ABSTRACT

Title of dissertation: TOWARDS VISUAL ANALYTICS IN VIRTUAL ENVIRONMENTS

Eric Krokos
Doctor of Philosophy, 2018

Dissertation directed by: Professor Amitabh Varshney
Department of Computer Science

Virtual reality (VR) is poised to become the new medium through which we engage, view, and consume content. In contrast to traditional 2D desktop displays, which restrict our interaction space onto an arbitrary 2D-plane with unnatural interaction mechanisms, VR expands the visualization and interaction space into our 3D domain, enabling natural observations and interactions with information. With the rise of Big Data, processing and visualizing such enormous datasets is of utmost importance and remains a difficult challenge. Machine learning, specifically deep learning, is rising to meet this challenge. In this work, we present several studies: (a) demonstrating the effectiveness of immersive environments over traditional desktops for memory recall, (b) quantifying cybersickness in virtual environments, (c) enabling human analysts and deep learning to support, refine, and enhance each other through visualization, and (d) visualizing root-DNS information, enabling analysts to find new and interesting anomalies and patterns.

In our first work, we conduct a user study where participants memorize and

recall a series of spatially-distributed faces on both a desktop and head-mounted display (HMD). We found that the use of virtual memory palaces in the HMD condition improves recall accuracy when compared to the traditional desktop condition. This improvement was statistically significant. Next, we present our work on quantifying cybersickness through EEG analysis. We found statistically significant correlations with increases in delta, theta, and alpha brain waves with self-reported sickness levels, enabling future virtual reality developers to design countermeasures. Third, we present our work on enabling domain experts to discover hidden labels and communities within unlabeled (or coarsely labeled) high-dimensional datasets using deep learning with visualization. Lastly, we present a 3D visualization of root-DNS traffic, revealing characteristics of a DDOS attack and changes in the distribution of queries received over time. Together, this work takes the first steps in bringing together machine learning, visual analytics, and virtual reality.

TOWARDS VISUAL ANALYTICS IN VIRTUAL ENVIRONMENTS

by

Eric Krokos

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor Amitabh Varshney, Chair/Advisor
Professor Joseph JaJa
Professor Matthias Zwicker
Professor John Dickerson
Dr. Kirsten Whitley

© Copyright by
Eric Krokos
2018

Table of Contents

List of Tables v

List of Figures vi

1 Introduction 1
- 1.1 Virtual Memory Palaces . . . 5
  - 1.1.1 Introduction . . . 5
  - 1.1.2 Approach . . . 6
  - 1.1.3 Results . . . 9
- 1.2 Characterizing Vection-Induced Cyber Sickness using EEG . . . 9
  - 1.2.1 Introduction . . . 9
  - 1.2.2 Approach . . . 10
  - 1.2.3 Results . . . 13
- 1.3 Enhancing Deep Learning with Visual Interactions . . . 14
  - 1.3.1 Introduction . . . 14
  - 1.3.2 Approach . . . 16
  - 1.3.3 Results . . . 17
- 1.4 Visual Analytics for Root DNS Data . . . 20
  - 1.4.1 Introduction . . . 20
  - 1.4.2 Approach . . . 20
  - 1.4.3 Results . . . 23

2 Virtual Memory Palaces: Immersion aids Recall 29
- 2.1 Introduction . . . 29
- 2.2 Related Work . . . 31
  - 2.2.1 Memory Palaces on a Desktop Monitor . . . 33
  - 2.2.2 Memory Palaces on Multiple Displays . . . 34
  - 2.2.3 Search and Recall in Head-mounted Displays . . . 36
  - 2.2.4 Embodied Interaction and Recall . . . 38
- 2.3 Method . . . 41
  - 2.3.1 Participants . . . 42
  - 2.3.2 Materials . . . 42

2.3.3 Design . . . 43
2.3.4 Procedure . . . 44
2.4 Results . . . 49
2.4.1 Task Performance . . . 50
2.4.2 Errors and Skips . . . 51
2.4.3 Confidence . . . 52
2.4.4 Ordering Effect . . . 54
2.5 Discussion . . . 55
2.5.1 Study Limitations . . . 57
2.5.2 Conclusions . . . 58
2.5.3 Future Work . . . 59

3 Interactive Characterization of Cybersickness in Virtual Environments using EEG 61
3.1 Introduction . . . 61
3.2 Related Work . . . 63
3.2.1 Self-reporting Cybersickness . . . 64
3.2.2 Measuring Motion Sickness with EEG . . . 65
3.3 Materials and Methods . . . 67
3.3.1 Participants . . . 68
3.3.2 Experimental Protocol . . . 69
3.3.3 Signal Acquisition and Pre-Processing . . . 70
3.3.4 Independent Component Analysis . . . 71
3.3.5 Time-Frequency Analysis . . . 73
3.4 Results . . . 76
3.4.1 Self-Reported Cybersickness . . . 76
3.4.2 Spectral Differences . . . 79
3.4.3 Time-Frequency with User input signals . . . 81
3.4.4 External Factors . . . 85
3.5 Conclusions and Future Work . . . 86

4 Enhancing Deep Learning with Visual Interactions 88
4.1 Introduction . . . 88
4.2 Related Work . . . 90
4.2.1 Dimensionality Reduction . . . 91
4.2.2 High-Dimensional Community Visualization . . . 93
4.2.3 Interactive Analysis of High-Dimensional Data . . . 94
4.2.4 Label Generation . . . 96
4.2.5 Deep Learning Semi-Supervised Classification . . . 99
4.2.6 Interactive Intelligent Systems and Active Learning . . . 101
4.3 Our Approach . . . 103
4.3.1 Point-Distribution Generation . . . 105
4.3.2 Cluster Visualization and Manipulation . . . 109
4.4 Results . . . 119
4.4.1 Pavia University Dataset . . . 122

4.4.2 Salinas Valley . . . 124
4.4.3 User Study . . . 125
4.4.4 Interpreting Discovered Labels . . . 128
4.4.5 DNS Query Dataset . . . 131
4.5 Discussion . . . 133
4.6 Conclusions . . . 136

5 Visual Analytics for Root DNS Data 138
5.1 Introduction . . . 138
5.2 Background . . . 141
5.2.1 Traditional 2D Network and DNS Visualization . . . 142
5.2.2 3D Network Visualization . . . 144
5.3 Problem and Solution . . . 146
5.3.1 The Challenge . . . 147
5.3.2 Approach Overview . . . 149
5.3.3 Flow-Map IP-Space Visualization . . . 149
5.3.4 IP-Space Observations . . . 151
5.3.5 Deep Learning Driven Query Space Visualization . . . 153
5.3.5.1 3D Query Flow Visualization . . . 156
5.3.6 Dual IP-Query Visualization Interaction . . . 159
5.4 Empirical Validation . . . 160
5.4.1 DNS Expert A . . . 161
5.4.2 DNS Expert B . . . 164
5.4.3 DNS Expert C . . . 166
5.5 Conclusions . . . 167

6 Conclusions and Future Work 169
6.1 Conclusions . . . 169
6.2 Future Work . . . 171
6.3 Peer Reviewed Publications . . . 172

Bibliography 179

List of Tables

3.1 Correlations (Pearson R-values) between average ERSP values for the four frequency bands and the self-reported cybersickness levels. All the correlations are statistically significant ($p < 0.001$). The graphs of the various frequency bands for clusters A can be seen in Figure 3.10. 83

1 Trend Scores for each face for Face Set 1 from Google Trends, with the average trend score of 30.5 and standard deviation of 21.86. The data was collected from April, May, June, and July of 2015. 174

2 Trend Scores for each face for Face Set 2 from Google Trends, with the average trend score of 29.83 and standard deviation of 18.32. The data was collected from April, May, June, and July of 2015. 175

3 Angular Resolution of Faces in the Town and Palace scenes, with the average and standard deviation of the angular resolutions of the set of faces for each scene. The difference in angular resolutions between the two scenes was not statistically significant with ($p = 0.44 > 0.05$). 176

List of Figures

1.1 One of the virtual memory palace scenes used in our user study (left) an ornate palace showing some of the faces used, and (right) the same ornate palace, with the faces replaced with numbers. 6

1.2 The overall average recall performance of participants using a HMD was 8.8% higher compared to a desktop. The median recall accuracy percentage for HMD was 90.48% and for desktop display was 78.57%. The figure shows the first and third quartiles for each display modality. 7

1.3 The overall confidence scores of participants using a HMD and a desktop. Each participant gave a confidence score between 1 and 10 for each face they recalled. Those in the HMD are slightly more confident about their answers than those on the desktop. 7

1.4 The distribution of incorrect answers for each display modality showing the median, first, and third quartiles. 8

1.5 The distribution of Simulator Sickness Questionnaire (SSQ) scores obtained from participants after participating in the experiment. The plot shows the median, first and third quartiles (orange and grey respectively), with the minimum and maximum shown as error bars. . 11

1.6 The names and locations of the 14 EEG electrodes in the Emotiv Epoc headset. 12

1.7 Comparison of the EEG power spectra between the baseline (blue) and virtual flythrough (green) for ICA cluster A. The paired t-test with Bonferroni-correction between the two spectra reveal $p < 0.001$ for much of the frequency ranges, and with $p < 0.05$ for most ranges. 12

1.8 Average over four frequency bands for Cluster A compared with the average self-reported cybersickness (in green). 13

1.9 A brief illustration of the difference between traditional deep learning techniques and our approach. Deep learning traditionally requires a large, time-consuming, and precisely labeled dataset for training. For several reasons, such datasets may be inappropriately labeled. In our approach, we start with coarse labels (that are typically far easier to construct) and then refine them through an iterative process, involving visual interactions and deep learning. 15

1.10 A comparison of the initial coarse labeling, to the generated refined labels, and the precise (true) labels of the Pavia university dataset after 3 iterations. Starting from the three initial categories: natural surfaces, roads, and buildings, we were able to reconstruct the distribution of the 9 labels with an accuracy of 88.2%. 17
1.11 A comparison of the initial coarse labeling, the generated refined labels, and the precise (true) distribution of labels, of the Salinas valley dataset after five iterations. Starting with 6 initial coarse labels, we were able to reconstruct the distribution of the 16 labels with an accuracy of 97.4%. 18
1.12 The evolution of the query-space representation over eight iterations, showing the influence of the iterative labeling. Note that for some clusters certain colors have been re-used due to the high-number of groups. 19
1.13 Our root-DNS dual-visualization that provides both high and low-level overviews and interactions of the IP and query spaces. IP packet traffic is visualized on the left, revealing hidden patterns, IP distributions, and a real TCP-SYN flood attack. A two-dimensional query-space generated using deep learning portrays a spatial distribution of received queries and counts. The right image portrays the spatial distribution of queries as they change over time, revealing the diminished number of received queries due to a DDOS. 22
1.14 Overview of the process from raw pcap files to The Flow-Map IP-Space visualization. Starting from a binary pcap file, we extract and count the occurrence of each IPv4 IP-Address and type of packet. Next, the IPs are converted from a 4D to a 2D grid representation, with glyphs scaled and colored based on the number and type of packets. This process repeats for each time slice, with slices stacked along the z-axis. The result is then visualized using 3D accelerated rendering, which allows for high-level structure and low-level analysis, to help analysts establish a sense of normalcy (central blue image), identify outliers (green TCP burst), classify and characterize attacks (top right), measure attack impacts (middle right), and monitor after effects (lower right). 23
1.15 In the lower-left, the 2D query-space clustered into semantic query categories. There is a general trend with alphabet-based queries towards the left, and number-based queries the right, along with a general trend to alphanumeric characters at the top, and non-alphanumeric characters at the bottom. The lower-right shows the temporal query visualization, portraying a high-level temporal overview of the distribution of queries. The top reveals a selection of interesting observations, such as rapidly diminishing groups of queries at the start, temporally repeating groups of queries, the large reduction in queries during the attack, and an overall decrease in the number of queries over time. 24

1.16 The region of the query-space consisting of different distributions of random characters. . . . 26
1.17 The region of the query-space consisting of different distributions of IP addresses, fragments, and expressions. . . . 27
1.18 The region of the query-space consisting of unusual characters and queries. . . . 28

2.1 Giulio Camillo's depiction of a memory palace (1511 AD). Memory palaces like this have been used since the classical times as a spatial mnemonic. . . . 31
2.2 The two Virtual Memory Palace scenes used in our user study (a) an ornate palace, and (b) a medieval town, as seen from the view of the participants. . . . 45
2.3 Virtual memory palace: recall phase . . . 47
2.4 Locations of faces and numbers in the Virtual Memory Palaces used in our user study (a) an ornate palace, and (b) a medieval town. Note that this is not the view the participants had during the experiment, these pictures are used to convey the distribution of the face locations. The participants would have been placed in the middle of these scenes surrounded by the faces as seen in Figure 2.2. . . . 47
2.5 The overall average recall performance of participants in the HMD condition was 8.8% higher compared to the desktop condition. The median recall accuracy percentage for HMD was 90.48% and for desktop display was 78.57%. The figure shows the first and third quartiles for each display modality. . . . 50
2.6 The distribution of incorrect answers for each display modality showing the median, first, and third quartiles. . . . 51
2.7 The distribution of faces skipped during recall for each display modality showing the median, first, and third quartiles. . . . 52
2.8 The overall confidence scores of participants in the HMD condition and the desktop condition. Each participant gave a confidence score between 1 and 10 for each face they recalled. Those in the HMD condition are slightly more confident about their answers than those in the desktop condition. . . . 53
2.9 The number of errors made for each display condition for various confidence levels. . . . 54
2.10 The performance of participants going from a desktop to a HMD and from a HMD to a desktop, showing the median, first and third quartiles. . . . 55

3.1 A still from the virtual spaceport flythrough used in our cybersickness study. . . . 68
3.2 Averaged scalp maps of clustered independent components. The scalp map which correlated with cybersickness is shown in the black box. . . . 72

3.3 The names and locations of the 14 EEG electrodes in the Emotiv Epoc headset. . . . 73
3.4 The virtual camera flythrough of the spaceport that each participant in our study experienced. Note how the above correspond to the self-reported cybersickness levels in Figure 3.5. . . . 75
3.5 The self-reported cybersickness levels, using joystick, for each participant are shown in the thin colored curves. The bold black curve shows the average of all the participants' self-reported cybersickness levels. . . . 77
3.6 Participant Simulator Sickness Questionnaire (SSQ) scores after the experiment are shown here. The plot shows the median, first and third quartiles (orange and grey respectively), with the minimum and maximum shown as error bars. . . . 78
3.7 A comparison of the average score as reported by the joystick with the SSQ sum for each participant. The SSQ score and the self-reported cybersickness using the joystick have a Pearson Correlation r-value of 0.49. . . . 79
3.8 Comparison of the EEG power spectra between the baseline (blue) and virtual flythrough (green) for ICA cluster A. The paired t-test with Bonferroni-correction between the two spectra reveal $p < 0.001$ for much of the frequency ranges. . . . 81
3.9 Time-Frequency visualization of cluster A. The average self-reported cybersickness levels are shown below in red. . . . 82
3.10 Average over four frequency Bands for Cluster A compared with the average self-reported cybersickness (in green). . . . 84
3.11 Visualization of the ERSP from a cluster A participant with self-reported cybersickness levels. Note how the changes in ERSP values, especially for Delta and Theta bands, align with the participant's self-reported cybersickness. . . . 84

4.1 A brief illustration of the difference between traditional deep learning techniques and our approach. deep learning traditionally requires a large, time-consuming, and precisely labeled dataset for training. For many different reasons, such datasets may be inappropriately labeled. In our approach, we start with coarse labels (that are typically far easier to construct) and then refine them through an iterative process, involving visual interactions and deep learning. . . . 104
4.2 Equations used in calculating the Constrastive Loss for the Siamese Network . . . 107

