasible. (Ch. 5)
2. We show that identifiability substantially decreases between sessions spaced one week apart, but still remain well above random. (Ch. 5)
3. We demonstrate that representing data as velocity rather than position further reduces identifiability. (Ch. 5)
4. We contribute deeper understanding to identifiability as it is affected by the task being performed by the user, and the data that the system has available. (Ch. 6)

## CHAPTER 2: LITERATURE REVIEW

### 2.1 Training Outcomes

Machine learning has been applied to predict knowledge acquisition in each of the three domains of learning: cognitive, psychomotor, and affective learning[110]. Much of this research has focused on creating an “online” classifier: one that can evaluate or predict students’ performance in real-time during teaching without the need for separate testing. In this section, we discuss prior work that has investigated the prediction of training outcomes, both by the their learning domain (Table 2.1) and their types of models and whether they’re predicting retention (Table 2.2)

#### 2.1.1 Predicting Cognitive Outcomes

There has been some work in the past on predicting cognitive learning outcomes in training and education environments, particularly in the realms of declarative and procedural knowledge. While

Table 2.1: An overview of prior work that have used machine learning to predict learning outcomes.

Paper	System	Context	Input Data	Predicted Outcome
<b>Chapter 3</b>	<b>Immersive tutor</b>	<b>Surgical robot troubleshooting</b>	<b>Head and hand tracking</b>	<b>Procedural knowledge (C)</b>
[2]	Desktop tutor	Math	Eye tracking, context events	Procedural knowledge (C)
[106]	Immersive tutor	Ear anatomy	Skeletal tracking, context events	Declarative knowledge (C)
[123]	Dyadic teaching	Environmental principles	Skeletal tracking	Declarative knowledge (C)
[109]	Surgical robot	Mastoidectomy	Force and instrument tracking	Fine motor skills (PM)
[26]	Surgical simulator	Bone marrow harvesting	Force and instrument tracking	Fine motor skills (PM)
[30]	Dance trainer	Forró partner dancing	Phone accelerometer, song BPM	Gross motor skills (PM)
[126]	Classroom	Synthesized	Skeletal, face, and eye tracking	Attention (A)
[117]	Classroom	Mechanical engineering	System events, eye tracking	Attention (A)
[29]	Desktop tutor	Computer literacy	Skeletal tracking, Dialogue	Engagement, confusion, frustration (A)
[90]	Desktop game	Constraint satisfaction	Posture tracking	Engagement (A)
[31]	Textbook	Biology	Transcript, audio	Engagement (A)

C = cognitive, PM = psychomotor, A = affective

the field of ITSs has many examples of predicting cognitive learning outcomes based on system and contextual events, we focus our discussion on research that makes use of physiological data for prediction.

The prediction of declarative knowledge gains based on physiological data has been an active area of research for some time. Schneider and Blikstein [106] developed an ear anatomy trainer with a tangible user interface and used ML methods to predict the learning gains of students. Based on a "rough" median split of their participants, they were able to achieve 100% accuracy with their SVM using a Multilayer Perceptron kernel [97]. Although they did not find a direct correlation between knowledge acquisition and posture, they determined that student posture had predictive value for classifiers by showing that it was a useful feature for their system. Won et al. [123] also researched prediction of declarative knowledge acquisition, specifically looking at the learning gains of environmental principals in a teacher-student dyad. They made use of skeletal data in the form of joint angles tracked from a Microsoft Kinect, a full-body motion sensing device that makes use of structured light and time-of-flight, to predict the learning gains. While the system showed little predictive power over all dyads, decision trees yielded a classification accuracy of 85.7% when comparing the top seven and bottom seven teacher-student dyads.

Beyond these approaches for predicting declarative knowledge gains, researchers have also investigated the prediction of procedural knowledge gains based on physiological data. Amershi and Conati [2] designed a desktop tutor that taught mathematics using a clustering approach that used distilled eye-tracking features, such as gaze shifts and interface events. This ML model was able to classify students based on high and low knowledge acquisition with an average accuracy of 86.3%. Recently, we also developed an ML approach that partitions users into high and low knowledge acquisition groups [86]. We made use of velocity data derived from HMD and controller tracking data to predict the learning gains from a robotic operating room training scenario based on a posttest knowledge test and found a mean accuracy of 93.1% among our examined approaches.

Unlike our recent work, in this paper, we investigate both position and velocity data, and we also investigate using ML models to predict retention outcomes, in addition to learning outcomes.

Amershi and Conati [2] showed that eye-tracking data can be used to predict procedural knowledge acquisition. Similarly, Won et al. [123] and Schneider et al. [106] found that body posture is useful in predicting declarative knowledge acquisition. These results clearly indicate that physiology can be used as an indicator of cognitive learning outcomes. To the best of our knowledge, our work was one of the first to successfully use HMD and controller tracking data to predict procedural knowledge acquisition.

The research studies above indicate that physiologically based data, such as body postures [106, 123], eye tracking [2], and body movements [86], can be used to predict cognitive learning outcomes, such as declarative and procedural knowledge acquisition.

### *2.1.2 Predicting Cognitive Retention Outcomes*

In addition to predicting the immediate learning outcomes, research has also been conducted to predict longer-term knowledge retention. Wang and Beck [121] examined using logistic regression to predict knowledge retention for a period of five to ten days after the student's last practice. Their approach makes use of features representing prior performance, when the student last practiced, and student response time. Work by Choffin et al. [20] has looked at modeling student learning and forgetting of skills by leveraging the Knowledge Tracing Machines framework [118] to embed features from multiple skills together. Their results suggest that incorporating both item-skill relationships and forgetting effects provides better learner models than only incorporating one or the other. Li et al. [66] found speed of mastery to be a relevant feature for predicting performance in week-later assessments within their Automatic Reassessment and Relearning System intelligent tutor.

By looking at the retention after a period of time, these works attempt to measure the "broader notion of knowing a skill" [121]. These works all found that the retention of knowledge appears to vary by skill and is possibly unique to each student [121, 20, 118] and generally requires additional considerations beyond prediction of knowledge acquisition.

### *2.1.3 Predicting Psychomotor Outcomes*

In addition to predicting cognitive learning outcomes, there has been research into predicting psychomotor learning outcomes from physiological data.

DeMoraes et al. [26] developed an online training assessment system for a bone marrow harvesting simulator that made use of a Gaussian Naïve-Bayes classifier. This system was able to match expert classification with a Kappa coefficient of 80% by utilizing position, velocity, forces, and time as features. Ultimately, their system made mistakes in only 20 of 150 cases. Similarly, Sewell et al. [109] performed classification in a mastoidectomy simulator via a Hidden Markov Model. This system used position, distance, tool force, and suction position features, and yielded 85% correct classification for both novices and experts. In more recent work that attempts to predict learning of gross motor movements, dos Santos [30] investigated rhythm skill prediction in a dancing trainer. This system made use of data from the student's phone and the song tempo in beats per minute (BPM) to classify their movements as faster, correct, slower, or mixed and was able to correctly identify 74% of sessions with an F1 score of 0.79.

The works by DeMoraes et al. [26] and Sewell et al. [109] indicate that motion data collected during training can be used to predict fine motor skill learning of complex tasks. Additionally, dos Santos [30] has shown that gross motor movement learning can also be predicted and used for feedback. These results indicate that psychomotor outcomes can be estimated by using positional tracking data.

#### 2.1.4 *Predicting Affective Outcomes*

Machine learning has also been employed for attention estimation in intelligent tutoring systems. Zaletelj et al. [126] have researched the use of skeletal, face, and eye tracking features through a Kinect to estimate attention in a classroom setting. Using bagged trees, they were able to achieve 86.9% prediction accuracy. Veliyath [117] also investigated classroom attention assessment using only eye-tracking, time, and whether the foreground application on the student's computer was relevant. Extreme Gradient Boost was able to predict whether a student was paying attention at 77% accuracy.

Other affective outcomes have been estimated as well. D'Mello and Graesser [29] found that dialogue and posture features could discriminate between states of boredom, confusion, engagement, and frustration in a desktop tutoring environment. Their system only averaged 50% accuracy for a four-way classification among their 6 classifiers, which is better than random chance, but not excellent. Another classifier that attempts to predict engagement was researched by Mota and Picard [90], and made use of posture detected by weight distribution. This Hidden Markov Model based system was able to discriminate between three states (high interest, low interest, and taking a break) with a reported average accuracy of 82.25%.

Drummond and Litman [31] investigated predicting zoning out through the use of transcript features (e.g., number of words read, disfluencies) and audio features (e.g., percent silence and minimum pitch). They found that a decision tree trained on the minimum pitch alone was able to predict with 64.3% accuracy high or low zoning out states.

The work by Zaletelj et al. [126] and Veliyath [117] indicate that machine learning can be used to estimate attention by using student physiology as input. Work by others indicates that body posture is a useful feature for classifying students into affective states, though as D'Mello and Graesser [29]



Table 2.2: An overview of related work that have predicted learning and retention outcomes.

<b>Prediction</b>	<b>Paper</b>	<b>System</b>	<b>Input</b>	<b>Model</b>
Cognitive learning	[2]	Math tutor	Eye tracking, Context events	K-means
	[123]	Classroom monitor	Skeletal tracking	Decision tree
	[106]	Anatomy tutor	Skeletal tracking, Context events	SVM
	[86]	Surgical simulation	Head tracking, Hand tracking	SVM
Psychomotor learning	[109]	Surgical simulation	Instrument tracking	HMM
	[26]	Surgical simulation	Instrument tracking	Naive Bayes
	[30]	Dance monitor	Phone accelerometer, Song BPM	Decision tree, kNN, Logistic regression Naive Bayes, Neural network, Random forest, SVM
Cognitive retention	[121]	Math tutor	System events	Logistic regression
	[66]	Math tutor	System events	Logistic regression
	[20]	Math tutor	System events	Logistic regression
<b>Cognitive learning, Cognitive retention, Psychomotor retention</b>	<b>Ours</b>	<b>Surgical simulation</b>	<b>Head tracking, Hand tracking</b>	<b>SVM</b>

indicate, categorizing into more than two subsets of affect can still present challenges compared to categorization of one affective state like Mota and Picard [90]. These works, along with that by Drummond and Litman [31] show that it is possible to automatically predict much about a learner’s affective state.

When this research was conducted, our work was the only one that made use of HMD and controller tracking data to predict learning outcomes and one of few works to predict procedural knowledge outcomes. Though other works do make use of skeletal tracking or posture tracking (see Table 2.1), these made use of additional tracking systems such as the Microsoft Kinect. While training systems may non-intrusively make use of such tracking, HMD and controller tracking data is readily available in consumer VR systems.

## 2.2 Identifiability and Obfuscation

The recent availability of larger datasets has been key for the improvement of machine learning techniques. Goodfellow et al. note that “the learning algorithms reaching human performance on complex tasks today are nearly identical to the learning algorithms that struggled to solve toy problems in the 1980’s... The most important new development is that today we can provide these algorithms with the resources they need to succeed.” [41]. VR offers unique access to a set of biometric data at high frequencies from users, due to the inherent need to track their movements to render the virtual world.

### *2.2.1 Machine Learning Biometrics for Authentication*

There has been much research into the field of behavioral biometrics to investigate means of improving authentication security while attempting to remain less intrusive than physiological biometrics. Work has been done to investigate passive authentication in the form of keystroke dynamics analysis while entering a password [101], continuous voice authentication for voice assistants [37], and sensor-based passive smartphone authentication [64]. Specific device contexts can allow access to biometric data that may normally be difficult to gather passively. Zhang et al. [127] recently investigated electrocardiogram-based authentication for healthcare systems. Eberz et al. [32] have looked at the ability to spoof user data for an electrocardiogram-based authentication device, and there has also been research into the ability to transform recorded heart data across contexts, increasing the ability to spoof electrocardiogram data [33].

### 2.2.2 *Machine Learning VR Tracking Data*

Similar to collecting electrocardiogram biometrics for authentication while a user is already wearing a sensor [33], VR provides a device context that allows software to incorporate several pieces of behavioral biometric data that is normally inaccessible. At a minimum, most VR hardware consists of a HMD with orientation tracking of the head, and common systems can incorporate positional and orientation tracking of the HMD and multiple controllers. Some VR systems, such as the HTC Vive Pro Eye, have embedded eye-trackers as well. The streams of data from these devices which are used for displaying and interacting with virtual environments can also be a source of rich biometric data, and there has been increasing research in using this data to identify and authenticate users.

#### 2.2.2.1 *VR Tracking-based Authentication*

Kupin et al.[60] investigated authentication through a controlled ball-throwing task by analyzing the trajectory of the position of the controller over the entire task. This trajectory was then mapped to existing trajectories with a symmetric sum-squared distance evaluation function to attain accuracy of about 90% among 14 participants. Ajit et al. [1] built upon this work by looking at new match metrics, including dynamic time warping, and using a perceptron to identify optimal match weights. They also made use of the position and orientation of both controllers and the HMD. Ajit et al. [1] were able to get accuracies around 93% for their 33 participants. Finally, Miller et al. [81] extended on the same authentication task by developing a system that sends matches to a perceptron to report confidence in identity, in order to create a system that should be able to detect when a different user has hijacked a VR session.

Similarly, Mustafa et al. [91] made use of feature vectors derived from time and frequency domain

derivatives of the  $x$ ,  $y$ ,  $z$ , and magnitude values from each accelerometer and gyroscope sensor, as well as features such as correlations across multiple streams to authenticate users. Mustafa et al. [91] compared the performance of Logistic Regression to SVM for their authentication system, and ultimately built a model capable of equal error rates among their 23 users as low as 7% in their application that consisted of free head movements in response to random events.

#### 2.2.2.2 *VR Tracking-based Identification*

Rogers et al. [103] found they were able to uniquely identify users from their involuntary blinks and head movements when presented with images of numbers and letters. They made use of 96 total features derived from blinking and head movements, registered from the infrared (IR) eye tracker, gyroscope, and accelerometer of their device. After an enrollment phase consisting of showing the user a sequence of images, they were able to identify users with an RF model, getting a balanced accuracy of about 94% for their 20 users.

Pfeuffer et al. [99] recently conducted a study in which 22 participants performed controlled VR tasks over two sessions and, by making use of head, hand, and eye-tracking data, were able to identify participants at rates of about 40%. They explored 4 generic tasks (pointing, grabbing, walking, and typing) in a controlled setting and compared several feature representations, including tracker positions, distance between trackers, and distances between trackers and targets, over individual tasks. By comparing the classification accuracy between different tasks and feature sets, they found which features were optimal for each of their tasks. Pfeuffer et al.'s [99] maximum accuracy of about 64% was in pointing tasks with features based on the maximum dominant hand angular velocity, the average distance between the two hands, as well as six other features.

Similarly, work by Miller et al. [80] has shown positive identification rates above 95% when identifying among 511 participants viewing 360° videos and answering questionnaires in a stationary

position. They made use of a position and orientation feature vector derived from each second of data. Unlike Pfeuffer et al.'s [99] work, Miller et al. [80] explored identification within a single session only. Our current work is based on the work of Miller et al. [80].

Most recently, Liebers et al. [67] explored user identification with body normalization. In their work, they developed two controlled identification tasks (bowling and archery) and employed featurization derived from the position and orientation of the HMD and controllers, fed to a Recurrent Neural Network consisting of three Long-Short Term Memory layers, and a multilayer perceptron. Their highest overall accuracy was 90% for archery and 68% for bowling among their 16 participants.

### *2.2.3 Machine Learning Velocity for User Experiences*

In our current work, we investigate using velocity to reduce identifiability. This is motivated by recent work that finds utility in applying ML to velocity data for predicting user experience outcomes. Padmanaban et al. [96] used velocity-based video features for predicting simulator sickness in 360° stereoscopic videos with relatively low root mean squares. Similarly, Moore et al. [86] made use of velocity components of the HMD and controllers in a VR training environment to predict procedural knowledge gain with accuracies up to 93%. Finally, David-John et al. [25] were able to successfully predict the onset of 3D interactions with gaze dynamics, achieving a maximum area under the Receiver-Operator Curve of 0.92. These works made successful use of velocity-based data representations rather than position data for predicting user experience outcomes. Hence, we decided to investigate velocity-based input features in our current work.

## 2.3 Identifiability and Authentication Research by Inclusion of XR Tracking Features

In this section, we examine some works organized by their inclusion of different features, and the comparisons in performance that they make by models trained on those different feature inclusion conditions.

### *2.3.1 Identification and Authentication with Eye-tracking*

Several commercially available HMDs for augmented or virtual reality afford eye-tracking. Often the hardware necessary for this is included as a means of collecting user input or improving the hardware performance through techniques like foveated rendering. As this stream of data has become more available, several researchers have begun exploring its usefulness as a means to identify or authenticate users.

In recent work by David-John et al. [24], the authors recognize that while eye-tracking data can be useful as a means of input, it also runs a heightened risk of allowing systems identify users, for example, by correlating a users' data across multiple accounts. They propose a privacy-preserving means of streaming eye-tracking data by gatekeeping at the API level. Their approach was capable of reducing identifiability from 85% to approximately 30% while preserving a system's ability to use gaze data for input like foveated rendering.

In another recent work by Liebers et al [68], the authors make use of eye-tracking features and HMD orientation and provided participants with a stimulus designed to elicit smooth pursuit head and eye movements. These movements were tracked by logging the reported HMD Euler angles, as well as the pupil position. After preprocessing to determine additional eye-tracking events such as saccades and pursuits, The authors then investigate both kNN as well as a set of 10 Deep Learning Neural Network approaches. Ultimately the authors found that inclusion of HMD-based

Table 2.3: A comparisons of the input features used by related works. Each letter under a given paper indicates a condition that was evaluated in that paper consisting of the data for each row in which the letter appears. The chart also shows what kinds of classifiers each work explored as well as the number of participants, and the highest accuracy attained by their best-performing classifier among the explored representations.

		[24]	[68]	[115]	[94]	[3]	[91]	[80]	[105]	[67]	[87]	[99]	Ours (Pos)	Ours (Vel)	
Input Features	Eye	2D Pos	A	A		A									
		3D Dir					A								
		Others	A		A		A								
	Head	Position			AB	A			A	A	A	A	AEP	AGJK PQSU	
		Velocity										B	BEP	AGJK PQSU	
		Distance											CEP		
		Rotation		A	AB	A		A	A	AB	A	A	DEP	BGMN PQTU	
		Ang Vel								C		B		BGMN PQTU	
		Dominant Hand	Position			A	A			A	AB	AB	A	FJP	CHJL PRSU
	Velocity									C		B	GJP	CHJL PRSU	
	Distance												HJP		
	Rotation				A	A			A	AB	AB	A	IJP	DHMO PRTU	
	Ang Vel									C		B		DHMO PRTU	
	Off Hand	Position			A	A			A	AB	AB	A	KOP	EIKL QRSU	
		Velocity								C		B	LOP	EIKL QRSU	
		Distance											MOP		
		Rotation			A	A			A	AB	AB	A	NOP	FINO QRTU	
		Ang Vel								C		B		FINO QRTU	
	Classifiers		RBF	DNN kNN	LogR DT, RF	kNN	CNN kNN LSTM RF	LogR SVM	GBM kNN RF	RF, MLP FRNN LSTM GRU	MLP RNN	GBM kNN RF	RF SVM	GBM kNN RF	GBM kNN RF
	Participants		18	12	35	15	34	23	511	34	16	60	22	45	45
Highest Accuracy		85%	100%	96%	98%	98%	93%	95%	100%	90%	90%	44%	96%	68%	

data increased accuracy of their ML model from 45% to 90%, as well as their best-performing deep learning model from 96% to 100%.

Expanding on the number of tasks examined, Tricomi et al. [115] present a pre-print investigating the identifiability of participants in both VR and AR tasks. The participants were exposed to 5

types of tasks in the AR condition and 7 types of tasks in the VR condition. They make use of automated systems to determine salient features of the raw data, then use machine learning to attempt to identify and profile their participants. The authors found 96% identifiability accuracy.

Olade et al. [94] also examined similar body and eye tracking features, but for both continuous identification as well as authentication. They used a data set of 15 participants, and investigated various attack types could one could be conducted and what the risk was among their participants. Ultimately, their accuracy was 98.6%.

Finally, one more paper that investigates identifiability with the inclusion of eye-tracking data by Asish et al., [3] focused exclusively on eye-tracking data. This paper has VR session divided into 4 separate experiences and examines identifiability across those sessions. The authors also train their models on 3 of the 4 experienced sessions and find 98% overall accuracy with their deep learning models.

### *2.3.2 Identification and Authentication without Eye-tracking*

While eye-tracking has been incorporated in multiple HMDs, many headsets do not support it as a form of input. Several works have investigated identifiability or authentication by making use only of the sensors that provide the tracking needed for a virtual reality experience. The benefit to restricting oneself to this set of data is that the results are applicable to a broader range of hardware than only those that afford eye-tracking.

Mustafa et al. [91] explore authentication in a VR experience making use of only features derived from the orientation of the HMD. In their work, they generated authentication models that gave equal-error rates around 7%, showing feasibility in such an approach for authenticating in a scenario with 23 users. One caveat pointed out by the authors is that their results were task-specific



due to encoding features of the virtual environment, and so authentication in another environment would require the creation of new models which may perform differently.

In recent work by Miller et al. [80], the authors presented a study involving 511 participants with a virtual environment that displayed 360° video clips with questionnaires presented between. They made use of only the positions and orientations of the controllers and head-mounted display for passive identification of users. They conducted an ablation study examining the removal of subcomponents of tracking data and found that removal of the HMD Y value (which is strongly correlated with height) resulted in the largest drop in identification accuracy.

Schell et al. [105] also investigate identifiability on motion from a restricted set of data points. In this pre-print, the authors investigated identifiability by using the open Talking with Hands dataset [63]. Schell et al. examined creating head-relative values for the hand position and orientations, as well as their time-derivatives from this conversational dataset. This subset of the data was chosen as it corresponds to the tracking data that a typical room-scale system affords. They found that with majority voting on increasingly long test sequences, they were ultimately able to attain 100% accuracy with several of their explored models. While the data this analysis is conducted on is from in-person human-human conversational dyads outside of VR, the authors suggest that their results further contribute to the body of research showing potential in the use of this data for identifying individuals in VR experiences.

In another recent work by Liebers et al. [67], the authors investigate the identifiability of users performing two different tasks in VR. In their study, they had 16 participants perform repetitions of prescribed tasks over two days. By training their models on data from a single day's session and evaluating on a different day, they ensured that their models weren't encoding data that may be session-specific. They found that for their, they were able to attain accuracy up to 90% with a motion mapped to a normalized human body model with a Recursive Neural Network based on

LSTMs and a Multilayer Perceptron.

Another work that investigated identifiability by Moore et al. [87] examined the passive identification of users across a week delay between sessions in an interactive training environment. They found that the identification accuracy across sessions was greatly reduced from around 90% to near 32%. The authors hypothesize a few reasons for this finding including variability in the presentation of the VR experience, the potential for the user to be in a substantially different physical and mental state, and wearing different clothing. The authors additionally examined using velocity-based features and found that those further reduced identification accuracy.

Finally, Pfeuffer et al. [99] also investigate identification between sessions with a minimum 3-day period between exposure. They examined identification making use of features derived from the head, hand, eye, among 22 participants. They also looked at 4 different types of simple interactions to identify users with. With multiple samples of each atomic interaction, they were able to achieve 44% identification accuracy between sessions.

# **CHAPTER 3: EXTRACTING VELOCITY-BASED USER-TRACKING FEATURES TO PREDICT LEARNING GAINS IN A VIRTUAL REALITY TRAINING APPLICATION**

*NOTE:* This chapter is a modified format version of the paper previously published.

Material from: A. G. Moore, R. P. McMahan, H. Dong, and N. Ruozzi. Extracting velocity-based user-tracking features to predict learning gains in a virtual reality training application. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 881–890, Los Alamitos, CA, USA, nov 2020. IEEE Computer Society. doi: 10.1109/ISMAR50242.2020.00099. URL <https://doi.ieeecomputersociety.org/10.1109/ISMAR50242.2020.00099>

### 3.1 Introduction

Virtual Reality (VR) has a long history of being used for training and educational purposes [10]. It has been used for *cognitive*-based learning (i.e., knowledge acquisition [57]), such as learning about World War I [125], spatial understanding of geometry [61], and haul truck inspection procedures [75]. For example, it has been used to learn how to use calipers and micrometers [7], how to don personal protective equipment [36], and how to perform prescribed visual scanning strategies [100]. Finally, VR has also been used for education in the *affective* domain (e.g., attitudes, emotions, and feelings [56]), such as interpersonal skills [53] and therapy [102]. With the introduction of consumer VR systems, these learning-focused uses of VR have gained more attention recently [16].

Closely related to VR training and education is the Intelligent Tutoring Systems (ITSs) field. The goal of an ITS is to engage learners in a sustained learning activity and to interact with each learner based on a deep understanding of that learner's behaviors [22]. To understand the learner's behaviors, ITSs incorporate techniques from the Artificial Intelligence (AI) community to provide tutoring systems that know “*what* they teach, *who* they teach, and *how* to teach it” through user models [93]. These user models enable ITSs to provide adaptive support for learners by using an abstract representation of the learner in terms of relevant traits, such as knowledge, meta-cognitive ability, and learning behaviors [2]. Two forms of adaptivity for ITSs include *scaffolding* (i.e., proactively providing guidance) and *feedback* (i.e., reactively providing guidance) [116]. However, feedback is the more common approach to creating an adaptive learning environment [116], and most feedback approaches use simple and static rules that employ particular feedback for specific types of detected errors [42].

In this chapter, we investigate the feasibility of using a machine-learning approach to classify users of a VR training system into groups of low-learning (LL) and high-learning (HL) gains,

based on their head-mounted display (HMD) and controller tracking data. By using tracking data, as opposed to events or errors [42], our approach affords new possibilities for real-time adaptivity, before events or errors even occur. Unlike most prior approaches that are only capable of feedback [116], this tracking-based, machine-learning approach can potentially be used for both feedback and scaffolding, such as guiding the user’s attention [23], providing interaction cues [50], and simplifying the interactions [74]. As discussed in Section 3.2, our machine-learning work is the first to use HMD and controller tracking data to predict cognitive learning gains.

After discussing related work, we present a user study, in which participants used a VR training simulation to learn the procedure for troubleshooting a surgical robot and then a knowledge test was administered to assess their cognitive learning gains. We used the results of the knowledge test to separate participants into two groups (LL and HL), with learning gains one standard deviation or above the mean being classified as HL. We then present the results of a machine-learning experiment that employed support-vector machines (SVMs) [9] to predict which group (LL or HL) each participant belonged to, based on their tracking data. For the experiment, we compared the accuracy (i.e., correctness of the prediction) and confidence of the models learned using three different sets of input features and two feature representations. For the input features, we investigated: a) linear and angular velocity vectors of the HMD and controllers (HLA+CLA), b) linear and angular velocities of the HMD and linear velocities of the controllers (HLA+CL), and c) linear velocities of the HMD and controllers (HL+CL). For the feature representations, we investigated: a) Principal Component Analysis (PCA) [119] and b) convex matrix factorizations (CVX) [8].

The results of our machine-learning experiment indicate that the velocities of the HMD and controllers yielded high mean accuracies for both the training data (85.7%) and the testing data (93.1%). There were no major differences among the input features and feature representations on the testing data. However, we did find that the combination of HLA+CL input features and the CVX feature representation yielded 100% accuracy for the testing data.

In order to understand how the SVMs classified participants into LL and HL groups with such high accuracies, we have visually inspected segments of tracking data that were correctly identified by the SVMs. Novel visualizations of the segments show that participants with HL gains moved with better economies of motion than those participants with LL gains. These results indicate that it is feasible to use the velocities of the HMD and controllers to develop new real-time adaptivity techniques for improved scaffolding and feedback in VR training systems.

We anticipate that our work will be useful in the development of systems that predict learning gains, and subsequently respond dynamically when users may not be learning as well as possible. Such systems may make use of real-time classifiers to provide additional scaffolding and feedback to better support the end-user's learning. Through this work, we hope to begin to answer RQ1: Can readily available VR tracking data be used to predict cognitive outcomes from a VR training scenario?

We present the following contributions in this chapter:

- We collected HMD and controller tracking data from 61 participants using a VR training application for troubleshooting surgical robots and measured their procedural knowledge acquisition using a posttest.
- We conducted a machine learning experiment investigating three sets of velocity-based input features (HLA+CLA, HLA+CL, HL+CL) and two feature representations (PCA, CVX).
- Our results indicate that it is feasible to predict LL and HL gains with high accuracies and confidences using our investigated machine learning methods.
- We found that participants with LL gains moved less smoothly but also more slowly than participants with HL gains, using our novel visual inspection technique.

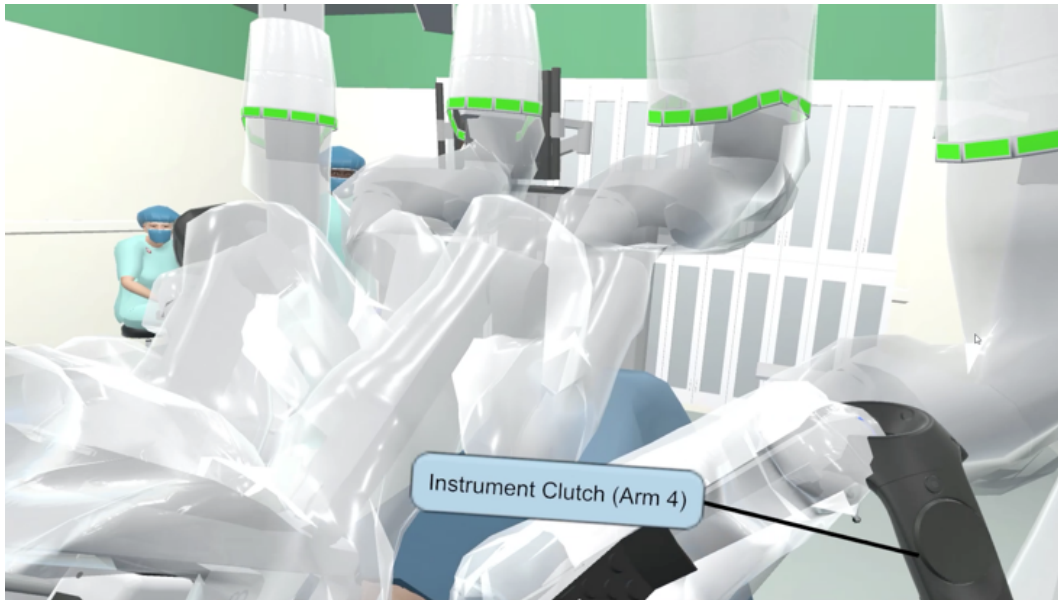


Figure 3.1: The VR training simulation used in our user study.

### 3.2 VR Training User Study

We used an existing VR application that was designed to train first assists who work with surgical robots [19]. The training scenario involves interacting with a virtual surgeon and non-sterile staff member to troubleshoot a surgical robot with a faulted instrument arm. In brief, the troubleshooting procedure involves attempting to restart the system, using a tool to open the instrument forceps that are stuck grasping patient tissue, removing the instrument, stowing the arm, and disabling it. To accomplish these tasks, a trainee must complete several steps that require various actions, including moving around the virtual operating room (OR), looking at the endoscopic monitor, selecting dialogue options to converse with the virtual agents, and manipulating objects, such as the instrument clutch (see Figure 3.1).

### 3.2.1 *Materials*

We used an HTC Vive system, including the head-mounted display (HMD) and both handheld controllers, to run the VR training application. The Vive HMD provided a 110° diagonal field of view with a display resolution of 1080 x 1200 pixels per eye and a 90Hz refresh rate. The HMD was retrofitted with a Vive audio strap. The VR training application was developed using Unity and maintained framerates of 90 frames per second to match the Vive’s refresh rate. The SteamVR plugin for Unity was used to process the Vive’s input data.

During the procedure, the VR training application collected head and hand tracking data every frame. Each data point consisted of the frame’s timestamp, the global positions and rotations of the HMD and handheld controllers, and the status of each controller’s trigger button, which was the only button used for interactions. We are using three attributes (i.e., a 3-vector) to describe each global position and four attributes (i.e., a quaternion) to describe each global rotation.

### 3.2.2 *Procedure*

The following procedure was reviewed and approved by the University Institutional Review Board (IRB).

After informed consent, the session began with a background survey on the participant’s demographics, education, and technology experience. The experimenter would then help the participant put on the HTC Vive and run the SteamVR tutorial to train the participant how to use the Vive. The participant would then experience the VR training application. After completing the training session, the participant was administered the Simulator Sickness Questionnaire (SSQ) [55], the Spatial Presence Experience Scale (SPES) [44], the System Usability Scale (SUS) [14], and questionnaires regarding scenario fidelity [73] and body ownership aspects [72] of the VR training



experience. Finally, participants were administered a knowledge test consisting of 20 multiple-choice questions pertaining to the training scenario. The session lasted approximately 60 minutes.

### *3.2.3 Participants*

A total of 61 participants (11 females, 50 males) were recruited through university mailing lists and completed our study. As an exclusion criterion, none of the participants had prior experience or knowledge of surgical robots. The mean age of the participants was  $22.7 \pm 4.23$  years old. Based on self-reported background data, 18 participants regularly played video games five or more hours each week and 32 participants have prior experience with consumer VR HMDs.

### *3.2.4 Descriptive Statistics*

The mean time required for the participants to complete the VR training application was 12.45 minutes ( $\sigma = 3.29$  minutes). The mean total simulator sickness score was 21.09 ( $\sigma = 18.42$ ), which is similar to or better than recently reported SSQ results for consumer VR HMDs [122, 114, 69]. The mean presence score was 4.01 ( $\sigma = 0.70$ ), which is similar to recent SPES results that also used the HTC Vive [34]. The mean perceived usability score was 73.81 ( $\sigma = 13.26$ ), which is greater than the average SUS score of 68 [14]. The mean scenario fidelity score was 3.88 ( $\sigma = 0.64$ ) out of 5.00, and the mean body ownership score was 3.65 ( $\sigma = 0.79$ ) out of 5.00. Generally, it appears the scenario fidelity and body ownership aspects of the VR training simulation were acceptable.

Finally, the mean knowledge test score was 8.38 ( $\sigma = 3.07$ ) out of 20.00. While this appears to be low, it is important to keep in mind that the participants have no prior experience or knowledge of surgical robots. Additionally, the participants experienced the troubleshooting procedure only once. Finally, recent research indicates that people can generally only remember  $4 \pm 1$  items at a

time [54]. Considering all of this, a mean score of 8.38 out of 20 questions seems acceptable. At the same time, it also clearly indicates that there is room to improve the VR training simulation with real-time adaptivity techniques.

### 3.3 Machine Learning Experiment

In this section, our aim is to fit machine learning models to the collected VR data. In particular, we are interested in answering the following question: *Can we predict whether or not a trainee is experiencing high learning gains or low learning gains using only the head and hand tracking data?*

As we are operating in a data limited setting, care must be taken when fitting ML models: naïve applications of even simple ML models can easily overfit the data and result in poor predictive performance. Consequently, we need to proceed with caution when processing the data, extracting features, and fitting the ML model in order to minimize overfitting.