4.3 An example of the result of running the Variational Autoencoder and then refining that result using the Siamese Network. By running the Siamese network after autoencoder, the generated clusters tend to be tighter with more space in-between, making the individual clusters easier to identify. The user has the ability to adjust the number of iterations the Siamese network runs which affects the tightness of the clusters. 108
4.4 A visual representation of the network structure used in both the Variational Autoencoder and the Siamese network. The same network structure and weights are shared across the networks. First the Variational Autoencoder runs, and the Siamese network continues using the network weights generated by the autoencoder. After the Siamese network has run, the resulting network is used to generated the 2D distribution of points. The network weights are also saved, reused, and refined in the following iterations. 109
4.5 Initial view of points, with all points given the same color, which are only assigned a color once selected or activated by the user. 112
4.6 The menu interface presented when a user selects a point/group. . . . 113
4.7 Selection of points through manual paint-brush and circle selection interaction. 114
4.8 Equations used to compute normalized cut. 114
4.9 Segmentation of a set of points using normalized cut (Ncut). 115
4.10 Common Interaction Techniques 117
4.11 Handling of overlapping sets of points. 118
4.12 An example of where a few points are selected from a parent cluster, and over time the points are separated away from the parent cluster to form their own cluster. 119
4.13 The resilience of the algorithm to labeling mistakes. 120
4.14 A comparison of the initial coarse labeling, to the generated refined labels, and the precise (true) labels of the Pavia university dataset after 3 iterations. Starting from the three initial categories: natural surfaces, roads, and buildings, we were able to reconstruct the distribution of the 9 labels with an accuracy of 88.2%. 122
4.15 A comparison between the labeling generated by our system and the ground truth labeling. The labeling generated by our system is driven by the clearly distinct group off the main body of points above it. This difference is supported by a visual difference in the aerial view. 124
4.16 A comparison of the initial coarse labeling, the generated refined labels, and the precise (true) distribution of labels, of the Salinas valley dataset after five iterations. Starting with 6 initial coarse labels, we were able to reconstruct the distribution of the 16 labels with an accuracy of 97.4%. 126
4.17 Participants accuracy and timings for the Pavia University dataset. . 128
4.18 Participants accuracy and timings for the Salinas Valley dataset. . . . 129

4.19 A potential newly discovered sub-group within the painted metal sheets label in the Pavia University dataset. On the left shows the point distribution and spatial representation of the labeling as generated by our technique. On the right shows another iteration showing if we had sub-divided the current points and the resulting spatial representation. 131
4.20 A potential newly discovered sub-group within the untrained grapes label in the Salinas Valley dataset. The left figure shows the meta information used, the aerial view of the valley. The middle image shows the labeling as generated using our approach. The right image shows the labeling as given by the dataset. The yellow labeling portion in the middle image matches with a discoloring in the aerial view, which suggests that there may be a different material there. . . 132
4.21 The evolution of the query-space representation over eight iterations, showing the influence of the iterative labeling. Note that for some clusters certain colors have been re-used due to the high-number of groups. 134
4.22 The output of the DNS query investigation coarsely labeled to identify the meaning of the individual clusters and their relation to each other. Note that for some clusters certain colors have been re-used due to the high-number of groups. 135

5.1 Overview of the process from raw pcap files to The Flow-Map IP-Space visualization. Starting from a binary pcap file, we extract and count the occurrence of each IPv4 IP-Address and type of packet. Next, the IPs are converted from a 4D to a 2D grid representation, with glyphs scaled and colored based on the number and type of packets. This process repeats for each time slice, with slices stacked along the z-axis. The result is then visualized using 3D accelerated rendering, which allows for high-level structure and low-level analysis, to help analysts establish a sense of normalcy (central blue image), identify outliers (green TCP burst), classify and characterize attacks (top right), measure attack impacts (middle right), and monitor after effects (lower right). 146
5.2 An example analysis in Wireshark, a widely used pcap analyzer. . . . 147
5.3 A overview of the process from raw pcap files to Query-Space visualization. Starting from a binary pcap file, we extract and count each query. Next, each query is converted to a TF-IDF (Term Frequency, Inverse Document Frequency) character-level feature vector. A deep learning autoencoder is trained using all queries, to generate a visually coherent spatial distribution of queries when projected into a 2D space. This distribution of queries is then visualized using 3D accelerated rendering, which allows for high-level temporal structure (top-right) and low-level query analysis (bottom right). 152

5.4 Interesting self-similar patterns of intra-IP-bin queries and across-IP-bin traffic (from the TCP-Syn DDOS) over time in D-Root traffic. . . 152
5.5 In the lower-left, the 2D query-space clustered into semantic query categories. There is a general trend with alphabetic-based queries towards the left, and numeric-based queries the right, along with a general trend to normal characters at the top, and unusual characters at the bottom. The lower-right shows the temporal query visualization, portraying a high-level temporal overview of the distribution of queries. The top reveals a selection of interesting observations, such as, rapidly diminishing groups of queries at the start, temporally repeating groups of queries, the large reduction in queries during the attack, and an overall decrease in the number of queries over time. . . 154
5.6 Two selected regions of queries. The top selection indicates the queries originate from a wide range of IPs, while the bottom selection indicates those queries came from very few IPs. Non-included IPs may be set transparent (bottom) or left un-transparent(top). The bottom-right image shows the resulting queries included from an entire IP-bin. 161
5.7 The region of the query-space consisting of different distributions of random characters. 162
5.8 The region of the query-space consisting of different distributions of IP addresses, fragments, and expressions. 164
5.9 The region of the query-space consisting of unusual characters and queries. 167

1 Face Set 1, containing 21 faces. 177
2 Face Set 2, containing 21 faces. 178

Chapter 1: Introduction

The work presented in this dissertation is geared towards enabling visual analytics within virtual environments. A successful virtual reality-based analytical visualization should consist of several key components: provide the analyst with an intuitive view of and interaction with the data, leverage and support the respective advantages of humans and computers, and be capable of presenting vast amounts of abstract data enabling the analysis and discovery of actionable intelligence. This dissertation presents advances in each of those areas, justifying and enabling the use of head-mounted displays, leveraging and enhancing machine-learning using human pattern recognition and domain knowledge for data discovery and analysis, and the development of a three-dimensional visualization for the exploration and characterization of vast amounts of abstract DNS network data.

The world is amidst a global cyber war. Conducting the analysis of vast amounts of network data for monitoring and safeguarding a core pillar of the internet, the root DNS, is an enormous challenge. It is critical to the continuing stability and operation of the Internet that we discover, understand, and thwart attacks. The current state-of-the-art tools are in essence log-file editors, only portraying specific details about specific Internet packets, with no sense of global structure, spatial

orientation, or temporal connections. Humans, in contrast, are adept at organizing and interacting with the world in a spatial manner.

While the monitors and traditional displays are generally familiar and comfortable to use, the benefit of head-mounted displays is their ability to leverage and connect our kinesthetic intelligence (our natural inclination for movement and spatial interaction) with the content displayed. Rather than mapping data to a large 2D array of monitors, head-mounted displays allow for an immersive and natural interaction space. Discovering and quantifying the benefits of head-mounted displays over traditional desktops are therefore very important. In our first study we ask, what are the benefits of using a head-mounted display over a traditional desktop display? Our study showed that there was a statistically significant improvement in the recall of spatially-distributed information when using a head-mounted display, as compared to a traditional desktop. We present the summary of our research in section 1.2, and give details in Chapter 2. This result shows that using head-mounted displays has advantages over traditional desktop when engaging with spatial content.

A crucial problem with head-mounted displays that still prevents their wide adoption is the onset of cybersickness. It has been reported [1] that a large proportion of the population, after an extended usage of head-mounted displays, experience feelings of nausea, headache, fatigue, and other negative ailments, which are collectively referred as cybersickness. This means there are analysts who are unable to use virtual reality for data visualization. At present, there is little understanding of the causes of cybersickness, let alone how to mitigate it. One of the prevailing theories on the cause of cybersickness is the sensory conflict theory, which attributes

it to the dissonance between the visual and the vestibular sensory cues [1,2]. Before we are able to treat and correct for cybersickness, we need to be able to quantify and measure it, motivating our second study. Our goal is to develop a technique for the objective measurement and quantification of cybersickness. Using an electroencephalogram (EEG) we validate our measurements by correlating the EEG signals with a continuous self-diagnosis of cybersickness from our participants using a joystick. We found that the delta, theta, and alpha wave bands were correlative with the reported levels of cybersickness. We present the summary of our research in section 1.3, and give details in Chapter 3. This result provides the framework to evaluate the onset and mitigation of cybersickness in future applications and head-mounted displays.

Having established that head-mounted displays have advantages over desktops, and a framework to detect and quantify cybersickness, we shift focus towards visual analytics. The ability of our technology to collect large amounts of data has far surpassed our ability to process and generate insights and conclusions using existing methods. Traditionally, the visualization of large datasets on desktops is done with large lists, tables, or conventional plots (such as line and bar plots). Each representation has its disadvantages, but their weaknesses are most evident when the datasets visualized are large and high dimensional. These weaknesses make interpreting and drawing conclusions from such datasets very difficult. Rather than relying on pure visualization for analysis, analysts often turn to machine learning for information discovery and analysis. Deep learning, a branch of machine learning, has proved itself extremely successful for many different application areas of

science and visualization. However, deep learning requires a tremendous amount of precisely-labeled training data in order to function properly. In the real world, such training sets are not readily available or easily created, such as in the case of training a classifier to detect attacks on DNS. An approach where machines and humans are able to leverage the strengths of the other is desirable. This is the motivation for our third study, to develop a technique which leverages the pattern recognition ability of humans with the analytical processing power of machines and deep learning, with the goal of discovering hidden communities and labels within sparsely labeled, high-dimensional, non-spatial datasets. To evaluate our technique, we treated two finely labeled, high-dimensional, spatial datasets as ground-truth. Then, we started the iterative discovery process by stripping away the spatial context, and greatly reduced the number of labels by merging many of the precise labels into initial coarse labels. For the study, our goal was to reconstruct, to the best of our ability, the original labeling distribution of the test datasets. For each of the datasets, we were able to rediscover the hidden labels and accurately label many of the data points appropriately. We further present the summary of our research in section 1.4, and give details in Chapter 4. This result shows we can discover latent clusters and information through an iterative process, while simultaneously improving the capabilities of the deep learning training model with human guidance.

A virtual reality-based visualization will inherently leverage a 3D environment. While 2D visualizations are often regarded as easier to create and understand (in terms of time required for comprehension), recent research has shown there are many benefits to 3D visualizations over 2D for abstract data visualiza-

tion [3–5], including clearer spatial separation, reduced over-plotting, and enabling faster construction and deeper mental models. These ideas have inspired our work, summarized in section 1.5, and presented in more detail in Chapter 5, where we develop a three-dimensional virtual-environment data visualization capable of portraying large amounts of abstract network data. For our visualization, we sought to tackle a real-world problem, enabling the analysis of the vast amounts of network traffic that flows through the root-DNS. Traditional network tools largely consist of advanced text editors, simply listing every aspect of every packet. Although such visualizations provide unparalleled detail, they inhibit the potential of detecting anomalous trends, generally require analysts to know of a search target, and do not scale to handle the ever-increasing amounts of data to be reviewed. Our visualization has been carefully constructed with DNS industry experts to leverage our pattern recognition and spatial recognition capabilities. We show that our visualization is capable of providing analysts with a comprehensible analysis of the data of the spatio-temporal data, enabling a definition of normalcy, detecting previously unknown anomalies and clusters, and characterizing large-scale real-world attacks for future attack identification and mitigation.

1.1 Virtual Memory Palaces

1.1.1 Introduction

Since classical times, people have used memory palaces as a spatial mnemonic to help remember information by organizing it in an environment and associating it

with salient features in that environment. Virtual reality affords a new medium for exploring large datasets in an embodied way. We present here a summary of Chapter 2, where we explore whether using virtual memory palaces in a head-mounted display (HMD) will allow a user to be able to better recall information than with a traditional desktop display.



Figure 1.1: One of the virtual memory palace scenes used in our user study (left) an ornate palace showing some of the faces used, and (right) the same ornate palace, with the faces replaced with numbers.

1.1.2 Approach

For this study, we focused on analyzing and leveraging the spatial component of human memory and the inherently spatial nature of immersive virtual reality head-mounted displays. Our hypothesis is that the added immersion of virtual reality will improve the recall performance of human subjects compared with desktop displays. To test this hypothesis, we prepared two 3D realistically rendered environments and

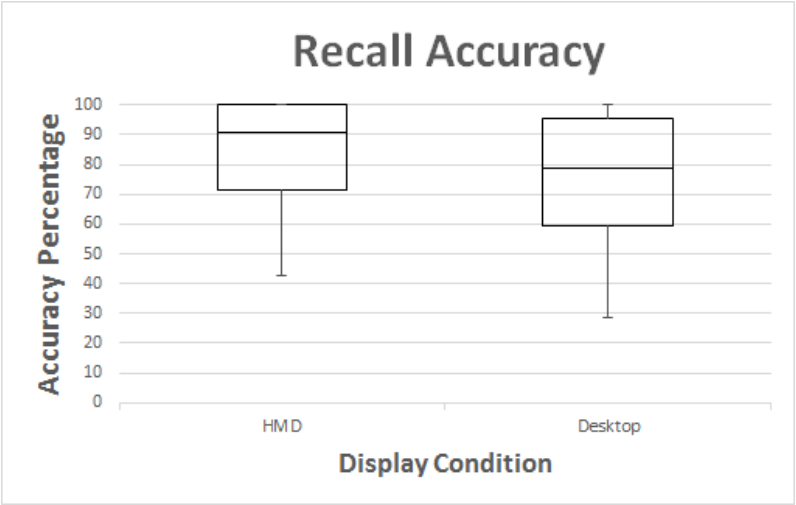


Figure 1.2: The overall average recall performance of participants using a HMD was 8.8% higher compared to a desktop. The median recall accuracy percentage for HMD was 90.48% and for desktop display was 78.57%. The figure shows the first and third quartiles for each display modality.

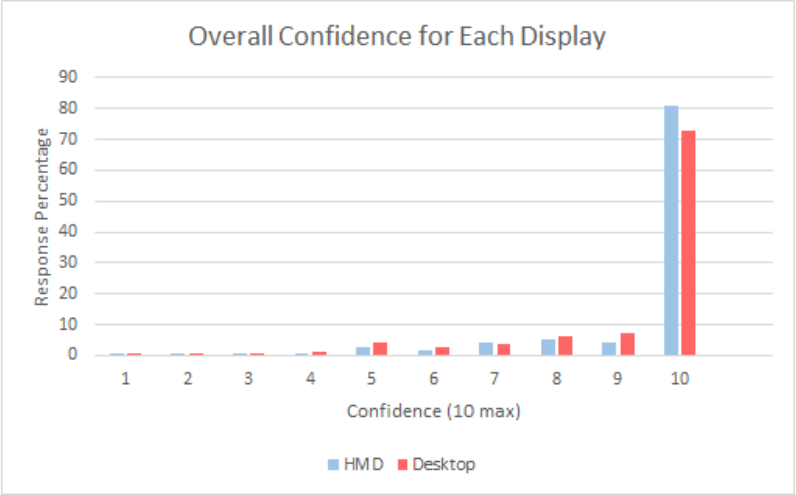


Figure 1.3: The overall confidence scores of participants using a HMD and a desktop. Each participant gave a confidence score between 1 and 10 for each face they recalled. Those in the HMD are slightly more confident about their answers than those on the desktop.

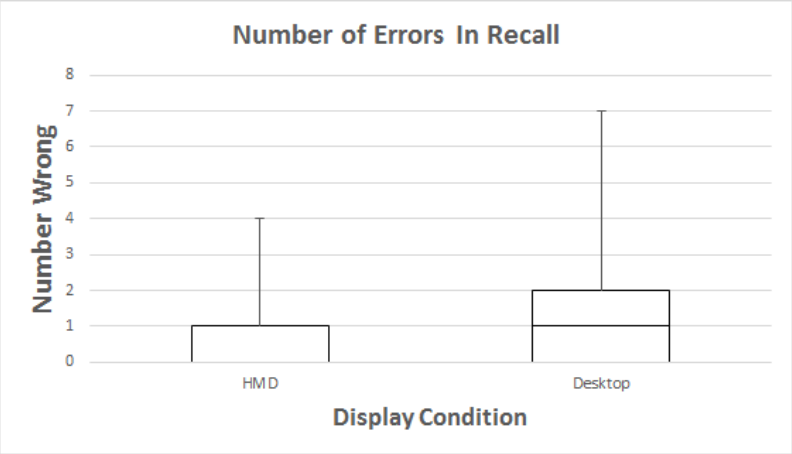


Figure 1.4: The distribution of incorrect answers for each display modality showing the median, first, and third quartiles.

placed faces of well-known people within them. The faces were distributed around a central location where we position our study participants. After our participants familiarize themselves with the faces and their names, the participants first use either a desktop or a head-mounted display, in one of the two scenes, and view one of two sets of faces. We give the participants five minutes to familiarize themselves with the scene and the faces distributed around the scene. After a two-minute break, we place participants back in the environment, with the faces replaced with numbers. The participants next recall the names of the faces which were previously at each numbered location, and to give a confidence in their answer $(0 - 10)$, with 10 as fully confident. This process then repeats for the alternate display, the other environment, and with the alternate set of faces. An example of what one of the scenes looked like with faces and with numbers can be seen in Figure 1.1.

1.1.3 Results

Our study, with 40 participants, found that virtual memory palaces viewed in an HMD provide a superior memory recall ability compared to a traditional desktop display. Specifically, we found a statistically significant increase of 8.8% in the recall accuracies of the HMD as compared to a desktop, and that 40% of our participants had at least a 10% advantage while using the HMD. The distribution of the accuracies for each display can be seen in Figure 1.2. In addition, we also found a statistically significant increase in the confidence of our participants in their answers when in the HMD as compared to using the desktop (as can be seen in Figure 1.3). This increase in confidence was supported by a statistically significant decrease in the number of errors for the HMD as compared to the desktop (See Figure 1.4). We believe that these types of virtual environments can create natural and more memorable experiences that enhance productivity for better recall and understanding of large amounts of information.

1.2 Characterizing Vection-Induced Cyber Sickness using EEG

1.2.1 Introduction

If virtual reality is to revolutionize the way we view and interact with machines, data, and each other, mitigating and understanding the onset of cybersickness is critical. Virtual and augmented reality are poised to fundamentally shift the way we consume visual information and interact with technology. However, the adaptation

of this new method for viewing and interacting with information is hampered by the fact that a large proportion of people suffer from what has been named cyber sickness [1]. Common symptoms of cybersickness include nausea, increased heart rate, disorientation, sweating, eye strain, and headaches [6,7]. One of the prevailing theories on the cause of cybersickness (also referred as simulator sickness or visual fatigue) is the sensory conflict theory, which attributes it to the dissonance between the visual and the vestibular sensory cues [1, 2]. Our goal is to develop a simple, passive, and objective technique which quantifies cybersickness. We present a summary of our study here, and cover its design and results in more detail in Chapter 3.

1.2.2 Approach

We conducted a user study which involves taking EEG measurements of participants, using the Emotiv Epoc 14-channel 128Hz EEG headset, while the participants are exposed to an experience designed to induce cybersickness in virtual reality. After taking a baseline EEG reading, the participants fly through a virtual spaceport for approximately a minute. During the fly-through, participants report their current level of cybersickness by tilting a joystick, indicating higher sickness with more tilt. After each session, each participant completes a Simulator Sickness Questionnaire (SSQ), which subjectively assesses the different symptoms related to cybersickness, giving each a score from 0 to 4.

For the user study, we recruited 44 participants, of which we were able to use

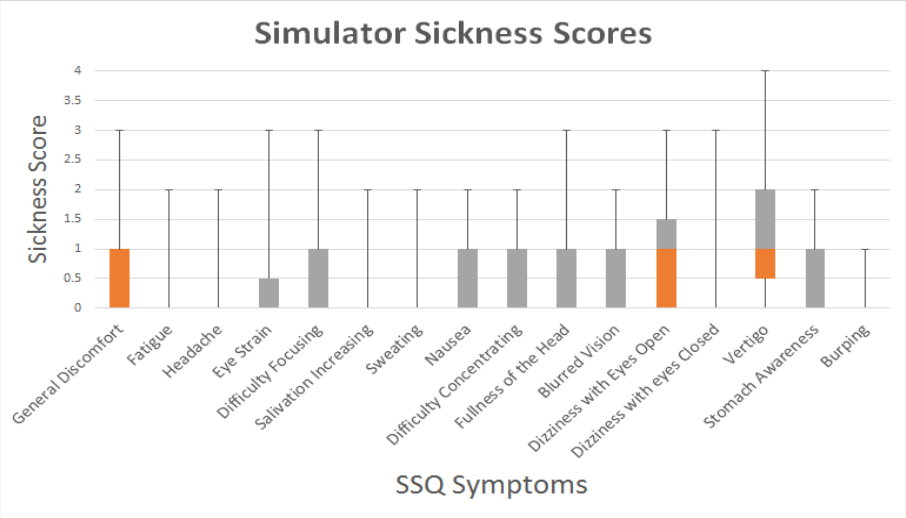


Figure 1.5: The distribution of Simulator Sickness Questionnaire (SSQ) scores obtained from participants after participating in the experiment. The plot shows the median, first and third quartiles (orange and grey respectively), with the minimum and maximum shown as error bars.

the data of 43 participants. First, we confirm that our participants experienced cybersickness through analysis of the SSQ scores. The distribution of the SSQ scores can be seen in Figure 1.5. By looking at the distribution of the SSQ scores, it is clear that our participants, on average, claimed to experience some cybersickness symptoms. To process the collected EEG data we use the EEG toolkit EEGLab (`https://sccn.ucsd.edu/eeglab/`). EEGlab works by taking each participant's EEG signals and performing independent component analysis (ICA), generating 14 independent components (ICs) for each participant. The goal of ICA is to deconstruct the original signals into component signals such that each of the component signals represents independent sources of the resulting signal, such as eye-blinks, noise, etc. Using the calculated independent components from each of the partic-

ipants, we cluster them using EEGLab's built-in K-means functionality, such that similarly attributed components across participants are together.

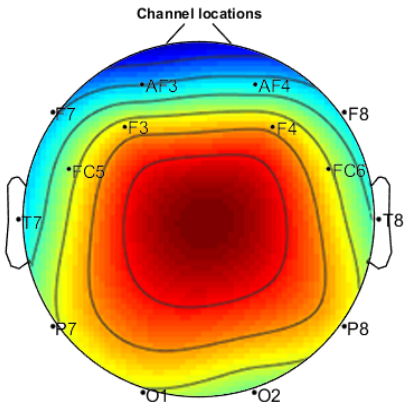


Figure 1.6: The names and locations of the 14 EEG electrodes in the Emotiv Epoc headset.

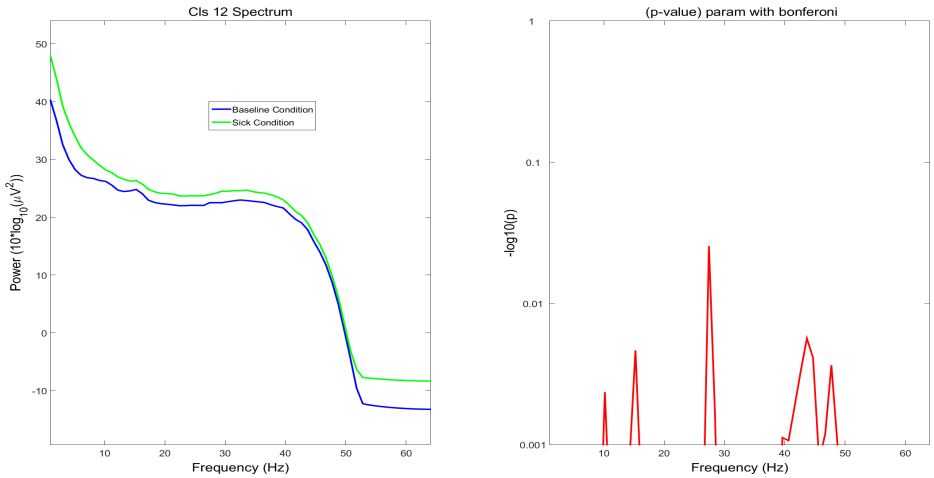


Figure 1.7: Comparison of the EEG power spectra between the baseline (blue) and virtual flythrough (green) for ICA cluster A. The paired t-test with Bonferroni-correction between the two spectra reveal $p < 0.001$ for much of the frequency ranges, and with $p < 0.05$ for most ranges.

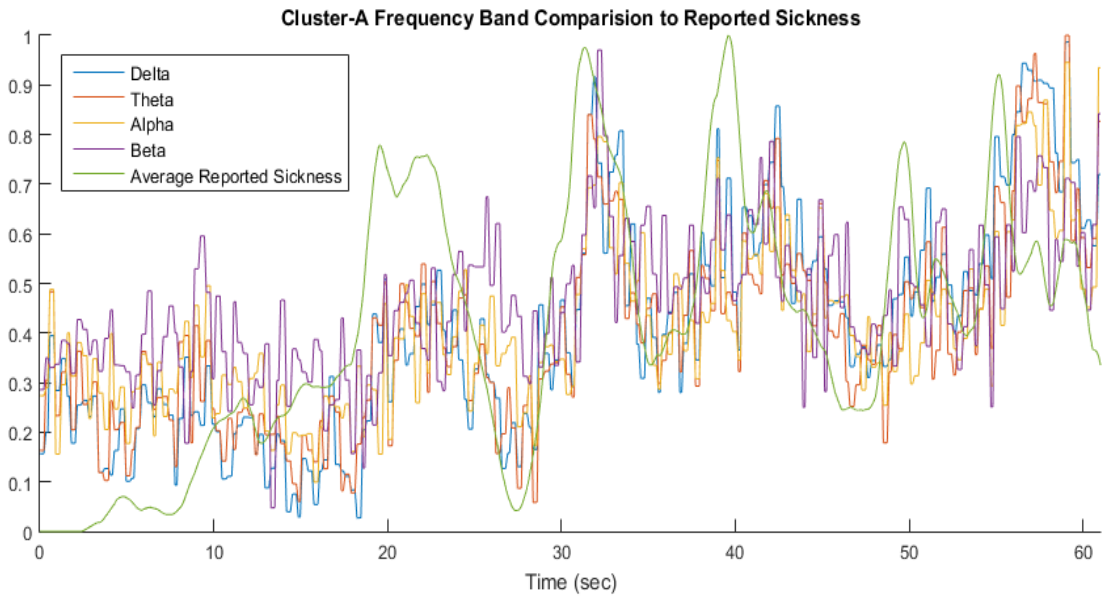


Figure 1.8: Average over four frequency bands for Cluster A compared with the average self-reported cybersickness (in green).

1.2.3 Results

From the 14 generated independent component (IC) clusters, we identified one of the clusters (cluster 12 which we call cluster A) as representative of cybersickness, which is shown in Figure 1.6. For this cluster, we found that there was a statistically significant power increase across many frequency ranges for the participants in the sick condition (when flying through the spaceport), as compared to the baseline condition (stationary), which can be seen in Figure 1.7. To validate our results further, we compare the clustered EEG data for the sick condition with the collected joystick information. For the different EEG frequency bands, we see a high correlation between the delta-wave $(1.0 - 4.0)$ Hz, theta-wave $(4.0 - 7.0)$ Hz, and Alpha-wave $(7.0 - 13.0)$ Hz signals, which can be seen in Figure 1.8.

One of the first and most crucial steps to mitigating cyber-sickness is to figure out how to quantify the characteristics of cybersickness. The result of this study offers an objective method to measure and quantify cybersickness, which can be used for further work on mitigating cybersickness.

1.3 Enhancing Deep Learning with Visual Interactions

1.3.1 Introduction

Recent advances in deep learning have led to impressive advances in many areas of science and visualization. However, deep learning requires huge amounts of finely and precisely labeled training data to function. Such complete training datasets are rare due to their, high cost, large amounts of time to create, erroneous labels, and labeling subjectivity. In addition, most datasets are abstract or non-spatial, further increasing the difficulty of their annotation. In Chapter 4, we present how we leverage the strengths of human pattern recognition and deep learning analytics to facilitate the exploration and interaction of high-dimensional datasets that have coarse labels, through an interactive and iterative process of refinement, re-training, and visualization. Our approach can be used to (a) alleviate the burden of intensive manual labeling that captures the fine nuances in a high-dimensional dataset by simple visual interactions, (b) replace a complicated (and therefore difficult to design) labeling algorithm by a simpler (but coarse) labeling algorithm supplemented by user interaction to refine the labeling, or (c) use low-dimensional features (such as the RGB colors) for coarse labeling and turn to higher-dimensional latent structures,

that are progressively revealed by deep learning, for fine labeling.

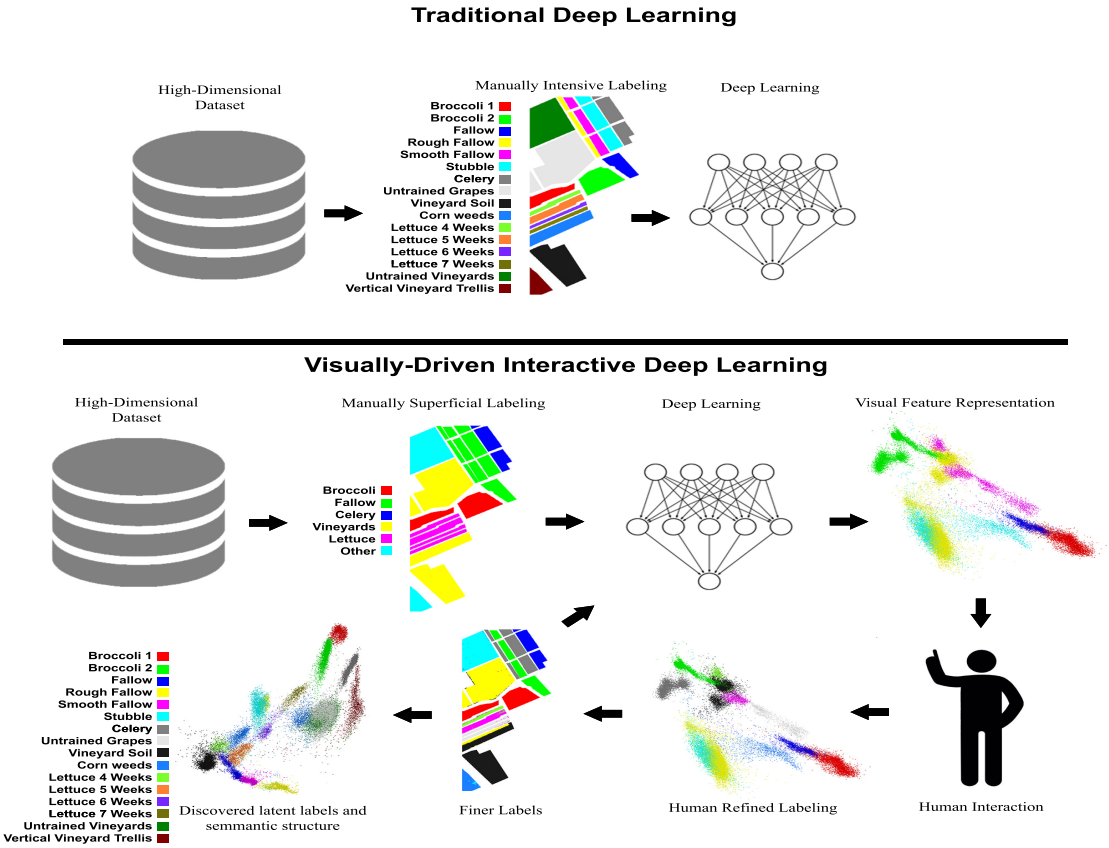


Figure 1.9: A brief illustration of the difference between traditional deep learning techniques and our approach. Deep learning traditionally requires a large, time-consuming, and precisely labeled dataset for training. For several reasons, such datasets may be inappropriately labeled. In our approach, we start with coarse labels (that are typically far easier to construct) and then refine them through an iterative process, involving visual interactions and deep learning.