In the following sections, we will describe how we converted the collected data into a feature representation that is amenable to standard ML methods, explain how the data was divided into subsets for training and testing, specify how the model was fit to the training data, and show how the learned model can be used to answer the above question. All code for these procedures was written in Python using scikit-learn to fit the machine learning models. The featurized version of the data set and the Python code for fitting the models will be made publicly available on GitHub<sup>1</sup>.

---

<sup>1</sup>[GitHub.com/LeonDong1993/learning-gain-prediction](https://github.com/LeonDong1993/learning-gain-prediction)

### 3.3.1 Data Pre-processing

Below we discuss how we organized participants into groups of low- and high-learning gains. We also describe how we segmented the participant data into two data sets: one for training our machine-learning models and one for testing our trained models. We then discuss how we pre-processed the tracking data and ultimately converted it to the linear and angular velocities of the head and hand tracking data. Finally, we discuss how we pre-processed the tracking data and ultimately investigated the linear and angular velocities of the head and hand tracking data.

#### 3.3.1.1 Low- and High-Learning Gains

In order to label participant data as **low-learning (LL)** or **high-learning (HL)** gains, we first fit a Gaussian distribution to the observed knowledge test scores, creating a normal distribution with a mean score of 8.38 ( $\sigma = 3.07$ ). We then classified all test scores one standard deviation above the mean, i.e.,  $\geq 12$ , as HL gains. All other test scores and participants were classified as LL gains. This resulted in 51 LL participants and 10 HL participants.

#### 3.3.1.2 Training and Testing Data Sets

We divided our complete data set into a training data set and a testing data set: the training data is used to build the ML models while the testing data is used to evaluate the performance of the learned model on novel data observations. For both groups, we used an 80:20 ratio for dividing the data for training and testing, which is based on the Pareto Principle [111]. That is, we randomly selected 80% of the participants from each class for the training data set and used the remaining 20% for the testing data set. This resulted in a train-test split in which the training set contained 41 LL participants and 8 HL participants and a testing data set with 10 LL participants and 2 HL

participants.

### 3.3.1.3 *Velocity-Based Input Features*

As described in Section 3.2.1, the raw input data from the VR training application included the global positions and rotations of the HMD and handheld controllers, and the status of each controller’s trigger button, per timestamped frame. However, there were several issues with the raw data upon inspection. First, the relationship between the button and tracking inputs were highly correlated with the tasks of the VR training application. For example, during the task of manipulating the instrument clutch (see Figure 3.1), button inputs occurred approximately in the same tracked positions due to the position of the instrument clutch before the manipulation began. For the same reason, the positions and rotations of the HMD and handheld controllers were also highly correlated with task execution. Additionally, the heights (i.e., z-axis values in Unity) of the positions were highly dependent upon the heights of the participants. Including such information can cause machine learning models to overfit the training data set, e.g., by learning specific tasks and participants, which often results in poor testing performance [58].

To address the above issues, we converted the raw positions and rotations to linear and angular velocities. First, we sampled every 15 frames of the original data, which resulted in approximately 6 frames of data per second. We did this to reduce the effects of tracking jitter [73]. We then computed 3-vector linear velocities for each new frame of data by dividing the differences in the 3-vector positions, compared to the previous frame, by the time difference between the two frame’s timestamps. We computed 4-vector angular velocities from the 4-vector rotations in a similar manner, except we first converted the quaternion representations to angle-axis representations to calculate the differences and then back to quaternion representations. In summary, these conversions yielded  $1/15^{th}$  of the original frames, with 3-vector linear velocities and 4-vector angular

velocities for the HMD and both handheld controllers each frame.

Note, as part of our machine learning experiment, we investigated three sets of input features: a) the **HMD linear and angular velocity and the controller linear and angular velocities (HLA+CLA)**, b) the **HMD linear and angular velocity and the controller linear velocities (HLA+CL)**, and c) the **HMD linear velocity and the controller linear velocities (HL+CL)**.

#### *3.3.1.4 Data Segments*

The velocity sequence data for each user was divided into multiple short velocity segments of the same fixed length using the sliding window method. Note that we allow those velocity segments overlap with each other (this is particularly important for feature selection with CVX as we do not know in advance which segments correspond to important patterns in the data). In other words, when creating the next segment, the amount we shift the sliding window right is less than the size of segment length (i.e., the size of sliding window) itself. The shift size and the length of the velocity segments are considered as hyperparameters of our model. The ML process will automatically select the best values among the given candidates based on predictive performance as part of the cross-validation procedure. More details about the hyperparameter selection process and the range of values we considered for each hyperparameter can be found in Section 3.3.2.1 and 3.3.3, respectively.

#### *3.3.1.5 Feature Representations*

The data obtained after pre-processing could be directly plugged into a learning procedure. For example, we could construct a data vector by flattening the velocity segment in a row order. As the data vectors produced this way are usually high dimensional and quite noisy, we expect poor

performance from the learned classifier due to our limited number of participants, particularly our limited number of HL gains. Instead, we consider two feature extraction procedures to produce feature input vectors for our machine-learning models: a) Principal Component Analysis and b) convex-combination representations.

**Principal Component Analysis (PCA)** aims to explain the data as *a linear combination of input features*. This is done by finding the eigenvectors and corresponding eigenvalues of the data covariance matrix, where each eigenvector is considered as one component and the corresponding eigenvalues tell us how much of the variance is explained by that eigenvector. We can construct a lower dimensional feature representation by projecting onto the eigenvectors of the top  $k$  most informative components.

As the features produced by PCA are linear combinations of the input data, and hence difficult to interpret, we were motivated to investigate an alternative robust feature selection strategy based on non-negative matrix factorization that aims to provide interpretable feature representations. The motivation for interpretable features is our belief that certain patterns of velocities are more typical of low- or high-learning gains. As a result, it makes sense to build feature vectors that are based on canonical velocity segments.

Specifically, we consider the **convex-combination (CVX)** feature extraction technique, which aims to explain the data as *convex combinations of a subset of the input features* [8]. In this approach, each data vector is considered as a point in a high-dimensional space, and the objective of the CVX algorithm is to find  $k$  points whose convex hull well-approximates the convex hull of the entire data set. The  $k$  selected points are necessarily extreme points of the data set and can be interpreted as those data points that contain patterns whose convex combinations well explain the remaining data set. Note that every data point inside the convex hull can be exactly represented as the convex combination of these  $k$  data points. For data points that are not inside the convex

hull (often a small number in practice), we compute an approximation by projecting them onto the convex hull. Similar to PCA, the coefficients, obtained by projection onto the hull, that describe the approximation form the new feature vector. Note that the sum of the nonnegative coefficients is guaranteed to be one, where the value of each coefficient tells us to what extent the data point is similar to the corresponding selected point.

### 3.3.2 Model Setup

We fit support vector machines (SVMs) [9] with Gaussian kernels to the featurized data. The resulting linear separator will yield a classification of LL/HL for every velocity segment. One possible way to extend the segment classification to an entire user is to determine the user’s label by majority vote over the predicted labels of the user’s velocity segments. This turns out to be a bad idea in our case. To see why, observe that approximately 20% to 30% of the velocity segments are common between LL and HL. In other words, these velocity segments are not informative and the classifier should yield a low confidence, i.e., essentially random, prediction for each of them. As a result, the user label will likely be skewed by these segments if we do not take the confidence of the segment-level classifier into consideration.

For example, consider an HL user that has 5 velocity segments whose predicted labels and corresponding confidence scores are given as follows:  $(1, 2.4), (0, 1.7), (0, 0.2), (1, 2.7), (0, 1.3)$ , where 0/1 stands for LL/HL. Here, the third velocity segment is an uninformative segment where the classifier predicts the label as LL. If we determine the label by majority vote, we would incorrectly predict LL. However, if we take the confidence of the classifier into account, we see that the classifier is more confidently predicting an HL label. As a result, although only 2 out of 5 velocity segments are predicted as HL, the overall confidence in HL is  $\frac{2.4+2.7}{2.4+2.7+1.3+1.7+.2} = 61.4\%$ , which gives the correct prediction result.

To overcome the limitations of majority vote, we propose to extend the classifier over single velocity segments to a classifier over users as follows. First, recall that, for the SVM classifier, the prediction is generated by a decision function  $f$ . For a given input (feature vector)  $v$ , the sign of  $f(v)$  indicates the class (LL/HL) of  $v$  while the absolute value  $|f(v)|$  is proportional to distance to the decision boundary. We say that the classifier is more confident for points further from the boundary between LL and HL as small changes in data points that are far from the decision boundary are very unlikely to change the prediction.

Let  $l_{u,i}$  and  $d_{u,i}$  denote, respectively, the predicted label and corresponding distance for the  $i^{\text{th}}$  velocity segment of user  $u$ . We define the user level confidence that  $u$  belongs to class  $c$  as

$$\text{conf}_{u,c} = \frac{\sum_i \text{st. } l_{u,i}=c d_{u,i}}{\sum_{i=1}^N d_{u,i}},$$

where  $N$  is the number of velocity segments that belong to user  $u$ . Then, the predicted label of the user is determined by selecting the class  $c$  that maximizes the confidence.

### 3.3.2.1 Hyperparameter Selection

There are five hyperparameters that need to be selected during the training process: the length of the velocity segments, the shift size of sliding window, the number of approximate hull points (CVX) / the number of selected eigenvectors (PCA)  $k$ , a parameter  $\gamma$  that controls the expressivity of the Gaussian kernel, and the slack penalty coefficient  $C$  that controls the degree to which misclassification of the velocity segments is penalized. The larger the value of  $C$ , the more the important it is to classify all velocity segments in the training split perfectly.

In our experiments, we used cross-validation at the user level to jointly tune those parameters. Unlike normal fold-based cross-validation, we do not split our data into equal sized folds and train



Table 3.1: Value range of hyperparameters considered.

HyperParameter	Seg. Length	Shift Size	Components	Penalty	Expressivity
Minimum Value	30	5	10	1	0.001
Maximum Value	120	13	80	10000	0.09

a classifier on all folds except one, which is left out for evaluating model fit. Instead, we divide all users in the train split into two groups such that the larger group accounts for 75% of all users. We train an SVM classifier under each hyperparameter configuration using all velocity segments belonging to users in the larger group, and we evaluate the performance of the trained classifier on all velocity segments belonging to users in the smaller group. After that, we redivide the users in a different way and conduct the above evaluation again. This process is repeated eight times and the hyperparameter configuration that achieves the highest average evaluation score is selected as the best hyperparameter.

Finally, the model fit with each configuration of the hyperparameters is evaluated using a scoring function. A common approach would be to select the classifier with the highest F1-score, which is the harmonic mean of the precision and recall of the classifier’s prediction results. This may not be ideal in our setup as the F1-score only measures the segment level performance while what we really cares about is the user level performance. In other words, we should favor a classifier that can distinguish LL and HL gains with high accuracy as well as high confidence.

Based on the same idea of balancing precision and recall used by the F1-score, we propose the following *user level* scoring function. First, define  $P_L \equiv LLAcc \times LLConf$  and  $P_H \equiv HLAcc \times LLConf$  as the performance measure of the trained classifier for the LL and HL classes respectively. Our scoring function is then the harmonic mean of these quantities.

$$score = \frac{2P_L P_H}{P_L + P_H}.$$

Table 3.2: Best hyperparameter for each model.

Attributes	Feature	Seg. Length	Shift Size	Components (k)	Penalty C	Expressivity $\gamma$
HMD linear and angular velocity	PCA	105	7	10	10	0.001
Controller linear and angular velocity	CVX	90	5	10	1000	0.001
HMD linear and angular velocity	PCA	120	7	30	10	0.001
Controller linear velocity	CVX	105	7	20	10000	0.05
HMD linear velocity	PCA	90	7	80	1	0.003
Controller linear velocity	CVX	120	7	50	10000	0.07

### 3.3.2.2 Evaluation Metrics

Once we select the best hyperparameters via cross-validation, we train an SVM classifier with a Gaussian kernel over the entire training split using these hyperparameters. The resulting model is then evaluated on both the training and test splits. In the following experiments, we assess the performance of our learned models with respect to three metrics: (1) the accuracy over velocity segments  $SegAcc$ , which is the percentage of velocity segments that are correctly classified; (2) the accuracy over users for both classes  $LLAcc$  and  $HLLAcc$ , which is the percentage of users that are correctly classified for their respective labels. (3) the average confidence of the user level prediction for both classes  $LLConf$  and  $HLLConf$ . This is calculated by taking the mean of the user level prediction confidence. Note that we only consider the users that are correctly classified, thus the average confidence should always be greater than 0.5 since we have only two classes. We will report all evaluation metrics for both the train and test splits.

### 3.3.3 Results

We consider two different feature extraction algorithms as well as three different sets of input attributes. The range of values we considered for each hyperparameter are shown in Table 3.1 and the corresponding best hyperparameters found via cross-validation are shown in Table 3.2. The

Table 3.3: Experimental results on *Train* split using different attributes combination and feature extraction technique.

Attributes	Feature	SegAcc	LL Acc	LL Conf	HL Acc	HL Conf
HMD linear and angular velocity	PCA	0.737	32/41	0.936	5/8	0.905
Controller linear and angular velocity	CVX	0.743	33/41	0.934	5/8	0.858
HMD linear and angular velocity	PCA	0.731	30/41	0.903	6/8	0.788
Controller linear velocity	CVX	0.794	41/41	0.883	6/8	0.822
HMD linear velocity	PCA	0.771	41/41	0.865	5/8	0.785
Controller linear velocity	CVX	0.865	41/41	0.939	7/8	0.867

Table 3.4: Experimental results on *Test* split using different attributes combination and feature extraction technique.

Attributes	Feature	SegAcc	LL Acc	LL Conf	HL Acc	HL Conf
HMD linear and angular velocity	PCA	0.842	9/10	0.92	2/2	0.838
Controller linear and angular velocity	CVX	0.842	9/10	0.939	2/2	0.776
HMD linear and angular velocity	PCA	0.818	9/10	0.904	2/2	0.956
Controller linear velocity	CVX	0.827	10/10	0.889	2/2	0.688
HMD linear velocity	PCA	0.798	9/10	0.872	2/2	0.856
Controller linear velocity	CVX	0.781	9/10	0.904	2/2	0.871

performance of the models trained using the best hyperparameters for the train and test split are shown in Table 3.3 and Table 3.4, respectively.

### 3.3.3.1 Comparison of Feature Representations

In this section, we compare the performance of the feature selection strategies PCA and CVX. We make the following observations from the results in Tables 3.3 and 3.4. (1) The model trained using the CVX feature representation achieves higher accuracy on the training split in all cases and comparable levels of accuracy on the test split compared to the models fit using the PCA feature selection technique. In addition, the models trained with the CVX feature representation always

correctly identify at least as many LL and HL gain users compared to model trained with the PCA feature representation on both the training and test splits. (2) The CVX feature representation tends to use fewer components, the hyperparameter  $k$ , to explain the data as compared with PCA as shown in Table 3.2. This indicates that CVX feature extraction results in a more compact and robust feature representation, which usually leads to better generalization performance, i.e., performance on held-out data. (3) The models fit with the PCA feature representation tend to have a smaller “Expressivity,”  $\gamma$ , and slack penalty,  $C$ , hyperparameters. This suggests that the best model under the PCA features is not close to a perfect classifier on the training data. The larger slack penalty for the optimal models under CVX features suggests that misclassification is highly penalized in the optimal models, but the hyperparameter is slightly larger suggesting that the representation in the kernel space is slightly more complicated.

The above observations suggest that the feature representation produced by CVX may better capture the important patterns inside the data compared to the features extracted by PCA. Recall that the CVX feature representation also provides interpretability: the selected components correspond to real motion segments. Still, both methods perform quite in this limited data setting.

### 3.3.3.2 *Noisy Data and Head Orientation*

We also investigated the effect that the angular velocity attributes had on the quality of the learned classifier. Our initial experiments suggested that the head orientation data was quite noisy. This observation also motivated us to investigate whether or not head orientation contributes significantly to the predictive performance of the ML model. To do this, we first removed the angular velocity attributes of both hands from all velocity segments and then we applied the PCA and CVX feature extraction algorithms as above. The same procedure is repeated when we further remove the angular velocity attributes of the head. The results of these experiments are shown in Table 3.3

and Table 3.4 for train and test splits, respectively.

On the training split, we can see that the models trained with either the PCA or CVX feature representations fail to identify a large percentage of LL and HL gains when the angular velocity of the head and both hands are included. When the angular velocity of both hands are removed, the ML model trained using CVX features achieves much better performance but the the model trained on PCA features is still poor. This makes sense if the knowledge of angular velocity of the hands either is not useful for predicting learning gains and/or the angular velocity of the hands adds a significant amount of noise, which results in some eigenvectors in PCA that simply fit the noise. Once we remove the angular velocity of the head as well as the hands, the model trained using PCA features can achieve comparable results to the model trained with CVX features on the training set.

On the test split, we can see that the model performance is quite robust no matter whether we include the angular velocities or not, though the affect of these features on the confidence of the ML models' predictions is less clear. This suggests that the additional angular velocity attributes add little generalization power to the ML model and only make it more difficult to train. In other words, we can, and likely should, drop the angular velocity features to improve performance. However, given our limited data setting, we view the above as preliminary observations; a larger data set is necessary to draw stronger conclusions.

### 3.4 Visual Inspection of Machine Learning Results

Given the promising results of our machine learning experiment, we decided to visually inspect segments of tracking data that were correctly identified as LL or HL by the CVX representation, in order to understand how the SVMs classified participants with such high accuracies. Even

though segments are independent of the VR system’s tasks, we wanted to visualize LL and HL segments corresponding to the same VR system task to reduce confounds for the visual inspections. However, because the participants completed the VR tasks at different times within the application, we had to manually inspect the tracking data of segments to find those with corresponding tasks. For both groups of learning gains, we manually inspected the 50 segments with the greatest HL+CL CVX confidences that were correctly identified. We identified a total of 20 pairs of LL and HL segments with corresponding VR tasks. See Figure 2 for visualizations of one of these paired segments and a description of their underlying tasks.

In order to compare each pair of corresponding LL and HL segments, we needed a method for visualizing their tracking data. A number of prior works have used top-down views of tracked positions to visualize trajectories of VR travel [4, 21, 48, 92]. However, none of these prior works have visualized handheld controller data, in addition to HMD data. In the robotic surgery domain, Hung et al. [51] have used 3D perspectives to visualize the trajectories of tracked handheld instruments and camera.

Having considered prior works, we developed our own technique for visualizing HMD and handheld controller tracking data. Like some prior works [4, 92], we use a top-down view with the virtual environment serving as a background for the visualization. We render the tracked HMD, left-, and right-handheld controllers as purple, yellow, and cyan trajectories, respectively. To convey the temporal relationships among these lines, we chose to render two lines from the HMD position to the two controller positions, using a temporal blue-red color gradient, where blue represents the oldest data and red represents the newest. To avoid visual clutter, we only rendered these HMD-controller lines on certain frames. To determine which ones, we chose to render those frames in which the linear velocity of any tracked object exceeded 1 m/s, which was inspired by our machine learning results.

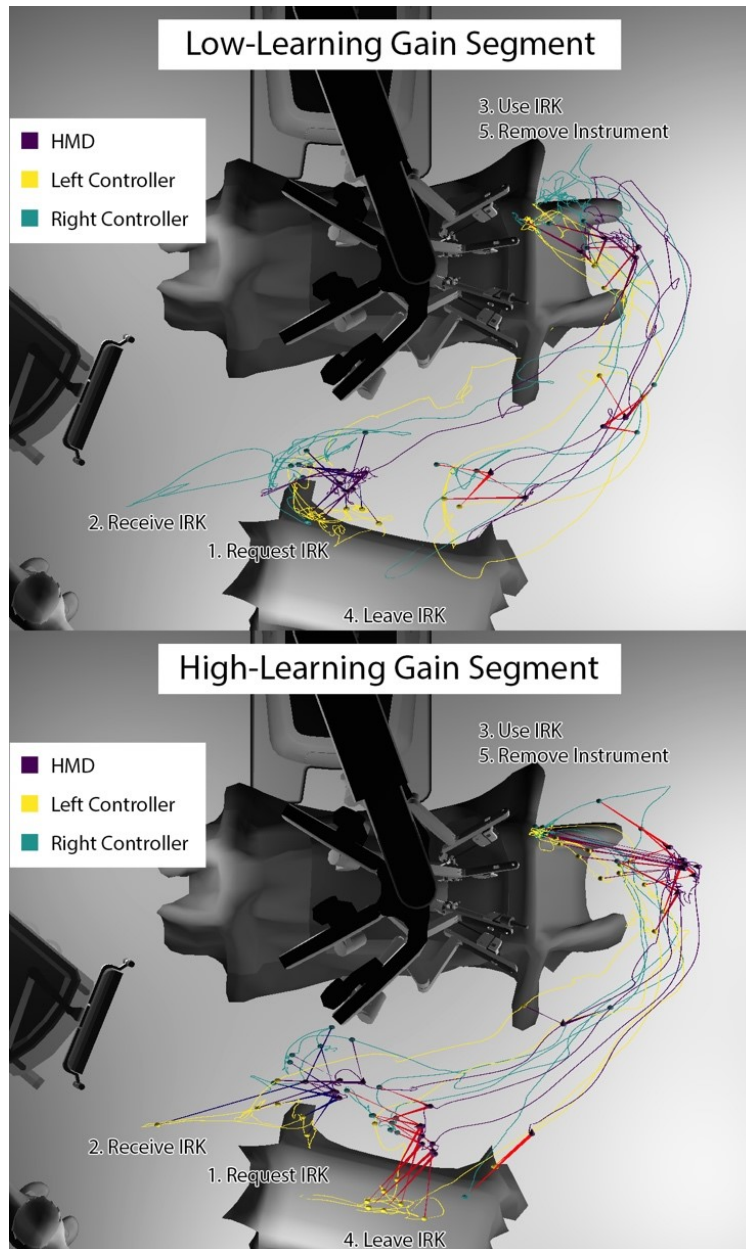


Figure 3.2: Visualizations of a pair of corresponding low- and high-learning gain segments. The underlying VR tasks involved: 1) Requesting an instrument release kit (IRK) via a floating dialog menu, 2) Receiving the IRK from a virtual agent, 3) Using the IRK on the instrument arm, 4) Leaving the IRK on the back table, and 5) Removing the instrument. The blue-red color gradient of the connected HMD-controller lines indicates the oldest (blue) and newest (red) frames in which a linear velocity exceeded 1 m/s.

Upon visual inspection of our visualizations, we quickly noticed two major differences between our paired LL and HL segments. First, we noticed that the LL segments contained more changes in positional direction than the HL segments, for all three tracked devices (see Figure 3.2). This result indicates that the LL participants were more haphazard with their movements while the HL participants maintained more control over theirs. Second, we noticed that the HL segments contained many more velocity-based, HMD-controller lines than the LL segments (see Figure 3.2). This result indicates that when the HL participants moved, they moved with more speed and likely intention, as opposed to the LL participants, who moved more slowly and were possibly unsure of their movements.

Considering our visualization outcomes indicating that the HL participants moved with more control and more deliberately while the LL participants moved more haphazardly but more timidly, we decided to investigate whether prior VR experience was the cause. We conducted a Mann-Whitney U test to compare learning label (LL or HL) differences between those participants with prior VR experience and those without. However, we did not find a significant difference between the two groups ( $U = 441$ ,  $p = 0.605$ ), which indicates that prior VR experience did not have a significant effect on learning gains for the VR training system.

### 3.5 Discussion

In the following sections, we discuss predicting learning gains and the movements of learners during VR training.



### *3.5.1 Feasibility of Predicting Learning Gains*

The results of our machine learning experiment indicate that it is feasible to predict learning gains during a VR training application, based on the velocities of the HMD and handheld controllers. We found that the investigated machine-learning approaches yielded a mean accuracy of 93.1% on our test data for predicting LL and HL gains and a mean overall accuracy (i.e., testing and training data) of 87.2%. We also found that the approaches yielded a mean test accuracy of 91.7% and a mean overall accuracy of 89.2% for predicting LL gains. This is important since the motivation for predicting learning gains is to develop adaptivity techniques for learners that are struggling.

Once we have established a larger data set to refine our machine-learning models on, the next step will be to validate our classifiers in practice. We plan to conduct another large user study, in which we will use our classifiers to predict each participant's learning gains at the end of the VR training application. We will then compare these predictions to the post-training knowledge test results, in order to assess and validate the accuracies of our classifiers in practice. Accurate prediction of learning gains will enable trainers to identify users that likely require more training, reducing the time cost of separate evaluations when training to proficiency.

Finally, once we have validated the ability to predict learning gains with a high-degree of accuracy, we will then need to investigate how to modify our machine-learning classifiers to be used in real time. This capability is necessary in order to provide real-time adaptivity techniques, such as scaffolding and feedback. A system could potentially identify users with LL gains during training and dynamically adjust the prompts, animations, and cues provided to intervene and reduce the time spent training while learning poorly. Tentatively, we expect to use a vote-oriented approach [83]. This would require investigating a proper vote weighting method, confirmation and election methods, and expiration parameters.

### 3.5.2 *Movements of Learners During VR Training*

An interesting result from our research were the differences in movements between participants with LL and HL gains. Using a novel visualization technique (see Section 3.4), we were able to visually inspect the tracking segments correctly identified by our machine-learning models. We found that participants with LL gains moved less smoothly but also more slowly than participants with HL gains. These results indicate that participants with LL gains may have been unsure of their movements while those with HL gains may have been more confident in their movements. These outcomes also mirror results from the robotic surgery domain, in which experienced robotic surgeons are found to have better economies of motion (i.e., fewer overall motions that are generally smooth and continuous except during fast ballistic motions) than less-experienced robotic surgeons [35, 65].

We also investigated whether these results pertaining to economy of motion were due to prior VR experiences. However, we found no significant differences between participants with LL and HL gains due to prior VR experiences. Hence, we believe the observed economies of motion were indicative of cognitive performance, rather than prior VR experience.

### 3.5.3 *Limitations*

Despite these promising results, more research needs to be conducted to achieve our goal. First, it is important to note that we had a small number of HL gains to train and test our machine-learning approaches on, due to the complexity of the VR training system, as discussed in Section 3.4. The next step in our research is to conduct a much larger user study, in order to hopefully find a greater number of HL examples. This will help refine our machine-learning models and avoid overfitting our data set, which would negatively impact any future uses of our machine-learning classifiers.

Another limitation is that while there clearly appears to be a link between user motion and knowledge test performance, it is unclear how relevant the HL/LL split is with regard to training performance. It is possible that a human trainer may view training performances quite differently, compared to knowledge test performances. Additionally, while our HL/LL split was based on the mean and standard deviation of the entire population, this wouldn't necessarily be feasible in a real-world scenario where any such division would have to be based on training data alone. In our case, the HL/LL split for the entire population was 8.38 ( $\sigma = 3.07$ ), while for just the training population, it was 8.57 ( $\sigma = 2.85$ ). So, we would still have used a score of 12 as the threshold for HL/LL gains.

It's worth noting that our evaluation with the test data yielded better results than the evaluation using the training data as seen in Tables 3.3 and 3.4. This is likely because there were greater variations within the training data than the testing data due to its larger sample size. These greater variations resulted in the testing data yielding better results than the training data.

Additionally, the selected training scenario required significant interactions and movements. This may have aided the classifier in its ability to distinguish learning performance compared to a virtual environment consisting of only minimal interactions. Further research should be conducted to determine to what degree the magnitude of required movements in a training scenario affects the capability of a learning gains classifier.

### 3.6 Conclusion

In this chapter, we investigated the feasibility of predicting learning gains during a VR training experience, based on a user's HMD and controller movements. We conducted a user study with a VR training simulation for learning how to troubleshoot a surgical robot and used a post-training

knowledge test to assess learning gains. Based on the results of the knowledge test, we separated participants into low- and high-learning gains, which we then randomly assigned to training and testing data sets. We converted the positional and rotational tracking data from the VR training system to linear and angular velocity vectors, which we used as input features. We used SVM classifiers and two different feature representations (Principal Component Analysis and convex matrix factorizations) to model learning gains based on the velocity vectors. Given its highly accurate results, our machine-learning experiment indicates that it is feasible to predict learning gains during a VR training application, based on the velocities of the HMD and handheld controllers. Visual inspection of these results indicates that participants with high-learning gains exhibit better economies of motion than those with low-learning gains.

# **CHAPTER 4: EXPLORATION OF FEATURE REPRESENTATIONS FOR PREDICTING LEARNING AND RETENTION OUTCOMES IN A VIRTUAL REALITY TRAINING SCENARIO**

*NOTE:* This chapter is a modified format version of the paper previously published.

Material from: A. G. Moore, R. P. McMahan, and N. Ruozzi. Exploration of feature representations for predicting learning and retention outcomes in a vr training scenario. *Big Data and Cognitive Computing*, 5(3), 2021. ISSN 2504-2289. doi: 10.3390/bdcc5030029. URL <https://www.mdpi.com/2504-2289/5/3/29>

## 4.1 Introduction

With the positive results we identified in the previous chapter for the prediction of learning gains in virtual environments, we continued to investigate other cognitive outcomes that we may be able to predict. Our motivation for continuing to pursue this line of research was based on the increasing use of VR for education and training [10], hoping to further our understanding of RQ1, and to begin to answer RQ2: Can readily available VR tracking data be used to predict psychomotor outcomes from a VR training scenario?

VR practitioners have developed educational VR environments for knowledge acquisition, such as learning about geometry [61], World War I [125], and how to inspect a haul truck [75]. VR has also been used for training psychomotor skills, such as how to visually scan for threats [100], use measurement tools [7], and put on personal protective equipment [36]. As consumer VR systems have become more easily available, these educational uses of VR have gained more attention [16].

The field of Intelligent Tutoring Systems (ITSs) is closely related to VR training and education. The goal of an ITS is to develop a deep understanding of each learner’s behavior to allow for intelligent engagement in sustained learning activities [22]. While much work in the field of ITSs is focused on immediate knowledge, some research has made strides towards attempting to understand and predict longer-term retention of information, which is a better indicator of mastery than short-term retention [121]. ITSs incorporate Artificial Intelligence (AI) techniques to allow for tutoring systems that know “*what* they teach, *who* they teach, and *how* to teach it” through user models [93]. ITSs can use these user models to provide adaptive support for learners because they provide a representation of the learner in terms of relevant traits like learning behaviors and meta-cognitive ability [2]. This adaptive support can be implemented via proactive guidance (*scaffolding*) or reactive guidance (*feedback*) [116]. Most ITSs that create adaptive learning environments employ feedback [116], which is usually implemented with static rules for specific types

of detected errors [42].

In this chapter, we investigate using a machine learning (ML) approach to classify users of a VR training application into groups of low and high learning and retention outcomes, expanding upon our previous work which only sought to predict learning outcomes [86]. Our work in this chapter makes use only of the head-mounted display (HMD) and handheld controller tracking data, as opposed to events or errors [42], which makes our approach viable for real-time scaffolding, before events or errors even occur. Unlike most prior approaches that are only capable of feedback [116], this tracking-based ML approach can potentially be used for both feedback and scaffolding, such as providing interaction cues [50], guiding the user's attention [23], or even simplifying the interactions [74]. Our work explores using HMD and controller tracking data as a means for predicting knowledge acquisition, knowledge retention, and performance retention.

After discussing related work, we present a user study in which participants used a VR training application to learn the procedure for troubleshooting a surgical robot [85], then returned a week later to perform this task again in VR. Participants were administered a knowledge test after the initial training session and before the week-later retention session to evaluate knowledge acquisition and retention, respectively. Errors were tracked in the retention simulation to evaluate performance retention. For each learning and retention outcome, we used the results to separate participants into two groups: high-performance participants scoring at least one standard deviation above the mean and low-performance participants (i.e., all remaining participants that did not score at least one standard deviation above the mean).

We then present the results of an ML experiment that employed support-vector machines (SVMs) [9] to predict which group each participant belonged to, based on their tracking data. For the experiment, we compared six different sets of input features based whether the data represented positions or velocities and three different combinations of HMD and controller data: a) linear-and-angular

features for both the HMD and controllers, b) linear-and-angular features for the HMD and linear-only features for the controllers, and c) linear-only features for both the HMD and controllers. We compared the accuracy (i.e., correctness of the prediction) and Matthews correlation coefficient (MCC) (i.e., a measure of the quality of binary classifications [18]) of each model learned from these six different sets of input features. For the feature representations, we applied Principal Component Analysis (PCA) [119]. The results of our ML experiment indicate that this approach can yield high degrees of accuracy for predicting learning and retention outcomes, with our maximum observed overall accuracy at 96.7%. However, we note varying accuracy across our different input features, which reinforces the usefulness of exploring these features as hyperparameters when building models for similar educational purposes.

In order to understand how the SVMs classified participants into their high and low-performance groups with such high degrees of accuracy, we have visually inspected tracking data segments that were correctly identified by the SVMs. Visualizations show that participants with high learning and retention outcomes moved with better economies of motion than those participants with lower outcomes. These results show that it is feasible to use HMD and controller tracking data to develop new real-time scaffolding and feedback techniques for VR training applications. We anticipate that our work will be useful in the development of systems that predict learning and retention outcomes, and subsequently respond dynamically when learning can be improved. Such systems may make use of real-time classifiers to provide scaffolding and feedback that better support the user's learning.

In this chapter, we present the following research activities:

- A study that collected VR tracking data and results pertaining to knowledge acquisition, knowledge retention, and VR-based performance retention from 60 participants across two VR training sessions, separated by one week.



- An ML experiment investigating six sets of input features for predicting the three different outcomes. This is the first such experiment to investigate both learning and retention, particularly psychomotor retention.

We also present the following research results:

- Our results indicate that our velocity-based ML models generally outperformed our position-based models for predicting all three outcomes.
- Our results also indicate that VR tracking data can be better used to predict psychomotor-based outcomes than cognitive-based outcomes.

## 4.2 VR Learning and Retention User Study

We made use of an existing VR application designed for training first assistants how to troubleshoot a faulted arm on a surgical robot [85]. This robotic operating room (OR) application makes use of virtual hand-based selections and manipulations [84, 78] to support multiple equipment interactions, as well as communicating with a virtual surgeon and staff member by selecting dialog options. In order to complete the scenario, the user must perform an ordered set of steps involving these interactions while moving around the virtual OR by physically walking [62] and periodically looking at a vision cart screen. Table 4.1 shows a list of these subtasks and the types of interactions that they require.

We slightly varied the virtual training application between the learning and retention sessions. During the learning session, it provided interaction cues, which convey actions to take [49], for each step. These interaction cues consisted of verbal instructions and visual animations showing

Table 4.1: The subtasks and their required interactions involved in the VR training simulation.