1.3.2 Approach

Our goal is to take a coarsely labeled, non-spatial, high-dimensional dataset, and generate a low-dimensional representation that is easily interpretable by an analyst. To generate the low-dimensional representation, we employ two convolutional deep neural networks, a variational autoencoder, and a Siamese network. The variational autoencoder attempts to project the high-dimensional data to a two-dimensional representation in an unsupervised manner, relying entirely on the underlying distribution and patterns within the data. The variational autoencoder is constrained by requiring the network to reconstruct the original high-dimensional input data as accurately as possible from the projected two-dimensional layout. This provides analysts with a starting 2D data layout, derived from the data itself. Once the network generates the 2D distribution of data points, an analyst visually analyzes the distribution of the points, colored based on their currently assigned label (if provided), and manually selects points to assign or reassign their labeling. A point's label may change for many reasons, such as the point's proximity to other points of a different label and group evolution. The advantage here is the ability of humans to identify spatial patterns and provide that information to the network. Once the analyst has adjusted the labeling of the data points to their satisfaction, they are fed into the deep learning Siamese network system. The Siamese network consists of an internal network exactly similar with both internal structure and weights to the autoencoder, but compares how two input points are projected into the two-dimensional space based on their given labels. If the points are close

but share different labels, the network attempts to push them further apart in the projection, and if the points share a label, the network attempts to push them closer together. With this new model, a new distribution of points is generated using both the autoencoder and Siamese networks. This iterative process of projection and relabeling continues until the generated labeling and distribution is satisfactory. An overview of this process can be seen in Figure 1.9.

1.3.3 Results

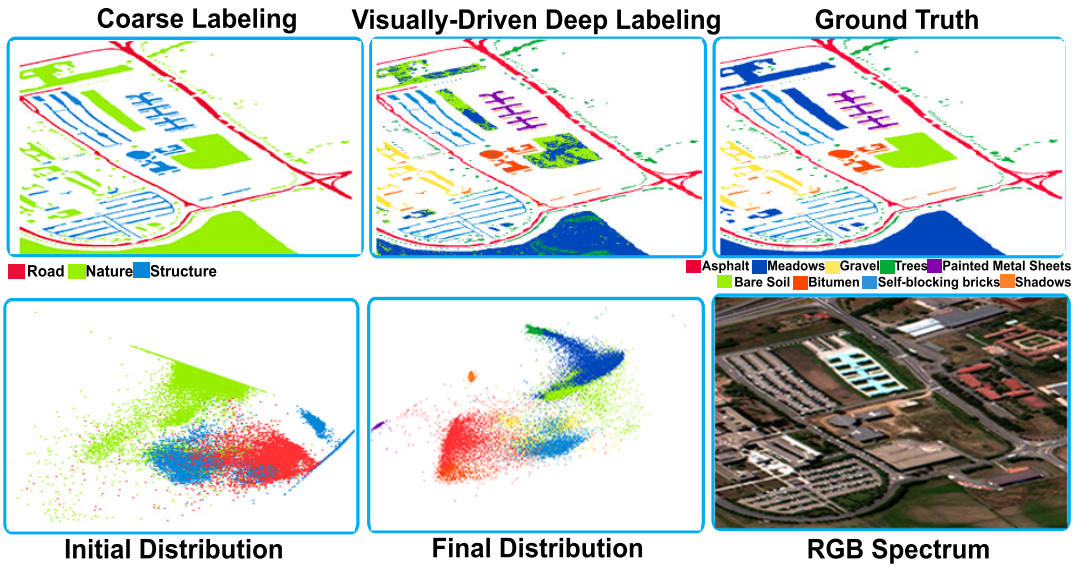


Figure 1.10: A comparison of the initial coarse labeling, to the generated refined labels, and the precise (true) labels of the Pavia university dataset after 3 iterations. Starting from the three initial categories: natural surfaces, roads, and buildings, we were able to reconstruct the distribution of the 9 labels with an accuracy of 88.2%.

To illustrate our process, we evaluate our technique on three datasets, a hyperspectral image of Pavia University [8], a hyperspectral image of Salinas Valley [9],

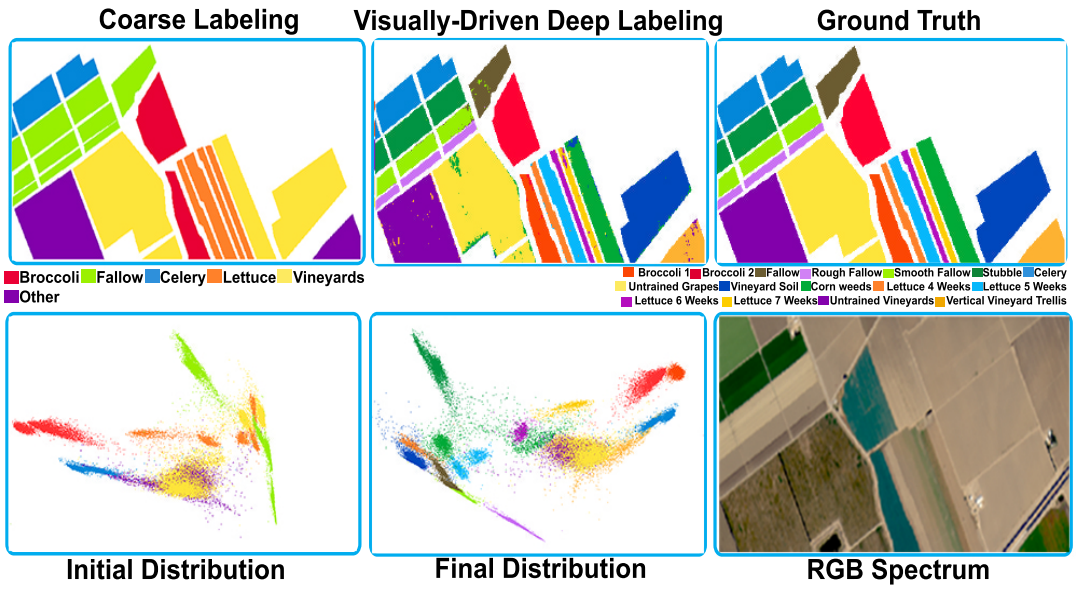


Figure 1.11: A comparison of the initial coarse labeling, the generated refined labels, and the precise (true) distribution of labels, of the Salinas valley dataset after five iterations. Starting with 6 initial coarse labels, we were able to reconstruct the distribution of the 16 labels with an accuracy of 97.4%.

and a collection of text-based queries from the D-Root DNS authority at the University of Maryland. For our purposes, we combined and reduced the number of known labels to start, with the goal of reconstructing the hidden ground truth labels for the first two datasets. The third DNS dataset is completely unlabeled. The initial and final distribution of the data with labeling can be seen for the first two datasets in Figure 1.10 and Figure 1.11 along with the evolution of the final dataset labeling distribution in Figure 1.12, demonstrating the ability of our approach to discover latent clusters and categories.

Our approach offers two main contributions. First, we are able to improve and enhance deep learning by introducing an approach that leverages human pat-

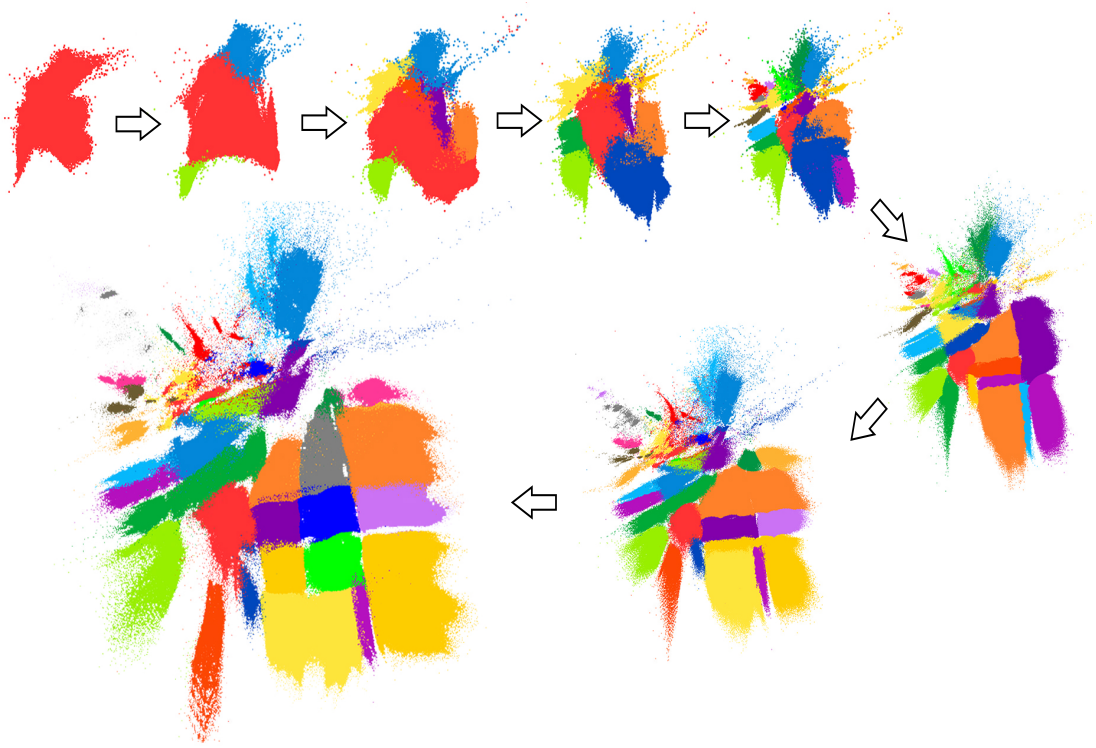


Figure 1.12: The evolution of the query-space representation over eight iterations, showing the influence of the iterative labeling. Note that for some clusters certain colors have been re-used due to the high-number of groups.

tern recognition to improve the data that is used to train the deep learning model. Second, we show that we are able to construct spatially-meaningful representations of abstract high-dimensional data that can be easily interpreted and manipulated. The next work is an expansion of the ideas established here, organized in a virtual three-dimensional environment, and evaluated using a real-world dataset and experts.

1.4 Visual Analytics for Root DNS Data

1.4.1 Introduction

The analysis of vast amounts of network data for monitoring and safeguarding the root DNS, a core pillar of the internet, is an enormous challenge. Understanding the distribution of the queries received by the root DNS, and how those queries change over time, in an intuitive manner is sought. Traditional query analysis is performed packet by packet, lacking global, temporal, and visual coherence, obscuring latent trends and clusters. In Chapter 5 we present our approach that leverages human pattern recognition and computational power of deep learning with 2D and 3D rendering techniques for quick and easy interpretation and interaction with vast amount of root DNS network traffic. Working with real-world DNS experts, we developed a visualization that reveals several surprising latent clusters of queries, potentially malicious and benign, discover previously unknown characteristics of a real-world root DNS DDOS attack, and uncover unforeseen changes in the distribution of queries received over time. These discoveries will provide DNS analysts with a deeper understanding of the nature of the DNS traffic under their charge, which will help them safeguard the root DNS against future attack.

1.4.2 Approach

Our visualization consists of three main components, an IP-space visualization, a 2D query-space visualization, and a 3D query-space visualization. The purpose

of these visualizations is to provide analysts with a well-rounded representation of the DNS packets they receive and process, providing a high-level overview while preserving low-level salient details. The first visualization, portraying the IP-space distribution of the received packets, as shown in the left of Figure 1.13, with more detail provided in Figure 1.14, conveys a 2D representation of the 4D IPv4 space, with the X-axis as a linear combination of the first two IPv4 octets, and the Y-axis as the last two IPv4 octets. Within each cell, a five-second period of packet accumulation is presented, conveying the number and variety of packets received using the size and color of glyphs. As these packet streams aggregate over time along the Z-axis they reveal patterns, anomalies, and distinct characteristics of DNS attacks. This representation provides DNS analysts with an unparalleled level of temporal and visual coherence for understanding the changing characteristics of their data.

The second visualization, as shown in the middle of Figure 1.13, leverages deep learning to project and organize the distribution of high-dimensional non-spatial DNS queries in an easy-to-interpret two-dimensional spatial layout. In this representation, distinct clusters of queries emerge, enabling analysts to gain a sense of the normal and abnormal distribution of queries along with the number of times each of those queries was received. Using this projection, a set of semantic axes arises, with alphabet-based queries appearing on the left, and number-based queries appearing on the right of the space, along with a general usage of alphanumeric characters on the top, and non-alphanumeric characters on the bottom. Through our investigation with DNS experts, we have identified several interesting general

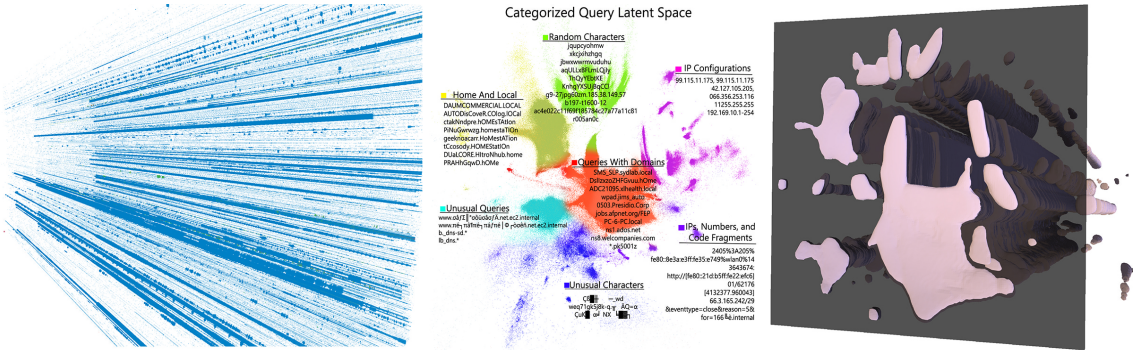


Figure 1.13: Our root-DNS dual-visualization that provides both high and low-level overviews and interactions of the IP and query spaces. IP packet traffic is visualized on the left, revealing hidden patterns, IP distributions, and a real TCP-SYN flood attack. A two-dimensional query-space generated using deep learning portrays a spatial distribution of received queries and counts. The right image portrays the spatial distribution of queries as they change over time, revealing the diminished number of received queries due to a DDOS.

groups of queries, as presented in the left of Figure 1.15. Such a query-based spatial visualization was previously unknown to our experts, who achieved new insights into their traffic using the system. The third visualization, as shown in the right of Figure 1.13 and Figure 1.15, is a 3D temporal expansion of the 2D query-space visualization, providing additional information such as the evolution of the distribution of queries over time, revealing patterns and anomalies. Through such visualizations, our analysts have been able to make connections between the traffic in the IP-space and the query-space, discovering previously unknown clusters, repeating patterns, and anomalies.

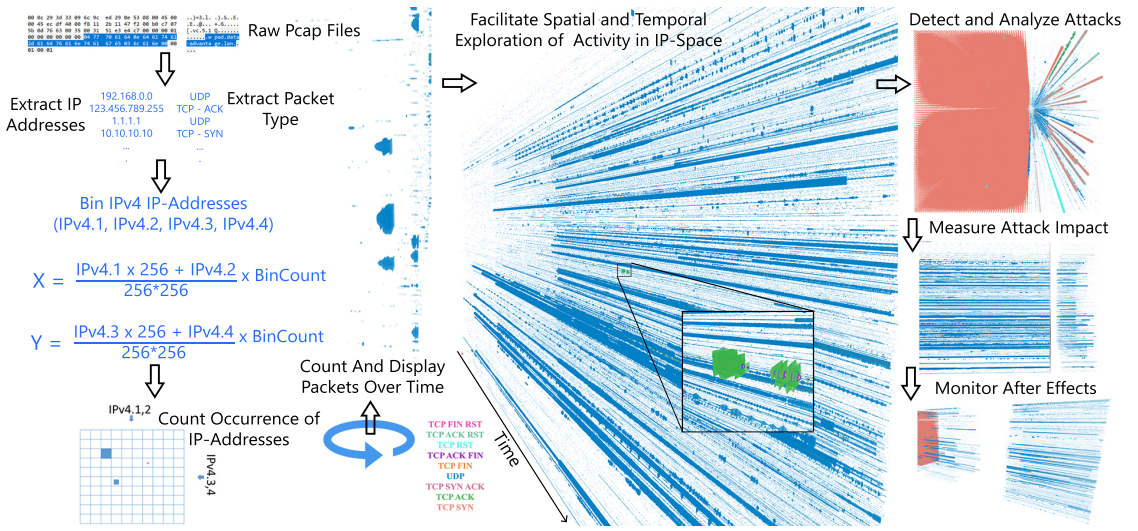


Figure 1.14: Overview of the process from raw pcap files to The Flow-Map IP-Space visualization. Starting from a binary pcap file, we extract and count the occurrence of each IPv4 IP-Address and type of packet. Next, the IPs are converted from a 4D to a 2D grid representation, with glyphs scaled and colored based on the number and type of packets. This process repeats for each time slice, with slices stacked along the z-axis. The result is then visualized using 3D accelerated rendering, which allows for high-level structure and low-level analysis, to help analysts establish a sense of normalcy (central blue image), identify outliers (green TCP burst), classify and characterize attacks (top right), measure attack impacts (middle right), and monitor after effects (lower right).

1.4.3 Results

We have validated our approach on a real-world dataset that includes a DDOS attack from one of the servers under the domain of the D-Root DNS authority. In the IP-space visualization, a baseline before the attack is established, with the ma-

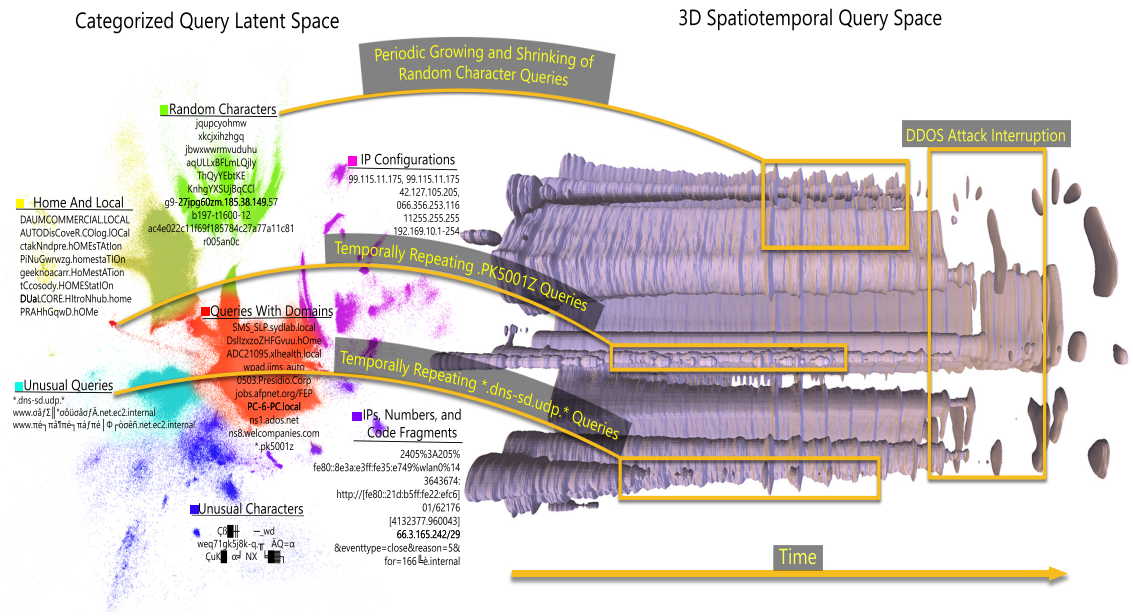


Figure 1.15: In the lower-left, the 2D query-space clustered into semantic query categories. There is a general trend with alphabet-based queries towards the left, and number-based queries the right, along with a general trend to alphanumeric characters at the top, and non-alphanumeric characters at the bottom. The lower-right shows the temporal query visualization, portraying a high-level temporal overview of the distribution of queries. The top reveals a selection of interesting observations, such as rapidly diminishing groups of queries at the start, temporally repeating groups of queries, the large reduction in queries during the attack, and an overall decrease in the number of queries over time.

jority of the IP-space consistently empty, and many of the used IP-bins falling into one of two categories, consisting of low or high volume traffic. Within the flow of the traffic, as shown in Figure 5.1 several bursts of short-lived anomalies emerge, and other temporally repeating bins of traffic are observed. In addition, the scale, duration, and distinct characteristics of a real DDOS attack are revealed,

such as the distinct range of IP-bins used, and the oscillatory nature of the flow of packets. The characterization of this DDOS attack through our visualization provided more detail than the official report on this attack released by the Root Servers (http://root-servers.org/news/events-of-20160625.txt). The two and three-dimensional visualizations of the query distributions reveal the wide variety of received queries for the DNS to resolve. Historically, queries are generally ignored in packet analysis due to their unstructured, high-dimensional, and abstract nature. In this visualization, the distribution, count, and evolving behavior of these queries are revealed. In Chapter 5 we go into more detail about the various discoveries made by the DNS experts. There were three categories of discoveries made, (1) those involving the distribution of the random characters, as shown in Figure 1.16, (2) a distribution of numeric and code-fragments (Figure 1.17), and (3) a distribution of queries containing non-alphanumeric characters (Figure 1.18). Through joint interactions between the query-space and IP-space visualizations, analysts can see from which IP-bins and when certain queries arrive, indicating if a particular query or set of queries comes from a wide or small set of users and the variety of queries sent from a particular bin. Using these visualizations, our DNS experts were able to learn many new aspects of the data that passes through their system.

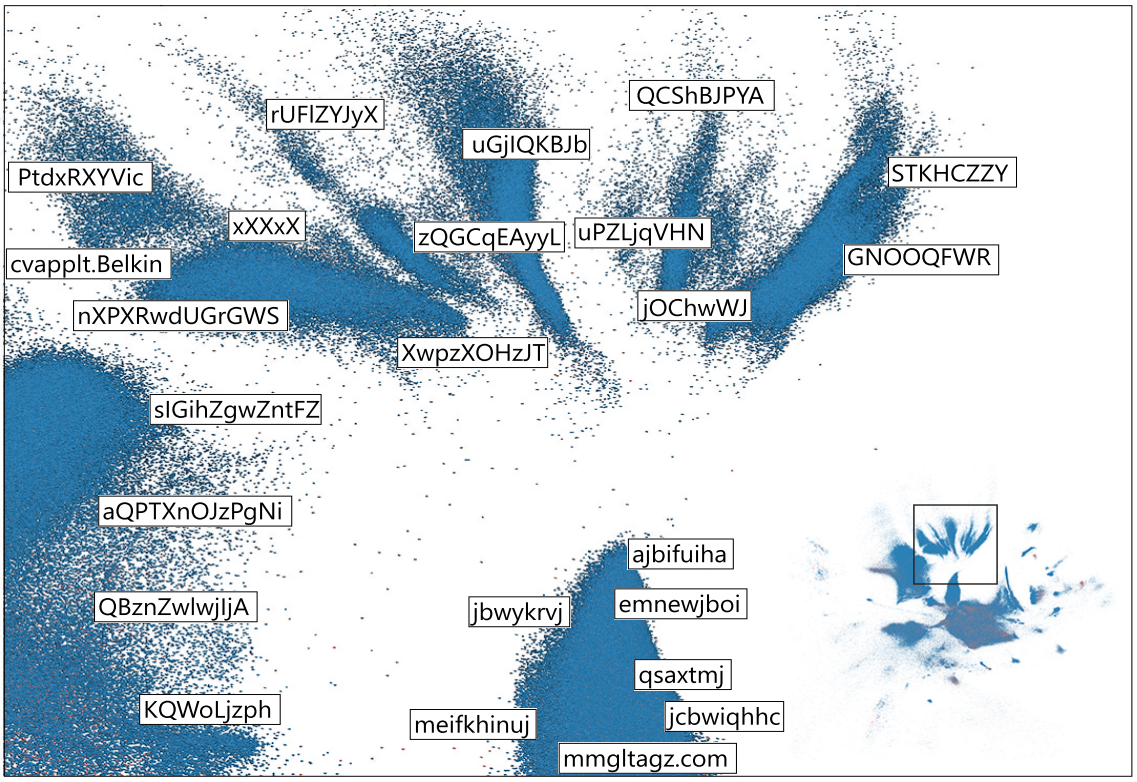


Figure 1.16: The region of the query-space consisting of different distributions of random characters.

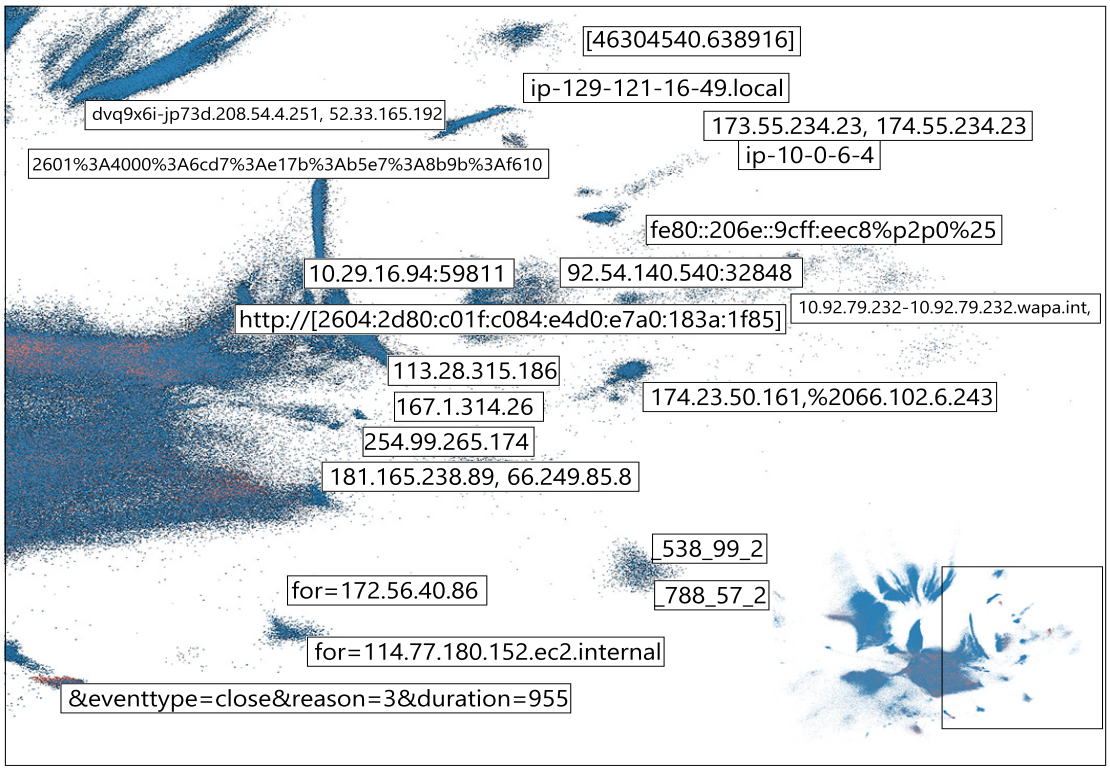


Figure 1.17: The region of the query-space consisting of different distributions of IP addresses, fragments, and expressions.

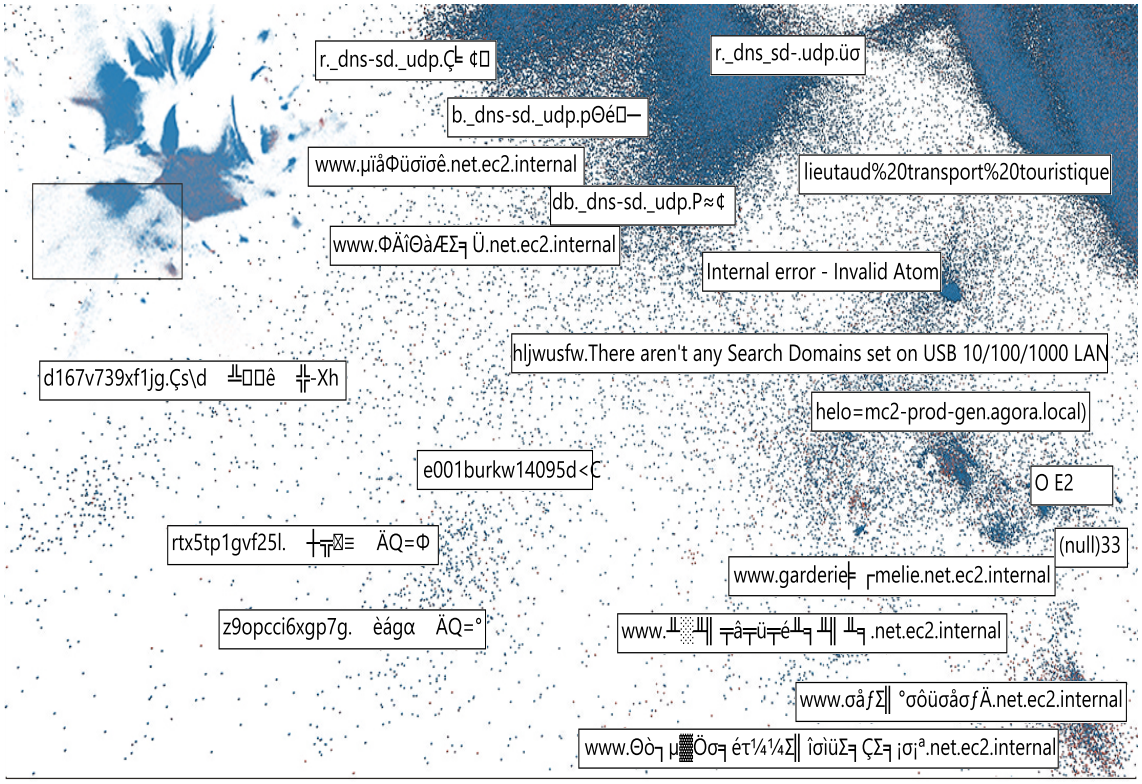


Figure 1.18: The region of the query-space consisting of unusual characters and queries.

Chapter 2: Virtual Memory Palaces: Immersion aids Recall

2.1 Introduction

Throughout history, humans have relied on technology to help us remember information. From cave paintings, clay tablets, and papyrus, to modern paper, audio, and video, we have used technology to encode and recall information. This chapter addresses the question of whether virtual environments could be the next step in our quest for better tools to help us memorize and recall information. Virtual reality displays, in contrast to traditional displays, can combine visually immersive spatial representations of data with our vestibular and proprioceptive senses. The technique of memory palaces provides a natural spatial mnemonic to assist in recall. Since classical times, people have used memory palaces (Method of Loci), by taking advantage of the brain's ability to spatially organize thoughts and concepts [10–12]. In a memory palace, one mentally navigates an imagined structure to recall information [13, 14]. Even the Roman orator Cicero is believed to have used the memory palace technique by visualizing his speeches and poems as spatial locations within the auditorium he was in [13, 15]. Spatial Intelligence has been associated with a heightened sense of situational awareness and of relationships in one's own surroundings [16, 17].

Research in cognitive psychology has shown that recall is superior in the same environment in which the learning took place [18]. Such findings of context-dependent memory have interesting implications for virtual environments that have not yet been fully explored. Imagine, for instance, a victim of a street aggression being asked to recall the appearance details of their assailant. Virtual environments that mirror the scene of the crime could provide superior assistance in recall by placing the victim back into such an environment.

In this chapter we present the results of a user study that examined if virtual memory palaces could assist in superior recall of faces and their spatial locations aided by the *context-dependent* immersion afforded by a head-tracked head-mounted display (HMD condition) as compared to using a traditional desktop display with a mouse-based interaction (desktop condition). To explore this question we designed an experiment where participants were asked to recall specific information in the two environments: the HMD condition and the desktop condition. We created the virtual memory palaces prior to the start of the study. Our hypotheses are as follows:

- Hypothesis 1: The participant memory recall accuracy will be higher in the HMD condition as compared to the desktop condition due to the increased immersion.