#	Subtask	Interaction
1	Check error message	Walk + Look
2	Consult Surgeon	Select (dialog)
3	Ask to press power down	Select (dialog)
4	Ask to press power up	Select (dialog)
5	Ask to call support	Select (dialog)
6	Ask for release wrench	Select (dialog)
7	Grab release wrench	Select (wrench)
8	Ask for emergency stop	Select (dialog)
9	Hold instrument carriage	Select (carriage)
10	Insert wrench	Position (wrench)
11	Rotate wrench	Rotate (wrench)
12	Check vision monitor	Look
13	Remove wrench	Position (wrench)
14	Remove instrument	Position (instrument)
15	Give instrument to staff	Position (instrument)
16	Ask to recover fault	Select (dialog)
17	Ask to disable arm	Select (dialog)
18	Check error message	Walk + Look
19	Use cannula lever	Select (lever)
20	Use instrument clutch	Position (clutch)
21	Use port clutch	Position (clutch)
22	Ask to confirm disable	Select (dialog)
23	Check error message	Look
24	Ask to press recover fault	Select (dialog)

perceived affordances and feedforward information. Selection cues used semi-transparent green controller models that continuously linearly interpolated from the user's controller to the target dialog option or object to be selected (see Figure 4.1). Manipulation cues used semi-transparent green copies of the objects being manipulated and linearly interpolated these copies to the target positions for the manipulations. Finally, for travel cues, these animations consisted of semi-transparent green boots that linearly interpolated from the user's position to a icon representing the travel destination that read "Stand Here".

In the learning version of our VR training application, the interaction cues described were preemptively presented to the users, in order to demonstrate and train how to perform the troubleshooting task. However, for the retention-session version, these cues were not presented, unless the user



Figure 4.1: A first-person perspective of the VR training application used for our research.

committed an error or was inactive for 30 seconds.

#### *4.2.1 Materials*

The VR hardware for this study was HTC Vive system, consisting of an HMD and two handheld controllers, which were used to interact with the VR training application. The display of the Vive HMD has a resolution of 1080x1200 pixels per eye, a 90Hz refresh rate, and affords a 110° diagonal field of view (FOV). We fitted the HMD with the Vive audio strap that integrates over-the-ear headphones. The VR application maintained 90 frames per second to match the Vive's refresh rate and was developed in Unity. The input data from the Vive was processed with the

SteamVR plugin. For every frame in the VR training application, the HMD and controller tracking data was logged. This data consisted of the global positions and quaternions of both the HMD and the controllers, as well as the frame's timestamp.

#### 4.2.2 Procedure

The following procedure was reviewed and approved by the University of Texas at Dallas Institutional Review Board (IRB).

The human subjects study consisted of one learning and one retention session for each participant. The duration of the sessions was approximately 60 minutes for the learning session and 30 minutes for the retention session. The retention session occurred one week after the learning session.

Participants first gave informed consent, then the learning session began with a background survey on the demographics, education, and technology experience of the participant. To train the participant on how to use the HTC Vive, the experimenter would help the participant put on the HMD and run the SteamVR tutorial. After this, the experimenter would then run the participant through the VR training application. Once the participant finished, they were then administered a number of questionnaires regarding their VR experience. Finally, participants were administered a knowledge test consisting of multiple-choice questions pertaining to the training scenario.

One week later (restricted to the same day of the week to avoid confounds), a participant would begin the retention session with the experimenter administering the same knowledge test to measure knowledge retention. After completing the test, the experimenter would help the participant to put on the HTC Vive, and the participant would then experience the retention version of the VR application. After completing the retention application, the participant was given a free-response exit survey and compensated \$15 USD.

### 4.2.3 *Participants*

A total of 61 participants were recruited through university mailing lists and completed the initial training session. However, one participant did not return to complete the retention session. Thus, our data consists of 60 participants (11 females, 49 males). None of our participants had prior knowledge of or experience with surgical robots or the training task. The mean age of the participants was  $22.6 \pm 4.2$  years.

## 4.3 Machine Learning Experiments

In this section, we discuss the learning and retention outcomes, experimental design, input features, input feature representation, hyperparameters, and scoring method.

### 4.3.1 *Learning and Retention Outcomes*

For this research, we are concerned with three learning and retention outcomes: Knowledge Acquisition, Knowledge Retention, and Performance Retention. We measure Knowledge Acquisition and Knowledge Retention with the knowledge tests that were administered at the end of the learning session and beginning of the retention session, respectively. We measure Performance Retention by tracking errors and completion time at the subtask level. If participants made no errors and completed a subtask within 30 seconds (i.e., the period of inactivity allowed before an interaction cue was presented), they are regarded as having successfully completed that subtask. In order to predict these different outcomes in a consistent way, we choose to implement a similar approach to that by Moore et al. [86]. For each outcome factor, we fit a Gaussian distribution to the observed scores, then classify all scores one standard deviation above the mean as high performance, and all others as low performance. These splits are described in Table 4.2.

Table 4.2: Evaluation metrics and categorizations of participants.

Metric	Value Range	Mean Score	SD	# High	# Low
Knowledge Acquisition	[0, 20]	8.45	3.04	10	50
Knowledge Retention	[0, 20]	10.50	3.60	9	51
Performance Retention	[0, 24]	8.47	3.05	9	51

### 4.3.2 Experimental Design

In order to evaluate the performance of our models, we conduct an exhaustive grid-search with four-fold, participant-level cross-validation across all hyperparameter options. This is performed by first segmenting our data into an 80/20 train/test split for later evaluation. On the training data, we create four roughly equal partitions to iteratively train on three of the partitions and evaluate on the fourth, retaining the average score for comparison against other hyperparameter configurations. We then select the configuration with the highest average score, train on the entirety of the training data, and then test on both the training and testing data, separately and overall.

### 4.3.3 Input Features

Our data consists of the position and rotation of the tracked HMD and handheld controllers at a rate of 90Hz, the frame rate of the training application. We encode this tracking data in the form of a three-value vector for position and a four-value quaternion for rotation. To mitigate possible noise introduced by including the rotation data, we compare three sets of position and rotation data: a) linear-and-angular features for both the HMD and controllers, b) linear-and-angular features for the HMD and linear-only features for the controllers, and c) linear-only features for both the HMD and controllers. Because of recent success seen in making use of velocity rather than position for predicting success [86, 96], we also look at the linear and angular velocities derived from the position and rotation data, thus resulting in six total conditions for each evaluation metric.

#### 4.3.4 *Input Feature Representation*

In our analysis, we take the 90Hz data and use a sliding window approach to chunk it into spans that are more meaningful than an instant of data. This window is made by selecting a time span for the segment length, and another for the shift size. Depending on these parameters, there can be a significant amount of overlap between subsequent windows. The frames of data within a window are concatenated to create a high-dimensional vector that encodes the information. These vectors are then processed with Principal Component Analysis (PCA) to extract salient features. This process greatly reduces the amount of data and forms the input vectors used for training our models.

#### 4.3.5 *Hyperparameters*

We consider several hyperparameters for both the representation of the input and the support vector machine (SVM). The six described options for Input Feature Representation can be regarded as hyperparameters within an evaluation metric prediction, however, in this chapter, we will treat them as separate experiments as we are interested in discussing the differences in their performance. The sliding window approach for chunking our data is determined by two hyperparameters: the segment length and shift size. The PCA feature extraction also has a hyperparameter for determining the number of features extracted. Finally, the SVM that we use in our methodology has two hyperparameters: penalty and expressivity. The range of values that we explore for these five hyperparameters are shown in Table 4.3.

Table 4.3: Value range of hyperparameters considered.

Hyperparameter	Seg. Length	Shift Size	PCA Features	Penalty	Expressivity
Min Value	30	5	10	1	0.001
Max Value	120	13	80	10000	0.090

#### 4.3.6 Scoring Method

We fit SVMs [9] with Gaussian kernels to the featurized data. The resulting linear separator will yield a classification of low or high for each window segment. One possible way to extend the segment classification to an entire user is to determine the user’s label by majority vote over the predicted labels of the user’s velocity segments. This approach is reasonable, but doesn’t accommodate well for cases where several vectors are incorrectly classified with low confidence. Thus, we weight each vote based on its distance from the decision boundary.

Let  $l_{u,i}$  and  $d_{u,i}$  denote the predicted label and corresponding distance for the  $i^{th}$  segment of participant  $p$ . We define the participant-level confidence that  $p$  belongs to class  $c$  as:

$$\text{conf}_{p,c} = \frac{\sum_i \text{st. } l_{p,i}=c d_{p,i}}{\sum_{i=1}^N d_{p,i}},$$

where  $N$  is the number of segments that belong to participant  $p$ . Then, the predicted label of the user is determined by selecting the class  $c$  that maximizes the confidence.

## 4.4 Machine Learning Results

In this section, we will detail the results of our ML experiments across our learning metrics. We choose to analyze the performance of each model in terms of the Matthew’s Correlation Coefficient (MCC) score. This value is a special case of the Pearson’s Correlation Coefficient for two cate-



gories and is valued from -1 (full counter-correlation) to 1 (full correlation). Unlike metrics such as accuracy, MCC takes into account the relative size of each category when assigning a score, thus balancing the weight of disparately sized categories [18]. This is particularly useful for our experiments in which the relative category sizes have approximately a 5:1 ratio of low-performance to high-performance participants.

#### *4.4.1 Knowledge Acquisition Results*

Table 4.4 shows the full results of our Knowledge Acquisition experiment. First, it is clear from the discrepancies in high-performance accuracy between the training (High Acc = 1.000) and testing (High Acc = 0.000) datasets and the MCC scores for the testing data (MCC = -0.135) that the first and third sets of input features, both position-based, suffered from overfitting the high-performance training data. Similarly, the sixth set, the velocity-based, linear-only input features set, also suffered from overfitting the high-performance training data, resulting in a low MCC score (MCC = 0.000).

Interestingly, the two best performing models were the second and fifth sets of input features (i.e., the position-based and velocity-based linear-and-angular HMD features and linear-only controller features, respectively). The velocity-based fifth set yielded the best test MCC of 0.674 and a better overall accuracy of 0.833 than the position-based second set, which yielded a test MCC of 0.400 and an overall accuracy of 0.783. This position-based second set was more conservative in identifying low-performance participants in both the training (Low Acc = 0.725) and testing (Low Acc = 0.900) datasets than the velocity-based fifth set, which was more accurate for both training (Low Acc = 0.825) and testing (Low Acc = 1.000).

Table 4.4: Knowledge Acquisition prediction results. For each of the six sets of input features, the accuracy of the low-performance and high-performance predictions and the MCC are shown for both the training and testing data. Accuracy and MCC are also shown for each set of input features across the entire dataset (i.e., both the training and testing data).

Set	Position / Velocity	HMD	Controllers	Train / Test	Low Acc	High Acc	MCC	Overall Acc	Overall MCC
1	Position	Lin+Ang	Lin+Ang	Train	0.800	1.000	0.632	0.817	0.513
	Position	Lin+Ang	Lin+Ang	Test	0.900	0.000	-0.135		
2	Position	Lin+Ang	Linear	Train	0.725	1.000	0.553	0.783	0.516
	Position	Lin+Ang	Linear	Test	0.900	0.500	0.400		
3	Position	Linear	Linear	Train	0.829	1.000	0.665	0.836	0.541
	Position	Linear	Linear	Test	0.900	0.000	-0.135		
4	Velocity	Lin+Ang	Lin+Ang	Train	0.700	0.750	0.346	0.700	0.309
	Velocity	Lin+Ang	Lin+Ang	Test	0.700	0.500	0.158		
5	Velocity	Lin+Ang	Linear	Train	0.825	0.750	0.482	0.833	0.493
	Velocity	Lin+Ang	Linear	Test	1.000	0.500	0.674		
6	Velocity	Linear	Linear	Train	1.000	0.625	0.762	0.917	0.674
	Velocity	Linear	Linear	Test	1.000	0.000	0.000		

Table 4.5: Knowledge Retention prediction results. For each of the six sets of input features, the accuracy of the low-performance and high-performance predictions and the MCC are shown for both the training and testing data. Accuracy and MCC are also shown for each set of input features across the entire dataset (i.e., both the training and testing data).

Set	Position / Velocity	HMD	Controllers	Train / Test	Low Acc	High Acc	MCC	Overall Acc	Overall MCC
1	Position	Lin+Ang	Lin+Ang	Train	0.732	0.857	0.435	0.717	0.358
	Position	Lin+Ang	Lin+Ang	Test	0.600	0.500	0.076		
2	Position	Lin+Ang	Linear	Train	0.732	0.857	0.435	0.733	0.377
	Position	Lin+Ang	Linear	Test	0.700	0.500	0.158		
3	Position	Linear	Linear	Train	0.756	0.857	0.459	0.717	0.358
	Position	Linear	Linear	Test	0.500	0.500	0.000		
4	Velocity	Lin+Ang	Lin+Ang	Train	0.854	0.714	0.477	0.817	0.430
	Velocity	Lin+Ang	Lin+Ang	Test	0.800	0.500	0.258		
5	Velocity	Lin+Ang	Linear	Train	0.610	0.857	0.331	0.617	0.327
	Velocity	Lin+Ang	Linear	Test	0.400	1.000	0.316		
6	Velocity	Linear	Linear	Train	0.805	0.714	0.412	0.783	0.380
	Velocity	Linear	Linear	Test	0.800	0.500	0.258		

#### 4.4.2 Knowledge Retention Results

Table 4.5 shows the complete results of our Knowledge Retention experiment. Unlike the Knowledge Acquisition experiment, there are no clear results indicating that any of the six models overfitted the training data for low or high-performance outcomes. However, it is also the case that none of the six models performed as well on the Knowledge Retention results, as they performed on the Knowledge Acquisition results. For Knowledge Acquisition, five of the six models yielded overall MCC scores ranging from 0.493 to 0.674. However, for Knowledge Retention, the six models only yielded overall MCC scores ranging from 0.327 to 0.430. In terms of test MCC scores, two of the models, the first and third sets, yielded chance-like results (i.e., near zero MCC scores).

The best performing model for predicting Knowledge Retention was the fourth set of velocity-based, linear-and-angular HMD and controller input features. This model yielded the best overall MCC of 0.430 and the second best test MCC of 0.258, which would have dramatically improved had it identified both high-performance participants instead of just one (i.e., High Acc = 0.500). The sixth set of velocity-based, linear-only HMD and controller input features performed similarly with an overall MCC of 0.380 and the same test MCC of 0.258. Finally, the fifth set of velocity-based, linear-and-angular HMD and linear-only controller features performed well with an overall MCC of 0.327 and the best test MCC of 0.316. However, this model was overly generous with high-performance labels, particularly in the testing data (High Acc = 1.000), which yielded poor low-performance identification (Low Acc = 0.400).

#### 4.4.3 Performance Retention Results

Table 4.6 shows the complete results of our Performance Retention experiment. Like the Knowledge Retention results, there are no clear results indicating that any of the six models overfitted the

Table 4.6: Performance Retention prediction results. For each of the six sets of input features, the accuracy of the low-performance and high-performance predictions and the MCC are shown for both the training and testing data. Accuracy and MCC are also shown for each set of input features across the entire dataset (i.e., both the training and testing data).

<b>Set</b>	<b>Position / Velocity</b>	<b>HMD</b>	<b>Controllers</b>	<b>Train / Test</b>	<b>Low Acc</b>	<b>High Acc</b>	<b>MCC</b>	<b>Overall Acc</b>	<b>Overall MCC</b>
<b>1</b>	Position	Lin+Ang	Lin+Ang	Train	0.878	1.000	0.716	0.867	0.620
	Position	Lin+Ang	Lin+Ang	Test	0.800	0.500	0.258		
<b>2</b>	Position	Lin+Ang	Linear	Train	0.927	1.000	0.805	0.900	0.685
	Position	Lin+Ang	Linear	Test	0.800	0.500	0.258		
<b>3</b>	Position	Linear	Linear	Train	0.780	1.000	0.584	0.800	0.572
	Position	Linear	Linear	Test	0.700	1.000	0.529		
<b>4</b>	Velocity	Lin+Ang	Lin+Ang	Train	0.683	0.714	0.290	0.700	0.341
	Velocity	Lin+Ang	Lin+Ang	Test	0.700	1.000	0.529		
<b>5</b>	Velocity	Lin+Ang	Linear	Train	0.780	1.000	0.584	0.800	0.517
	Velocity	Lin+Ang	Linear	Test	0.800	0.500	0.258		
<b>6</b>	Velocity	Linear	Linear	Train	1.000	1.000	1.000	0.967	0.869
	Velocity	Linear	Linear	Test	0.900	0.500	0.400		

training data for low or high-performance outcomes. However, unlike the Knowledge Retention results, the six models performed relatively well on the Performance Retention results, yielding overall MCC scores ranging from 0.341 to 0.869 and test MCC scores ranging from 0.258 to 0.529.

The best performing model for predicting Performance Retention in the VR training application was the sixth set of velocity-based, linear-only HMD and controller input features. It yielded the highest overall MCC score of 0.869 (with an overall accuracy of 0.967), and it yielded one of the top test MCC scores of 0.400, incorrectly predicting exactly one low-performance participant and one high-performance participant. The third and fourth sets of input features yielded the best test MCC scores of 0.529. However, these models were generous with high-performance labels in the testing data (High Acc = 1.000), which yielded conservative low-performance identification (Low Acc = 0.700).

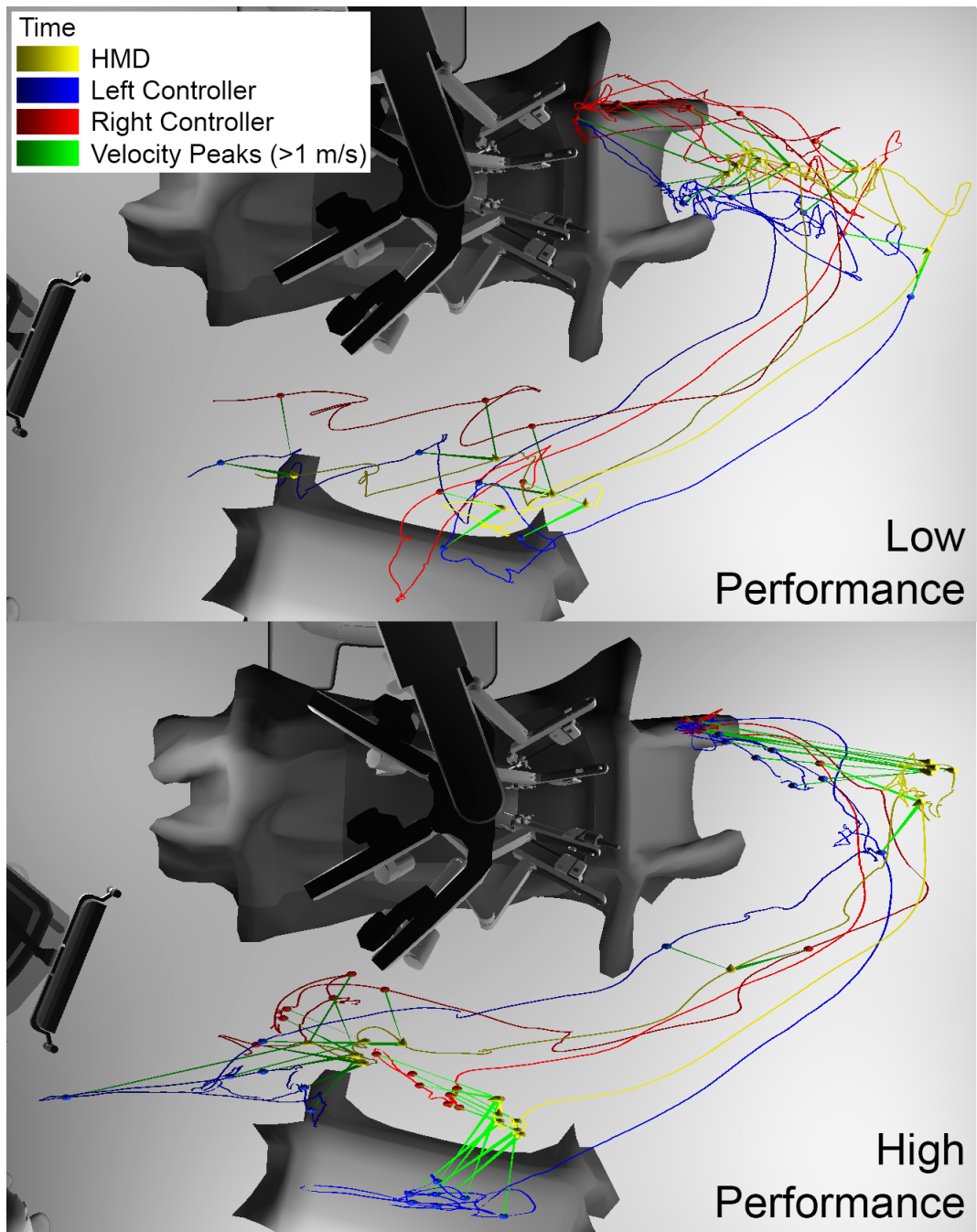


Figure 4.2: Visualizations of motion data from a low-performance participant and a high-performance participant, based on all three metrics, for the same set of actions.

## 4.5 Visual Inspection of Results

Given the highly accurate results of our ML experiments, we decided to conduct visual inspections of the motions of participants, in order to try and understand how the SVMs are classifying participants with such high degrees of accuracy. We use an approach similar to our recent work [86], rendering the positions and high-velocity moments of the tracking data within the environment, in a top-down orthographic view. We show the HMD tracking data in yellow, and the left and right controller data in blue and red, respectively. To convey high-velocity moments, green lines connecting the three tracked traces are rendered for frames in which the velocity of any tracked device exceeded 1 m/s, with a minimum 15 frame break since the last rendered line to prevent visual clutter. To convey time, we adjust the brightness of all four traces from dim to bright over the course of the segment. For example, Figure 4.2 shows the visualizations of the same set of actions for a participant with low performances on all three metrics (i.e., Knowledge Acquisition, Knowledge Retention, and Performance Retention) and a participant with high performances on all three metrics.

Through our visual inspections, we have noticed that the motions of low-performance participants generally have more directional changes and return more often to prior positions than the motions of high-performance participants. Additionally, we have observed that low-performance participants exhibit fewer velocity peaks (i.e., moments in which velocity exceeds 1 m/s) than high-performance participants. Together, these results suggest that low-performance participants moved more haphazardly and with less certainty. In contrast, high-performance participants moved more smoothly and with greater certainty, and likely, intention. These results are similar to prior results from the robotic surgery domain, in which less-experienced robotic surgeons often demonstrate worse economies of motion (i.e., excessive motions) than more-experienced robotic surgeons [35, 65].

Table 4.7: Results of Mann-Whitney U tests comparing the distribution differences of low and high performances among participants with or without prior VR experiences.

Metric	Prior VR Exp	No VR Exp	U	p
	Low:High Ratio	Low:High Ratio		
Knowledge Acquisition	28:6	22:4	432.0	0.888
Knowledge Retention	29:5	22:4	439.0	0.968
Performance Retention	27:7	24:2	385.0	0.401

Considering that our visualizations indicated that low-performance participants moved more haphazardly and timidly, we decided to investigate whether prior VR experience had a significant effect on whether a participant would be a low or high-performance learner. We conducted a Mann-Whitney U test for each metric to compare the distribution differences of low and high performances among participants with and without prior VR experiences. Table 4.7 shows the results of these tests. The results clearly indicate that prior VR experience did not have a significant effect on Knowledge Acquisition, Knowledge Retention, or even Performance Retention.

## 4.6 Discussion

In this section, we will discuss our results and what knowledge has been gained through this work.

### 4.6.1 Position-based versus Velocity-based Models

Overall, we found that our velocity-based models generally outperformed our position-based models. For all three metrics, we found that one of our velocity-based models yielded the best results. Our velocity-based, linear-and-angular HMD with linear-only controller model was the best for predicting Knowledge Acquisition. Our velocity-based, linear-and-angular HMD and controller input features had the best results for predicting Knowledge Retention. Finally, our velocity-based,

linear-only HMD and controller input features were the best for predicting Performance Retention.

In contrast, we observed that our position-based models produced some of the worst results. For predicting Knowledge Acquisition, we found that two of the position-based models suffered from overfitting the high-performance training data. Similarly, we found that the same two position-based models, the linear-and-angular HMD and controller input features and the linear-only HMD and controller input features, yielded chance-like results for predicting Knowledge Retention. However, all three position-based models did perform moderately well for predicting Performance Retention, which is based on psychomotor skills as opposed to cognitive-only skills.

These results strongly support future research into the use of velocity-based models for predicting learning and retention outcomes, both cognitive and psychomotor ones. On the other hand, particularly if computing resources are limited, future researchers can likely omit position-based models for predicting cognitive learning and retention outcomes, like our Knowledge Acquisition and Knowledge Retention metrics. However, we believe that position-based models are still viable for predicting psychomotor outcomes, based on the results of our Performance Retention experiment.

#### 4.6.2 *Linear-and-Angular versus Linear-Only Models*

In general, we did not find any results indicating that linear-and-angular or linear-only models performed better than the other. In our experiments, we investigated three different combinations of these features: a) linear-and-angular features for both the HMD and controllers, b) linear-and-angular features for the HMD and linear-only features for the controllers, and c) linear-only features for both the HMD and controllers. We found that a velocity-based version of each combination performed the best for one of our three metric predictions. We found that the velocity-based, linear-and-angular HMD and controller combination performed the best overall for predict-



ing Knowledge Retention. We found that the velocity-based, linear-and-angular HMD and linear-only controller model performed the best overall, without overfitting, for predicting Knowledge Acquisition. Finally, we found that the velocity-based, linear-only HMD and controller features performed the best overall for predicting Performance Retention.

These results suggest that there is still much research to be conducted into the investigation of these linear and angular features. It is likely that one combination is not superior to the others, and that researchers should investigate each to identify the best model for their prediction metric. Furthermore, it is important to note that we did not investigate angular-only features, which might be viable models for some types of prediction, such as simulator sickness [55]. There are also other types of features that should be investigated, such as the linear distances between the HMD and handheld controllers [99].

#### 4.6.3 *Cognitive versus Psychomotor Models*

We found that our models produced the best results for the psychomotor-based Performance Retention metric, as opposed to the cognitive-based Knowledge Acquisition and Knowledge Retention metrics. The Performance Retention models yielded generally higher overall MCC scores, with five of the six models ranging from 0.517 to 0.869. In contrast, four of the six Knowledge Acquisition models had overall MCC scores ranging from 0.309 to 0.516 and the best Knowledge Retention model had an overall MCC score of 0.430. The psychomotor-based Performance Retention metric also yielded the best overall accuracy of 0.967, out of all 18 models evaluated, and the only perfect MCC score of 1.000 for the velocity-based, linear-only HMD and controller input features model.

These results indicate that VR tracking data can be better used to predict psychomotor-based learning and retention outcomes than to predict cognitive-based outcomes. This is intuitive as VR

tracking data is directly generated by psychomotor-based actions. Hence, it is a more-direct representation of the psychomotor-based mental models that participants have.

These results also imply that VR technologies are most likely more useful for training psychomotor-based skills, such as troubleshooting a surgical robot, as opposed to cognitive-only skills, such as mathematical calculations. However, more research is necessary to investigate these potential differences in usefulness. In particular, real-world efficacy evaluations of skills transfer is necessary. In our research, we were only able to assess psychomotor retention by having participants use the retention version of our VR training application. Ideally, we would have assessed psychomotor retention by having participants demonstrate a real-world transfer of those psychomotor-based skills to a physical surgical robot and OR environment. However, this was not feasible due to the limited availability of such robotic ORs in hospitals [85].

#### *4.6.4 Limitations*

In our ML experiments, we chose to omit predicting performance during the learning session of our study. This decision was made because while there may be utility in such a classifier, it would be trying to predict values derived in part from actions the participant has already undertaken, and wouldn't be as directly comparable.

Additionally, in this study, we performed PCA to reduce the dimensionality of our input vectors before using them to train the SVM models. While this is a generally accepted practice, for completeness, a future study would ideally also treat rotation algorithms for the PCA, as well as other feature extraction techniques such as convex-hull representations [86], and other types of ML models as hyperparameters, and evaluate performance among those. The decision not to explore those here was due in part to the computational complexity of such a broad exploration.

Finally, the positional data was originally encoded in terms of VR world space, and while this remained consistent with the real-world space in terms of scale, orientation, and being stationary, it decreased the model's ability to encode salient features such as head-to-hand distance. Directly encoding such salient features would likely have been beneficial for our models' predictive power as Pfeuffer et al. [99] found, however evaluating all possible salient features would be intractable, and selecting a subset of features to evaluate over may have introduced bias.

#### 4.7 Conclusion

In this chapter, we explored the feasibility of employing ML models based on different sets of VR tracking features to predict learning and retention outcomes from a VR training application. Our results show that such models can be used to predict such educational outcomes with high degrees of accuracy. Furthermore, our results indicate that velocity-based models are likely better predictors of learning and retention outcomes, particularly cognitive-based outcomes, than position-based models. However, our results did not indicate that any particular combination of linear-and-angular or linear-only conditions yielded better results. Hence, we generally recommend investigating different linear and angular combinations of input features, in addition to considering some features not investigated in this work, such as head-to-hand distances. Finally, our results clearly indicate that VR tracking features can be better used to predict psychomotor outcomes than cognitive ones.

Through the course of this research, we have demonstrated that this data may be valuable to XR practitioners, particularly those in the field of VR-based intelligent tutoring systems. However, without an understanding of the potential privacy implications of collecting and maintaining this data, informed decisions about its use cannot yet be made. In the next two chapters, I will begin to approach this topic.

# **CHAPTER 5: PERSONAL IDENTIFIABILITY AND OBFUSCATION OF USER TRACKING DATA FROM VIRTUAL REALITY TRAINING SESSIONS**

*NOTE:* This chapter is a modified format version of the paper previously published.

Material from: A. G. Moore, R. P. McMahan, H. Dong, and N. Ruozzi. Personal identifiability and obfuscation of user tracking data from vr training sessions. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 221–228, 2021. doi: 10.1109/ISMAR52148.2021.00037

## 5.1 Introduction

In the previous two chapters, value has been demonstrated in collecting Virtual Reality (VR) tracking data for the uses of predicting learning outcomes in training environments. While this data can be valuable, further understanding of its potential role in privacy and authentication would be helpful to allow developers of VR training environments to make informed decisions on the collection of this data, which has already seen interest in its use for authentication and identification. With this chapter, we hope to investigate RQ4: How does identifiability of readily available VR tracking data change between tasks?

Authentication is the process of determining that someone is who they claim to be. For example, recent work by Kupin et al. [60] focuses on task-driven authentication by having users perform a task multiple times, creating a library of samples to authenticate against, then matching new samples from a user against their library. This process is similar to providing multiple samples of a fingerprint when setting up fingerprint authentication.

While authentication will generally have the user indicate who they are and provide some proof of that (e.g., a password, token, or biometric data), identification is the process of associating a person with an identity [6]. Identification can be a non-intrusive process performed by a system without awareness on the part of the user. It can both be done positively for the user's benefit (e.g., to passively determine an identity to authenticate against), or antagonistically, in the form of identifying a user without their consent.

Recent work on identification by Pfeuffer et al. [99] considered the problem of determining a user's identity over two sessions based on motion data. This work relied on recording the motion of participants' head-mounted displays (HMDs) and controllers through several iterations of a set of 4 controlled tasks, designed to represent generic VR tasks. The researchers then attempted to iden-

tify participants using the data collected from these tasks using a variety of different features for training their Machine Learning (ML) models [99]. Overall, Pfeuffer et al. [99] report accuracies around 40% when identifying users across sessions (N=22) via Random Forest (RF) classifiers.

In another recent work, Miller et al. [80] were able to identify participants (N=511) with accuracies up to 95% by training SVM, RF, and Gradient Boost Machine (GBM) models on HMD and controller data captured from participants observing 360° videos within a single session. Miller et al. [80] described two major limitations of their work in their paper. First, the VR experiences were primarily stationary with interactions largely limited to standing with little motion. Second, all the data for a participant was recorded during a single period of time without the user removing the VR head-mounted display (HMD), setting down the handheld controllers, or restarting the VR application. Too much homogeneity in the data collection process could increase the identifiability of individual users as it can limit the realism of the collected data. As an example, consider a facial recognition task in which the training data consists solely of images of faces of individuals under carefully controlled lighting, background, and orientation. Under such carefully controlled conditions, the task is much simpler than if the faces are in random orientations with a variety of backgrounds and lighting conditions that are more typical in real-world settings.

If we are to understand motion data from the user's HMD and controllers as uniquely identifying across a broad range of VR experiences, VR practitioners will need to take special considerations to maintain compliance with the European Union's General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA), even when the data is fully disassociated from other identifying information.

To address both of the prior limitations of Miller et al. [80], we follow the same approach with default hyperparameters for identification of VR participants using motion data that we previously captured from a two-session user study (N=60), but with two substantial differences to the overall

methodology. First, the VR experience in our user study involved training the participant how to troubleshoot a surgical robot, which involved numerous tasks with lots of motion, including moving to inspect equipment, taking tools from virtual teammates, and manipulating instruments. Hence, our motion data is not primarily stationary, addressing the first limitation of Miller et al. [80]. Second, our study included two sessions of motion data for each participant: the first session focused on learning how to troubleshoot, and the second session, which was conducted one week later, was focused on evaluating the participant's retention of the troubleshooting procedure. Therefore, our data spans multiple days, not just sessions, in which the participants doffed the VR hardware and the VR application was restarted, addressing the second limitation of the data analyzed by Miller et al. [80]. This approach also explores some of the potential improvements identified by Pfeuffer et al. [99], by considering feature vectors generated over time and identifications determined by aggregations over multiple samples. Finally, in contrast to their work, our work explores identification of users based on an ecologically valid VR training task, as opposed to a controlled interaction task.

After discussing related work, we first describe our two-session user study, which involved a learning session and then a retention session, one week later. We then present our implementation of the single-session approach and models presented by Miller et al. [80] on the motion data from our separate learning and retention sessions. On our learning session data, we found that two of the three ML models performed noticeably worse than the results reported by Miller et al. [80]. This is surprising considering that our data set is much smaller, i.e., random selection has higher probability of identifying the correct participant ( $1/60$ ) than the original study ( $1/511$ ). Hence, we believe this indicates that our VR experience is different in terms of motion than the experiences captured by Miller et al. [80]. On our retention session data, we found that only one of the three models performed worse, which indicates potential differences between the VR experiences of our two sessions.

Next, we applied the classifiers learned from each session to the other session to address the issue of multiple sessions and doffing the VR hardware. By applying the classifiers from our learning session to our retention session data, we found that the accuracy of all three models decreased by approximately 30-50%, to levels similar to those found by Pfeuffer et al. [99]. In this instance, our results are surprising now because compared to Pfeuffer et al. [99], we have a larger dataset, and thus a smaller random chance of success (1/60) than the original study (1/22). In applying the classifier from our retention session to our learning session data, we found that the accuracy of the models decreased by 40-65%. Hence, these results appear to indicate that identifying participants across multiple sessions is more challenging, though still possible in some cases.

Finally, we demonstrate how motion tracking data can be obfuscated to reduce identifiability by encoding positional data as velocity data, which we believe should reduce the encoding of anatomical features, such as height and arm-length. We believe that obfuscating data and reducing identifiability are important areas of future research, in order to protect VR users from being identified without their consent. Using the single-session approach on our learning session data, we found that encoding velocity data instead of positional data caused the accuracy of the models to decrease by 45-65%. Furthermore, when applied to the other session data, we found that the velocity representation caused accuracy to decrease by approximately 65-80%. Because prior work has successfully made use of similar velocity features to predict user experience outcomes like simulator sickness [96] and knowledge acquisition [86], we believe this approach to obfuscation could be a viable method for reducing the identifiability of VR user tracking data while retaining useful information for other ML applications.