- Hypothesis 2: Participants will have higher confidence in their answers in the HMD condition as compared to the desktop condition.

The experiment was a within-subject, $2 \times 2 \times 2$ Latin-square design, ensuring all the different combinations of variables and factors were accounted for. The

experimental results of our study support both hypotheses.

2.2 Related Work

Memory palaces have been used since the classical times to aid recall by using spatial mappings and environmental attributes. Figure 2.1 shows a depiction of a memory palace attributed to Giulio Camillo in 1511. The idea was to map words or phrases onto a mental model of an environment (in this case an amphitheater), and then recall those phrases by mentally visualizing that part of the environment.

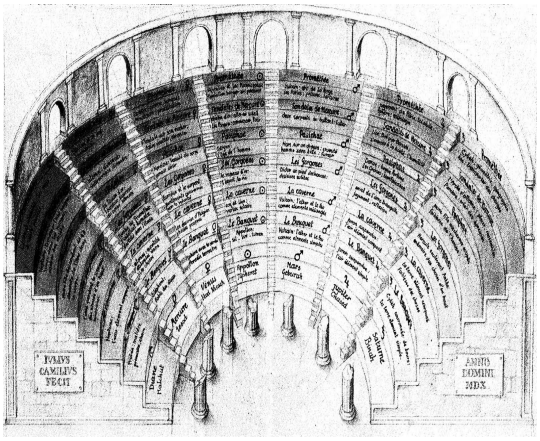


Figure 2.1: Giulio Camillo's depiction of a memory palace (1511 AD). Memory palaces like this have been used since the classical times as a spatial mnemonic.

An important component of the memory palace technique is the subjective experience of being virtually present in the palace, even when one is physically elsewhere. This notion of *presence* has long been considered central to virtual environments, for evaluation of their effectiveness as well as their quality [19]. More precisely, Slater *et al.* [20] developed the idea of place illusion (PI), referring to the aspects of presence "constrained by the sensorimotor contingencies afforded by the

virtual reality system". Sensorimotor contingencies are those actions which are used in the process of perceiving the virtual world, such as moving the head and eyes to change gaze direction or seeing around occluding objects to gain an understanding of the space o2001sensorimotor. Slater *et al.* [20] therefore concluded that establishing presence or "being there" for lower-order immersive systems such as desktops is not feasible. In contrast, the sensorimotor contingencies of walking and looking around facilitated by head-mounted displays contribute to their higher-order immersion and establishing presence.

Recent research in cognitive psychology [21] suggests that the mind is inherently embodied. The way we create and recall mental constructs is influenced by the way we perceive and move [22, 23]. The memory system that encodes, stores, recognizes, embodies, and recalls spatial information about the environment is called spatial memory [24]. Several studies have found that embodied navigation and memory are closely connected [25, 26]. Madl *et al.* [24] states that there are several different types of brain mechanisms involved in processing spatial representations in the brain. *Grid cells* in the entorhinal cortex, used for path integration, are activated by changes in movement direction and speed [27, 28]. *Head-direction cells* activate in the medial parietal cortex when the head points in a given direction, providing information on viewing direction [29]. *Border cells* and *boundary vector cells* in the subiculum and entorhinal cortex activate in close proximity to environment boundaries, depending on head direction [28, 30]. Lastly, *place cells* in the hippocampus activate in specific spatial locations, independent of orientation, providing an internal representation of the environment [31, 32]. It is believed that place

cell fields arise from groups of grid and boundary cells which activate for different spatial scales and environmental geometry to provide a sense of location [33, 34]. In addition, these hippocampal cells also provide information about place-object associations, associating place cell representations of specific locations with the representations of specific objects in recognition memory [35, 36]. This leads us to the possibility that a spatial virtual memory palace, experienced in an immersive virtual environment, could enhance learning and recall by leveraging the integration of vestibular and proprioceptive inputs (overall sense of body position, movement, and acceleration) [32].

2.2.1 Memory Palaces on a Desktop Monitor

Legge *et al.* [37] have compared the use of the traditional method of Loci using a mental environment against a 3D graphics desktop environment. In this study, the subjects were divided into three groups. The first group was instructed to use a mental location or scene, the second group a 3D graphics scene, and the third (control) group was not informed on the use of any mnemonic device. The subjects in the three groups were given 10 to 11 uncorrelated words and asked to memorize the words with their mnemonic device, if any. The users then recalled the words serially. This study found that the users who used a graphics desktop environment as the basis for their method of Loci performed better than those using a mental scene of their choice, and those who were not instructed on a memory strategy did not perform as well as those who were instructed to use the memory strategy.

Fassbender and Heiden [38] compared the ability of users to recall a list of 10 words when using a desktop compared to memorizing the word list. The authors created a navigable 3D castle with 4 sections and 10 objects, where each object has a visual and audible component, with the idea that a user will associate a word with that object. First, each user was given 10 words to memorize and then were asked to recall as many as they could after a two-minute distraction task. Next, each user was explained and shown the 3D castle on a desktop. After being given time to learn the associations between the words, images, and audio, the users were evaluated on their ability to recall the words in the 3D castle on the desktop. The study found that there was no significant difference between the users' ability to immediately recall the words after a two-minute break, but after one week there was a 25% difference in recall in favor of the 3D graphics desktop memory palace environment condition. The above studies show that compared to a purely mental mnemonic, a graphics-desktop setup is better in assisting retention and recall.

Both of these studies have been carried out on desktops and not in immersive HMDs. In our study we compare the performance of users on a desktop compared with an immersive HMD.

2.2.2 Memory Palaces on Multiple Displays

The efficacy of varying immersion levels by changing the field of view has also been studied in the context of procedural training [39]. Sowndararajan *et al.* [40] compared subject performance for a simple and complex procedural task (involving a

different number of steps and interactions), but with two different fields of view – one with a laptop and the other with a large rear-projected L-shaped display. The study had participants trained on two procedures and the performance with the two levels of immersion was compared. The study found that higher levels of immersion (in this case, field of view) were more effective in learning complex procedures that reference spatial locations. In addition, there was no statistical difference in performance for the simple task for the different levels of immersion. Ragan *et al.* [41] carried out a user study in which participants were asked to memorize and recall the sequence of placement of virtual objects on a grid shown on three rear-projected screens (one front and two side screens). The participants were divided into multiple groups that performed the task with different fields of view and fields of regard. The field of view is the size of the visual field seen in one instant while the field of regard is the total size of the visual field that *can* be seen by a user [39]. Both are measured in degrees of visual angle. Ragan *et al.* found that higher field of view and field of regard produced a statistically-significant performance improvement.

The above studies examined the effectiveness of memory recall of objects, their locations, and the sequence of placement actions, in a limited field of view and field of regard in monoscopic display environments with multiple monitors. The field of regard in these studies did not surround the viewer completely. In our study we wanted to examine the effectiveness of stereoscopic, spherical-field of regard afforded by modern HMDs compared to a desktop for memory recall of objects and their spatial locations.

2.2.3 Search and Recall in Head-mounted Displays

Pausch *et al.* [42] studied if immersion in a virtual environment using a HMD aids in searching and detection of information. For their study they created a virtual room with letters distributed on walls, ceiling, and floor. A user was placed in the center of this room and was asked if a set of letters was present or not. The test was conducted using a HMD and a traditional display with a mouse and keyboard. They found that when the search target was present, the HMD and the traditional display had no statistically significant difference in performance. However, when the target was absent, the users were able to confirm its absence faster in the HMD than on the traditional display. In addition, the users that used the HMD first and then moved to a traditional desktop had better performance than those who used the desktop first and then the HMD. This suggests a positive transfer effect from the HMD to a desktop. Our user study is highly influenced by the study of [42], but in our study, users perform recall rather than search.

Ruddle *et al.* [43] compared user navigation time and relative straight-line distance accuracy (amount of wasteful navigational movement) between a HMD and a traditional desktop. Users were then asked to learn the layout of two virtual buildings, one using a HMD, and the other using a desktop. After familiarizing themselves with the buildings, each user was placed in the lobby of that building and were told to go to each of five named rooms and then return to the lobby. They found that the users wearing the HMD had faster navigation times, less waste-full movement, and were more accurately able to estimate distances, compared to those

using a desktop.

Mania *et al.* [44] examined accuracy and confidence levels associated with recall and cognitive awareness in a room filled with objects such as pyramids, spheres, and cubes. Participants were exposed to one of the following scenarios: (a) a virtual room using a HMD, (b) a rendered room on a desktop, or (c) a real room experienced through glasses designed to restrict the field of view to 30° to match that of the HMD and desktop. All the four walls of the room were distinct. After three minutes of exposure, the participants were given a paper containing a representation of the room which included numbered positions of objects in the various locations. The participants were asked to recall which objects were present and where they were located in the room, and to give a confidence and awareness state with each answer. The study evaluated the participants immediately after the exposure and then again after one week. The study found that immediately after the exposure the participants had the most accurate recall in the real-world scene, were slightly less accurate and confident in the HMD, and least accurate and confident on the desktop. After one week, the overall scores and confidence levels dropped consistently across the board, with the viewing condition having no effect on the relative reduction in performance. In this inspirational study, the participants only experienced one display. In our study the participants were exposed to both the desktop and the HMD. This makes it possible to compare recall for the same user across the two display modalities. Further, to use the context provided by immersion, the participants in our study were asked to recall the information while viewing the same virtual scenes on the same display, rather than recording their answers on a

representation of the scene on paper.

Harman *et al.* [45] explored immersive virtual environments for memory recall by having participants take on the role of a boarding an airplane in a virtual airport. After the experience, the participants were asked about the tasks they performed. The participants who experienced the virtual airport in a HMD had more accurate recall than those who used the desktop. In this study each participant used either a HMD or a desktop. Also, the evaluation of the memory recall was done outside of the visual experience, through a questionnaire. In our study, not only do participants experience the virtual environment in both, HMD and desktop, but are also asked to recall in the same environment in which they experienced the information.

2.2.4 Embodied Interaction and Recall

Virtual walk-throughs have been one of the earliest applications of virtual worlds [46]. Brooks *et al.* [47] studied if active participants had superior recall of the layout of a 3D virtual house on a desktop compared to passive participants. Active participants controlled camera navigation via a joystick, while passive participants observed the navigation. They found that active participants had a superior environment layout recall compared to those who were passive. However, they also found that there was no statistically significant difference between the recall or recognition of objects (such as furniture or entrances and exits of a room) or their positions within the environment between the active and passive participants. This suggests that memory was only enhanced for those aspects of the environment that

were interacted with directly – particularly the environment which was navigated.

Richardson *et al.* [48] had users learn the layout of a complex building through either 2D maps, physically walking through the real building, or through a 3D virtual representation of that building built using the Doom II engine and shown on a desktop. The study found that when the building was a single floor, the real-world and virtual-environment-trained users had comparable results. However, when the building had two floors, relative view orientation during learning and testing mattered. If the participants were in the same orientation that they had used during learning, they were able to navigate the environment just as well as those who were physically in the environment. However, participants were susceptible to disorientation if their starting-out views were different between their training and testing. The authors concluded that training in the virtual and real-world environments likely used similar cognitive mechanisms.

Wraga *et al.* [49] compared the effectiveness of vestibular and proprioceptive rotations in assisting recall by having participants recall on which of the four walls was a object located relative to their orientation before and after rotation. Participants were placed in a virtual room with four distinctly colored alcoves on four walls and given time to learn and recognize the alcoves. Participants would then rotate, either using the HMD accelerometer or a joystick, to find a certain object on one of the alcoves as described by the tester. Once the user was looking at that object on one of the alcoves, their view would be frozen and the tester would ask the participant to state where a particular (different) alcove was relative to their orientation. They found that users in a HMD were better able to keep track of the

objects by rotating their heads as compared to using a joystick. In another experiment, the authors also found that users in a HMD who controlled their bearing in a virtual world by actively rotating in a swivel chair were better able to keep track of an object than those that were being rotated by a tester. In our study we expect vestibular and proprioceptive inputs to improve performance in the HMD. We study how well people can recall information regardless of their orientation. In addition, our objects are distributed in more than four unique locations.

Perrault *et al.* [50] leveraged the method of Loci technique by allowing participants to link gestural commands, that would control some system, to physical objects within a real room. They compared their interaction technique to a mid-air swipe menu which relies on directional swiping gestures. Their idea was to leverage spatial, object, and semantic memory to help users learn and recall a large number of gestures and commands. In a home environment, participants were shown a command (or stimulus) on a television and then performed a motion that a Microsoft Kinect would track and record as representing that command. For the mid-air swipe the participant would perform a 2-segment marking menu gesture. For the physical Loci, the participant would simply point at an object in the environment that they wanted associated with the command, such as a chair or poster. Once the gestures and physical Loci were trained, the participants went into the recall phase. In this phase, a command would be presented on the screen and the user had to quickly and accurately perform the corresponding gesture. The system would then show if the participant performed the correct gesture or pointed at the correct Loci object that they originally assigned for that command. The authors found that users, when us-

ing their physical Loci technique, had superior command recall and was more robust compared to the more traditional mid-air swipe menu.

2.3 Method

A memory palace is a spatial mnemonic technique where information is associated with different aspects of the imagined environment, such as people, objects, or rooms, to assist in their recall [13,14]. The goal of our user study was to examine if a virtual memory palace, experienced immersively in a head-tracked stereoscopic HMD, can assist in recall better than a mouse-based interaction on a traditional, non-immersive, monoscopic desktop display. Previous work has examined the role of spatial organization, immersion, and interaction in assisting recall.

This study is different from the previous work in several ways. First, we are focusing on spatial memory using a 3D model of a virtual memory palace, rather than relying on other forms of memory (such as temporal/episodic). Second, both the training and testing (recall) phases take place within the same virtual memory palace. Third, participants used both the desktop and HMD displays, which allows us to compare each participant's recall across displays. Lastly, the content used in previous studies was either abstract, verbal, textual, visually simplistic, low in diversity, or time-based, whereas our study uses faces, with unique and diverse characteristics.

2.3.1 Participants

Our user study for this research was carried out under IRB ID $751321 - 1$ approved on August 7, 2015 by the University of Maryland College Park IRB board. In this study, we recruited 40 participants, 30 male and 10 female, from our campus and surrounding community. Each participant had normal or corrected-to-normal vision (self reported). The study session for each participant lasted around 45 minutes.

2.3.2 Materials

For this study we used a traditional desktop with a 30 inch (76.2) cm - diagonal monitor and an Oculus DK2 HMD. The rendering for the desktop was configured to match that of the Oculus with a resolution of 1920×1080 pixels (across the two eyes) with a rendering field-of-view (FOV) of $100°$. In order to give the desktop display the same field-of-view as the HMD, the participants were positioned with their heads 10 inches (25.4 cm) away from the monitor. The software used to render the 3D environments on both the desktop and HMD was identical and was designed in-house using C++ and OpenGL accelerated rendering. The rendering was designed to replicate a realistic looking environment as closely as possible, incorporating realistic-lighting, shadows, and textures. The models (the medieval town and palace), were purchased through the 3D modeling distribution website TurboSquid [51, 52].

2.3.3 Design

The participants were shown two scenes, on two display conditions (head-tracked HMD and a mouse-based interaction desktop), and two sets of faces (within-subject design), all treated as independent variables, with the measured accuracy of recall as the dependent variable. The two scenes (Virtual Memory Palaces) consist of pre-constructed palace and medieval town environments filled with faces. We decided to use faces given the previous work [53, 54] showing the effectiveness of memory palaces aiding users in recalling face-name pairs. We used faces as the objects to be memorized and carefully partitioned them into two sets of roughly equal familiarity. We quantified the familiarity of the faces using Google trends data over the four months preceding the study. The faces are shown in the appendix (at the very end of the dissertation) in Figures 1 and 2, and the Google trends statistics are presented in Tables 1 and 2. There was no statistically significant difference between the two sets of Google trends data: $p = 0.45 > 0.05$.