To summarize, in this work we explore the following:

- Identifiability of participants within one session of a VR training application.
- Identifiability of participants from one session of a VR training application to another.



- Reduction in identifiability of participants by representing data as velocity rather than position.

## 5.2 VR Learning and Retention User Study

In order to investigate identifiability in an *ecologically valid* context, we used an existing VR application that was designed to train first assists how to troubleshoot a surgical robot in a robotic operating room (OR) [85]. This application makes use of virtual hand selections and manipulations [84, 78] to support multiple interactions with equipment, as well as selecting dialog options to communicate with a virtual surgeon and non-sterile staff member, in order to fix a faulted arm on the robot. In order to complete the scenario, the user must perform a variety of ordered steps involving these interactions and must additionally move about the virtual OR using real walking [62] and look at notifications and an endoscopic monitor displayed on a vision cart. Table 5.1 shows a list of subtasks and the types of interactions they require to complete.

During the learning session, this VR training application provided interaction cues, which convey actions to take [49], for each step that consisted of verbal instructions and visual animations showing perceived affordances and feedforward information. For travel cues, these animations consisted of semi-transparent green boots that linearly interpolated from the user's current position to a "Stand Here" icon representing the travel destination. Selection cues made use of semi-transparent green controller models that also continuously linearly interpolated from the user's controller to the target dialog option or object to be selected. Manipulation cues made use of semi-transparent green clones of the object, and linearly interpolated to the position they needed to go, in order for the user to progress (an example is visible in Figure 5.1). For the retention session, these cues were not provided until the user made an error or took no action for 30 seconds. In both sessions, errors and inaction caused a virtual agent to verbally instruct the participant what to do,

while the interaction cue for that step would enable.

### 5.2.1 Materials

This study used the HTC Vive system, which includes an HMD and two handheld controllers to interact with the VR training application. The Vive HMD has a 90Hz refresh rate, a display resolution of 1080x1200 pixels per eye, and affords a 110° diagonal field of view. The HMD

Table 5.1: The subtasks and their associated required interactions involved in our VR training simulation.

#	Subtask	Interaction
1	Check error message	Walk + Look
2	Consult Surgeon	Select (dialog)
3	Ask to press power down	Select (dialog)
4	Ask to press power up	Select (dialog)
5	Ask to call support	Select (dialog)
6	Ask for release wrench	Select (dialog)
7	Grab release wrench	Select (wrench)
8	Ask for emergency stop	Select (dialog)
9	Hold instrument carriage	Select (carriage)
10	Insert wrench	Position (wrench)
11	Rotate wrench	Rotate (wrench)
12	Check vision monitor	Look
13	Remove wrench	Position (wrench)
14	Remove instrument	Position (instrument)
15	Give instrument to staff	Position (instrument)
16	Ask to recover fault	Select (dialog)
17	Ask to disable arm	Select (dialog)
18	Check error message	Walk + Look
19	Use cannula lever	Select (lever)
20	Use instrument clutch	Position (clutch)
21	Use port clutch	Position (clutch)
22	Ask to confirm disable	Select (dialog)
23	Check error message	Look
24	Ask to press recover fault	Select (dialog)



Figure 5.1: A first-person perspective of the VR training application.

was fitted with the Vive audio strap that integrates over-the-ear headphones. The VR application was developed in Unity and maintained 90 frames per second to match the Vive's refresh rate. SteamVR was used to process the Vive's input data. The VR training application also collected HMD and controller tracking data every frame. This data consisted of the frame's timestamp and the positions and orientations of the HMD and controllers every frame.

### 5.2.2 Procedure

The following procedure was reviewed and approved by the University Institutional Review Board (IRB).

The study consisted of two sessions for each participant: the learning session and the retention session. The learning session lasted approximately 60 minutes. The retention session occurred one

week later and lasted approximately 30 minutes.

After informed consent, the learning session began with a background survey on the participant's demographics, education, and technology experience. The experimenter would then help the participant put on the HTC Vive and run the SteamVR tutorial to train the participant how to use the Vive. The participant would then experience the VR training application. After completing the training session, the participant was administered a number of questionnaires regarding their VR experience. Finally, participants were administered a knowledge test consisting of multiple-choice questions pertaining to the training scenario.

One week later (restricted to the same day of the week to avoid confounds), the retention session began with the experimenter helping the same participant put on the HTC Vive. The participant would then experience the VR retention application. After completing the retention session, the participant was given a free-response exit survey and compensated \$15 USD.

### 5.2.3 Participants

A total of 61 participants were recruited through university mailing lists and completed the initial training session. Due to a system issue, one participant was unable to complete the retention session, leaving 60 participants (11 females, 49 males). None of our participants had prior experience or knowledge of surgical robots. The mean age of the participants was  $22.6 \pm 4.2$  years old.

## 5.3 Identifying Participants in the Same Session

In this section, our aim is to fit machine learning models to identify participants based on their motion data. *Can we identify participants based on tracking data taken during a VR training*

*scenario?*

For purposes of comparison, we followed the basic approach described by Miller et al. [80] in terms of feature representation and general ML approaches (some important deviations will be discussed later). The motion data from our VR training system contained 6-degree-of-freedom (6-DOF) data from both of the controllers and the HMD for 18 total values ( $\{x, y, z, \text{roll}, \text{pitch}, \text{yaw}\} \times 3$  tracked objects) at 90 frames per second. This data was then combined into a descriptive feature vector for each second, consisting of the minimum, maximum, mean, median, and standard deviation (i.e., 5 metrics) for each of the 18 values, resulting in a 90-valued feature vector for each second,  $\vec{v} \in \mathbb{R}^{90}$ . These sets of feature vectors were then partitioned into 10 sequential subsessions for each participant.

Again, in keeping with Miller et al. [80], we trained three different ML models: k-Nearest-Neighbors (kNN), random forests (RF), and gradient boosting machines (GBM). The kNN classifier treats all training vectors as points in a high dimensional space (in our case,  $\mathbb{R}^{90}$ ), and when asked to predict the category of a new vector, finds the  $k$  nearest training vectors (usually by Euclidian distance), and predicts whichever category occurs most frequently among these neighbors. Note that for kNN, we do normalize these vectors to  $M = 0, SD = 1$  to prevent the scale of input along different axes from affecting the prediction, a known limitation of kNN. RF was implemented with scikit-learn defaults, which are the same as R defaults, but with 100 estimators rather than 10 [98]. GBM was also implemented with scikit-learn defaults: 100 estimators, a maximum depth of 3, and a minimum of 1 observation per leaf. These hyperparameters match those of Miller et al. [80], with the exception of the number of estimators for GBM, for which they used 20. While training kNN simply involves embedding the training vectors and their labels into a high dimensional space, random forest and GBM are trained by building ensembles of decision trees that try to map vectors to labels. After being trained, all three of these classifiers attempt to solve our multiclass classification problem – they take a single feature vector as input, and return a prediction as to

which participant generated that feature vector.

We applied 20-fold leave-one-out cross-validation to compare the performance of the three approaches: we retained one subsession per participant for evaluating the performance of the classifier trained on the other 9 subsessions, and we performed this training and testing process with 20 different random Monte-Carlo selections. While the classifiers provided a prediction for each feature vector (i.e. one prediction per second of data), we used the plurality of these predictions at the subsession level, saying each classifier ultimately identified the most commonly predicted identity for that subsession. Each training and validation process yields a percentage of correct identifications out of all 60 participants, and we averaged this value over all 20 cross-validations to get a mean accuracy rating for each classifier. We found that GBM performed the best with an average accuracy of 90.83% (fig. 5.2). Our GBM accuracy was substantially higher than that reported by Miller et al. [80], which averaged 68.2%. This is most likely due to our much smaller sample size (N=60), compared to the original study's sample size (N=511), which increases the probability of correct identifications, as well as our larger number of estimators, made possible by our smaller sample size. However, our kNN and RF classifiers performed notably worse than those of Miller et al. [80]. Specifically, our kNN averaged 80.42% accuracy compared to their kNN's average accuracy of 92.3%, and our RF averaged 89.33% compared to their RF average accuracy of 95.3%. We believe these results indicate that it is more difficult to identify participants from our VR training application than the 360° video and survey experiences used by Miller et al. [80], due to differences in the motion data.

For completeness, we also trained and evaluated the three ML models on the retention session data. We began by dividing the retention session data from the participants into 10 equal subsessions, as we had done for the learning session models. We then analyzed the performance of the three models using the same 20-fold leave-one-out Monte-Carlo analyses as above. These results based on the retention session motion data perform better than those based on the learning session motion

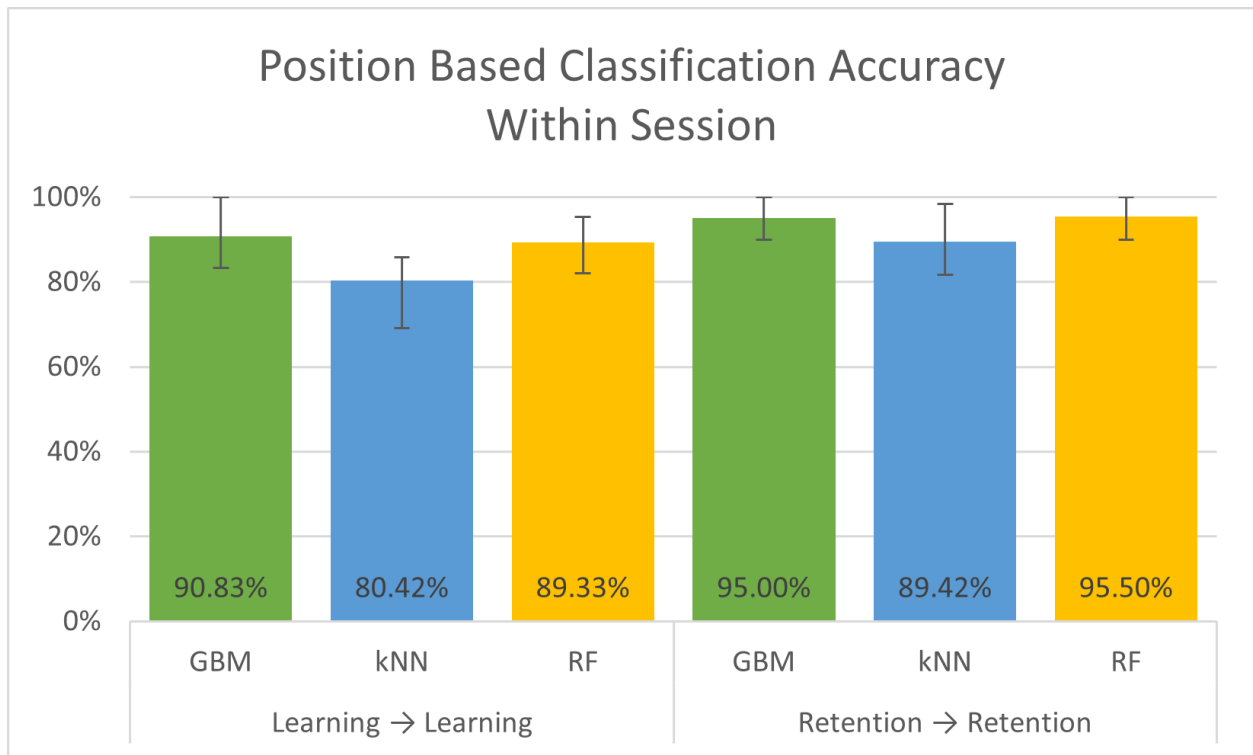


Figure 5.2: Classification accuracy using position feature vectors, averaged over 20 Monte-Carlo cross-validations within session data for each participant. Error bars indicate range of values.

data, as shown in Figure 5.2. Compared to Miller et al. [80], our GBM and RF results for our retention session data are better (GBM: 95.00% vs. 68.2%, RF: 95.50% vs. 95.3%). Again, this is likely due to our much smaller sample size and larger number of estimators. However, our kNN results were again worse than those reported by Miller et al. [80] (kNN: 89.42% vs. 92.3%).

#### 5.4 Identifying Participants in a Different Session

In this section, our goal is to use ML models trained on the motion data in one session to identify participants with motion data from another session. *Can we identify participants in a session that takes place at a different time from the session used to train the classifiers?*

Inherently, the problem presented in the previous section should be a relatively easy identification task. In our study, participants did not doff the HMD between subsessions, so the fit should have remained roughly consistent. Additionally, 8 of the 10 subsessions, when selected for the Monte-Carlo cross-validation, would have been chronologically bound on both ends by subsessions that the classifier was directly trained upon, leading us to expect several feature vectors on either end of a subsession being easy to identify. Showing how well participants can be identified one week later (or earlier), when a multitude of factors affecting the physiology of a participant, as well as the fit of the HMD and grip on the controllers, may be different, can give us insight into how much we should consider VR motion data to be personally identifiable information.

To address this issue, we first used **all of the data from the learning session** to train our three classifiers. For our analysis, we attempted to identify within our 10 equal subsessions of the retention session and averaged the classification accuracy across all 10, which yielded worse performance for all three classifiers (see Figure 5.3). We found that the accuracy of each classifier decreased by around 50%, compared to their learning session average accuracy. More specifically, GBM decreased from 90.83% to 41.50%, kNN decreased from 80.42% to 28.00%, and RF decreased from 89.83% to 42.33%. Out of curiosity, we also looked at the accuracy of the classifiers when being used to predict identity for the full retention session (as opposed to one subsession) and found that with more data, our classifiers were able to eventually attain accuracies closer to 50%. Figure 5.4 shows the percent of correct identifications each classifier achieves with more data and suggests that adding even more data would not result in further increases in accuracy. The final values of the classifiers in this condition were 48.33% for GBM, 46.67% for kNN, and 51.67% for RF.

For completeness, we also used **all of the data from the retention session** to train our three classifiers and attempted to identify participants from the data from each of the 10 equal subsessions of the learning session data. Averaging the accuracy of these again, we find substantially worse performance for all three classifiers when compared to their performance for the within retention-



session data (see Figure 5.3). Each classifier in this scenario decreased in their average accuracy by over 50%, specifically GBM decreased from 95.00% to 37.67%, kNN decreased from 89.42% to 24.83%, and RF decreased from 95.50% to 25.17%.

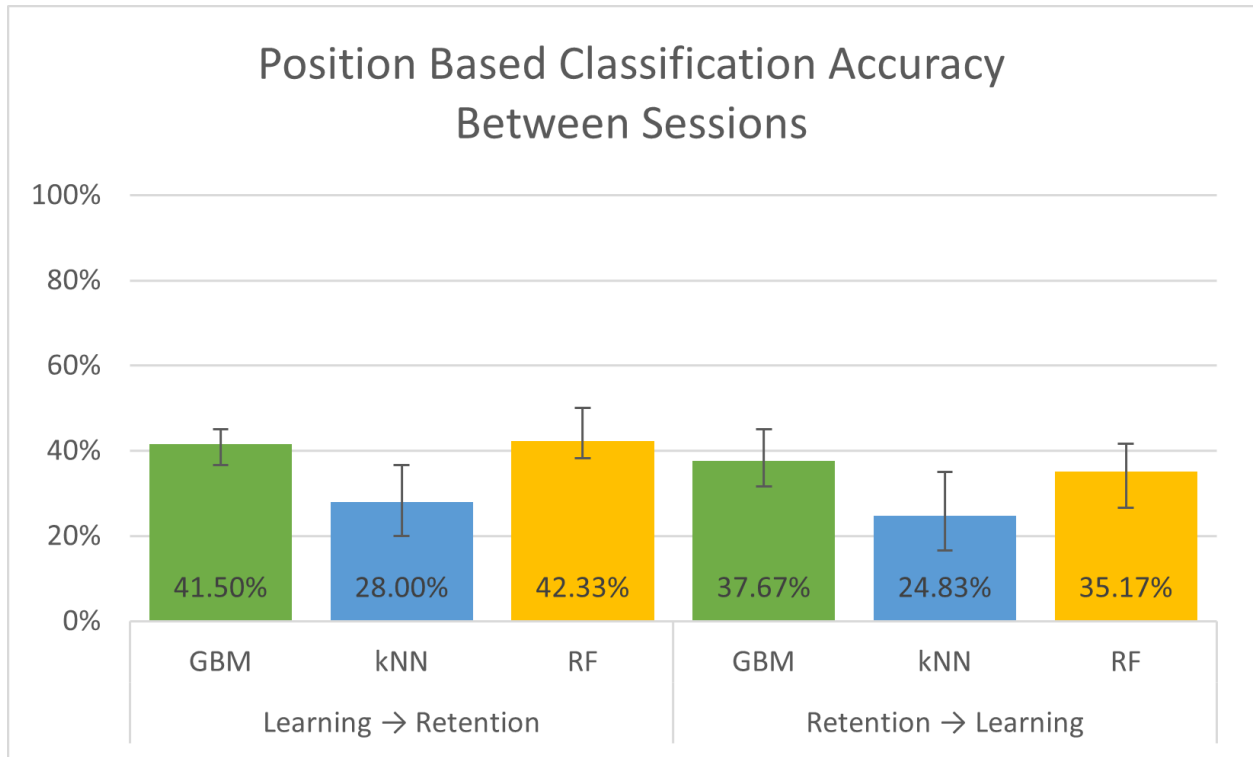


Figure 5.3: Classification accuracy using position feature vectors, averaged over 10 subsessions for each participant, across sessions. Error bars indicate range of values.

### 5.5 Obfuscating Identities With Velocity-based Data

The results in the previous section show that the ML algorithms, if provided with several minutes of data, can achieve approximately 50% accuracy in identifying participants between different sessions. As researchers, this brings up the question of how we can share data to improve the field of VR and ML without potentially compromising the identities of participants. *Can we reduce the*

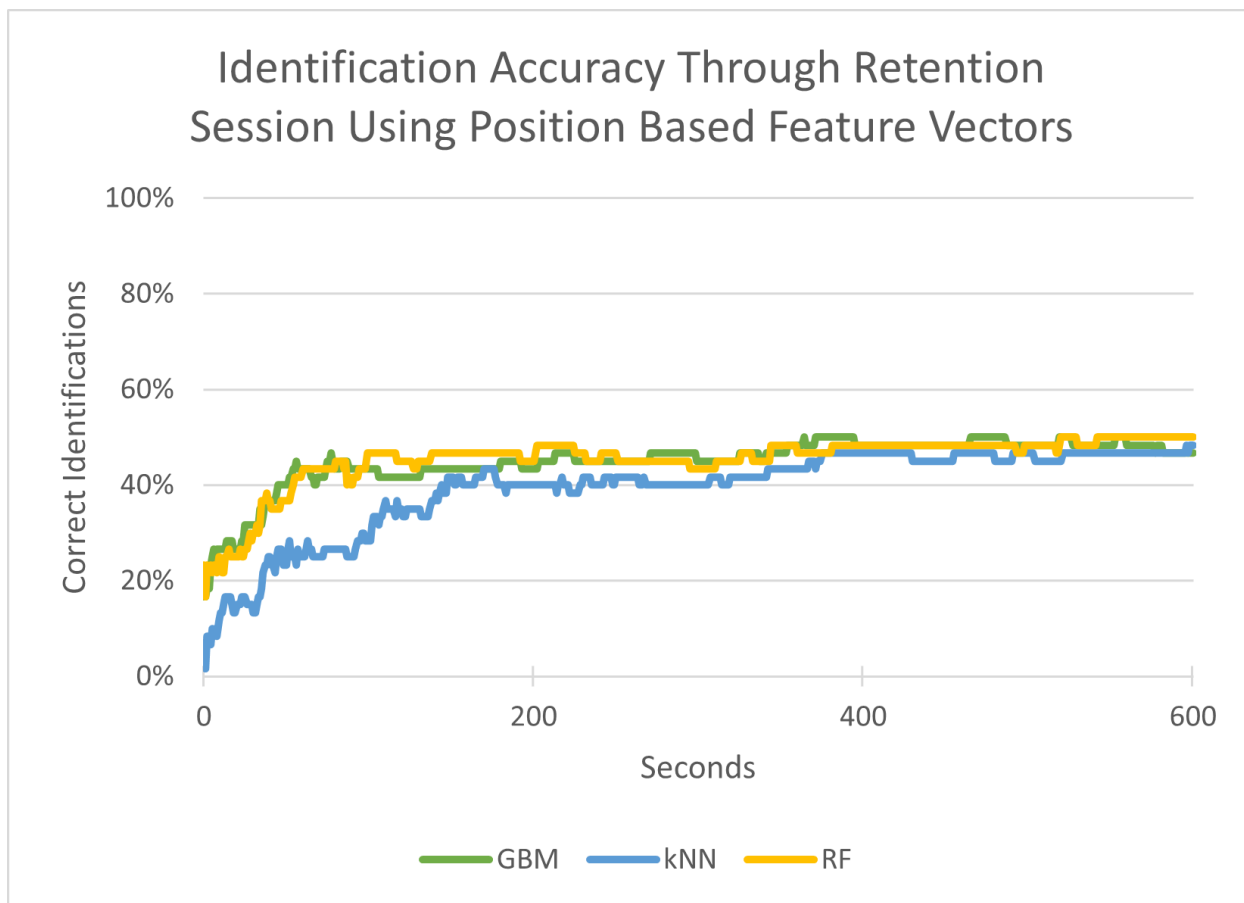


Figure 5.4: Correct identifications for each additional second of position-based features for the first 10 minutes of data per participant.

*identifiability of motion data while still retaining usefulness for research?*

The way in which the data is featurized could have a significant impact on identifiability. For example, positional data from the VR hardware can encode the height and reach of each participant. More importantly, data is often featurized in a specific way in order to make it useful for a particular task. In this sense, the featurization used by Miller et al. [80] and in the experiments above may not be realistic for ML tasks of interest beyond authentication and identification. And, of course, different featurizations may be more or less difficult to de-anonymize.

To look at the ability to de-anonymize data that has been encoded for predicting other factors of the participants, we considered the prior work of Moore et al. [86], which makes use of velocity-based feature vectors to predict participant knowledge gains in a training scenario. We choose to look at the capabilities of the same classifiers presented in the prior two sections when trained and tested on feature vectors based on velocity rather than position. Each positional data value ( $6\text{-DOF} \times \{\text{HMD, left controller, right controller}\}$ ) was interpolated to a velocity value based on each frame's timestamp. We then made similar feature vectors consisting of the min, max, median, mean, and standard deviation of each value, for a new 90-valued vector  $\vec{v} \in \mathbb{R}^{90}$ .

We then re-conducted the 20-fold leave-one-out Monte-Carlo cross-validation procedure, trained on 9 of the 10 learning subsessions per participant, to determine to what degree of accuracy these classifiers are able to predict the removed subsessions. Our data shows that these classifiers had much more difficulty identifying the participants in this scenario, with the best performing classifier, kNN, reaching only an average accuracy of 35.17% (see Figure 5.5).

For completeness, we also chose to inspect the ability of these classifiers to predict the identity of participants during the retention session. Again, we trained the classifiers on all of the learning session data, and analyzed the prediction accuracy across the 10 retention subsessions for each participant and we found even worse performance. Compared to our position learning-session to retention-session condition, we found each velocity-based classifier to be less than half as accurate. Specifically the decreases were 41.50% to 11.83% for GBM, 28.00% to 12.33% for kNN, and 42.33% to 13.83% for RF (see Figure 5.5). Again we were curious as to how additional data affected the accuracy, so we examined how well the classifiers could identify participants using the entire retention session's data (as opposed to one subsession). Figure 5.6 shows the percent of correct identifications each classifier achieves with more data and again suggests that more data would not result in increases in accuracy. The final values of the classifiers in this condition were 16.67% for GBM, 15.00% for kNN, and 18.33% for RF.

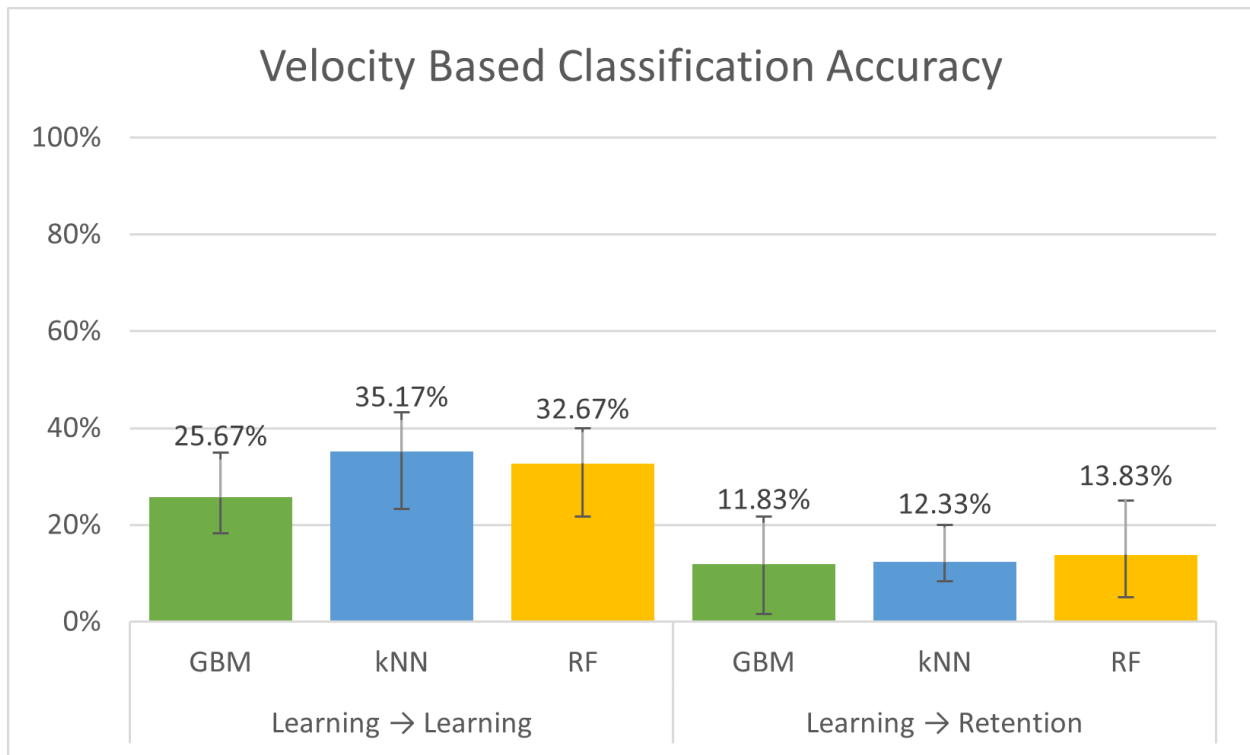


Figure 5.5: Classification accuracy using velocity feature vectors, averaged over 20 Monte-Carlo cross-validations for each participant. Error bars indicate range of values.

## 5.6 Discussion

In this section, we discuss the implications of our within-session identification, between-session identification, and velocity-based obfuscation results. We also discuss sharing our resources for future replication studies, the limitations of our current work, and our plans for future work.

### 5.6.1 Validation of Within-Session Identification

In contrast to prior work by Pfeuffer et al. [99], we investigated identification of users based on an ecologically valid VR training task. In section 5.3, we showed that our classifiers were

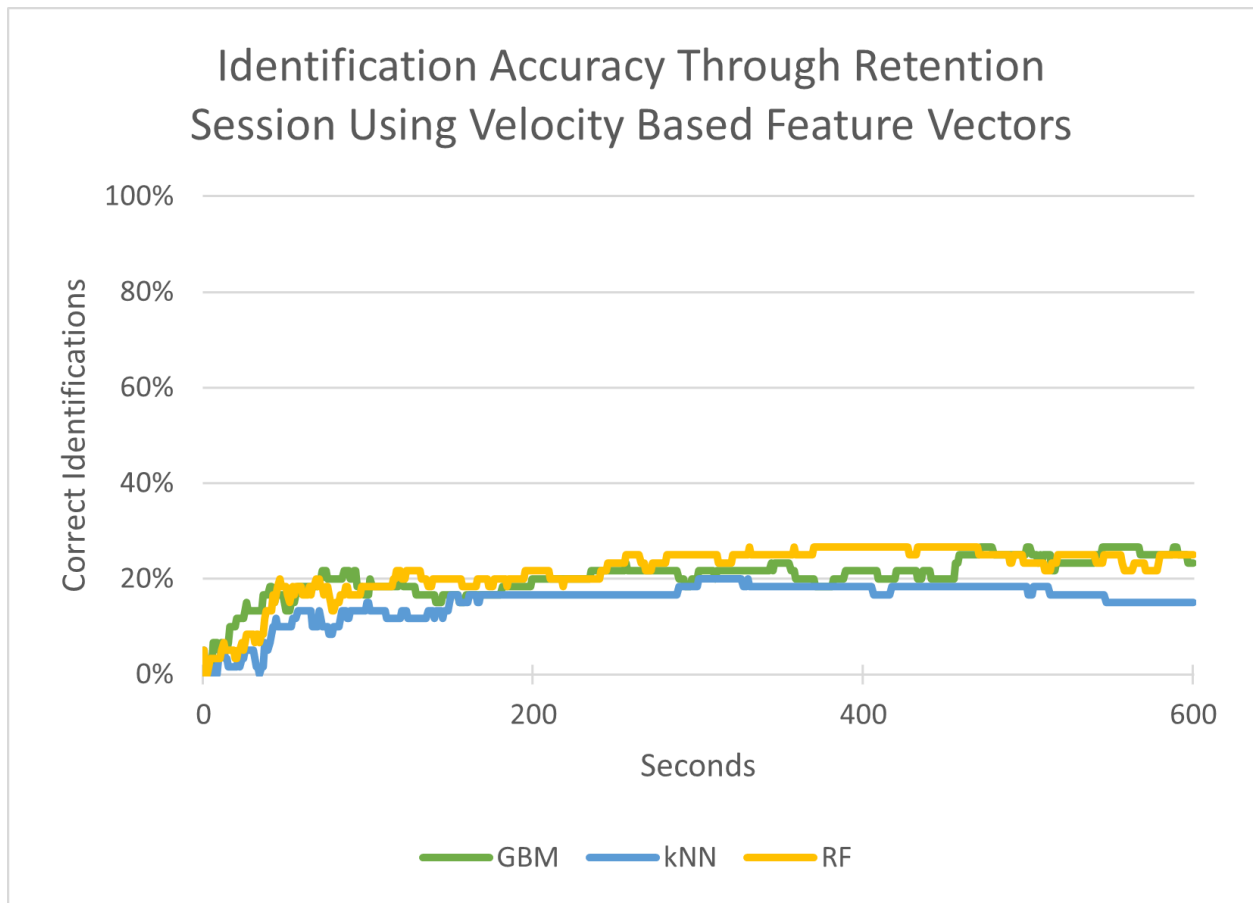


Figure 5.6: Correct identifications for each additional second of velocity-based features for the first 10 minutes of data per participant.

able to achieve fairly accurate identification of users. These results are much higher than chance (1/60), however, when compared to the results and chance (1/511) of Miller et al. [80], our results were underwhelming. In their paper, they also present the results of their classification procedure on a sample of 50 randomly selected participants, achieving 98% accuracy, demonstrating the improvement in accuracy one might expect with a smaller sample size [80].

We believe that the primary contributor of this disparity is the large difference in VR experiences presented to the users. While our experiment made use of a VR training scenario with several virtual

object manipulations that required movement around the virtual space, the environment presented in the Miller et al. [80] consisted of five 360° videos and five in-VR questionnaires, presented one after each video. Indeed, the authors stated that “a limitation of this work is that both tasks captured standing users with little motion. . . the generalizability beyond the tasks demonstrated here should be tested.”[80]. Our results seem to indicate that while it’s still quite possible to identify a user in a scenario with more movement, the classifiers seem to perform less well.

### 5.6.2 *Feasibility of Between-Session Identification*

In contrast to Miller et al. [80], we investigated identification of users across multiple sessions. In section 5.4, we showed that the same classifiers that performed well within the learning-session data performed worse when evaluated on the retention session data, but still performed much better than chance. Three likely contributing factors to this change in performance are differences between the VR learning and retention experiences, potential physiological differences for the participant after a week, and changes in the donned positions of the VR hardware. While the virtual robotic OR application required users to complete the same tasks in both the learning and the retention sessions, the learning session provided unprompted interaction cues, while the retention session only displayed these after the user made an error or was inactive for 30 seconds. We believe that this caused user behaviors and motions to be subtly different, as they still had to complete the same tasks and interact with the same objects that were located in the same positions. This as a contributing factor is supported by observed performance differences between the same-session classifiers for the training and retention sessions.

We also think that physiological differences contributed to the observed change in classifier performance. After a week, participants may be in a different physiological, emotional, and mental state. Finally, the fit of the HMD on their head and the grip with which they held the controllers may be

significantly different too. Miller et al. [80] identified that one limitation of their study was that participants did not doff the headset at any time between sessions. We think that our results help show, again, that participant identification is certainly still possible, but once again, a little more difficult – likely because the ML models are somewhat overfit to differences between participants that are not consistent across different days.

While not a perfect comparison, we choose to examine our results with respect to recent work by Pfeuffer et al. [99] that also looked at identification across sessions. We find that our classification accuracy is roughly on par with their results despite having a lower random chance of success (1/60 for ours compared to 1/22 for theirs). This should not necessarily be interpreted as indicating that identification across sessions is easier within an ecologically valid context than with an artificial context as there are a large number of factors that differentiate our approach with theirs, including that our vectors are determined at the per-second level, as opposed to on a per-action level, and we only have one example of each subtask per participant embedded in our model. Finally, we are also using the plurality of predictions to determine identity, as Miller et al. [80] and Liebers et al. [67] did, and as Pfeuffer et al. [99] recommended to potentially improve accuracy. All of these differences likely contributed to the improved performance of our classifiers in light of Pfeuffer et al.’s [99] results.

### 5.6.3 *Feasibility of Velocity-based Obfuscation*

Finally, in section 5.5, we looked at the possibility of obfuscating motion data as a series of featurized summary vectors of linear and angular velocities rather than position and orientation to attempt to reduce identifiability, while maintaining usefulness for VR researchers. This direction was chosen based on prior work by Moore et al. [86] that indicated procedural knowledge transfer can be predicted using velocity-based features. We similarly computed velocity feature vectors that

matched our position-based feature vectors, but made use of the first-order derivative with respect to time. We found this approach yielded much lower identification accuracy for our within-session condition, e.g. GBM, our highest performing classifier for the positional representation at 90.83% accuracy, degraded to 25.67% accuracy when given velocity-based data. We observed a similar decrease in accuracy for our between-session condition, wherein RF decreased from 42.33% to 13.83%.

Ultimately, we determined that the accuracy of participant identification remains better than chance and may still contribute to identification, but is much more difficult for the classifiers we examined, particularly in our most difficult condition, identifying participants during a different session from velocity-based vectors. Looking at the comparisons between Figures 5.4 and 5.6, it appears as though the classifiers quickly approached their approximate final accuracy rate when given position-based feature vectors, while the velocity-based classifiers needed more data before they reached their peaks, before dropping to a lower final accuracy.

Interestingly, our presumption is that because the classifiers were less able to embed anatomical data, such as participant height or arm-length, they performed worse. This stands in contrast to recent work by Liebers et al. [67], in which they found that normalizing participants' height led to higher identification accuracy, saying that "instead of recognizing different users by their physiology... we force it to focus on the subtle changes in behavior between participants." We believe that this disparity is due in part to the difference in the design of tasks: where our data was captured passively during a training scenario, their participants performed several repetitions of sports actions, which would likely yield more identifying movements.



#### 5.6.4 *Sharing Resources for Replication*

The code we use for our ML approach is implemented in Python with the Scikit-Learn library [98] and is available on GitHub<sup>1</sup>. Due to the wording of our IRB-approved informed consent, we are unable to share participant data. However, we are working with our IRB to develop an informed consent form that will allow us to share VR tracking data like this in the future.

#### 5.6.5 *Limitations*

We make two observations that limit the generalizability of this work, as well as that of Miller et al. [80]. First, in both this work and that of Miller et al. [80], cross-validation should have been used to determine the hyperparameters of the different models (e.g., the  $k$  in kNN, the number and depth of trees in the RF algorithm). In particular, the decision to discretize the features over one-second intervals is a bit arbitrary, and a better value may be able to be found with proper cross-validation. Note that in both this work and in Miller et al. [80], the ML models are quite limited in that they do not take advantage of the fact that this data is actually temporal (i.e., each one-second data point is effectively classified in isolation). As a result, the scale at which the data is characterized can have a significant impact on predictive performance of the models. Similarly, in this work, we divided the motion data of each participant into ten equal size pieces—a somewhat arbitrary slicing of their motion data.

Second, we note that, if our aim were to actually identify participants from motion data in practice, we would use cross-validation to select the best model/hyperparameters on, say, the learning data set, train a model on the full learning data set using the selected model/hyperparameters, and then apply this model to a new unseen data set, say, the retention data set. That is, care should be taken

---

<sup>1</sup> [github.com/tapiralec/Identifiability-and-Obfuscation-VR-Training](https://github.com/tapiralec/Identifiability-and-Obfuscation-VR-Training)

not to use performance on the test set to draw conclusions about the best model to pick. As an example, we note that when training on the “survey” data and then testing on the “video” data, Miller et al. [80] remark that the model is still accurate. However, this observation overfits the test set. An improved procedure that limits this overfitting would be to train on “survey”, note that kNN had the highest accuracy in the 20-fold cross validation procedure, and then apply a kNN model trained on the whole of “survey” to the “video” data set. This is important because, in practice, if we were trying to de-anonymize the data, we would not have the labels on the test data set to know which model we should have chosen and can only rely on performance on data used to train the models to select between them. Furthermore, one set of hyperparameters we would evaluate over would be the duration of time used for the segments, as a real-time classifier would not know the total session duration and thus would be unable to partition based on the total session.

Finally, it is worth mentioning that although we found our models to have degraded performance when attempting to identify across sessions, there are other possible confounding variables that can have contributed to this decrease as well. Notably, while participants were required to perform the same ordered set of twenty 3D interaction tasks in the same space for both sessions, the specific presentation of manipulation cues varied in order to maintain the ecological validity of a training and evaluation scenario. Even without this as a confound, however, learning effects would also contribute to different behaviors between sessions.

## 5.7 Conclusion

In this chapter, we investigated the feasibility of implementing the approach presented by Miller et al. [80] to identify users in a VR scenario that involved more than stationary users and little motion, and attempted to identify them between sessions, one week later. We ultimately determined that while these classifiers performed much better than random chance at correctly identifying par-

ticipants, their performance is worse than the prior work's results. While there are several factors contributing to this discrepancy, the identifiability of participants through their motion data appears to be dependant on the underlying VR experience and the representation of the data. We also showed that identification between sessions is a more difficult task, but that rates above random chance are still achievable. Unlike prior work by Pfeuffer et al. [99], which also explored identification between sessions, we show that identification is possible even with data collected passively in the background of a VR training scenario, without a dedicated enrollment phase to build a library of known actions. The next chapter continues this research with an investigation into how this identifiability changes between two similar, but distinct tasks.

**CHAPTER 6: A SYSTEMATIC INVESTIGATION OF DEVICE  
COMBINATIONS AND SPATIAL REPRESENTATIONS FOR  
IDENTIFYING VIRTUAL REALITY USERS IN AN ASSEMBLY  
TRAINING ENVIRONMENT**

*NOTE:* This chapter contains material that has been modified from a paper submitted for publication, and has not yet been peer reviewed. Authors: Alec G. Moore, Tiffany D. Do, Nicholas Ruozzi, Ryan P. McMahan.

## 6.1 Introduction

Continuing our investigation into identifiability, we choose to now focus on identifying people between similar, but distinct tasks, trying to improve our understanding of RQ4: How does identifiability of readily available VR tracking data change between tasks? There have been several works recently that investigate the identifiability of virtual reality (VR) users based on properties of their biomarkers. Some works, such as Pfeuffer et al. [99], focus on an authentication model, where the user proposes their identity, and the system verifies that they are indeed that person. Other works, like those by Miller et al. [80] and Moore et al. [88] look at the question of passive identifiability. That is, given a person's biomarkers data in a population, can they be identified afterwards from a new sample of their VR usage data.

While the question of identifiability has been explored in several contexts, recent work by Liebers et al. [67], examines identifiability within a gameified context. Other environments, such as the one used by Asish et al. [3], are built as educational experiences, while some are developed specifically to elicit identifiable motions like one used in a paper by Liebers et al. [68]. In work by Mustafa et al. [91], the authors recognize that their approaches may yield different results when applied to a different context.

We expand upon a shortcoming of prior works by providing an analysis into the identifiability of users across two separate VR training sessions, where both samples are captured within a single hour span of time. Prior work by Miller et al.[80] found very high identification rates, they were between single-session samples of data. The work by Moore et al.[87], on the other hand, showed lower identification rates, but between sessions collected a week apart, introducing potential confounds due to the participants' mental state, clothes, physical health, and other potential changes. By collecting data within the same period of time, but separate VR sessions, we can isolate and begin to understand to what degree identifiability diminishes as a result of exiting and reentering

VR.

In this work, we choose a simple training task as our ecologically valid environment since this domain has seen expanded use as of late [124]. We developed a virtual environment to train users how to construct simple objects out of a set of toy pipes and connectors, allowing us to emulate a simple assembly task. By using a prescribed set of instructions that the user had to replicate, we reduce the likelihood that our within-session models overfit to features of the participant's experience, because we know that all participants completed the same steps in the same order for the same tasks.

Using this training task, we conducted a study with 45 participants, yielding 1.7 million frames of data over more than 5 hours. As far as research on identifiability with VR experiences this amounts to less than the 60 participants of data Moore et al. [87] had, and far fewer than the 511 participants of the work by Miller et al. [80]. Unlike those works, however, our informed consent allows us to publish this dataset to make it openly available for future research.

Finally, we conduct 4 machine learning experiments to examine the identifiability of this data, moderating the inclusion of data, its representation and the models, yielding results for 1176 total conditions. We choose to examine only the machine learning models Random Forest (RF), Gradient Boosting Machine (GBM), and k-nearest neighbors (kNN) rather than deep learning models due to their reduced likelihood of overfitting. In our best-performing within-session RF model, we find over 95% accuracy, but that same condition drops in accuracy to around 46% when trained on one session and evaluated on another. We also examine a set of featurizations based on the 1st order time derivative of the data and find it to perform moderately well, but with a different set of data inclusion.

In this chapter, we will describe the development of the virtual environment and the experimental methodologies, discuss our findings, and finally conclude with the limitations of this work and

some future directions. The primary contributions presented herein are:

1. An analysis of identifiability between VR sessions within a short span of time.
2. An exhaustive exploration of data inclusion and representation conditions and how they affect identifiability.
3. We provide a novel dataset featuring high-framerate capture of the tracked devices across two VR sessions for 45 participants. A dataset containing all data used in this chapter will be made available at GitHub:

## 6.2 Full-scale Assembly Benchmark

There were a few motivating factors for developing a new testbed for this research. Primarily, we wanted to create a platform that was both usable and reproducible. We also wanted to investigate assembly, since it enables us to maintain a consistent context while modifying the specifics of the task that the participant will have to undergo. Another benefit of using an assembly training testbed is that we can moderate the duration of the experience by changing the complexity of the object that the participant has to build.

We are making use of a relatively inexpensive and readily available tube and connector toy as our real-world task. We decided on this specific type of toy because the individual pieces are large enough to elicit the gross motor movements that VR easily affords, and the individual pieces are relatively simple objects and are easy to model with sufficient realism (Figure 6.1). To ensure accurate reproduction, the toys were measured with digital calipers to measure the thickness and positioning of each physical feature. I then recreated these using Blender, ignoring minor features such as injection mold flashing. Finally the Unity standard shader was applied to the models and values were tuned to achieve a similar visual appearance to their real-world counterparts.



Figure 6.1: Two of the types of connectors in the toy set and their corresponding models.

In parallel with modelling, I also developed an assembly system that allows the individual pieces to be attached to each other, as well as read, write, and modify sets of assembly instructions. This system was then used in the editor to design some models and create associated sets of instructions. At runtime, users could use virtual hand enhanced with VOTE [83] to select and manipulate the virtual toy pieces. Because all of the pieces were inherently symmetrical about at least one axis, care was taken to ensure that the system would recognize connections that were visually identical as being the same. Generally, each step consisted of attaching a new piece to the current work in progress, adding a screw to that connection, then using the key to turn the screw 90°, as seen in Figure 6.2.

While the environment was structured in a way that required the users to "turn" each screw, this interaction was identified as having a disproportionately high level of dexterity needed for realism, with interactions occurring at a scale afforded poorly by our hardware's lack of finger-tracking. Thus, participants were required only to use the key to touch the head of the screw, at which time, the system would animate the key turning the screw, then return the key to the participant's hand<sup>1</sup>.

---

<sup>1</sup>This design decision was met with mixed opinions from our participants as reflected in free-response questions that did not prime discussion of the screw-turning behavior. Some participants specifically appreciated that it was easier than the real-world, others disliked the suspension of realism compared to the rest of the environment.



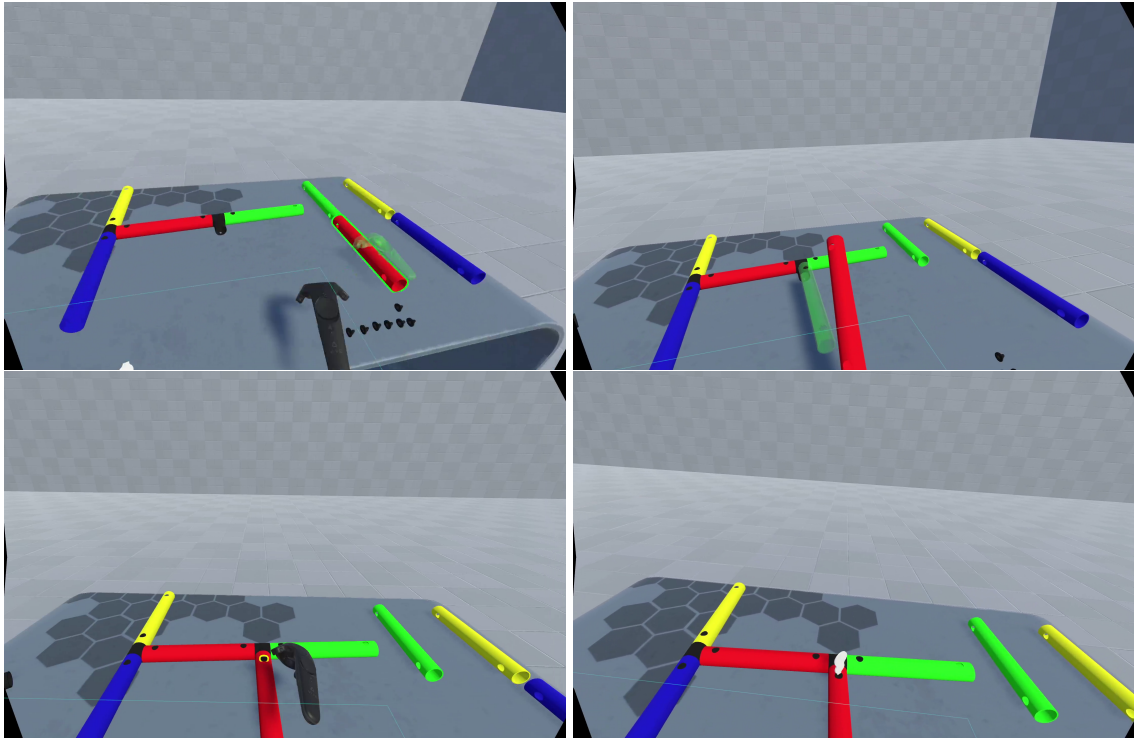


Figure 6.2: Attaching a tube in the Full-scale Assembly Benchmark. From top left, grabbing a pipe, attaching it to the work in progress, inserting a screw, and turning the screw with the key.

When in use, the system would iterate through the instruction steps, presenting a set of interaction cues [49] (consisting of animated transparent copies of objects to convey their intended motion as well as static outlines to indicate objects to interact with and arrows to indicate screws to turn) appropriately for each step. Additionally, as a form of feedback, a "pop" noise was added to audibly confirm when pieces were attached together.

Finally, similarly to the Robotic Operating Room environment, the system was programmed to record several features of movement including the positions, orientations, and button states of each tracked object in the scene at 90Hz, when piece-attach events occurred, a first-person perspective video recording of the activity, what objects were being observed and hovered over by the hands, and times when the HMD was removed.

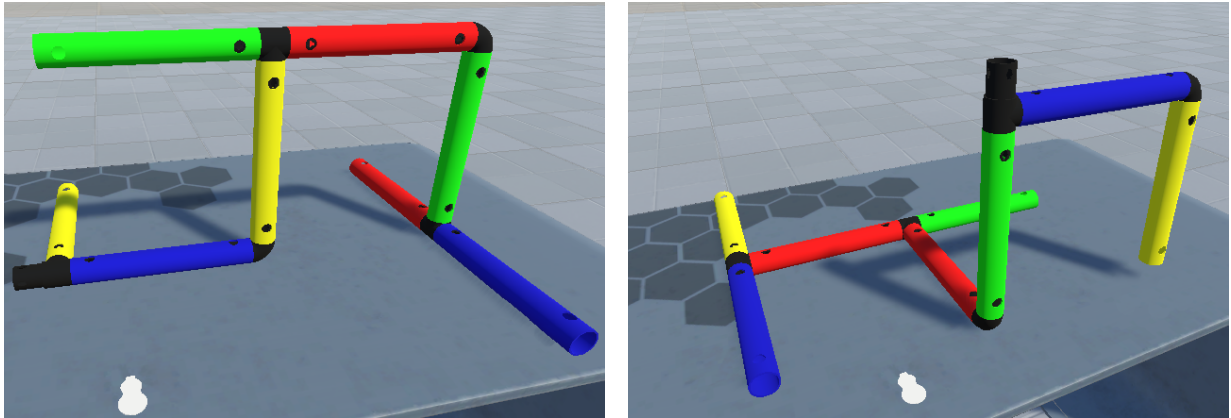


Figure 6.3: The completed structures. Build A is shown on the left, and build B on the right. Screws do not appear as an artifact of the way the screenshots were captured, but were visible in the application.

### 6.2.1 Experimental Design

With the newly developed Full-scale Assembly Benchmark, we prepared two structures for participants to build: build A and build B (see Figure 6.3). For both, we avoided having obvious semantic meanings such as a house shaped structure. Both puzzles consisted of the same pieces: 3 “T Connectors,” 2 “L Connectors,” and two of each Red, Green, Yellow, and Blue pipes, as well as 12 screws, exactly the number of each part needed for all of the connections in both puzzles. At the beginning of the each experience, these pieces were laid out on the virtual table identically, as shown in Figure 6.4. The full set of instructions for each build is shown in Table 6.1.

In the study, participants were presented first with a simplified VR onboarding task that prepared them to complete the VR training scenario (see Figure 6.5). For this tutorial task, participants were placed into a simplified scene that required only two connections to be made. In addition to the interaction cues used throughout, the tutorial experience was annotated with both verbal and text instructions providing guidance on the actions that needed to be taken in VR as well as the physical



Figure 6.4: Both builds made use of the same pieces, in the same locations. They consisted of two of each color pipe (red, green, blue, yellow), two elbow connectors, three three-way connectors, and 12 screws. Also visible in the middle is the metallic key used to turn the screws to secure connections.

Table 6.1: The order of steps for builds A and B. The letters R, G, B, and Y represent red, green, blue, and yellow pipes respectively, T and L represent Three-way and Elbow joints, respectively.

Build	1	2	3	4	5	6	7	8	9	10	11	12
<b>A</b>	Y1	B1	L1	Y2	T2	G1	R1	L2	G2	T3	R2	B2
	T1	T1	B1	L1	Y2	T2	T2	R1	L2	G2	T3	T3
<b>B</b>	Y1	B1	R1	T2	G1	R2	L1	G2	T3	B2	L2	Y2
	T1	T1	T1	R1	T2	T2	R2	L1	G2	T3	B2	L2

actions needed to elicit them. This allowed the experimenter to have to only assist the participant with making the ergonomic adjustments needed to don the HMD and hold the controllers correctly.

After completing the tutorial, participants had to do the VR training environments for the A and B builds, which were presented in a counterbalanced manner. After completing each training environment, participants attempted to recreate the structure they just built in VR, but now with the physical toys. We made use of colored tape to mark the starting positions of the real-world toys so that they would closely match the starting positions of their VR counterparts. See Figure 6.6 for

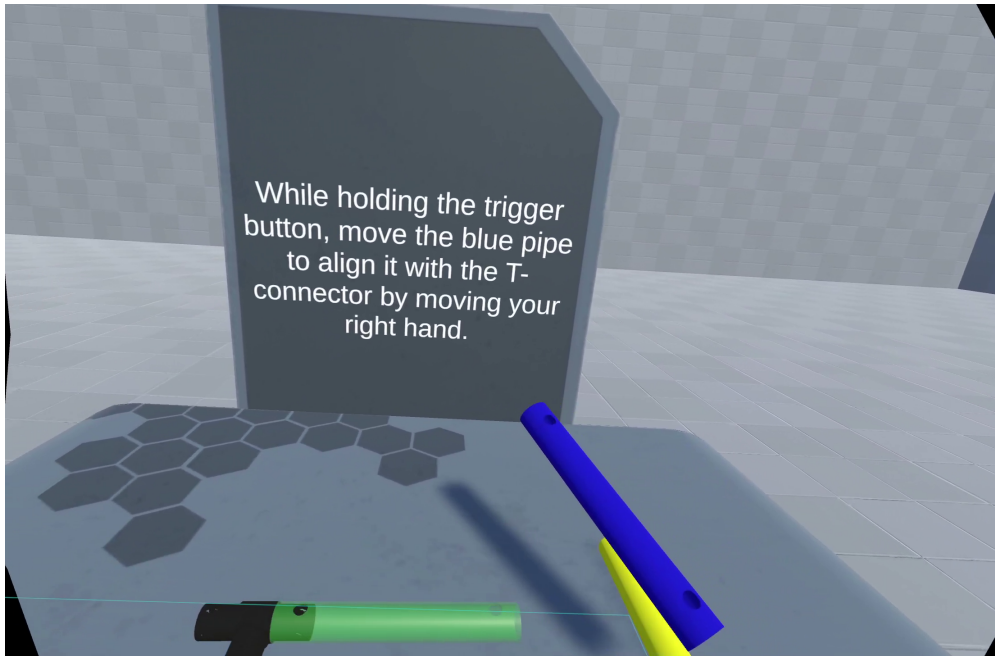


Figure 6.5: A still of the participant's perspective of the VR onboarding task.

an overview of the participant flow through the study.

### 6.2.2 Procedure

The following procedure was approved by the University of Central Florida Institutional Review Board.

A link to a pre-survey was made available through university mailing lists. This pre-survey first ensured that people responding to our survey were eligible according to our inclusion criteria. They were then asked for their consent and, if granted, were administered a demographics survey. Finally, the pre-survey automatically helped the participant schedule a 1-hour period of time to participate in the in-person portion of the study.

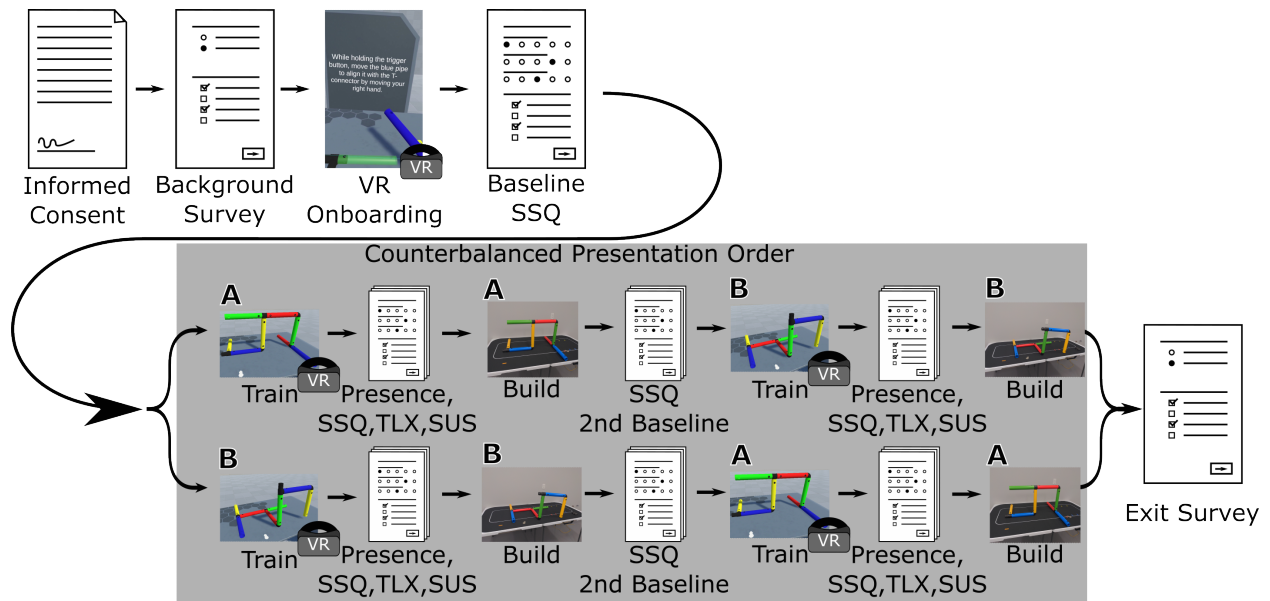


Figure 6.6: The full participant flow through the experiment.

Upon arrival, participants were asked for informed consent by the experimenter. Participants experienced the tutorial task, then VR training for the first build and replicating the build in the real world. For the real-world task, participants were asked to recreate the build as accurately as they could, and were asked to make use of all pieces in the event that they were stuck. This was followed by the VR training and real-world building for the other build, with the ordering presented in a counterbalanced manner. The result of the participant’s attempt to recreate the builds in the real world was photographed (without the participant visible) and the administering researcher timed the participant.

In addition to the physical tasks asked of our participants, they were given some common questionnaires throughout the study including the System Usability Scale (SUS) [14], Simulator Sickness Questionnaire (SSQ) [55], NASA Task Load Index (TLX) [43], and the Spatial Presence Experience Scale [44]. The ordering of the administration of these questionnaires was designed to ensure that we would acquire immediate before and after measures of SSQ, and per-build measures of

TLX, SUS, and SPES.

Finally, participants were given an exit survey after completing the second real-world build. The full presentation ordering of questionnaires, VR experiences, and assembly evaluation tasks can be seen in Figure 6.6. After completing the exit survey, the participant was thanked for their time and compensated \$25. Finally, the experimenter used a cleanbox to prepare the HMD and controllers for the next user.

### 6.2.3 *Materials*

We made use of the HTC Vive Pro Eye system which has 1440x1600 pixels per eye, a 110° field of view, integrated earphones, and provides accurate position and orientation tracking, eye-tracking, a front-facing camera, and microphone. We paired this with a computer capable of rendering and playing the VR training application at 90Hz as well as logging the tracked data to disk.

### 6.2.4 *Participants*

A total of 45 participants were recruited via university mailing lists. All participants (20 females, 25 males) had normal or corrected-to-normal vision with contacts, which were worn throughout the duration of the study. The mean age of our participants was  $22.1 \pm 4.2$  and 3 were left-hand dominant.

## 6.3 Machine Learning Experiment 1

In this machine learning experiment, we analyze the identifiability of motion data within each session. That is to say that we make use of data from a given session for training our models, and

Table 6.2: An overview of the conditions compared in this chapter. Explored conditions include the 0th and 1st order time derivative, within- and between-session predictions, the inclusion and exclusion of individual trackers, the type of data used from those trackers, and the model used.

Trackers Used			Included Data		Model Type		Session		Time Derivative
Head	DomH	OffH	Position	Euler					
Head	DomH		Position	Quaternion					
Head		OffH	Position	SixD		RF		A → A	
	DomH	OffH	× Position		×	GBM	×	B → B	×
Head				Euler		kNN		A → B	
	DomH			Quaternion				B → A	
		OffH		SixD					

some retained data for evaluating their accuracy. Several works explore the identifiability of users within a single VR session, such as that by Miller et al. [80].

For each session the participants experienced, the system tracked the position and orientation of the HMD and controllers at a rate of 90Hz. Because the application was built in the Unity Engine for the HTC Vive Pro Eye, this was the framerate at which the application executed its event loop, and thus the rate at which position and orientation data was provided to Unity to ensure the application updated the rendered view for the HMD.

Across all analyzed conditions, we considered the inclusion or exclusion of each tracked object (the HMD on the head, the controller in the dominant hand, and the controller in the non-dominant hand), as well as inclusion of position or orientation. Although the inclusion and exclusion of each individual value from each tracked object could be toggled independently (such as including or excluding the Y component of position), the combination of these features would result in an untenable search space. We choose to moderate the inclusion and exclusion of data by tracked object to allow us to explore how each tracked object is contributing to identifiability. Likewise, we moderate the inclusion of position and orientation data so we can evaluate if either are too noisy

Table 6.3: Within-session identification accuracy for Random Forest, with position and orientation data, trained and evaluated with data from session A.

<b>Position A → A RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	95.1%	95.3%	95.8%	85.6%	85.9%	85.4%	87.8%
<b>Head + DomH</b>	94.7%	93.4%	94.3%	78.3%	74.4%	73.0%	75.6%
<b>Head + OffH</b>	93.9%	93.7%	94.3%	79.7%	76.6%	76.8%	77.6%
<b>DomH + OffH</b>	88.7%	87.9%	88.4%	64.4%	71.0%	72.8%	71.3%
<b>Head</b>	85.1%	85.0%	86.0%	53.2%	42.9%	40.7%	44.2%
<b>DomH</b>	66.6%	67.0%	67.4%	41.7%	46.7%	45.7%	45.7%
<b>OffH</b>	71.2%	72.2%	71.9%	39.4%	49.9%	45.7%	47.8%

Table 6.4: Within-session identification accuracy for Random Forest, with position and orientation data, trained and evaluated with data from session B.

<b>Position B → B RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	94.2%	95.7%	95.0%	81.8%	83.1%	83.0%	84.1%
<b>Head + DomH</b>	91.8%	91.0%	91.0%	72.2%	70.6%	68.6%	69.3%
<b>Head + OffH</b>	94.4%	93.2%	94.4%	77.3%	75.7%	75.3%	76.6%
<b>DomH + OffH</b>	86.0%	87.3%	85.9%	59.6%	71.8%	73.8%	70.8%
<b>Head</b>	84.7%	81.3%	83.8%	48.0%	37.4%	36.9%	40.1%
<b>DomH</b>	61.6%	61.2%	59.7%	32.2%	47.3%	49.2%	49.4%
<b>OffH</b>	71.0%	71.7%	70.4%	40.1%	56.0%	56.2%	51.8%



for the models to fit well.

Beyond the inclusion of orientation data, we also considered three orientation representations: Euler angles, quaternions, and a six-dimensional representation. While the authors are not aware of a six-dimensional representation being applied specifically to data collected in VR experiences, this rotation representation has recently shown promise for machine learning models because of the lack of discontinuities among continuous data [128]. We also examined Euler angles and quaternions as alternatives due to their ready availability in the Unity engine and to help with comparisons with prior works [86] [91].

The models we chose to evaluate were k-Nearest Neighbors, Random Forests, and Gradient-Boosting Machine. These have been used previously by Miller et al. [80] effectively for identifying participants. Because of the large number of hyperparameters we are already evaluating across we choose to make use of default intrinsic parameter values for each model in the Scikit-Learn library [98]. These values were  $k=3$  for kNN, 100 estimators for Random Forest, and 100 estimators and a learning rate of 0.1 for GBM. For kNN, training data was normalized and the same scaling function was applied to the evaluation data to avoid issues with the scales along different axes. We chose not to examine deep learning models due to the likelihood of overfitting to the relatively short samples of data from each participant.

The full set of hyperparameter conditions explored in this chapter are described in Table 6.2. For this section, we discuss the results of moderating the trackers used, included data, model type and the first two values in the session column.

For a given set of conditions, we computed a set of per-second feature vectors. This feature vector described per-second statistics of each available field in the data by calculating the minimum, maximum, mean, median, and standard deviation. As an example, if head position was included in the raw data, a corresponding feature vector would include these statistics for the x, y, and z

values. These feature vectors from each session were then partitioned into 10 subsessions. For the within-session identifiability analyses, we evaluated 20 Monte-Carlo shuffles of the data by training on 9 random subsessions and evaluating the accuracy on the remaining subsession. While the subsessions retained for evaluation for each participant varied within shuffles, the shuffles themselves remained consistent across conditions. The models would predict a participant ID label per 1-second feature vector provided. These 1-second level predictions were aggregated together to yield a final predicted label for a given participant's evaluation data by selecting the most-predicted label.

For positional data, within-sessions, we found our best performance among conditions including both the head and at least one hand, as well as incorporating both positional and rotational data (Figure 6.3,6.4). For session A, this was 95.8% accuracy with the RF model, all three trackers, using both position and 6-dimensional rotations. The same condition performed the best with the GBM model at 95.7% accuracy, and the kNN model at 90.1%. In this chapter, I present the full set of results for the Random Forest classifiers. All results for GBM and kNN across the machine learning experiments can be found in Appendix B. For session B, the best performance was also with all three trackers, but with position and quaternion rotations, at 95.7%. Again this condition was the best for GBM as well at 94.0% accuracy, for kNN the best was all three trackers with position and 6-dimensional rotations at 87.4% accuracy.

## 6.4 Machine Learning Experiment 2

For our next machine learning experiment, we investigate the between-session accuracy of these models. This is motivated by some prior works that indicate that the task of identifying an individual across multiple sessions is more difficult than within one session. Miller et al. [80] mention it as a limitation of their study, and while Liebers et al. [67], Moore et al. [89], and Pfeuffer et

Table 6.5: Between-session identification accuracy for Random Forest, with position and orientation data, trained on A and evaluated on B.

<b>Position A → B RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	51.1%	44.4%	46.7%	42.2%	46.7%	40.0%	44.4%
<b>Head + DomH</b>	53.3%	53.3%	48.9%	40.0%	35.6%	40.0%	33.3%
<b>Head + OffH</b>	57.8%	46.7%	51.1%	44.4%	37.8%	40.0%	40.0%
<b>DomH + OffH</b>	33.3%	33.3%	35.6%	22.2%	33.3%	33.3%	31.1%
<b>Head</b>	75.6%	71.1%	75.6%	42.2%	40.0%	31.1%	31.1%
<b>DomH</b>	31.1%	35.6%	31.1%	20.0%	31.1%	28.9%	26.7%
<b>OffH</b>	26.7%	24.4%	31.1%	15.6%	17.8%	17.8%	22.2%

al. [99] all make use of datasets involving multiple sessions, the sessions are spaced at least a day apart, introducing some additional variability due to the change in state of the participant.

For our Between-session Identifiability analyses, we evaluate identification accuracy across the same set of variables as the previous section. We moderate the inclusion of trackers, the position and orientation data of those trackers as well as the orientation representation, and model type, as shown in Table 6.2. In this section, we now look at training models on one session and evaluating on another.

For this analysis, we developed the same per-second feature vectors as previously described. Our models were then trained on the entirety of a given session and evaluated on the entirety of the other session, aggregating predictions at the second level to create a prediction for that session. This approach was chosen as it would be an ecologically valid approach for identifying users

Table 6.6: Between-session identification accuracy for Random Forest, with position and orientation data, trained on B and evaluated on A.

<b>Position B → A RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	51.1%	51.1%	53.3%	40.0%	46.7%	46.7%	44.4%
<b>Head + DomH</b>	57.8%	51.1%	55.6%	40.0%	42.2%	40.0%	40.0%
<b>Head + OffH</b>	62.2%	62.2%	62.2%	37.8%	40.0%	40.0%	40.0%
<b>DomH + OffH</b>	40.0%	37.8%	37.8%	31.1%	33.3%	31.1%	33.3%
<b>Head</b>	80.0%	75.6%	82.2%	33.3%	51.1%	31.1%	42.2%
<b>DomH</b>	35.6%	35.6%	35.6%	15.6%	31.1%	31.1%	24.4%
<b>OffH</b>	31.1%	31.1%	31.1%	22.2%	24.4%	22.2%	22.2%

without their knowledge.

Looking at the between-session positional data, we now find our best performances still incorporated positional and rotation data, but now only made use of the head tracker (Table 6.5,6.6). For the models trained on A and evaluated on B, the best performance was head tracker, position, and either Euler or 6-dimensional rotational representation, at 75.6% for RF, 73.3% for GBM, and 75.6% for kNN. For the models trained on B and evaluated on A, the head tracker and position with 6-dimensional rotation performed at 82.2% for RF, and 80% for GBM. For B to A, kNN performed best with head, position and Euler angles at 71.1%.

## 6.5 Machine Learning Experiment 3

Beyond creating the feature vector by using the collected data as it was, we also considered computing its first-order time derivative, since velocity-based features have been demonstrated to be useful when predicting knowledge and performance retention [89]. This tracking data was then aggregated at the one-second level by computing the minimum, maximum, mean, median, and standard deviation for each component value, to create a feature vector to describe that second of motion. Depending on the tracker inclusion and position/orientation conditions, this per-second feature vector consisted of between 15 and 135 values.

In this section, we again divided the data from each session into 10 subsessions. We use a similar approach of creating 20 Monte-Carlo shuffles of the subsessions, allowing the models to train on 9 of them per participant, and evaluating their identification accuracy on the retained ones. This is essentially the same procedure as presented in Section 6.3, but now conducted on the feature vectors that were generated using the time-derivative data instead of the raw data. Again, RF and GBM performed on par with each other, and overall better than kNN, so we choose to show the results for RF (Table 6.7,6.8), with the full set of results available in Appendix B.