The faces in the palace and medieval town were hand positioned for each environment, before the start of the study, and remained consistent throughout the study. We distributed the faces at varying distances from the users' location (see Figure 2.4) so that they surrounded and faced the user. Since we used perspective projection, the sizes of the faces varied. However, the distribution of the angular resolution of the faces across the two sets/environments were not statistically different, with $p = 0.44 > 0.05$ (see Table 3 in the appendix).

Users were allowed to freely rotate their view but not translate. This effectively

simulated a stereoscopic spherical panoramic image with the participant at its center. Our motivation behind this study design decision was that if even this limited level of immersion could show an improvement in recall, it could lead to a better-informed exploration of how greater levels of immersion relate to varying levels of recall.

2.3.4 Procedure

First, each participant familiarized themselves with all the 42 faces and their names used in the study. The participants received a randomly permuted collection of printouts, each containing a face-name pair used in the study. Participants were given as much time as needed until they stated when they were comfortable with the faces. In general, participants did not spend more than 5 minutes on this familiarization.

Next, each participant was told about the training and testing procedure, including how many faces were going to be in each scene (21), how much time they had to view the faces (5 minutes), how the breaks would work, that the faces would be replaced with numbers in the recall phase, and that they were to give a name and confidence for their recalled faces for each numbered position. In almost every case we recorded the answer as the name explicitly recalled by the participant. However, in rare, exceptional circumstances, when the participants gave an extremely detailed and unambiguous description of the face ("fat, wore a wig, was King of France, and is not Napoleon" for King Louis), we marked it correct. Next, each participant was placed either in front of a desktop monitor with a mouse or inside a head-tracked



Figure 2.2: The two Virtual Memory Palace scenes used in our user study (a) an ornate palace, and (b) a medieval town, as seen from the view of the participants.

stereoscopic HMD. They were given as much time as they desired to get comfortable, looking around the scene without numbers or faces. The users rotated the scene on a desktop monitor with a mouse and in the HMD setup they rotated their head and body, but no further navigation was possible.

Once each participant was comfortable with the setup and the controls, a set of 21 faces were added to the 3D scene and distributed around the entire space as shown in Figure 2.4. We used two such scenes – a palace and a medieval town, shown in Figure 2.2. The faces were divided into two consistent sets used for the whole study; if a face appeared in one set (or scene) for a given participant, it would not be shown again in the second set or scene.

To cover all possible treatments of the $2 \times 2 \times 2$ Latin square design, each participant was tested in both scenes, both display conditions (HMD and desktop), and both sets of faces, with their relative ordering counterbalanced across partici-

pants. The 21 faces within the scene were presented to the participants all at once, and the participants were able to view and memorize the faces in any order of their choosing. The faces were deterministically placed in the same order for all participants. However, since the participants were free to look in any direction, the order of presentation of faces was self-determined. Each participant was given five minutes to memorize the faces and their locations within the scene. After the five-minute period, the display went blank and each participant was given a two-minute break in which they were asked a series of questions. Questions we asked included how each participant learned about the study, what their profession/major was, and what were their general hobbies or interests. In the second half of the study, during the break for the alternative display, we asked how often a participant used a computer, what their previous experience was with VR, and their general impressions of VR. We consistently asked these questions of each participant, but did not record the responses.

The reasons for these study design decisions are rooted in foundational research in psychology on memory. From the seminal work by [55] we learn that the working memory [56] can only retain 7 ± 2 items. According to Atkinson *et al.* [57] the information in the short-term memory decays and is lost within a period of $15 - 30$ seconds. We feel confident that having participants recall 21 faces after a two-minute break will engage their long-term memory.

After the two-minute break, the scene would reappear on the display with numbers having replaced the faces, as shown in Figure 2.3. Each participant was then asked to recall, in any order, which face had been at each numbered location.



Figure 2.3: Virtual memory palace: recall phase

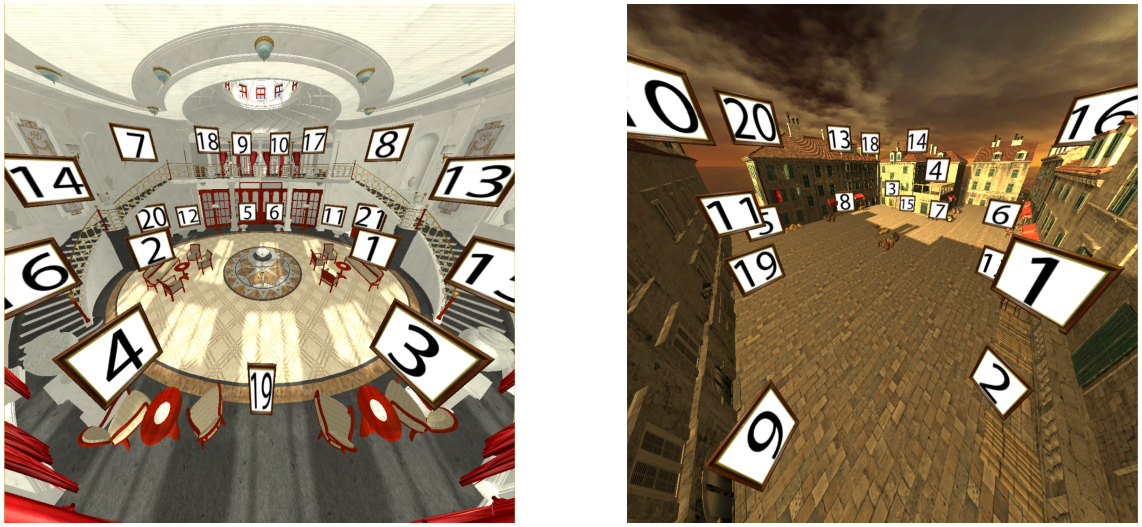


Figure 2.4: Locations of faces and numbers in the Virtual Memory Palaces used in our user study (a) an ornate palace, and (b) a medieval town. Note that this is not the view the participants had during the experiment, these pictures are used to convey the distribution of the face locations. The participants would have been placed in the middle of these scenes surrounded by the faces as seen in Figure 2.2.

During this recall phase, each participant could look around and explore the scene just as they did in the training phase, using the mouse on the desktop or rotating their head-tracked HMD. Each participant had up to five minutes to recall the names of all the faces in the scene. Once the participant was confident in all their answers, or the five-minute period had passed, the testing phase ended. After a break, each participant was placed in the other display that they had not previously tested with. The process was then repeated with a different scene and a different set of 21 faces to avoid information overlap from the previous test.

For each numbered location in the scene, the participants verbally recalled the name of the face at that location, as well as a confidence rating for their answer, ranging from 1 to 10, with 10 being certain. If a participant had no answer for a location, it was given a score of 0. The results were hand recorded by the study administrator, keeping track of the number, name, user confidence, and any changes in a previously given answer.

To mitigate any learning behavior from the first trial to the second, we employed a within-subject trial structure, using a 2 (HMD-condition to desktop-condition vs desktop-condition to HMD-condition) x 2 (Scene 1 vs Scene 2) x 2 (Face Set 1 vs Face Set 2) Latin-square design. By alternating between the displays shown first (2), the scenes (2), and the faces (2), we expect to mitigate any confounding effects. At the end, each participant was tested on the two display conditions, desktop and HMD, on two different scenes, and with two different sets of 21 faces. We note that participants could have used personal mnemonics to help remember the locations and ordering of faces. However, since we evaluated recall for each participant over

a desktop and a HMD, their performance should be counterbalanced between the two display conditions.

2.4 Results

Our hypothesis is that a virtual memory palace experienced in an immersive head-tracked HMD (the HMD condition) will lead to a more accurate recall than on a mouse-controlled desktop display (the desktop condition). In addition, we hypothesized that participants should be more confident in their answers in the headset and make fewer mistakes or errors in recall. Our null hypothesis is that there is no statistical difference between the accuracy and confidence of results between the HMD and desktop conditions, and that there is no statistical difference in the ordering of the display conditions.

We confirmed using a four-way mixed-ANOVA that there were no statistically significant effects on recall due to the scenes (palace and town) $F(1, 79) = 0.27, p > 0.05$, the two sets of 21 faces $F(1, 79) = 0.27, p > 0.05$, or the ordering of display conditions (HMD followed by desktop vs desktop followed by HMD) $F(1, 79) = 1.93, p > 0.05$. We found that there was a statistically significant effect for the display condition (HMD vs desktop) with $F(1, 79) = 4.6$ and $p < 0.05$. This means participants were able to recall better in the HMD condition as compared to the desktop condition, permitting us to reject the null hypothesis.

2.4.1 Task Performance

The overall average recall performance of participants in the HMD condition was 8.8% higher compared to the desktop condition with the mean recall accuracy percentage for HMD condition at 84.05% and the desktop condition at 75.24%. Using a paired t-test with Bonferroni-Holm correction, we calculated $p = 0.0017 < 0.05$ which shows that our result was statistically significant. In Figure 2.5 we present the overall performance of the users in the HMD condition as compared to the desktop condition.



Figure 2.5: The overall average recall performance of participants in the HMD condition was 8.8% higher compared to the desktop condition. The median recall accuracy percentage for HMD was 90.48% and for desktop display was 78.57%. The figure shows the first and third quartiles for each display modality.

2.4.2 Errors and Skips

The recall accuracy measures the number of correct answers. In addition, we kept track of when participants in our user studies made an error in recall (i.e. gave an incorrect answer) or skipped answering (i.e. did not provide an answer). We show the percentile distribution of the average number of erroneous answers per participant for each display modality in Figure 2.6. Participants in the HMD condition made on average fewer errors than those in the desktop condition. The total number of errors in the HMD condition for 40 people were 33 out of 840, and in the desktop condition was 56 also out of 840. In addition, the difference in the incorrect answers was statistically significant, shown using a paired t-test with Bonferroni-Holm correction resulting in $p = 0.0195 < 0.05$.

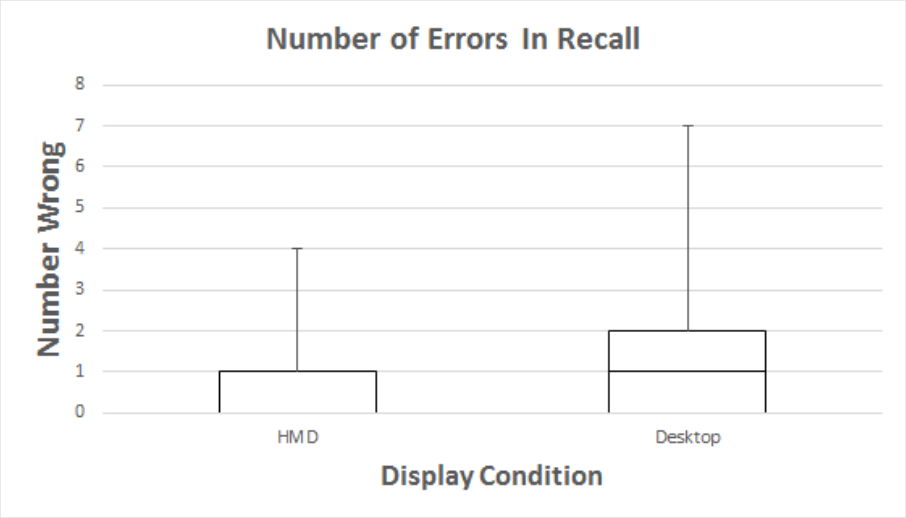


Figure 2.6: The distribution of incorrect answers for each display modality showing the median, first, and third quartiles.

In Figure 2.7, we showed that the number of faces for which participants

skipped an answer in the desktop condition was significantly higher than in the HMD condition. This was shown to be statistically significant using a paired t-test with Bonferroni-Holm correction with $p = 0.0062 < 0.05$, which reinforces that participants in the HMD had better recall than those on the desktop.

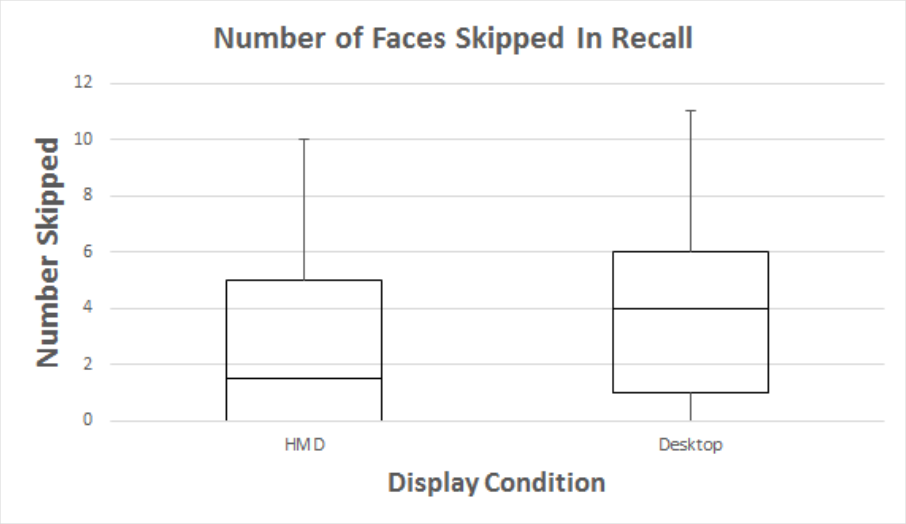


Figure 2.7: The distribution of faces skipped during recall for each display modality showing the median, first, and third quartiles.

2.4.3 Confidence

Previous work by Mania *et al.* [44, 58] examined user confidence with recall accuracy. This allows us to study not only the objective recall accuracy but also the subjective certainty of the user answers. We asked each participant to indicate their confidence on a scale of 1 to 10, with 10 being certain, as a measure of how certain they were in the correctness of their response, for each answer. The confidence scores aggregated across all the 40 participants and all the 42 faces that each studied are shown in Figure 2.8.

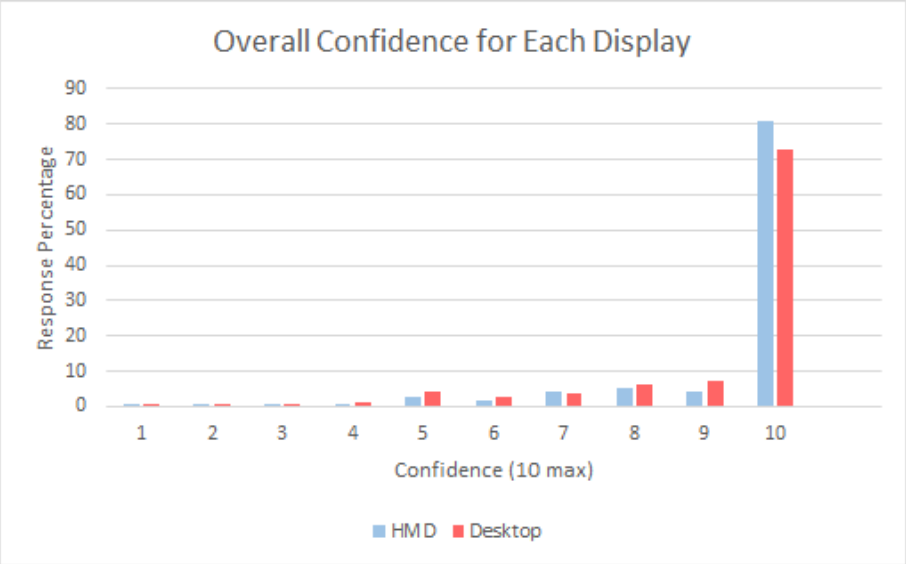


Figure 2.8: The overall confidence scores of participants in the HMD condition and the desktop condition. Each participant gave a confidence score between 1 and 10 for each face they recalled. Those in the HMD condition are slightly more confident about their answers than those in the desktop condition.

From Figure 2.8, we can see that users were slightly more confident in the HMD condition than on the desktop condition. The average confidence values for the HMD and desktop conditions were 9.4 and 9.1 respectively, ignoring skips. For the highest confidence, a confidence score equal to 10, there was a statistical difference between the number of correct answers given in the HMD and the desktop conditions, with $p = 0.009 < 0.05$ using a chi-square test, and with $p = 0.022 < 0.05$ including Yates community correction. However, confidence is not always an indication of correctness. We wanted to see if the HMD condition was giving a false sense of confidence. Figure 2.9 shows the number of errors given in each display based on the confidence of participant answers.