Examining the within-session velocity data, we found our highest identification accuracies among models involving six-dimensional rotation representations and incorporating at least two trackers. Across all models, for both the A session and B session, the best-performing conditions were those that made use of position and 6-dimensional rotations across all three trackers, yielding 64.1% accuracy for session A and 66.2% accuracy for session B. GBM maxed out at 64.3% for session A and 68.0% for B. Finally, kNN performed notably worse, with accuracies of 37.2% for A and 40.9% for B. k

Table 6.7: Within-session identification accuracy for Random Forest, with velocity and angular velocity data, trained and evaluated on A.

<b>Velocity A → A RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	54.2%	59.0%	64.1%	41.3%	45.7%	52.4%	62.0%
<b>Head + DomH</b>	48.6%	52.7%	60.3%	37.4%	40.4%	46.4%	54.8%
<b>Head + OffH</b>	44.8%	47.9%	53.8%	30.7%	36.2%	42.8%	48.0%
<b>DomH + OffH</b>	49.4%	56.0%	59.6%	39.3%	40.8%	51.0%	54.9%
<b>Head</b>	29.6%	32.2%	36.1%	18.3%	24.4%	28.1%	28.4%
<b>DomH</b>	42.4%	45.1%	49.1%	25.8%	32.3%	38.6%	43.7%
<b>OffH</b>	36.1%	45.1%	50.1%	27.7%	26.7%	37.7%	44.8%

Table 6.8: Within-session identification accuracy for Random Forest, with velocity and angular velocity data, trained and evaluated on B.

<b>Velocity B → B RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	58.7%	63.6%	66.2%	46.3%	53.0%	59.7%	64.2%
<b>Head + DomH</b>	51.8%	54.2%	58.3%	35.1%	41.7%	48.4%	54.6%
<b>Head + OffH</b>	50.9%	55.7%	60.8%	36.9%	41.4%	50.9%	56.3%
<b>DomH + OffH</b>	52.1%	58.7%	60.9%	36.7%	44.6%	53.4%	56.9%
<b>Head</b>	30.2%	31.9%	36.6%	20.2%	22.3%	25.6%	31.2%
<b>DomH</b>	41.6%	44.2%	46.4%	24.1%	27.4%	39.1%	43.2%
<b>OffH</b>	40.6%	49.0%	49.9%	26.2%	29.1%	40.1%	42.7%

Table 6.9: Between-session identification accuracy for Random Forest, with velocity and angular velocity data, trained on data from A and evaluated with data from session B.

<b>Velocity A → B RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	28.9%	28.9%	35.6%	17.8%	26.7%	28.9%	33.3%
<b>Head + DomH</b>	28.9%	24.4%	33.3%	15.6%	20.0%	28.9%	31.1%
<b>Head + OffH</b>	28.9%	31.1%	24.4%	15.6%	22.2%	22.2%	24.4%
<b>DomH + OffH</b>	22.2%	22.2%	26.7%	13.3%	17.8%	22.2%	24.4%
<b>Head</b>	33.3%	42.2%	44.4%	15.6%	28.9%	28.9%	33.3%
<b>DomH</b>	20.0%	20.0%	24.4%	15.6%	11.1%	11.1%	24.4%
<b>OffH</b>	24.4%	24.4%	20.0%	11.1%	20.0%	17.8%	15.6%

## 6.6 Machine Learning Experiment 4

Finally, we examine the identifiability of motion data, using the same velocity feature-vector procedure described in the previous section, but now between sessions. We conducted this exploration by allowing the models to train over the entirety of one session from all participants, then evaluated over the entirety of the other session. Again, because our models created predictions for each second, based on each per-second feature vector provided, we aggregate the predictions, saying that the model ultimately predicted an identity based on

Finally, when we look at the between-session velocity data, we find that no conditions exceeded 50% accuracy. Our best-performing models involved both the position and 6-dimensional representation of the data, with the head tracker only. For the RF model trained on session A and evaluated on session B, we found 44.4% accuracy, and for the RF model trained on B and evaluated on

Table 6.10: Between-session identification accuracy for Random Forest, with velocity and angular velocity data, trained on data from B and evaluated with data from session A.

<b>Velocity B → A RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	42.2%	44.4%	40.0%	33.3%	31.1%	40.0%	40.0%
<b>Head + DomH</b>	37.8%	35.6%	42.2%	28.9%	26.7%	28.9%	31.1%
<b>Head + OffH</b>	26.7%	35.6%	35.6%	17.8%	15.6%	35.6%	40.0%
<b>DomH + OffH</b>	31.1%	37.8%	28.9%	17.8%	22.2%	28.9%	26.7%
<b>Head</b>	28.9%	35.6%	46.7%	17.8%	17.8%	26.7%	33.3%
<b>DomH</b>	28.9%	26.7%	26.7%	13.3%	15.6%	20.0%	17.8%
<b>OffH</b>	22.2%	26.7%	24.4%	11.1%	11.1%	17.8%	20.0%

A, this yielded 46.7% accuracy. For that same condition, GBM had an accuracy of 46.7% for A to B. For GBM’s best B to A condition, we see head and dominant-hand, position and 6-dimensional rotations perform best at 48.9%. For kNN A to B, Head-only, position and 6D rotations is best at 31.1%, and no condition performed above 25% accuracy for B to A.

## 6.7 Discussion

### 6.7.1 *Tracking data from an unspecified task is likely not sufficient for identifying people against their will with these machine learning models*

Our findings generally show that across the board, it’s feasible to attain impressive results for identifying participants within the same task. The results of our analyses also demonstrate, however,



that even when models are provided with substantially more data to evaluate over, they still fail to identify participants at nearly the same level of accuracy when attempting to evaluate over a slightly different task. This result is important because it demonstrates that XR practitioners hoping to develop motion-based authentication should present users with the same task for encoding and authentication. Likewise, those hoping to de-anonymize XR usage data should, if possible, train their models on identified samples of users performing parts of the tasks in their target data set, because even similar tasks yield between sessions yield worse results.

Furthermore, it is worth noting that in an ecologically valid between-task identification scenario, where one knew the labels for session A, but not B, one might attempt to use the best-performing classifier in their labelled training data (i.e., all three trackers with position and quaternion rotations, which performed at 96% accuracy). This condition applied between sessions yields relatively poor performance, however, at only 48.9% accuracy.

#### *6.7.2 More tracked devices yields better identification accuracy within the same task*

As shown in Tables 6.3 and 6.4, across all position and orientation conditions, the inclusion of additional trackers yielded better accuracies. This suggests that within a given task and session, none of the tracked objects contributed data that caused the models to overfit to the training subset. Perhaps unsurprisingly, we also note that of the 2-device and 1-device conditions, those that included the head performed better than those without, when position was included. This is aligned with the results of the ablation results presented by [80], in which removal of the features related to the head Y-value resulted in the greatest drop in identification accuracy.

### *6.7.3 More tracked devices does not yield better identification accuracy across similar but slightly different tasks*

As shown in Tables 6.5 and 6.6, the inclusion of trackers other than the head yielded worse accuracies, suggesting that they contributed to noise in the training of the model. We were focused on the ecologically valid potential privacy issue in which a malicious actor has a sample of labeled, trained data. Because of our study design, we are unfortunately unable to separate out if this may have been due in part to the model encoding physiological features of the user such as their height, as opposed to features related to their environment as indicated by [105].

### *6.7.4 Using position and orientation generally yields higher identification accuracy*

Across both the within- and between-session conditions, the inclusion of both position and orientation generally yielded higher identification accuracy than using exclusively one or the other. These results indicate that broadly when including the data from a given set of trackers, it appears to be helpful to make use of both the position and orientation tracking, if afforded by the system.

Interestingly, we find that in many of our conditions, the six-dimensional representation proposed by Zhou et al. [128] not to perform significantly better. We posit that this may be partially a result of the manner in which we aggregate positions. Because each per-second feature vector was considered in isolation, moments where the quaternion or Euler values would have discontinuities was outnumbered by the samples of data without such discontinuities. It is possible that this representation may be more useful for a more time-dependent approach such as LSTM.

### *6.7.5 Identities can be partially obfuscated by encoding the velocities of tracked devices instead of positions*

While one might expect that using feature vectors generated from the first-order derivative to potentially encode data more specific to users and yield classifiers that are better capable of generalizing across tasks, we found that between-task performance was generally worse with the velocity-based features than with the positional ones, diminishing from 82.2% to less than 50% for Random Forest. While this is still much more than the random likelihood of a correct guess, this diminishing suggests that while velocity-based features may have some use for identification, they may need a specific featurization outside of the scope of this chapter to yield accuracies on par with our positional data.

### *6.7.6 Limitations*

In this work, we choose not to optimize some of the hyperparameters that are intrinsic to the models explored (such as the depth of trees in Random Forest, or  $k$  in kNN). This decision is due in part to the scope of the explored hyperparameters in this work requiring the evaluation of 1176 ( $2 * 4 * 7 * 7 * 3$ ) models, resulting in further hyperparameter exploration to be untenable. Additionally, our approach categorized each feature vector in isolation, not taking advantage of the temporal nature of our data.

Another limitation is that we examined the data from two similar but distinct tasks. It's possible that our results could be different with different builds or with fundamentally different tasks. We believe that this is a somewhat representative example of identifiability between similar tasks in which the user is performing a task that is prescriptively defined as a sequence of steps, but varying results may be found for environments in which the interactive objects are different, appear in

different locations, or lack the instructional context our application contains.

Further, while our total set of data may appear large (1.7 million samples), this represents the data of 45 users and a total of a little more than 5 hours of time. If we were to make use of Deep Learning methods on this amount of data, it would be likely to overfit to the individuals. While this amount of data is less than some previous work like that by Miller et al. [80] or Moore et al. [89], it is more than several of the other works exploring VR identifiability, and will be made available as a resource for future research.

One final limitation to address is the arguable presence of a confound due to differences in the builds in addition to the participants doffing and donning the VR equipment. Some existing work, such as that by Asish et al. [3] makes use of multiple scenes and contexts without removal of the headset. With the between-session portion of this work, our focus was on identifiability between two individual VR sessions. Because our context is in an ecologically valid training scenario, we anticipated learning affects to moderate participants' motions through the instructions. This would still result in unique behavior from session to session, so we opted for two unique, but slightly different builds.

## 6.8 Conclusion

In this chapter, we explored the identifiability of participants when presented with two distinct but similar tasks in VR. We first looked at identifiability within sessions and we examined the inclusion and exclusion of data across several conditions by tracked object as well as the kind of data contained therein. Our search over this space of data should help form an understanding of what is contributing to accuracy and what is overfitting for different training and testing conditions. Further, by examining temporally close data from two VR sessions, we hoped to expand

our understanding of identifiability as it relates to variables specific to a given session. While we found that within a VR session, increasing the sources of data generally improved the accuracy of the models identifications, we found some sets of data to be contributing data that overfits to the session, thus resulting in worse accuracy. Finally, by exploring a velocity representation, we find different sets of data to be useful for identification, warranting further exploration.

## CHAPTER 7: INSIGHTS AND FUTURE WORK

### 7.1 Introduction

The previous chapters discussed the findings related to two human-subjects studies. In the process of conducting this research, we have identified some gaps and new areas for research that are both interesting and valuable to investigate. In this chapter, I will describe both the upcoming research that the second human-subjects study enables, as well as some of the insights gained through the process of doing this research.

### 7.2 Insightful Conjectures

This research was broken into two major categories: the prediction of outcomes from VR training environments, and the identifiability of this data. Through this work, a few themes have emerged that may be of use to future researchers and XR practitioners.

#### *7.2.1 The Appropriateness of Time-derivative Data*

One of the comparisons that I examined was the performance of our models when trained on the unmodified data, or the derivative of that data. Taking the derivative with respect to time with rolling differences can accentuate noise by amplifying the high frequency components of the signal. While we assume the HTC Vive and HTC Vive Pro Eye systems used throughout this dissertation to yield accurate data with a relatively high signal to noise ratio, these effects were nevertheless magnified. Future effort could make use of filtering to produce better, filtered estimates of hidden states that are not directly measured (such as estimating velocity from position

with a Kalman filter). This could yield better results both in terms of identifiability as well as the prediction of learning outcomes with the time-derivative data.

Due to the combinatorial explosion of features, I was limited in what I could explore in these studies. I explored features with reference to the world-space, but other researchers like Pfeuffer et al. [99] have found certain velocity features to be helpful for identification or authentication when calculated when done in reference to other trackers. It's possible that a combination of positional, velocity, or even higher order derivatives of the data could be optimal for some of these tasks, but these potential future approaches will necessitate better filtering of high-frequency domain noise.

### *7.2.2 Velocity is an Indicator of Cognitive Processing*

In Chapters 3 and 4, we found the models trained on data based on the velocity of the the learner's movement to yield higher accuracy when predicting learning gains, knowledge retention, and performance retention. This suggests that the velocity of movement more closely indicates underlying cognitive processes than the positions themselves while learning. Our visual analysis presented in Chapter 3 and refined in Chapter 4 suggests that the classifiers may be identifying purposeful movement as part of its prediction. Since our participants had no prior experience with robotic surgery, we interpret this as indicating that the degree to which the user is encoding the training content is more predictable with this kinda of data.

### *7.2.3 Position Encodes Physiology which is a Key Feature for Identifiability*

When discussing our identifiability findings in Chapters 5 and 6, I noted that models trained on the positional representation of data consistently outperformed those trained on the velocity data. This

suggests that the position data, which is more closely related to the anatomy of the participants was more useful to our classifiers than the velocity, which may be more related to the kinetics. With this observation, one should note that VR users with disabilities likely have an increased risk of identification compared to existing online interactions, highlighting the importance of this ongoing field of research. As we see new techniques develop to attempt to reduce identifiability in virtual environments, such as remapping movements to a standard-scaled avatar, our understanding of adversarial identifiability will also have to evolve.

#### *7.2.4 Task Categorization*

One of the contributions to why we found positional characteristics to be more identifying and the kinematic data to be more helpful for predicting outcomes is due to the difference in the type of task being asked of the machine learning models. While identification or authentication tasks must seek to identify what is unique about a user's movement, performance prediction has to identify similarities in the movement, with individually identifying features potentially leading to detrimental overfitting.

For identification, based on the results presented in this dissertation, it appears that the overfitting risk is more related to factors intrinsic to the session or the content presented in the virtual environment, rather than the participant themselves. This suggests that there could be value in the development of a model that unifies the different physical aspects that feed into the readily available tracking data: the content of the presented scene, the physiology of the participant, the specifics of the VR hardware, and the goals of the participant. Such modeling could lead to better selection of features, filtering, or even development of more robust human performance modeling in the context of virtual environments.



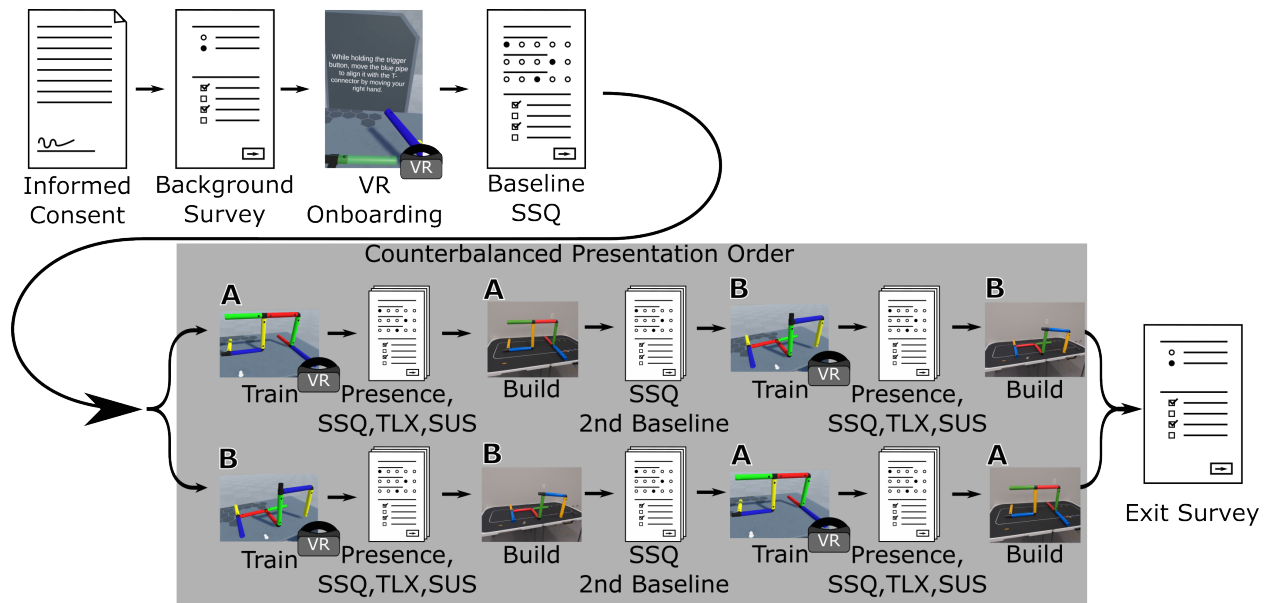


Figure 7.1: The participant flow as experienced through the experiment to support real-world transfer of skills

### 7.3 Future Work

While we presented some work that makes use of the Full-scale Assembly Benchmark (FAB) environment, we intend to continue using it for some additional work. In this section, I'll describe research that we're in the process of conducting, as well as some potential future work that could readily stem off of the research that I have presented in this dissertation.

#### 7.3.1 Predicting Real-world Transfer of Skills

We are currently in the process of conducting additional participants through the FAB environment with a study design that involves participants building a real-world version of each structure after learning how to construct it in VR (Figure 7.1). We intend to collect the data from 100 complete sessions in order to enable our planned analyses. Recall that after they completed their best attempt

at reconstructing the build, a photo was taken, allowing us to interpret the outcome quality of their real-world performance.

Since each piece in the toy set is limited in the way that it can be connected, we can design a heuristic to objectively measure the accuracy of the participants' performance by a measure of how different it is from what they were supposed to build. Similarly to the work presented in Chapter 4, we can build a classifier to attempt to predict this performance based on the data recorded while undergoing training.

While this data can allow us to predict the real-world transfer of skills, because we collected data from two unique tasks, we can look at additional questions, such as whether behavior when learning one task predict performance in a similar but distinct task, or if exposure to multiple tasks improve the performance of skill transfer prediction models. Additionally, beyond skill-transfer related questions, we can also look at predicting cybersickness and cognitive load, or modeling usability.

### 7.3.2 *Obfuscation*

While the FAB environment enables us to continue exploration into identifiability, there are a few potential future studies that would continue our understanding of the identifiability of this data. A further study we recommend be conducted includes a controlled study that compares the identifiability of users between tasks with little motion and tasks with much motion, specifically in guided and unguided interactions.

Another future study that would be valuable would be to analyze these approaches in the context of authentication, rather than identification, in an ecologically valid context, like the VR training application that we investigated. This could be used to augment existing work in continuous au-

thentication to detect session hijacking in order to retain your credentials between sessions until the system detects a different user has begun to use the machine.

Finally, this work demonstrated that training a classifier within a session of data can overfit to features specific to the task a participant is experiencing, and that using this data from one unspecified task may be insufficient to identify participants in other tasks against their will. In our dataset, participants were exposed to one session in VR, exited VR, then re-entered for the other task. Exploring multiple tasks within a single VR session, and alternatively the same task, longitudinally across multiple sessions will be an important next step for improving our understanding of the variability in tracking data across experiences and sessions. A further logical next step for this work would be to design additional experiences, and examine to what degree training classifiers on samples of data from multiple distinct tasks can improve accuracy in identifying people in unseen tasks.

## CHAPTER 8: CONCLUSION

In this dissertation, I presented some of my findings regarding the usefulness of readily available tracking data in virtual reality training experiences. My focus was primarily on prediction and estimation of the outcomes of those training environments at the user-level, as well as discussing identifiability as it relates to this data. I believe that it is clear that while the intrinsic tracking needed to convey VR experiences affords systems unique insight to their users, there is still much work to be done in this area to continue advancing our understanding of what this data is useful for and how it can be used.

Through the work described in Chapters 3 and 4, we found that readily available VR tracking data was useful for predicting both cognitive and psychomotor outcomes from VR training scenarios. Similarly, in Chapters 5 and 6, we determined that factors relating to the user's session and experience impacted the identifiability.

In seeking to answer these questions, we made some additional observations, like how identifiability isn't always improved by adding additional tracked data, and how models can be trained to predict the retention of psychomotor performance better than the immediate gains of cognitive knowledge. I am really excited to see where the future takes us for new applications of VR tracking data, because a system with a better understanding of the user is one that can hopefully be more equitable.

**APPENDIX A: A FORMATIVE EVALUATION METHODOLOGY FOR  
VIRTUAL REALITY TRAINING SIMULATIONS**

*NOTE:* This appendix is a modified format version of the paper previously published.

Material from: A. G. Moore, X. Hu, J. C. Eubanks, A. A. Aiyaz, and R. P. McMahan. A formative evaluation methodology for vr training simulations. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 125–132, 2020. doi: 10.1109/VRW50115.2020.00027

## A.1 Introduction

The term “formative evaluation” was originally coined by Michael Scriven [107] in 1967 to refer to “outcome evaluation at an intermediate stage in the development of the teaching instrument.” The term was quickly adopted by the educational research community [120] and popularized by Dick and Carey [27] in the area of instructional design. In the 1980s, the concept of “formative assessment” emerged as an opportunity to assess the learner’s progress, in addition to continually modifying instruction [5, 40]. In the 1990s, Carroll et al. [15] were among the first to use the term “formative evaluation” for a human-computer interaction (HCI) context, and shortly after, Hix and Hartson [46] defined it as the “evaluation of the interaction design as it is being developed early and continually throughout the interface development process.” Put simplest by Hartson and Pyla [45], *formative evaluation* helps you “form the design”, as opposed to *summative evaluation*, which helps you “sum up the design”.

In HCI, formative evaluations are empirical methods that use data collected from representative users interacting with the system to iteratively assess and improve the interaction design [47]. Numerous techniques have been employed to gather data during formative evaluations of traditional 2D user interfaces (UIs), including critical incidents, video and/or audio recordings, automated data logging, think aloud protocols, interviews, and questionnaires [45]. These techniques have also been applied to 3D UIs like VR [12]. However, in general, few evaluation techniques have

been developed specifically for 3D UIs[62].

In this appendix, we present a novel formative evaluation methodology designed specifically for improving the interaction design of VR systems that simulate real-world tasks, which we call *CARAI* (pronounced “kerē”). Our methodology involves five distinct stages: 1) *Capture* the real-world task and interactions; 2) *Automate* data logging for each subtask and interaction of the VR counterpart; 3) *Run* an empirical, rigorous user study; 4) *Analyze* user performances for each subtask; 5) *Inspect* significantly worse subtasks and interactions.

As a case study, we have used CARAI to formatively evaluate the interaction design of a VR system designed to train users how to troubleshoot a surgical robot during surgery. We used a videotape of a subject-matter expert (SME) troubleshooting a real-world robot to identify subtasks, the physical actions that each subtask involves, and how much time the SME needed to complete each one. During development of the VR application, we incorporated data logging features to automatically capture the time required to complete each subtask and any errors that could be potentially committed during that subtask. After completing development, we conducted a user study with 20 participants to capture usability data for all the subtasks and interactions. We then conducted a one-way repeated-measures analysis of variance (ANOVA) to identify subtasks that yielded significantly worse completion times or more errors. Finally, we inspected the implementations of these subpar subtasks and were able to identify important but nuanced issues in their interaction designs, which we have since addressed to improve the usability of our VR training application.

We present the following contributions in this appendix:

1. A formative evaluation methodology designed specifically for improving the interaction design of VR simulations.
2. A case study using our methodology to formatively evaluate and improve the usability of a

VR training simulation.

3. An investigation and discussion of how to optimally analyze user performances for each subtask of a VR simulation.

## A.2 Formative Evaluation

Several formative evaluation methodologies and approaches have been developed for conventional UIs [45], such as the Rapid Iterative Testing and Evaluation (RITE) method [79], which has been used to formatively evaluate VR systems (e.g., [61]). However, only a couple formative evaluation methods have been developed specifically for VR or 3D UIs [62].

Gabbard et al. [39] presented a user-centered design and evaluation methodology specifically for VR based on sequentially performing: 1) user task analysis; 2) expert guidelines-based evaluation; 3) formative user-centered evaluation; and finally, 4) summative comparative evaluations. While this sequential evaluation methodology could be applied generally to any type of interface, it employs application-specific guidelines, domain-specific representative users, and application-specific tasks to design and evaluate a useful VR system interface [62].

Bowman and McMahan [10] presented another evaluation approach based on investigating specific components of a VR system while keeping other components constant, in order to evaluate their effects on the usability of the system. This approach can be used either formatively, to decide upon unclear design choices, or summatively, to establish knowledge of the general effects of one or more components [62]. For identifying components to investigate, McMahan [73] has presented the user-system loop and identified several components pertaining to interaction [74], scenario [100], and display fidelity [77].



### A.3 New Formative Evaluation Methodology

In this section, we present our novel formative evaluation methodology that helps to identify usability and interaction issues in VR systems that simulate real-world tasks. This methodology emerged from the currently presented case study of a VR system designed to train users how to troubleshoot a surgical robot, which was initially developed using a user experience (UX) lifecycle of analyzing, designing, prototyping, and evaluating [8]. In the early stages of the lifecycle, we used conventional evaluation methods, including heuristic evaluations [20], SME interviews after demos [10], and a formative quasi-empirical evaluation with SMEs [8]. At this point, we redesigned the troubleshooting scenario to address several issues identified during the quasi-empirical study and re-developed the VR application from scratch. Additional SME interviews after demos of our new system indicated that we had fixed all of the previously identified issues. However, we observed some potential usability issues during some of the demos. To determine if these potential issues were actual usability problems, we devised the following formative evaluation method.

It is important to note that our methodology can be used at any stage in the UX lifecycle, despite us employing it in the late stages of our VR training application's lifecycle. We actually recommend using it at the start of the lifecycle, if possible.

Figure A.1 shows an overview of our methodology, which we refer to as CARAI. It involves five distinct stages: 1) Capture, 2) Automate, 3) Run, 4) Analyze, and 5) Inspect.

The capture stage occurs during the analyze phase of the UX lifecycle, and ideally, should occur during the first iteration. It involves videotaping one or more SMEs completing a physical task in the real world, which will eventually be simulated by the VR system to be developed. The most relevant output of this stage are subtask requirements, which clearly define the subtasks of the real-world task, the physical actions required for each subtask, potential errors that could be committed

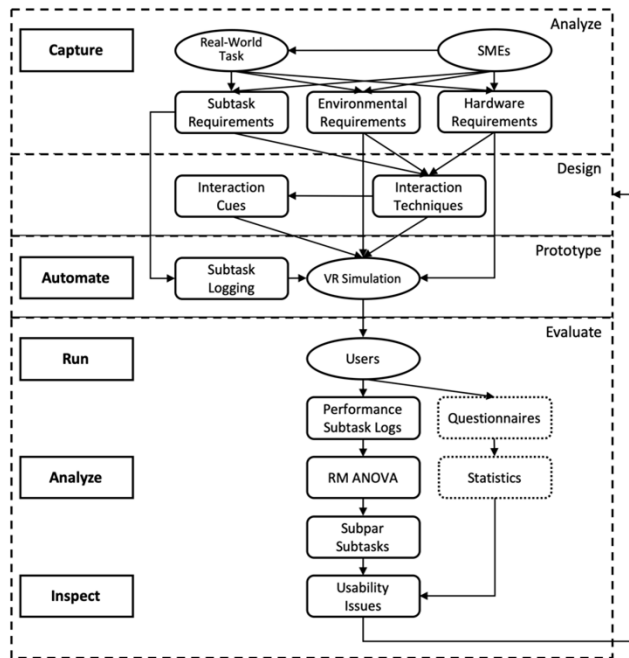


Figure A.1: Overview of our formative evaluation methodology.

during each, and metrics based on the performance of the SMEs, such as speed, accuracy, and precision [78]. Environmental requirements, such as what real-world objects need to be modeled, and hardware requirements, such as tracking capabilities, can also be derived from the real-world task, in addition to consulting the SMEs.

During the first iteration of the UX lifecycle, the design phase should include identifying interaction techniques (e.g., selection, manipulation, travel, and system control [62]) to employ, based on the subtask, environmental, and hardware requirements. In turn, interaction cues, which inform the user about an action to take [28], should be designed for the chosen interaction techniques.

The automate stage of our methodology occurs during the prototype phase of the UX lifecycle. It involves integrating automated data logging into the VR simulation to capture subtask performance metrics, such as time and interaction errors. In this appendix, we discuss how we formatted our

logs, but formatting requirements are highly dependent upon the subtask requirements identified during the capture stage, which are dependent upon the real-world task being simulated. Ideally, the automate stage would occur during the first iteration of the lifecycle; however, as discussed above, this requires time to implement, which detracts from developing other features or fixes, and may need to be reimplemented in later iterations, if the system is redesigned.

The run stage occurs during the evaluate phase of the lifecycle. It involves conducting an empirical, rigorous user study to capture performance data for each subtask implemented in the VR system. These metrics should be automatically captured by the automated data logging implemented in the automate stage. In addition to the performance logs, this stage can also be used to capture subjective data in the form of questionnaires. Early in the lifecycle, we recommend using custom usability questionnaires that address application-specific design requirements, such as the subtask, environmental, and hardware requirements identified during the capture stage. Later in the lifecycle, as was the case with our VR training system, we recommend using standardized questionnaires to obtain results that can be compared to other VR evaluations.

The analyze stage should immediately follow the run stage. It involves conducting a one-way (subtask) repeated measures (RM) ANOVA to identify subpar subtasks that were significantly worse than some other subtask, in terms of the performance metrics. This stage can also be used to analyze, and possibly compare, the statistics of any data collected via questionnaires.

The final inspect stage should follow the analyze stage. It involves inspecting the VR simulation's software, and possibly hardware, to identify usability issues pertaining to the subpar subtasks and to identify what causes them. In turn, these usability issues and their causes should be addressed by appropriately redesigning the interaction techniques and cues of the simulation. The formative iteration process of the UX lifecycle can continue by simply repeating the relevant stages of our methodology. We have provided a general overview of our formative evaluation methodology

above. In the next few sections, we discuss how we applied the five stages of CARAI to our VR training system for troubleshooting during robotic surgeries.

#### A.4 Case Study: The Real-World Task

Our VR simulation has been formatively designed and developed to train first assistants, who facilitate the procedures and perform instrument exchanges in robotic surgeries [19], to troubleshoot the robot during intra-operative complications, which is one of their most important responsibilities [59]. In the initial stage of our project, during the analyze phase, we collaborated with SMEs to identify the subtask, environmental, and hardware requirements for the VR simulation. A major part of this collaboration involved videotaping the SMEs executing the proper procedure for trouble-shooting the robot. This video “artifact” [45] allowed us to identify the subtasks required for the training, the physical actions to simulate, and potential errors that could be committed, in addition to some environmental requirements (see Figure A.2).

Though we did not employ CARAI early in the lifecycle of our VR simulation, we were able to reuse this video artifact later on to capture completion times in seconds for each subtask based on the actions of the SMEs ( $T_S$ ). See Table A.1 for the subtasks and times.

#### A.5 Case Study: The VR Simulation

During our initial analyze phase, we identified several design requirements for our VR simulation. The SMEs desired hardware that was easily replaceable and affordable (relative to hospitals), which constrained us to consumer VR technologies only (i.e., no specialized headsets or peripheral devices). The captured subtask requirements indicated that the system would need to support 3D manipulations outside of the user’s view, in order to rotate the wrench and check the vision moni-



Figure A.2: A frame of the video used to capture subtask metrics.

tor. This constrained us to choosing a room-scale technology with outside-in tracking, which only included the HTC Vive and Oculus Rift CV1 at the time. Finally, the environmental requirements indicated that a large tracking space would be necessary to access the virtual operating room by walking, without introducing a virtual travel technique and potentially simulator sickness. Hence, we settled on the HTC Vive, which provided the larger tracking space (4m x 4m).

As seen in Table A.1, several of the subtasks required the user, acting as a sterile first assistant, to communicate with the non-sterile surgeon and staff. Many of these communications involve requesting the surgeon or staff to interact with the non-sterile portions of the surgical robot, such as the touchscreen on the vision cart or the touchpad on the surgeon console. In order to simulate such communications and the ability to make requests of either a surgeon or staff member, we used a floating window with dialog options [62] for each virtual agent (see Figure A.3). We chose not

Table A.1: The subtasks for our VR simulation and their associated interactions and SME completion times (in seconds).

#	Subtask	Interaction	$T_S$
1	Check error message	Walk + Look	3.5
2	Consult surgeon	Select (dialog)	6
3	Ask to restart system	Select (dialog)	7
4	Ask to call support	Select (dialog)	7
5	Ask for release wrench	Select (dialog)	7
6	Ask to emergency stop	Select (dialog)	10
7	Hold instrument carriage	Select (carriage)	4
8	Insert wrench	Position (wrench)	5
9	Rotate wrench	Rotate (wrench)	4
10	Check vision monitor	Look	2
11	Remove wrench	Position (wrench)	4
12	Remove instrument	Position (instrument)	4
13	Give instrument to staff	Position (instrument)	7
14	Ask to recover fault	Select (dialog)	7
15	Ask to disable arm	Select (dialog)	7
16	Check error message	Walk + Look	3.5
17	Use cannula lever	Select (lever)	7
18	Use instrument clutch	Position (clutch)	8
19	Use port clutch	Position (clutch)	10
20	Ask to confirm disable	Select (dialog)	7

to use voice commands to reduce the extraneous cognitive load [95] induced by voice recognition errors [62]. In order to avoid clutter, we displayed only one window at a time, based on which agent was near to the center of the user’s field of view.

Outside of communicating with the surgeon and non-sterile staff, most of the remaining subtasks involved motor actions to grab, position, or rotate objects within the operating room. To increase the likelihood of successfully training these motor skills, we decided to use the simple virtual hand technique, which provides a high degree of interaction fidelity for such gross motor tasks [73] (see Figure A.4 for example). Again, to reduce extraneous cognitive load [95], we decided to use the same simple virtual hand technique for selecting the previously mentioned dialog options (see

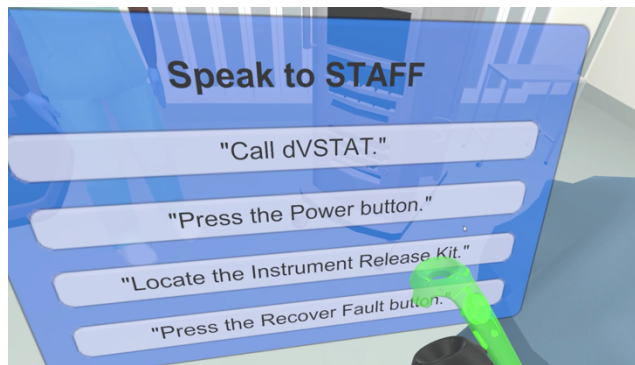


Figure A.3: One of the floating windows used for communicating with the virtual surgeon and non-sterile staff member.

Figure A.3), as opposed to introducing a second selection technique, such as ray-casting.

Finally, a number of the subtasks involved visually checking the vision monitor to either read error messages or to verify that the instrument is no longer grasping tissue. We implemented this by simply verifying that the vision monitor was in the center of the user's field of view. For reading the error messages, our subtask requirements indicated that, in the real world, the assistant should walk up to the vision cart to improve the legibility of the messages. Hence, we designed those subtasks to require walking to, but not touching (to avoid breaking sterility), the vision cart.

Because we were developing a VR simulation for training, we decided to use interaction cues to enable learning. An interaction cue is essentially a stimulus that informs the user about an action to take [28]. For our virtual hand interactions, we used green, semi-transparent animations to indicate what manual action to take (see Figures A.3 and A.4). For the walking and looking subtasks, we used a "Stand Here" icon to indicate where to walk to and a "Look Here" icon to indicate where to look (see Figure A.5).

In addition to the interaction cues, we included verbal repercussions for mistakes. If the user makes an error or incorrect decision, one of the virtual agents will verbally inform the user of the mistake.



Figure A.4: Example of using the simple virtual hand technique to rotate the grip release wrench.



Figure A.5: We required walking and looking to read error messages.

After implementing all of the interaction techniques and cues described above during our major re-design, we proceeded with the automate stage of CARAI and implemented automated data logging for each subtask. Completion times were measured in seconds from the moment the user was expected to act until the current subtask was successfully completed. For each subtask, we identified numerous errors that the user may make, including selecting the wrong dialog option, selecting the wrong object, positioning an object in an incorrect space, rotating the wrench the wrong direction, or at any point, touching a non-sterile object. Our data logs included a time column and an error count column for each subtask, formatted as a comma-separated value (.csv) file.



## A.6 Case Study: The Rigorous User Study

### A.6.1 *Independent Variable*

Our study's independent variable was the subtask completed, which included 20 levels or subtasks (see Table A.1) within subject.

### A.6.2 *Dependent Variables*

#### A.6.2.1 *Subtask Metrics*

As described above, we automatically collected two metrics for each subtask: completion time and number of errors committed.

#### A.6.2.2 *Questionnaires*

Because we conducted CARAI later in the lifecycle of our VR simulation, we decided to use standardized questionnaires for simulator sickness, presence, and usability. We specifically used the Simulator Sickness Questionnaire (SSQ) [55], the Spatial Presence Experience Scale (SPES) [44], and the System Usability Scale (SUS) [14]. We also administered a custom questionnaire addressing scenario fidelity since our VR system was designed to simulate the real-world task of troubleshooting a surgical robot. Participants ranked 10 aspects of the scenario from 1 ("I do not agree at all") to 5 ("I fully agree"). See Table A.5 for the 10 items.

### *A.6.2.3 Knowledge Posttest*

Because we had developed a VR simulation intended for training, we also included a knowledge posttest consisting of 20 multiple-choice questions correlating to each of the 20 subtasks.

### *A.6.3 Materials*

We used an HTC Vive system, including the HMD and both handheld controllers. The Vive HMD provided a 110° diagonal field of view with a display resolution of 1080 x 1200 pixels per eye and a 90Hz refresh rate. The HMD was retrofitted with a Vive audio strap. The VR training application was developed in Unity to maintain framerates of 90 frames per second to match the Vive's refresh rate. The SteamVR plugin for Unity was used to process the Vive's input data.

### *A.6.4 Procedure*

The following procedure was reviewed and approved by the University of Texas at Dallas Institutional Review Board (IRB). The study consisted of one session for each participant, which lasted approximately 60 minutes. After informed consent, each participant completed a background survey to capture the participant's demographics, education, and technology experience. The participant would then don the HTC Vive and experience the SteamVR tutorial to learn how to use the Vive. The participant would then experience the VR training application. After successfully completing the VR training application, the participant would be administered the SSQ, SPES, SUS, scenario fidelity questionnaire, and the knowledge test, which concluded the session.

### A.6.5 Participants

A total of 20 participants (4 females, 16 males) were recruited through university mailing lists for this study. As an exclusion criterion, none of the participants had prior experience or knowledge of surgical robots. The overall average age was  $23.6 \pm 5.8$  years, within a range of 19 to 45 years. Based on self-reported background data, 18 participants played video games on a regular basis (i.e., at least one hour per week) and 12 participants had prior VR experiences.

## A.7 Case Study: The Subtask Analyses and Results

Here, we describe the results of our study, including analyzing the subtasks of our VR simulation, investigating a useful threshold for the time metrics based on the SME completion times, and inspecting the results of our questionnaires and knowledge test.

### A.7.1 Subtask Analyses of Completion Times

Because the subtasks naturally varied in completion times in the real world (see Table A.1), we could not directly compare their completion times logged by the VR simulation. For example, subtask #1 (check error message) would most likely be significantly better than subtask #19 (use port clutch), even if it were poorly implemented. Hence, the original SME completion times ( $T_S$ ) would need to be used to establish thresholds for determining if a subtask passed or failed.

However, the question emerged on how best to establish a useful threshold for each subtask based on the original  $T_S$ . One option was to use  $1T_S$ , which would require the virtual subtasks to be completed as fast or faster than the real-world subtasks. Other possibilities that we considered were to use  $2T_S$ ,  $3T_S$ , and  $4T_S$ . For example, for subtask #19, which took 10s, these thresholds

would be 20s, 30s, and 40s, respectively.

For each threshold possibility, we conducted a one-way RM ANOVA at a 95% confidence level to investigate the main effect of subtask. Degrees of freedom were corrected using Greenhouse-Geisser estimates of sphericity when Mauchly's test of sphericity indicated that the assumption of sphericity had been violated.

For the  $1T_S$  threshold, we found a significant main effect of subtask,  $F(8.422, 160.009) = 11.415, p < 0.001, \eta^2 = 0.375$ . We also found significant main effects for the  $2T_S$  threshold,  $F(7.118, 135.249) = 7.538, p < 0.001, \eta^2 = 0.284$ , for the  $3T_S$  threshold,  $F(5.921, 112.496) = 6.590, p < 0.001, \eta^2 = 0.258$ , and for the  $4T_S$  threshold,  $F(4.746, 90.171) = 4.463, p = 0.001, \eta^2 = 0.190$ . Using Bonferroni post hoc tests, we found several subpar subtasks, which we define as any subtask significantly different from the best subtask. Table A.3 shows the results of these post hoc tests.

Interestingly, the  $1T_S$  threshold was the only one to yield more than one subpar subtask. The  $2T_S$  and  $3T_S$  thresholds both indicated that subtask #7 (hold instrument carriage) was a subpar subtask. Most interestingly, the post hoc tests for the  $4T_S$  threshold did not yield any significantly different pairs of subtasks, despite the RM ANOVA indicating statistical significance. This is due to the Bonferroni tests accounting for the numerous comparisons.

### A.7.2 *Subtask Analyses of Errors*

Another methodology question that emerged from our analyses was whether to analyze the number of errors per subtask, which our VR simulation automatically logged, or to analyze the percentage of participants that completed each subtask without error (i.e., convert any positive number of errors to 0 and any instances of no errors to 1). We followed the same statistical procedure to conduct a one-way RM ANOVA at 95% confidence level for each approach to considering errors.

Table A.2: The percentage of participants that successfully completed each subtask, based on the four proposed time thresholds. Asterisks indicate subpar subtasks.

Subtask	$1T_S$	$2T_S$	$3T_S$	$4T_S$
1	45%	75%	100%	100%
2	<b>*25%</b>	80%	85%	90%
3	75%	85%	95%	100%
4	85%	100%	100%	100%
5	85%	90%	95%	95%
6	80%	95%	95%	100%
7	<b>*0%</b>	<b>*25%</b>	<b>*40%</b>	100%
8	80%	95%	95%	95%
9	<b>*25%</b>	55%	70%	90%
10	70%	95%	100%	100%
11	70%	90%	90%	95%
12	<b>*0%</b>	55%	85%	95%
13	80%	95%	95%	95%
14	80%	100%	100%	100%
15	95%	100%	100%	100%
16	75%	85%	100%	100%
17	<b>*10%</b>	100%	100%	100%
18	55%	90%	95%	100%
19	45%	70%	75%	85%
20	60%	95%	100%	100%

For the mean number of errors, we found a significant main effect of subtask,  $F(3.724, 70.750) = 4.807, p = 0.002, \eta^2 = 0.202$ . For success percentages (i.e., no errors), we also found a significant main effect of subtask,  $F(6.074, 115.406) = 6.607, p < 0.001, \eta^2 = 0.258$ . Interestingly, for the mean number of errors, the Bonferroni post hoc tests did not yield any significantly different pairs of subtasks. However, they did indicate the subtask #9 (rotate wrench) was significantly different, based on the percentage of success (i.e., not committing an error).

Table A.3: The mean number of errors and percentage of participants that successfully (without errors) completed each subtask. Asterisks indicate subpar subtasks.

<b>Subtask</b>	<b>Errors</b>	<b>No Errors</b>
1	0.00	100%
2	0.30	80%
3	0.10	95%
4	0.15	85%
5	0.60	85%
6	0.15	85%
7	0.00	100%
8	0.00	100%
9	1.50	<b>*35%</b>
10	0.05	95%
11	0.45	65%
12	0.30	95%
13	0.05	95%
14	0.00	100%
15	0.00	100%
16	0.00	100%
17	0.05	95%
18	0.00	100%
19	0.10	90%
20	0.15	85%

### A.7.3 Questionnaire Results

Table A.4 shows the means and standard deviations for our SSQ, SPES, and SUS results. Our simulator sickness results were higher than the acceptable total score of 15 indicated by Kennedy et al. [55]; however, they were similar or better than recently reported SSQ results that used modern VR HMDs [122, 114, 70]. Our presence results were also similar to recent SPES results that also used the HTC Vive [34]. Finally, our perceived usability results were above the average SUS score

Table A.4: Descriptive statistics for standardized questionnaires.

<b>Questionnaire</b>	<b>Mean</b>	<b>Std. Dev</b>
Simulator Sickness Questionnaire	19.8	14.8
Spatial Presence Experience Scale	3.93	0.70
System Usability Scale	76.5	13.1

Table A.5: Descriptive statistics for scenario fidelity questions.

<b>Scenario Fidelity Aspect</b>	<b>Mean</b>	<b>Std. Dev</b>
1. The appearances of people were realistic.	3.05	1.15
2. The sounds made by people were realistic.	4.30	0.80
3. The movements made by people were realistic.	3.15	1.18
4. The interactions of people were realistic.	3.45	1.05
5. The people belonged in the environment.	4.30	0.73
6. The appearances of objects were realistic.	4.05	0.88
7. The sounds made by objects were realistic.	4.30	0.92
8. The movements made by objects were realistic.	4.15	0.93
9. The interactions of objects were realistic.	3.85	1.18
10. The objects belonged in the environment.	3.93	0.65

of 68 [14]. Table A.5 shows the results of our custom scenario fidelity questionnaire. Generally, it appears that the scenario fidelity of our simulation was acceptable.

#### *A.7.4 Knowledge Posttest Results*

For our knowledge posttest, we first examined the Cronbach's alpha for its results to determine its reliability. We found the alpha value to be 0.406, which is far below the acceptable value of 0.7 for such instruments [71]. We then inspected the percentage of participants that correctly answered

Table A.6: Comparison of success on the posttest, time, and errors.

<b>Subtask</b>	<b>Posttest</b>	<b>Time</b>	<b>No Errors</b>
1	25%	45%	100%
2	45%	25%	80%
3	70%	75%	95%
4	80%	85%	85%
5	40%	85%	85%
6	0%	80%	85%
7	30%	0%	100%
8	70%	80%	100%
9	90%	25%	35%
10	10%	70%	95%
11	70%	70%	65%
12	35%	0%	95%
13	55%	80%	95%
14	25%	80%	100%
15	80%	95%	100%
16	70%	75%	100%
17	35%	10%	95%
18	5%	55%	100%
19	15%	45%	90%
20	20%	60%	85%

each question and compared it to the prior percentages for completion times ( $1T_S$ ) and errors. Given the comparison (see Table A.6), it is clear that our knowledge test questions were highly variable and unreliable. Hence, we did not conduct a Kruskal-Wallis H test to determine whether the subtasks were significantly different with regard to the posttest.



## A.8 Case Study: The Interaction Inspection

Based on the previous subtask analyses, we identified five subpar subtasks (#2, #7, #9, #12, #17) to inspect for usability issues. For each inspection, we referenced any critical incidents noted by the evaluator for the subtask and reviewed the implementation of the interaction techniques and cues involved in the subtask.

### *A.8.1 Subtask #2 Inspection: Consult surgeon*

This is the first dialog selection subtask that the user makes in our VR simulation, so we expected them to perform slightly worse here. However, inspection of our software revealed that our attempt to avoid clutter, by displaying only one floating window at a time, was prioritizing the staff window if both agents were near the center of the user's field of view. For subtask #2, this naturally occurs due to the preceding subtask of walking to the vision cart and checking the error message.

An easy fix for this issue would be to simply prioritize the surgeon window over the staff window. However, we had observed that some participants did not realize that the dialog options between the windows differ. For example, the user can ask the staff for the grip release wrench, but not the surgeon. Similarly, the surgeon can be consulted about how to proceed, but not the staff. Hence, we have decided to reverse our design choice regarding clutter and to show both windows at the same time.

### *A.8.2 Subtask #7 Inspection: Hold instrument carriage*

This was the first subtask in our VR simulation that required using both hands. Due to the preceding subtask #5, the user is holding the grip release wrench in one of their virtual hands and must use

the other virtual hand to grab the instrument carriage. However, several participants did not realize that the interaction cue for this subtask was originating from their other virtual hand and instead attempted to use the virtual hand holding the wrench to grab the carriage. This would result in the wrench being dropped to the floor, which counts as an error due to breaking sterility.

In the evaluated version of our VR simulation, the virtual hands were simply represented by the HTC Vive controllers, which offer little feedback in terms of handedness. We have decided to incorporate left and right models of human hands holding each controller to better convey handedness for both the virtual hands and the virtual-hand interaction cues. We are also considering incorporating “L” and “R” labels into the representations to help users understand when bimanual interactions are expected.

### *A.8.3 Subtask #9 Inspection: Rotate wrench*

We observed that some participants would attempt to rotate the wrench immediately after inserting it and before the virtual agents prompted them to. This often would result in our VR simulation not recognizing their action or recognizing the action as an error. Upon inspecting our software, we discovered that the start angle for calculating the completion of this subtask was being set after the virtual agents prompted the subtask and not when the wrench is first inserted. Hence, this caused discrepancies between what the participants expected and what the system expected.

We have decided to set the start angle for the subtask rotation calculation when the preceding subtask is completed, as opposed to when this subtask is prompted.

#### *A.8.4 Subtask #12 Inspection: Remove instrument*

Based on our observations, this issue appeared to be caused by the interaction cue for removing the instrument animating too slowly. Upon inspection of our software, we discovered that the preceding interaction cue for grabbing the instrument was never being replaced by the interaction cue for removing the instrument. Due to participants grabbing the instrument off center, the cue for grabbing the instrument would appear to be a slow animation.

We have decided to fix this issue by replacing the interaction cue for grabbing the instrument with the intended interaction cue for removing the instrument.

#### *A.8.5 Subtask #17 Inspection: Use cannula lever*

From our study, we observed that participants had difficulty locating the cannula mount lever, which is largely occluded from most perspectives due to its inherent position in the space. Upon inspecting our software, we also determined that the collision volume used for interacting with the cannula mount level is the smallest interactable collider in the virtual scene (2.5cm radius).

We have decided to modify this subtask to include a walking component and “Stand Here” cue to provide the user with a better perspective to see the cannula mount lever. Additionally, we have decided to increase the size of the interactable collider and to use the VOTE technique [83] with closest intersection disambiguation [84] to address potential issues with multiple selections.

## A.9 Discussion

### *A.9.1 A Useful Formative Evaluation Methodology*

Through the presented case study, we have demonstrated that our formative evaluation methodology is useful for identifying usability issues and improving the interaction design of VR systems that simulate real-world tasks. By employing CARAI, we were able to identify five subpar subtasks in our VR simulation. Based upon inspections of our observations and software, we discovered that two subtasks were hindered by poor interaction techniques, two by poor interaction cues, and one by both poor interaction and cue.

Another useful aspect of our formative evaluation methodology is that it can be employed during any stage of the UX lifecycle. Most formative evaluation methodologies have been designed specifically for the early stages of UX lifecycle development. However, as demonstrated by our case study, CARAI can also be used in the late stages of the lifecycle.

### *A.9.2 Recommendations for Subtask Analyses*

In the presented case study, we investigated four potential thresholds for analyzing completion times based on SME real-world completion times ( $T_S$ ) and two approaches for analyzing subtask errors. Based on our current results, we recommend at least using and analyzing  $1T_S$  for subtask completion times. We also recommend analyzing the percentage of participants that completed each subtask without error, as opposed to analyzing the total number of errors per subtask.

### A.9.3 *Limitations*

One of the limitations of our work is that the participants recruited for our empirical, rigorous user study do not well represent our end users (first assistants). In order to conduct a large number of participants, we had to recruit from our university's student population. This resulted in nearly all of our participants having played video games on a regular basis and more than half of our participants having prior VR experiences, which are both uncommon among first assistants. These also likely do not represent a broad population. Furthermore, we had a low ratio of female participants to male participants (4 to 16).

Another limitation of our research was the poor reliability of the knowledge posttest that we administered in our study. In our prior formative quasi-empirical evaluation with SMEs, we did not administer a knowledge posttest. Hence, we did not have a validated instrument [71] prior to conducting our rigorous user study. Since completing the study and our analyses, we have discovered guidelines for creating test questions (e.g., [113, 13]). Upon inspection, our posttest questions clearly violated several of these guidelines, including:

1. Make sure the stem asks a clear question that can be answered without looking at the options.
2. Avoid placing information in the stem that is not required to answer the question.
3. Avoid overly wordy stems.
4. Avoid logical clues in the stem.
5. Make all options equal in length and amount of detail.

## A.10 Conclusion

We have presented a novel formative evaluation methodology designed specifically for identifying usability issues and improving the interaction design of VR systems that simulate real-world tasks and actions. We have also discussed the application of our methodology in a case study involving a VR simulation for training troubleshooting skills for surgical robots. Through this appendix, we have demonstrated that our methodology is capable and useful for identifying usability issues. We have also investigated how to analyze user performance based on subtasks and have provided recommendations for such analyses.

For future work, we plan to conduct a summative evaluation of our VR training simulation, using the same data collection and analyses techniques, to determine how effective our formative evaluation and improvements have been. We are also planning to use our formative evaluation methodology to improve another VR training simulation involving the inspection of haul trucks.

## A.11 Acknowledgments

This material is based on work partially supported by the National Science Foundation under Grant No. 1552344 – “CAREER: Leveraging the Virtualness of Virtual Reality for More-Effective Training.”

**APPENDIX B: ALL RESULTS FROM SYSTEMATIC INVESTIGATION  
OF DEVICE COMBINATIONS AND SPATIAL REPRESENTATIONS  
FOR IDENTIFYING VIRTUAL REALITY USERS IN AN ASSEMBLY  
TRAINING ENVIRONMENT**

This appendix contains the full set of results from the systematic investigation in chapter 6.

### B.1 Random Forest Results

<b>Position <math>A \rightarrow A</math> RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	95.1%	95.3%	95.8%	85.6%	85.9%	85.4%	87.8%
<b>Head + DomH</b>	94.7%	93.4%	94.3%	78.3%	74.4%	73.0%	75.6%
<b>Head + OffH</b>	93.9%	93.7%	94.3%	79.7%	76.6%	76.8%	77.6%
<b>DomH + OffH</b>	88.7%	87.9%	88.4%	64.4%	71.0%	72.8%	71.3%
<b>Head</b>	85.1%	85.0%	86.0%	53.2%	42.9%	40.7%	44.2%
<b>DomH</b>	66.6%	67.0%	67.4%	41.7%	46.7%	45.7%	45.7%
<b>OffH</b>	71.2%	72.2%	71.9%	39.4%	49.9%	45.7%	47.8%

<b>Position <math>B \rightarrow B</math> RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	94.2%	95.7%	95.0%	81.8%	83.1%	83.0%	84.1%
<b>Head + DomH</b>	91.8%	91.0%	91.0%	72.2%	70.6%	68.6%	69.3%
<b>Head + OffH</b>	94.4%	93.2%	94.4%	77.3%	75.7%	75.3%	76.6%
<b>DomH + OffH</b>	86.0%	87.3%	85.9%	59.6%	71.8%	73.8%	70.8%
<b>Head</b>	84.7%	81.3%	83.8%	48.0%	37.4%	36.9%	40.1%
<b>DomH</b>	61.6%	61.2%	59.7%	32.2%	47.3%	49.2%	49.4%
<b>OffH</b>	71.0%	71.7%	70.4%	40.1%	56.0%	56.2%	51.8%



<b>Position A <math>\rightarrow</math> B RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	51.1%	44.4%	46.7%	42.2%	46.7%	40.0%	44.4%
<b>Head + DomH</b>	53.3%	53.3%	48.9%	40.0%	35.6%	40.0%	33.3%
<b>Head + OffH</b>	57.8%	46.7%	51.1%	44.4%	37.8%	40.0%	40.0%
<b>DomH + OffH</b>	33.3%	33.3%	35.6%	22.2%	33.3%	33.3%	31.1%
<b>Head</b>	75.6%	71.1%	75.6%	42.2%	40.0%	31.1%	31.1%
<b>DomH</b>	31.1%	35.6%	31.1%	20.0%	31.1%	28.9%	26.7%
<b>OffH</b>	26.7%	24.4%	31.1%	15.6%	17.8%	17.8%	22.2%

<b>Position B <math>\rightarrow</math> A RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	51.1%	51.1%	53.3%	40.0%	46.7%	46.7%	44.4%
<b>Head + DomH</b>	57.8%	51.1%	55.6%	40.0%	42.2%	40.0%	40.0%
<b>Head + OffH</b>	62.2%	62.2%	62.2%	37.8%	40.0%	40.0%	40.0%
<b>DomH + OffH</b>	40.0%	37.8%	37.8%	31.1%	33.3%	31.1%	33.3%
<b>Head</b>	80.0%	75.6%	82.2%	33.3%	51.1%	31.1%	42.2%
<b>DomH</b>	35.6%	35.6%	35.6%	15.6%	31.1%	31.1%	24.4%
<b>OffH</b>	31.1%	31.1%	31.1%	22.2%	24.4%	22.2%	22.2%

<b>Velocity <math>A \rightarrow A</math> RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	54.2%	59.0%	64.1%	41.3%	45.7%	52.4%	62.0%
<b>Head + DomH</b>	48.6%	52.7%	60.3%	37.4%	40.4%	46.4%	54.8%
<b>Head + OffH</b>	44.8%	47.9%	53.8%	30.7%	36.2%	42.8%	48.0%
<b>DomH + OffH</b>	49.4%	56.0%	59.6%	39.3%	40.8%	51.0%	54.9%
<b>Head</b>	29.6%	32.2%	36.1%	18.3%	24.4%	28.1%	28.4%
<b>DomH</b>	42.4%	45.1%	49.1%	25.8%	32.3%	38.6%	43.7%
<b>OffH</b>	36.1%	45.1%	50.1%	27.7%	26.7%	37.7%	44.8%

<b>Velocity <math>B \rightarrow B</math> RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	58.7%	63.6%	66.2%	46.3%	53.0%	59.7%	64.2%
<b>Head + DomH</b>	51.8%	54.2%	58.3%	35.1%	41.7%	48.4%	54.6%
<b>Head + OffH</b>	50.9%	55.7%	60.8%	36.9%	41.4%	50.9%	56.3%
<b>DomH + OffH</b>	52.1%	58.7%	60.9%	36.7%	44.6%	53.4%	56.9%
<b>Head</b>	30.2%	31.9%	36.6%	20.2%	22.3%	25.6%	31.2%
<b>DomH</b>	41.6%	44.2%	46.4%	24.1%	27.4%	39.1%	43.2%
<b>OffH</b>	40.6%	49.0%	49.9%	26.2%	29.1%	40.1%	42.7%

<b>Velocity <math>A \rightarrow B</math> RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	28.9%	28.9%	35.6%	17.8%	26.7%	28.9%	33.3%
<b>Head + DomH</b>	28.9%	24.4%	33.3%	15.6%	20.0%	28.9%	31.1%
<b>Head + OffH</b>	28.9%	31.1%	24.4%	15.6%	22.2%	22.2%	24.4%
<b>DomH + OffH</b>	22.2%	22.2%	26.7%	13.3%	17.8%	22.2%	24.4%
<b>Head</b>	33.3%	42.2%	44.4%	15.6%	28.9%	28.9%	33.3%
<b>DomH</b>	20.0%	20.0%	24.4%	15.6%	11.1%	11.1%	24.4%
<b>OffH</b>	24.4%	24.4%	20.0%	11.1%	20.0%	17.8%	15.6%

<b>Velocity <math>B \rightarrow A</math> RF</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	42.2%	44.4%	40.0%	33.3%	31.1%	40.0%	40.0%
<b>Head + DomH</b>	37.8%	35.6%	42.2%	28.9%	26.7%	28.9%	31.1%
<b>Head + OffH</b>	26.7%	35.6%	35.6%	17.8%	15.6%	35.6%	40.0%
<b>DomH + OffH</b>	31.1%	37.8%	28.9%	17.8%	22.2%	28.9%	26.7%
<b>Head</b>	28.9%	35.6%	46.7%	17.8%	17.8%	26.7%	33.3%
<b>DomH</b>	28.9%	26.7%	26.7%	13.3%	15.6%	20.0%	17.8%
<b>OffH</b>	22.2%	26.7%	24.4%	11.1%	11.1%	17.8%	20.0%

## B.2 Gradient Boosting Machine Results

<b>Position A → A GBM</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	95.3%	96.0%	95.7%	85.0%	84.7%	82.3%	85.4%
<b>Head + DomH</b>	93.8%	93.1%	93.7%	75.3%	71.6%	68.3%	73.3%
<b>Head + OffH</b>	92.3%	92.0%	92.7%	75.9%	71.0%	72.3%	72.8%
<b>DomH + OffH</b>	87.1%	87.0%	88.3%	58.7%	68.1%	69.8%	70.0%
<b>Head</b>	82.7%	79.6%	83.4%	45.6%	35.9%	33.6%	36.0%
<b>DomH</b>	64.0%	62.7%	63.1%	36.2%	39.4%	43.2%	42.6%
<b>OffH</b>	68.7%	67.4%	66.4%	32.3%	41.9%	45.2%	42.7%

<b>Position B → B GBM</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	94.0%	93.2%	93.0%	80.6%	82.9%	81.0%	82.1%
<b>Head + DomH</b>	89.9%	88.7%	87.7%	67.6%	68.8%	67.2%	66.8%
<b>Head + OffH</b>	92.1%	91.8%	91.3%	73.0%	74.2%	70.6%	75.1%
<b>DomH + OffH</b>	83.9%	84.6%	83.0%	57.4%	70.2%	69.8%	69.1%
<b>Head</b>	79.9%	74.9%	79.1%	40.9%	36.1%	30.6%	36.6%
<b>DomH</b>	54.6%	55.8%	54.8%	28.7%	40.8%	44.1%	41.0%
<b>OffH</b>	66.9%	68.4%	65.6%	34.1%	46.7%	47.2%	49.0%

<b>Position A → B GBM</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	46.7%	48.9%	51.1%	44.4%	48.9%	46.7%	46.7%
<b>Head + DomH</b>	51.1%	51.1%	51.1%	40.0%	37.8%	37.8%	37.8%
<b>Head + OffH</b>	53.3%	53.3%	53.3%	44.4%	35.6%	40.0%	40.0%
<b>DomH + OffH</b>	35.6%	37.8%	35.6%	20.0%	33.3%	33.3%	35.6%
<b>Head</b>	73.3%	64.4%	71.1%	42.2%	26.7%	31.1%	46.7%
<b>DomH</b>	26.7%	31.1%	26.7%	15.6%	26.7%	26.7%	28.9%
<b>OffH</b>	24.4%	26.7%	31.1%	15.6%	15.6%	24.4%	13.3%

<b>Position B → A GBM</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	46.7%	46.7%	48.9%	46.7%	48.9%	48.9%	48.9%
<b>Head + DomH</b>	57.8%	53.3%	57.8%	46.7%	44.4%	42.2%	46.7%
<b>Head + OffH</b>	62.2%	57.8%	60.0%	42.2%	51.1%	44.4%	53.3%
<b>DomH + OffH</b>	44.4%	40.0%	42.2%	35.6%	35.6%	33.3%	33.3%
<b>Head</b>	73.3%	75.6%	80.0%	33.3%	35.6%	31.1%	42.2%
<b>DomH</b>	31.1%	35.6%	33.3%	24.4%	26.7%	20.0%	24.4%
<b>OffH</b>	35.6%	35.6%	28.9%	20.0%	20.0%	24.4%	17.8%

<b>Velocity <math>A \rightarrow A</math> GBM</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	55.2%	56.2%	64.3%	39.1%	44.9%	54.0%	62.3%
<b>Head + DomH</b>	47.3%	51.7%	59.3%	31.0%	36.1%	42.1%	50.8%
<b>Head + OffH</b>	44.4%	45.9%	53.3%	25.8%	31.2%	38.8%	49.8%
<b>DomH + OffH</b>	49.0%	50.4%	58.4%	31.8%	40.1%	44.2%	54.4%
<b>Head</b>	26.6%	29.3%	28.8%	14.4%	17.1%	20.7%	26.0%
<b>DomH</b>	34.3%	38.4%	44.6%	20.8%	24.8%	32.9%	37.1%
<b>OffH</b>	30.0%	35.8%	42.7%	18.6%	22.7%	29.9%	37.9%

<b>Velocity <math>B \rightarrow B</math> GBM</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	57.0%	62.7%	68.0%	37.0%	45.8%	55.2%	65.1%
<b>Head + DomH</b>	47.3%	49.3%	55.0%	27.6%	33.1%	40.8%	51.1%
<b>Head + OffH</b>	44.8%	51.3%	58.2%	27.8%	38.6%	40.2%	51.9%
<b>DomH + OffH</b>	46.4%	55.0%	56.2%	30.3%	37.8%	44.3%	53.7%
<b>Head</b>	25.6%	24.7%	31.2%	13.9%	17.9%	20.7%	26.2%
<b>DomH</b>	28.2%	35.0%	39.4%	17.9%	19.8%	29.1%	34.2%
<b>OffH</b>	31.6%	37.8%	42.8%	18.6%	24.1%	31.6%	35.0%

<b>Velocity A → B GBM</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	42.2%	44.4%	46.7%	26.7%	35.6%	40.0%	37.8%
<b>Head + DomH</b>	33.3%	44.4%	33.3%	13.3%	22.2%	28.9%	40.0%
<b>Head + OffH</b>	28.9%	37.8%	37.8%	15.6%	26.7%	31.1%	28.9%
<b>DomH + OffH</b>	33.3%	31.1%	31.1%	13.3%	15.6%	26.7%	26.7%
<b>Head</b>	28.9%	33.3%	37.8%	13.3%	22.2%	28.9%	28.9%
<b>DomH</b>	17.8%	20.0%	20.0%	13.3%	8.9%	11.1%	22.2%
<b>OffH</b>	15.6%	22.2%	22.2%	6.7%	13.3%	13.3%	13.3%

<b>Velocity B → A GBM</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	44.4%	46.7%	46.7%	26.7%	31.1%	37.8%	40.0%
<b>Head + DomH</b>	33.3%	37.8%	48.9%	17.8%	31.1%	31.1%	42.2%
<b>Head + OffH</b>	33.3%	42.2%	44.4%	17.8%	24.4%	33.3%	37.8%
<b>DomH + OffH</b>	33.3%	37.8%	35.6%	20.0%	26.7%	35.6%	28.9%
<b>Head</b>	31.1%	26.7%	40.0%	11.1%	17.8%	22.2%	31.1%
<b>DomH</b>	24.4%	17.8%	28.9%	8.9%	15.6%	15.6%	17.8%
<b>OffH</b>	24.4%	20.0%	22.2%	8.9%	13.3%	24.4%	17.8%

### B.3 K Nearest Neighbors Results

<b>Position <math>A \rightarrow A</math> kNN</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	90.3%	90.0%	91.3%	74.9%	82.0%	81.7%	81.3%
<b>Head + DomH</b>	85.8%	84.6%	84.8%	69.4%	64.2%	65.7%	66.1%
<b>Head + OffH</b>	87.3%	87.4%	86.8%	72.6%	68.2%	71.9%	71.0%
<b>DomH + OffH</b>	78.4%	79.3%	79.2%	48.8%	66.4%	64.3%	65.2%
<b>Head</b>	71.8%	69.6%	67.8%	43.2%	31.6%	27.1%	30.4%
<b>DomH</b>	57.7%	55.6%	57.8%	33.7%	38.7%	37.1%	37.3%
<b>OffH</b>	59.7%	60.6%	60.7%	29.6%	41.1%	40.2%	38.6%

<b>Position <math>B \rightarrow B</math> kNN</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	86.1%	86.3%	87.4%	66.2%	82.7%	80.6%	80.7%
<b>Head + DomH</b>	79.8%	79.2%	79.7%	60.6%	61.1%	58.4%	60.9%
<b>Head + OffH</b>	86.1%	86.8%	86.8%	65.2%	72.3%	72.3%	73.3%
<b>DomH + OffH</b>	72.9%	75.3%	76.3%	45.4%	68.4%	66.7%	66.2%
<b>Head</b>	66.7%	64.3%	64.0%	35.9%	28.4%	29.6%	29.9%
<b>DomH</b>	49.1%	52.9%	54.4%	26.6%	37.3%	36.7%	40.3%
<b>OffH</b>	67.3%	65.8%	64.4%	34.0%	43.3%	45.6%	45.0%



<b>Position <math>A \rightarrow B</math> kNN</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	48.9%	44.4%	46.7%	28.9%	48.9%	53.3%	51.1%
<b>Head + DomH</b>	46.7%	46.7%	44.4%	24.4%	44.4%	28.9%	40.0%
<b>Head + OffH</b>	44.4%	44.4%	44.4%	26.7%	37.8%	35.6%	35.6%
<b>DomH + OffH</b>	35.6%	33.3%	33.3%	24.4%	33.3%	35.6%	33.3%
<b>Head</b>	75.6%	62.2%	66.7%	33.3%	35.6%	17.8%	31.1%
<b>DomH</b>	28.9%	24.4%	24.4%	13.3%	24.4%	20.0%	24.4%
<b>OffH</b>	26.7%	24.4%	20.0%	15.6%	22.2%	15.6%	20.0%

<b>Position <math>B \rightarrow A</math> kNN</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	44.4%	46.7%	44.4%	37.8%	46.7%	46.7%	48.9%
<b>Head + DomH</b>	46.7%	44.4%	42.2%	40.0%	42.2%	33.3%	37.8%
<b>Head + OffH</b>	46.7%	46.7%	46.7%	31.1%	33.3%	35.6%	35.6%
<b>DomH + OffH</b>	35.6%	44.4%	42.2%	28.9%	35.6%	35.6%	33.3%
<b>Head</b>	71.1%	68.9%	68.9%	35.6%	26.7%	24.4%	20.0%
<b>DomH</b>	26.7%	24.4%	26.7%	15.6%	20.0%	20.0%	17.8%
<b>OffH</b>	26.7%	24.4%	26.7%	22.2%	13.3%	17.8%	17.8%

<b>Velocity <math>A \rightarrow A</math> kNN</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	29.6%	30.3%	37.2%	22.9%	24.1%	27.4%	34.9%
<b>Head + DomH</b>	25.1%	29.1%	35.4%	19.0%	18.3%	20.9%	31.1%
<b>Head + OffH</b>	25.0%	26.4%	33.8%	15.7%	18.9%	24.6%	31.1%
<b>DomH + OffH</b>	21.3%	26.8%	32.1%	19.9%	14.7%	21.1%	30.6%
<b>Head</b>	15.4%	16.2%	18.7%	9.3%	11.3%	12.9%	15.7%
<b>DomH</b>	18.7%	23.6%	32.1%	15.3%	16.1%	19.7%	30.0%
<b>OffH</b>	17.8%	21.8%	33.1%	12.9%	10.3%	20.4%	31.6%

<b>Velocity <math>B \rightarrow B</math> kNN</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	31.6%	30.6%	40.9%	24.0%	22.0%	27.8%	40.6%
<b>Head + DomH</b>	26.9%	28.7%	36.2%	17.6%	19.1%	22.3%	36.0%
<b>Head + OffH</b>	27.9%	28.8%	40.9%	22.0%	17.7%	23.3%	36.9%
<b>DomH + OffH</b>	24.9%	29.9%	35.6%	17.0%	15.1%	24.1%	31.8%
<b>Head</b>	15.9%	16.8%	21.4%	10.7%	13.0%	12.6%	20.3%
<b>DomH</b>	18.9%	25.6%	33.2%	11.0%	10.2%	18.2%	28.0%
<b>OffH</b>	21.2%	28.0%	36.4%	15.3%	13.4%	21.8%	30.3%

<b>Velocity <math>A \rightarrow B</math> kNN</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	15.6%	17.8%	22.2%	13.3%	15.6%	11.1%	17.8%
<b>Head + DomH</b>	22.2%	17.8%	22.2%	11.1%	17.8%	15.6%	17.8%
<b>Head + OffH</b>	15.6%	17.8%	24.4%	13.3%	15.6%	15.6%	20.0%
<b>DomH + OffH</b>	11.1%	17.8%	15.6%	8.9%	6.7%	11.1%	15.6%
<b>Head</b>	20.0%	22.2%	31.1%	15.6%	13.3%	17.8%	20.0%
<b>DomH</b>	13.3%	13.3%	13.3%	4.4%	6.7%	8.9%	11.1%
<b>OffH</b>	11.1%	13.3%	17.8%	4.4%	4.4%	8.9%	13.3%

<b>Velocity <math>B \rightarrow A</math> kNN</b>	<b>Position + Euler</b>	<b>Position + Quaternion</b>	<b>Position + SixD</b>	<b>Position</b>	<b>Euler</b>	<b>Quaternion</b>	<b>SixD</b>
<b>Head + DomH + OffH</b>	13.3%	17.8%	24.4%	6.7%	11.1%	11.1%	17.8%
<b>Head + DomH</b>	13.3%	11.1%	24.4%	13.3%	15.6%	11.1%	17.8%
<b>Head + OffH</b>	11.1%	20.0%	22.2%	6.7%	0.0%	15.6%	20.0%
<b>DomH + OffH</b>	8.9%	6.7%	11.1%	8.9%	4.4%	8.9%	13.3%
<b>Head</b>	17.8%	15.6%	20.0%	13.3%	11.1%	13.3%	24.4%
<b>DomH</b>	6.7%	8.9%	17.8%	4.4%	6.7%	11.1%	15.6%
<b>OffH</b>	11.1%	13.3%	13.3%	6.7%	4.4%	11.1%	13.3%

## **APPENDIX C: IRB APPROVAL**



UNIVERSITY OF CENTRAL FLORIDA

**Institutional Review Board**  
FWA00000351  
IRB00001138, IRB00012110  
Office of Research  
12201 Research Parkway  
Orlando, FL 32826-3246

APPROVAL

June 13, 2022

Dear Ryan McMahan:

On 6/13/2022, the IRB reviewed the following submission:

Type of Review:	Initial Study
Title:	Full-scale Assembly Benchmark (FAB) Study
Investigator:	Ryan McMahan
IRB ID:	STUDY00004329
Funding:	Name: Natl Science Fdn (NSF), Grant Office ID: AWD00000270, Funding Source ID: 2021607
Grant ID:	AWD00000270;
IND, IDE, or HDE:	None
Documents Reviewed:	<ul style="list-style-type: none"> <li>• Calendly Confirmation.pdf, Category: Other;</li> <li>• Exit Survey.pdf, Category: Survey / Questionnaire;</li> <li>• HRP-502 - FAB Study - Consent v3.pdf, Category: Consent Form;</li> <li>• HRP-503-FAB-Protocol v5.docx, Category: IRB Protocol;</li> <li>• NASA Task Load Index.pdf, Category: Survey / Questionnaire;</li> <li>• Online Pre-Screening Survey.pdf, Category: Survey / Questionnaire;</li> <li>• Recruitment Flyer.pdf, Category: Recruitment Materials;</li> <li>• Scheduling Email.pdf, Category: Other;</li> <li>• Simulator Sickness Questionnaire.pdf, Category: Survey / Questionnaire;</li> <li>• Spatial Presence Experience Scale.pdf, Category: Survey / Questionnaire;</li> <li>• System Usability Scale.pdf, Category: Survey / Questionnaire;</li> </ul>

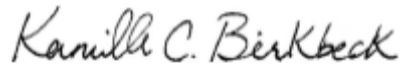
The IRB approved the protocol on 6/13/2022.

In conducting this protocol, you are required to follow the requirements listed in the Investigator Manual (HRP-103), which can be found by navigating to the IRB Library within the IRB system. Guidance on submitting Modifications and a

Continuing Review or Administrative Check-in are detailed in the manual. When you have completed your research, please submit a Study Closure request so that IRB records will be accurate.

If you have any questions, please contact the UCF IRB at 407-823-2901 or [irb@ucf.edu](mailto:irb@ucf.edu). Please include your project title and IRB number in all correspondence with this office.

Sincerely,

A handwritten signature in cursive script that reads "Kamille C. Birkbeck".

Kamille Birkbeck  
Designated Reviewer

## LIST OF REFERENCES

- [1] A. Ajit, N. K. Banerjee, and S. Banerjee. Combining pairwise feature matches from device trajectories for biometric authentication in virtual reality environments. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 9–97. IEEE Computer Society, 2019.
- [2] S. Amershi and C. Conati. Unsupervised and supervised machine learning in user modeling for intelligent learning environments. *International Conference on Intelligent User Interfaces, Proceedings IUI*, pages 72–81, 2007. doi: 10.1145/1216295.1216315.
- [3] S. M. Asish, A. K. Kulshreshth, and C. W. Borst. User identification utilizing minimal eye-gaze features in virtual reality applications. In *Virtual Worlds*, volume 1, pages 42–61. MDPI, 2022.
- [4] M. Azmandian, T. Grechkin, M. Bolas, and E. Suma. The redirected walking toolkit: a unified development platform for exploring large virtual environments. In *2016 IEEE 2nd Workshop on Everyday Virtual Reality (WEVR)*, pages 9–14. IEEE, 2016.
- [5] S. J. Bagnato, J. T. Neisworth, and A. Capone. Curriculum-based assessment for the young exceptional child: Rationale and review. *Topics in early childhood special education*, 6(2): 97–110, 1986.
- [6] S. P. Banerjee and D. L. Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1):116–139, 2012.
- [7] J. Bertrand, D. Brickler, S. Babu, K. Madathil, M. Zelaya, T. Wang, J. Wagner, A. Gramopadhye, and J. Luo. The role of dimensional symmetry on bimanual psychomo-

- tor skills education in immersive virtual environments. In *2015 IEEE Virtual Reality (VR)*, pages 3–10, 2015. ISBN VO -. doi: 10.1109/VR.2015.7223317.
- [8] A. Blum, S. Har-Peled, and B. Raichel. Sparse approximation via generating point sets. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 548–557, 2016.
- [9] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.
- [10] D. A. Bowman and R. P. McMahan. Virtual Reality: How Much Immersion Is Enough? *Computer*, 40(7):36–43, 2007.
- [11] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. An introduction to 3-d user interface design. *Presence*, 10(1):96–108, 2001.
- [12] D. A. Bowman, J. L. Gabbard, and D. Hix. A survey of usability evaluation in virtual environments: classification and comparison of methods. *Presence: Teleoperators & Virtual Environments*, 11(4):404–424, 2002.
- [13] J. Breakall, C. Randles, and R. Tasker. Development and use of a multiple-choice item writing flaws evaluation instrument in the context of general chemistry. *Chemistry Education Research and Practice*, 20(2):369–382, 2019.
- [14] J. Brooke. SUS: A Retrospective. *J. Usability Studies*, 8(2):29–40, feb 2013. ISSN 1931-3357.
- [15] J. M. Carroll, M. K. Singley, and M. B. Rosson. Integrating theory development with design evaluation. *Behaviour & Information Technology*, 11(5):247–255, 1992.



- [16] D. W. Carruth. Virtual reality for education and workforce training. In *2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 1–6, 2017. ISBN VO -. doi: 10.1109/ICETA.2017.8102472.
- [17] J. Chen, N. Cheng, G. Cacciamani, P. Oh, M. Lin-Brandt, D. Remulla, I. S. Gill, and A. J. Hung. Objective assessment of robotic surgical technical skill: a systematic review. *The Journal of urology*, 201(3):461–469, 2019.
- [18] D. Chicco and G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13, 2020.
- [19] W. R. Chitwood Jr, L. W. Nifong, W. H. Chapman, J. E. Felger, B. M. Bailey, T. Ballint, K. G. Mendleson, V. B. Kim, J. A. Young, and R. A. Albrecht. Robotic surgical training in an academic institution. *Annals of surgery*, 234(4):475–486, oct 2001. ISSN 0003-4932. doi: 10.1097/00000658-200110000-00007.
- [20] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie. Das3h: modeling student learning and forgetting for optimally scheduling distributed practice of skills. *arXiv preprint arXiv:1905.06873*, 2019.
- [21] G. Cirio, A.-H. Olivier, M. Marchal, and J. Pettre. Kinematic evaluation of virtual walking trajectories. *IEEE transactions on visualization and computer graphics*, 19(4):671–680, 2013.
- [22] A. T. Corbett, K. R. Koedinger, and J. R. Anderson. Chapter 37 - intelligent tutoring systems. In M. G. Helander, T. K. Landauer, and P. V. Prabhu, editors, *Handbook of Human-Computer Interaction (Second Edition)*, pages 849 – 874. North-Holland, Amsterdam, second edition edition, 1997. ISBN 978-0-444-81862-1. doi: <https://doi.org/10.1016/B978-0-444-81862-1>.

1016/B978-044481862-1.50103-5. URL <http://www.sciencedirect.com/science/article/pii/B9780444818621501035>.

- [23] F. Danieau, A. Guillo, and R. Doré. Attention guidance for immersive video content in head-mounted displays. In *2017 IEEE Virtual Reality (VR)*, pages 205–206, 2017. ISBN 2375-5334 VO -. doi: 10.1109/VR.2017.7892248.
- [24] B. David-John, D. Hosfelt, K. Butler, and E. Jain. A privacy-preserving approach to streaming eye-tracking data. *IEEE Transactions on Visualization and Computer Graphics*, 27(5): 2555–2565, 2021. doi: 10.1109/TVCG.2021.3067787.
- [25] B. David-John, C. Peacock, T. Zhang, T. S. Murdison, H. Benko, and T. R. Jonker. Towards gaze-based prediction of the intent to interact in virtual reality. In *ACM Symposium on Eye Tracking Research and Applications*, pages 1–7, 2021.
- [26] R. M. De Moraes and L. Dos Santos Machado. Online training assessment in virtual reality simulators based on Gaussian Naive Bayes. *World Scientific Proceedings Series on Computer Engineering and Information Science 1; Computational Intelligence in Decision and Control - Proceedings of the 8th International FLINS Conference*, 16:1147–1152, 2008. doi: 10.1142/9789812799470\_0188.
- [27] W. Dick and L. Carey. *The systematic design of instruction.*(editions 1 through 4.) new york, 1978.
- [28] K. R. Dillman, T. T. H. Mok, A. Tang, L. Oehlberg, and A. Mitchell. A visual interaction cue framework from video game environments for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.
- [29] S. D’Mello and A. Graesser. Mind and Body: Dialogue and Posture for Affect Detection

- in Learning Environments. *Artificial Intelligence in Education*, 158:161–168, 2007. ISSN 0922-6389.
- [30] A. D. P. dos Santos. *Using Motion Sensor and Machine Learning to Support the Assessment of Rhythmic Skills in Social Partner Dance: Bridging Teacher, Student and Machine Contexts*. PhD thesis, University of Sydney, 2019.
- [31] J. Drummond and D. Litman. In the zone: Towards detecting student zoning out using supervised machine learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6095 LNCS(PART 2): 306–308, 2010. ISSN 03029743. doi: 10.1007/978-3-642-13437-1\_53.
- [32] S. Eberz, N. Paoletti, M. Roeschlin, M. Kwiatkowska, I. Martinovic, and A. Patané. Broken hearted: How to attack ecg biometrics.(2017). URL: <http://www.nicolapaoletti.com/assets/papers/eberz2017broken.pdf>, 2017.
- [33] S. Eberz, G. Lovisotto, A. Patane, M. Kwiatkowska, V. Lenders, and I. Martinovic. When your fitness tracker betrays you: Quantifying the predictability of biometric features across contexts. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 889–905. IEEE, 2018.
- [34] B. Eckstein, E. Krapp, and B. Lugrin. Towards Serious Games and Applications in Smart Substitutional Reality. In *2018 10th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)*, pages 1–8, 2018. ISBN 2474-0470 VO -. doi: 10.1109/VS-Games.2018.8493444.
- [35] M. Ershad, Z. Koesters, R. Rege, and A. Majewicz. Meaningful assessment of surgical expertise: Semantic labeling with data and crowds. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 508–515. Springer, 2016.

- [36] J. C. Eubanks, V. Somareddy, R. P. McMahan, and A. A. Lopez. Full-body portable virtual reality for personal protective equipment training. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9740, pages 490–501, Cham, 2016. Springer International Publishing. ISBN 9783319399065. doi: 10.1007/978-3-319-39907-2\_47. URL [http://link.springer.com/10.1007/978-3-319-39907-2\\_{\\_}47](http://link.springer.com/10.1007/978-3-319-39907-2_{_}47).
- [37] H. Feng, K. Fawaz, and K. G. Shin. Continuous authentication for voice assistants. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, MobiCom '17*, page 343–355, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349161. doi: 10.1145/3117811.3117823. URL <https://doi.org/10.1145/3117811.3117823>.
- [38] R. Ferber, I. McClay Davis, D. Williams Iii, and C. Laughton. A comparison of within- and between-day reliability of discrete 3d lower extremity variables in runners. *Journal of Orthopaedic Research*, 20(6):1139–1145, 2002.
- [39] J. L. Gabbard, D. Hix, and J. E. Swan. User-centered design and evaluation of virtual environments. *IEEE computer Graphics and Applications*, 19(6):51–59, 1999.
- [40] M. Galliers. Assessment: An innovative approach. *The Vocational Aspect of Education*, 41(110):89–91, 1989.
- [41] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.
- [42] F. Gutierrez and J. Atkinson. Adaptive feedback selection for intelligent tutoring systems. *Expert Systems with Applications*, 38(5):6146–6152, 2011. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2010.11.058>.

- [43] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.
- [44] T. Hartmann, W. Wirth, H. Schramm, C. Klimmt, P. Vorderer, A. Gysbers, S. Böcking, N. Ravaja, J. Laarni, T. Saari, F. Gouveia, and A. Maria Sacau. The Spatial Presence Experience Scale (SPES). *Journal of Media Psychology*, 28(1):1–15, jan 2015. ISSN 1864-1105. doi: 10.1027/1864-1105/a000137.
- [45] R. Hartson and P. Pyla. The ux book—process and guidelines for ensuring a quality of user experience (vol. 1). *Elsevier*, 2012.
- [46] D. Hix and H. R. Hartson. *Developing user interfaces: ensuring usability through product & process*. John Wiley & Sons, Inc., 1993.
- [47] D. Hix, J. E. Swan, J. L. Gabbard, M. McGee, J. Durbin, and T. King. User-centered design and evaluation of a real-time battlefield visualization virtual environment. In *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*, pages 96–103. IEEE, 1999.
- [48] E. Hodgson, E. Bachmann, and D. Waller. Redirected walking to explore virtual environments: Assessing the potential for spatial interference. *ACM Transactions on Applied Perception (TAP)*, 8(4):1–22, 2008.
- [49] X. Hu, A. G. Moore, J. Coleman Eubanks, A. Aiyaz, and R. P. McMahan. Evaluating interaction cue purpose and timing for learning and retaining virtual reality training. In *Symposium on Spatial User Interaction*, pages 1–9, 2020.
- [50] X. Hu, A. G. Moore, J. C. Eubanks, A. A. Aiyaz, and R. P. McMahan. The Effects of Delayed Interaction Cues in Virtual Reality Training. In *2020 IEEE Conference on Virtual*

- Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 63–69, 2020. ISBN VO -. doi: 10.1109/VRW50115.2020.00019.
- [51] A. J. Hung, J. Chen, Z. Che, T. Nilanon, A. Jarc, M. Titus, P. J. Oh, I. S. Gill, and Y. Liu. Utilizing Machine Learning and Automated Performance Metrics to Evaluate Robot-Assisted Radical Prostatectomy Performance and Predict Outcomes. *Journal of Endourology*, 32(5): 438–444, 2018. ISSN 1557900X. doi: 10.1089/end.2018.0035.
- [52] M. S. Iqbal, V. Ramadoss, and M. Zoppi. Dynamic pose tracking performance evaluation of htc vive virtual reality system. *IEEE Access*, 9:3798–3815, 2020.
- [53] K. Johnsen, A. Raij, A. Stevens, D. S. Lind, and B. Lok. The Validity of a Virtual Human Experience for Interpersonal Skills Education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '07*, pages 1049–1058, New York, 2007. Association for Computing Machinery. ISBN 9781595935939. doi: 10.1145/1240624.1240784.
- [54] J. Johnson. *Designing with the Mind in Mind, Second Edition: Simple Guide to Understanding User Interface Design Guidelines*. Morgan Kaufmann Publishers Inc., San Francisco, 2nd edition, 2014. ISBN 0124079148, 9780124079144.
- [55] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator Sickness Questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, 1993. ISSN 1532-7108(Electronic),1050-8414(Print). doi: 10.1207/s15327108ijap0303\_3.
- [56] S. Kerry. Higher education for sustainability: seeking affective learning outcomes. *International Journal of Sustainability in Higher Education*, 9(1):87–98, jan 2008. ISSN 1467-6370. doi: 10.1108/14676370810842201.

- [57] D. R. Krathwohl. A Revision of Bloom’s Taxonomy: An Overview. *Theory Into Practice*, 41(4):212–218, may 2002. ISSN 00405841, 15430421.
- [58] M. Kuhn and K. Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.
- [59] R. Kumar and A. K. Hemal. The ‘scrubbed surgeon’ in robotic surgery. *World journal of urology*, 24(2):144–147, 2006.
- [60] A. Kupin, B. Moeller, Y. Jiang, N. K. Banerjee, and S. Banerjee. Task-driven biometric authentication of users in virtual reality (vr) environments. In *International conference on multimedia modeling*, pages 55–67. Springer, 2019.
- [61] C. Lai, R. P. McMahan, M. Kitagawa, and I. Connolly. Geometry explorer: Facilitating geometry education with virtual reality. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9740, pages 702–713, Cham, 2016. Springer International Publishing. ISBN 9783319399065. doi: 10.1007/978-3-319-39907-2\_67. URL [http://link.springer.com/10.1007/978-3-319-39907-2\\_{\\_}67](http://link.springer.com/10.1007/978-3-319-39907-2_{_}67).
- [62] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.
- [63] G. Lee, Z. Deng, S. Ma, T. Shiratori, S. S. Srinivasa, and Y. Sheikh. Talking with hands 16.2 m: A large-scale dataset of synchronized body-finger motion and audio for conversational motion analysis and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 763–772, 2019.
- [64] W.-H. Lee and R. B. Lee. Implicit smartphone user authentication with sensors and contextual machine learning. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 297–308. IEEE, 2017.

- [65] T. S. Lendvay, T. C. Brand, L. White, T. Kowalewski, S. Jonnadula, L. D. Mercer, D. Khor-sand, J. Andros, B. Hannaford, and R. M. Satava. Virtual reality robotic surgery warm-up improves task performance in a dry laboratory environment: a prospective randomized controlled study. *Journal of the American College of Surgeons*, 216(6):1181–1192, 2013.
- [66] S. Li, X. Xiong, and J. Beck. Modeling student retention in an environment with delayed testing. In *EDM*, pages 328–329, 2013.
- [67] J. Liebers, M. Abdelaziz, L. Mecke, A. Saad, J. Auda, U. Gruenefeld, F. Alt, and S. Schneegass. Understanding user identification in virtual reality through behavioral biometrics and the effect of body normalization. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. doi: 10.1145/3411764.3445528. URL <https://doi.org/10.1145/3411764.3445528>.
- [68] J. Liebers, P. Horn, C. Burschik, U. Gruenefeld, and S. Schneegass. Using gaze behavior and head orientation for implicit identification in virtual reality. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, pages 1–9, 2021.
- [69] S. Liu, N. Yu, L. Chan, Y. Peng, W. Sun, and M. Y. Chen. PhantomLegs: Reducing Virtual Reality Sickness Using Head-Worn Haptic Devices. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 817–826, 2019. ISBN 2642-5246 VO -. doi: 10.1109/VR.2019.8798158.
- [70] S.-H. Liu, N.-H. Yu, L. Chan, Y.-H. Peng, W.-Z. Sun, and M. Y. Chen. Phantomlegs: Reducing virtual reality sickness using head-worn haptic devices. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 817–826. IEEE, 2019.
- [71] X. López, J. Valenzuela, M. Nussbaum, and C.-C. Tsai. Some recommendations for the reporting of quantitative studies. 2015.



- [72] A. Maselli and M. Slater. The building blocks of the full body ownership illusion. *Frontiers in Human Neuroscience*, 7:83, 2013. ISSN 16625161. doi: 10.3389/fnhum.2013.00083.
- [73] R. P. McMahan. Virtual Reality System Fidelity. In N. Lee, editor, *Encyclopedia of Computer Graphics and Games*, pages 1–8. Springer, Cham, 2018.
- [74] R. P. McMahan and N. S. Herrera. AFFECT: Altered-fidelity framework for enhancing cognition and training. *Frontiers in ICT*, 3(NOV):29, 2016. ISSN 2297198X. doi: 10.3389/fict.2016.00029.
- [75] R. P. McMahan, D. A. Bowman, S. Schafrik, and M. Karmis. Virtual Environment Training for Preshift Inspections of Haul Trucks to Improve Mining Safety. In *First International Future Mining Conference and Exhibition*, pages 167–174, Sydney, 2008. The Australasian Institute of Mining and Metallurgy (AusIMM).
- [76] R. P. McMahan, A. J. D. Alon, S. Lazem, R. J. Beaton, D. Machaj, M. Schaefer, M. G. Silva, A. Leal, R. Hagan, and D. A. Bowman. Evaluating natural interaction techniques in video games. In *2010 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 11–14. IEEE, 2010.
- [77] R. P. McMahan, D. A. Bowman, D. J. Zielinski, and R. B. Brady. Evaluating display fidelity and interaction fidelity in a virtual reality game. *IEEE transactions on visualization and computer graphics*, 18(4):626–633, 2012.
- [78] R. P. McMahan, R. Kopper, and D. A. Bowman. Principles for designing effective 3d interaction techniques. In *Handbook of Virtual Environments*, pages 299–325. CRC Press, 2014.
- [79] M. C. Medlock, D. Wixon, M. Terrano, R. Romero, and B. Fulton. Using the rite method to

- improve products: A definition and a case study. *Usability Professionals Association*, 51: 1963813932–1562338474, 2002.
- [80] M. R. Miller, F. Herrera, H. Jun, J. A. Landay, and J. N. Bailenson. Personal identifiability of user tracking data during observation of 360-degree vr video. *Scientific Reports*, 10(1): 1–10, 2020.
- [81] R. Miller, A. Ajit, N. K. Banerjee, and S. Banerjee. Realtime behavior-based continual authentication of users in virtual reality environments. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 253–2531. IEEE, 2019.
- [82] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997. ISBN 978-0-07-042807-2.
- [83] A. G. Moore, J. G. Hatch, S. Kuehl, and R. P. McMahan. Vote: A ray-casting study of vote-oriented technique enhancements. *International Journal of Human-Computer Studies*, 120:36–48, 2018.
- [84] A. G. Moore, M. Kodeih, A. Singhanian, A. Wu, T. Bashir, and R. P. McMahan. The importance of intersection disambiguation for virtual hand techniques. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 310–317. IEEE, 2019.
- [85] A. G. Moore, X. Hu, J. C. Eubanks, A. A. Aiyaz, and R. P. McMahan. A formative evaluation methodology for vr training simulations. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 125–132, 2020. doi: 10.1109/VRW50115.2020.00027.
- [86] A. G. Moore, R. P. McMahan, H. Dong, and N. Ruozzi. Extracting velocity-based user-tracking features to predict learning gains in a virtual reality training application. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 881–890,

Los Alamitos, CA, USA, nov 2020. IEEE Computer Society. doi: 10.1109/ISMAR50242.2020.00099. URL <https://doi.ieeecomputersociety.org/10.1109/ISMAR50242.2020.00099>.

- [87] A. G. Moore, R. P. McMahan, H. Dong, and N. Ruozzi. Personal identifiability and obfuscation of user tracking data from vr training sessions. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 221–228, 2021. doi: 10.1109/ISMAR52148.2021.00037.
- [88] A. G. Moore, R. P. McMahan, H. Dong, and N. Ruozzi. Personal identifiability of user tracking data during vr training. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 556–557. IEEE, 2021.
- [89] A. G. Moore, R. P. McMahan, and N. Ruozzi. Exploration of feature representations for predicting learning and retention outcomes in a vr training scenario. *Big Data and Cognitive Computing*, 5(3), 2021. ISSN 2504-2289. doi: 10.3390/bdcc5030029. URL <https://www.mdpi.com/2504-2289/5/3/29>.
- [90] S. Mota and R. W. Picard. Automated Posture Analysis for Detecting Learner’s Interest Level. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 5:1–6, 2003. ISSN 21607516. doi: 10.1109/CVPRW.2003.10047.
- [91] T. Mustafa, R. Matovu, A. Serwadda, and N. Muirhead. Unsure how to authenticate on your vr headset? come on, use your head! In *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics, IWSPA ’18*, page 23–30, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356343. doi: 10.1145/3180445.3180450. URL <https://doi.org/10.1145/3180445.3180450>.
- [92] C. T. Neth, J. L. Souman, D. Engel, U. Kloos, H. H. Bulthoff, and B. J. Mohler. Velocity-

- dependent dynamic curvature gain for redirected walking. *IEEE transactions on visualization and computer graphics*, 18(7):1041–1052, 2012.
- [93] H. S. Nwana. Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, 4(4): 251–277, 1990. ISSN 1573-7462. doi: 10.1007/BF00168958.
- [94] I. Olade, C. Fleming, and H.-N. Liang. Biomove: Biometric user identification from human kinesiological movements for virtual reality systems. *Sensors*, 20(10), 2020. ISSN 1424-8220. doi: 10.3390/s20102944. URL <https://www.mdpi.com/1424-8220/20/10/2944>.
- [95] F. Paas, A. Renkl, and J. Sweller. Cognitive load theory and instructional design: Recent developments. *Educational psychologist*, 38(1):1–4, 2003.
- [96] N. Padmanaban, T. Ruban, V. Sitzmann, A. M. Norcia, and G. Wetzstein. Towards a machine-learning approach for sickness prediction in 360 stereoscopic videos. *IEEE transactions on visualization and computer graphics*, 24(4):1594–1603, 2018.
- [97] S. Pal and S. Mitra. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on Neural Networks*, 3(5):683–697, 1992. doi: 10.1109/72.159058.
- [98] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [99] K. Pfeuffer, M. J. Geiger, S. Prange, L. Mecke, D. Buschek, and F. Alt. Behavioural biometrics in vr: Identifying people from body motion and relations in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI ’19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery. ISBN

9781450359702. doi: 10.1145/3290605.3300340. URL <https://doi.org/10.1145/3290605.3300340>.

- [100] E. D. Ragan, D. A. Bowman, R. Kopper, C. Stinson, S. Scerbo, and R. P. McMahan. Effects of Field of View and Visual Complexity on Virtual Reality Training Effectiveness for a Visual Scanning Task. *IEEE Transactions on Visualization and Computer Graphics*, 21(7): 794–807, 2015. ISSN 1077-2626 VO - 21. doi: 10.1109/TVCG.2015.2403312.
- [101] K. Revett, F. Gorunescu, M. Gorunescu, M. Ene, S. Magalhaes, and H. Santos. A machine learning approach to keystroke dynamics based user authentication. *International Journal of Electronic Security and Digital Forensics*, 1(1):55–70, 2007.
- [102] A. Rizzo, G. Reger, G. Gahm, J. Difede, and B. O. Rothbaum. Virtual Reality Exposure Therapy for Combat-Related PTSD BT - Post-Traumatic Stress Disorder: Basic Science and Clinical Practice. In J. E. LeDoux, T. Keane, and P. Shiromani, editors, *Post-Traumatic Stress Disorder*, pages 375–399. Humana Press, Totowa, 2009. ISBN 978-1-60327-329-9. doi: 10.1007/978-1-60327-329-9\_18.
- [103] C. E. Rogers, A. W. Witt, A. D. Solomon, and K. K. Venkatasubramanian. An approach for user identification for head-mounted displays. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers, ISWC '15*, page 143–146, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450335782. doi: 10.1145/2802083.2808391. URL <https://doi.org/10.1145/2802083.2808391>.
- [104] G. Saposnik, T. Robert, M. Mamdani, D. Cheung, K. Thorpe, B. McIlroy, J. Willems, J. Hall, L. Cohen, and M. Bayley. Effectiveness of virtual reality using wii gaming technology in stroke rehabilitation (evrest): a randomized clinical trial and proof of principle. In *Stroke*, volume 41, pages E473–E473. LIPPINCOTT WILLIAMS & WILKINS 530 WALNUT ST, PHILADELPHIA, PA 19106-3621 USA, 2010.

- [105] C. Schell, A. Hotho, and M. E. Latoschik. Comparison of data representations and machine learning architectures for user identification on arbitrary motion sequences. *arXiv preprint arXiv:2210.00527*, 2022.
- [106] B. Schneider and P. Blikstein. Unraveling students’ interaction around a tangible interface using multimodal learning analytics. *Journal of Educational Data Mining*, 7(3):89–116, 2015.
- [107] M. Scriven. The methodology of evaluation. *Social Science Education Consortium*, 1967.
- [108] J. A. Self. Student models in computer-aided instruction. *International Journal of Man-machine studies*, 6(2):261–276, 1974.
- [109] C. Sewell, D. Morris, N. H. Blevins, S. Dutta, S. Agrawal, F. Barbagli, and K. Salisbury. Providing metrics and performance feedback in a surgical simulator. *Computer Aided Surgery*, 13(2):63–81, 2008. ISSN 1092-9088. doi: 10.3109/10929080801957712.
- [110] E. J. Simpson. The classification of educational objectives, psychomotor domain. 1966.
- [111] S. Suthaharan. Machine learning models and algorithms for big data classification. *Integr. Ser. Inf. Syst*, 36:1–12, 2016.
- [112] I. Sutherland. The ultimate display. 1965.
- [113] M. Tarrant and J. Ware. A framework for improving the quality of multiple-choice assessments. *Nurse Educator*, 37(3):98–104, 2012.
- [114] K. T. P. Tran, S. Jung, S. Hoermann, and R. W. Lindeman. MDI: A Multi-channel Dynamic Immersion Headset for Seamless Switching between Virtual and Real World Activities. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 350–358, 2019. ISBN 2642-5246 VO -. doi: 10.1109/VR.2019.8798240.

- [115] P. P. Tricomi, F. Nenna, L. Pajola, M. Conti, and L. Gamberini. You can't hide behind your headset: User profiling in augmented and virtual reality. *arXiv preprint arXiv:2209.10849*, 2022.
- [116] K. VanLehn. The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, 46(4):197–221, oct 2011. ISSN 0046-1520. doi: 10.1080/00461520.2011.611369.
- [117] N. Veliyath. *iFocus : A Framework for Non-intrusive Assessment of Student Attention Level in Classrooms*. PhD thesis, Georgia Southern University, 2019.
- [118] J.-J. Vie and H. Kashima. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 750–757, 2019.
- [119] M. E. Wall, A. Rechtsteiner, and L. M. Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.
- [120] J. Wallace. Some aspects of educational research, development and evaluation in the usa. *Educational Research*, 9(2):105–112, 1967.
- [121] Y. Wang and J. E. Beck. Using student modeling to estimate student knowledge retention. *International Educational Data Mining Society*, 2012.
- [122] T. Weissker, A. Kulik, and B. Froehlich. Multi-Ray Jumping: Comprehensible Group Navigation for Collocated Users in Immersive Virtual Reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 136–144, 2019. ISBN 2642-5246 VO -. doi: 10.1109/VR.2019.8797807.

- [123] A. S. Won, J. N. Bailenson, and J. H. Janssen. Automatic detection of nonverbal behavior predicts learning in dyadic interactions. *IEEE Transactions on Affective Computing*, 5(2): 112–125, 2014.
- [124] B. Xie, H. Liu, R. Alghofaili, Y. Zhang, Y. Jiang, F. D. Lobo, C. Li, W. Li, H. Huang, M. Akdere, et al. A review on virtual reality skill training applications. *Frontiers in Virtual Reality*, 2:645153, 2021.
- [125] R. Yu, Z. Duer, T. Ogle, D. A. Bowman, T. Tucker, D. Hicks, D. Choi, Z. Bush, H. Ngo, P. Nguyen, and X. Liu. Experiencing an Invisible World War I Battlefield Through Narrative-Driven Redirected Walking in Virtual Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 313–319, 2018. ISBN VO -. doi: 10.1109/VR.2018.8448288.
- [126] J. Zaletelj. Estimation of students’ attention in the classroom from kinect features. In *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis*, pages 220–224. IEEE, 2017.
- [127] Y. Zhang, R. Gravina, H. Lu, M. Villari, and G. Fortino. Pea: Parallel electrocardiogram-based authentication for smart healthcare systems. *Journal of Network and Computer Applications*, 117:10 – 16, 2018. ISSN 1084-8045. doi: <https://doi.org/10.1016/j.jnca.2018.05.007>. URL <http://www.sciencedirect.com/science/article/pii/S1084804518301693>.
- [128] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.