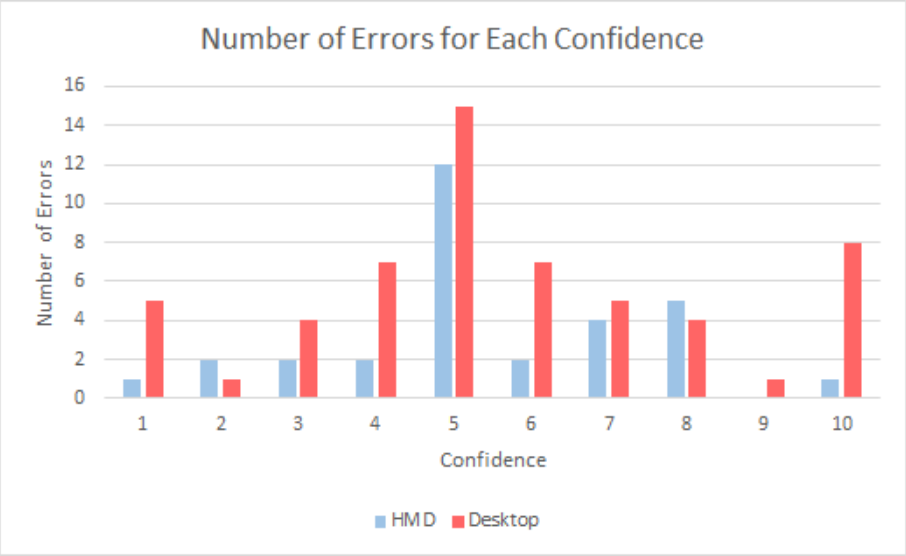


Figure 2.9: The number of errors made for each display condition for various confidence levels.

The results in Figure 2.9 show that when the users were less error-prone in the HMD condition, their confidence was better-grounded in the recall accuracy, than when in the desktop condition. In general, participants were more often correct in the HMD condition than for the desktop condition for a given confidence level.

2.4.4 Ordering Effect

In our study we alternated the order in which participants were exposed to the displays. Figure 2.10 shows the accuracy when using the desktop first followed by the HMD versus using the HMD first and then the desktop.

For both the desktop and HMD conditions, users started with roughly the same performance (accuracy) on both the desktop and HMD (desktop-1 and HMD-1 in Figure 2.10); but when going to the other display, the performance changed.

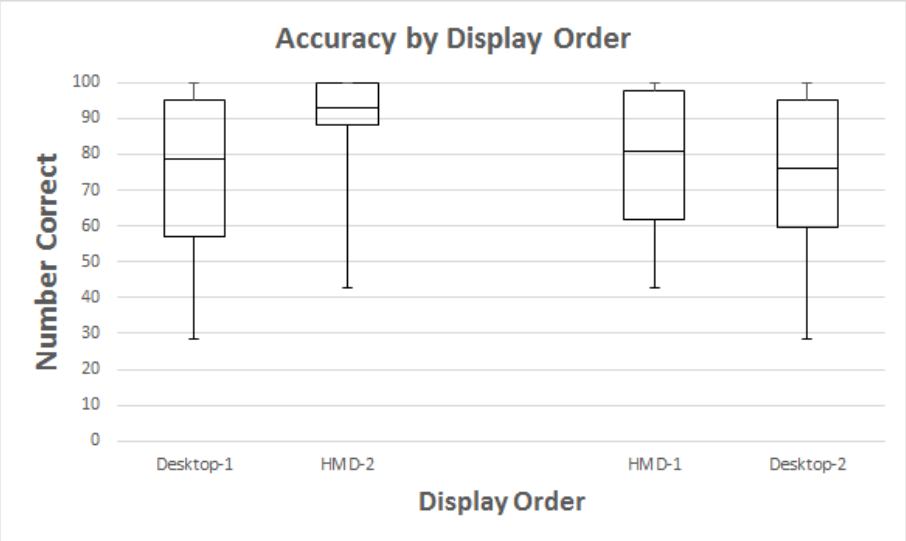


Figure 2.10: The performance of participants going from a desktop to a HMD and from a HMD to a desktop, showing the median, first and third quartiles.

When users went from a desktop to a HMD, their performance generally improved. However, when the users went from a HMD to a desktop, their performance surprisingly decreased. When comparing each participant's first trials, the desktop-1 and HMD-1, their distribution of recall scores were not significantly different with $p = 0.62 > 0.05$, but they were for the second trials, the HMD-2 and desktop-2, with $p = 0.025 < 0.05$.

2.5 Discussion

We next report some interesting observations based on a questionnaire the participants filled out after the study. All our participants were expert desktop users, but almost none had experienced a HMD before. We believe that if there were to be any implicit advantage it would lie with the desktop, given the overall familiarity

with it. Although we gave the participants enough time to get comfortable in the HMD before we began the study, we observed that many were not fully accustomed to the HMD, even though they performed better in it. We asked each participant which display they preferred for the given task of recall. We explicitly stated that their decision should not be based on the novelty or "coolness" of the display or the experience. All but two of the 40 participants stated they preferred the HMD for this task. They further stated that they felt more immersed in the scene and so were more focused on the task. In addition, a majority of the users (70%) reported that HMD afforded them a superior sense of the spatial awareness which they claimed was important to their success. Approximately a third mentioned that they actively used the virtual memory palace setup by associating the information relative to their own body. This ability to associate information with the spatial context around the body only adds to the benefit of increased immersion afforded by the HMD.

We note the interesting results we obtained with the display ordering. When starting with the desktop and then using the HMD we observed a significant improvement as compared to starting with the HMD and then using the desktop. A possible explanation for this could be that those who used the HMD first are able to benefit from the HMD's superior immersion, which they lose when they transfer to the desktop. However, when the users start on the desktop they invest a greater effort to memorize the information and therefore when they transfer to the HMD, they not only keep their dedication but also gain from the improved immersion.

2.5.1 Study Limitations

In general, it is a difficult design decision to balance the goals of experimental control and ecological validity. In our study, we placed the faces for a particular face set in the same locations for all participants. However, since the participants were free to look in any direction, the order of presentation of faces was self determined. We could have restricted the participants to look at the faces in a pre-determined order. However, we allowed the participants to look around freely, so that the results would achieve greater ecological validity. Randomization of faces could have led to unintended consequences; having the Dalai Lama's face next to Abraham Lincoln's in one instantiation could alter its memorability, as could the opportune positioning of the Dalai Lama on a roof-top background. To avoid such inter-object semantic saliency confounds, we decided to preserve the same ordering of faces for all participants that viewed the scene with a given set of faces. We recognize that not randomizing the stimuli in a within-subject design could introduce a bias. To make sure that this did not result in any significant effects, we carried out a four-way mixed ANOVA (reported at the beginning of Section 4) and we did not find any statistically significant effects on recall due to the scenes, face sets, or the ordering of the display conditions. Previous research, such as [59] points out the tradeoffs between experimental control and ecological validity for virtual environments. Parson *et al.* [60] persuasively argues for designing virtual environment studies that strike a balance between naturalistic observation and the need for exacting control over variables.

The modality of interactive exploration of the virtual environment in the two conditions was different (head tracking versus mouse tracking). Thus, differences in the recall performance may be explained by this diverse interaction modality. Our study did not attempt to distinguish the role of proprioceptive and vestibular information from visual stimuli, but examined them in the respective contexts of immersive HMD and desktop display conditions. It will be interesting to examine the relative advantage of the diverse interaction modalities with the same display modality, in future user studies.

2.5.2 Conclusions

We found that the use of virtual memory palaces in HMD condition improves recall accuracy when compared to using a traditional desktop condition. We had 40 participants memorize and recall faces on two display-interaction modalities for two virtual memory palaces, with two different sets of faces. The HMD condition was found to have 8.8% improvement in recall accuracy compared to the desktop condition, and this was found to be statistically significant. This suggests an exciting opportunity for the role of immersive virtual environments in assisting in recall. Given the results of our user study, we believe that virtual memory palaces offer us a fascinating insight into how we may be able to organize and structure large information spaces and navigate them in ways that assist in superior recall.

One of the strengths of virtual reality is the experience of *presence* through immersion that it provides [19, 61]. If memory recall could be enhanced through

immersively experiencing the environment in which the information was learned, it would suggest that virtual environments could serve as a valuable tool for various facets of retrospective cognizance, including retention and recall.

2.5.3 Future Work

Our study provides a tantalizing glimpse into what may lie ahead in virtual-environment-based tools to enhance human memory. The next steps will be to identify and characterize what elements of virtual memory palaces are most effective in eliciting a superior information recall. At present, we have only studied the effect of in-place stereoscopic immersion, in which the participants were allowed to freely rotate their viewpoint but not translate. It will be valuable to study how the addition of translation impacts information recall in a virtual memory palace.

Other directions of future studies could include elements in the architecture of the virtual memory palaces such as their design, the visual saliency of the structure of model [62], their type, and various kinds of layouts and distribution of content that could help with recall. Another interesting future work would be to allow people to build their own virtual memory palaces, manipulate and organize the content on their own, and then ask them to recall that information. If their active participation in the organization of the data in virtual memory palaces makes a meaningful difference, then that could be further useful in designing interaction-based virtual environments that could one day assist in far superior information management and recall tools than those currently available to us. Yet another

interesting future direction of research could be to compare elements of virtual memory palaces that are highly personal versus those that could be used by larger groups. Much as textbooks and videos are used today for knowledge dissemination, it could be possible for virtual memory palaces to be used one day for effective transfer of mnemonic devices amongst humans in virtual environments.

Chapter 3: Interactive Characterization of Cybersickness in Virtual Environments using EEG

3.1 Introduction

With the resurgence of virtual reality (VR), cybersickness has become a growing concern for researchers, developers, and users alike. Previous studies have shown that a large portion of the population ($40\% - 60\%$ according to a survey by Kolasinski [1]) may experience moderate to severe cybersickness in virtual environments. While there are several theories on reasons underlying cybersickness, there does not exist an easy or systematic method of measuring and quantifying cybersickness from one moment to another. Without the existence of a reliable tool to measure and interactively quantify cybersickness, understanding and mitigating it remains a challenge. Early work on studying cybersickness and motion-sickness relied on examining physiological changes such as sweating and increased heart rate. Eventually, this led to a standardized self-evaluation form for determining the intensity of sickness the person experienced, the Simulator Sickness Questionnaire [63]. A limitation of this approach is that measuring the effects of cybersickness requires either interrupting the subject during the experience (thereby affecting the experi-

ence itself and thus the results) or waiting until the end of the experience to assess their symptoms, which relies on the subject accurately recalling their sickness. This survey-based qualitative approach is unable to provide real-time quantitative measurements, making it difficult to objectively assess real-time cybersickness in the virtual environment.

In this chapter, we present the results of a user study that measures and examines cybersickness experienced by participants wearing a commercially available HMD and EEG headset. For this study, we designed a 3D environment and a camera path that was likely to evoke a moderate degree of cybersickness among participants. During this experience, the subjects' brain activity is measured using an EEG device and compared against a baseline EEG, when the scene is stationary. In addition, we also had participants continuously self-report their level of sickness with a joystick interface. We compared the self-reported data with their time-frequency spectral EEG information showing a correlation between the EEG data and the self-report data.

This chapter makes the following contributions to understanding and quantifying cybersickness in virtual environments:

- We establish that cybersickness in an immersive HMD is correlated with brainwave activity measured by EEG;
- We find a statistically-significant correlation of Delta, Theta, and Alpha-waves with self-reported cybersickness;
- Our approach facilitates ease of measurement and characterization of cyber-

sickness by using inexpensive, commodity off-the-shelf devices for VR headsets and EEG devices.

3.2 Related Work

LaViola [6] and Holmes *et al.* [7] found that common symptoms of cybersickness include nausea, increased heart rate, disorientation, sweating, eye strain, and headaches. One of the prevailing theories on the cause of cybersickness (also referred as simulator sickness or visual fatigue) is the sensory conflict theory, which attributes it to the dissonance between the visual and the vestibular sensory cues [1, 2]. This happens, for instance, when a user is immersed in a moving virtual environment, while stationary in the real world. The sensory conflict between what the eyes see and what the body feels is believed to lead to a physiological sense of discomfort and associated cybersickness. Cybersickness is closely related to motion sickness. Motion sickness is often induced by the unsettling movement, such as travel in vehicles or aircraft or amusement rides, but can also be caused with a mismatch of visual and vestibular sensation. Some of the techniques to mitigate cybersickness have therefore relied on minimizing this mismatch. A highly creative solution to resolving this mismatch was devised by Maeda *et al.* [64], who used galvanic vestibular stimulation to produce the sensation of vection or movement. Riecke *et al.* [65] reduced motion sickness by increasing a user's sense of self-motion without physically moving. This was elegantly accomplished through auditory cues, seat vibrations, and the introduction of subtle scratches in the periphery of the projection screen.

In contrast to the above, a highly innovative research direction has been in examining the role of peripheral vision in cybersickness. Rebenitsch and Owen [66] presented a thorough review of modern techniques to detect and measure cybersickness and urge for more research in the minimally understood subject. In their review, they state that the usage of EEG for such an endeavor is rare, noting only one related previous work. In a seminal study, Lin *et al.* [67] found that a user's visual field of view was positively correlated to their simulator sickness (SSQ) scores. More recently, Fernandes *et al.* [68] devised a clever solution to mitigating cybersickness by strategically and automatically manipulating the field of view of the wearer of a HMD based on virtual camera movement (full field of view when stationary and narrow field of view when in motion).

3.2.1 Self-reporting Cybersickness

The most common method for measuring cybersickness is to measure the severity of the users' symptoms using subjective self-reporting surveys [69]. A commonly-used survey is the Simulator Sickness Questionnaire (SSQ) by Kennedy *et al.* [63] which assesses sixteen symptoms, with each item rated on a scale of four (none, slight, moderate, and severe). These symptoms have been further grouped into three categories: oculomotor, disorientation, and nausea. Oculomotor symptoms include effects such as fatigue, eyestrain, and difficulty in focusing. Disorientation includes vertigo, dizziness, and blurred vision. Lastly, the nausea category includes symptoms such as sweating, burping, salivation, and nausea [6]. While the self-

reporting surveys are quite informative, they have the shortcoming that they can be administered only at the end of the simulator session or require the interruption of an experiment for a study participant to fill out the questionnaire. Waiting till the end loses the fine temporal granularity of cybersickness reporting. At the same time, interrupting the participant in a continuous experiment may be undesirable or even impossible. Further, an interruption may result in alteration of physiological symptoms in the study participant which may impact their reporting. For instance, the interruption could result in recovery from motion sickness due to the passage of time and lack of sickness-inducing stimuli. Therefore, passive, but continuous, approaches to measuring cybersickness are highly desirable.

Several biological metrics have been used to detect and measure the presence of motion sickness and cybersickness. These include heart rate, respiratory rate, finger-pulse volume, skin conductance, and gastric tachyarrhythmia [70]. A challenge with these metrics is that not all people suffer from these symptoms when experiencing cybersickness, and cybersickness is not the only cause of these symptoms. Other studies use a user-driven metric, where a participant uses a clicker or a joystick to continuously indicate when and how much cybersickness the participant is feeling at that moment [71].

3.2.2 Measuring Motion Sickness with EEG

EEG has been used to measure motion sickness [71–75]. Previous papers have focused on four frequency ranges, Delta ($1.0 - 4.0$ Hz), Theta ($4.0 - 7.0$ Hz), Alpha

(7.0−13.0 Hz) and Beta (13.0−25.0 Hz). Kim *et al.* [76] found that an increase in the delta power with a decrease in beta power was indicative of cybersickness during an object-finding VR experiment which used a rear-projected CAVE (cave automatic virtual environment) display system to show the visual stimuli. Another study by Min *et al.* [77] concluded that a decrease in delta power was indicative of a visually-induced motion sickness in a car-driving experiment which used a standard rear-projected display to show the visual stimuli. Chen *et al.* [71] built a driving simulator using a motion platform inside a 360° rear-projection display, in order to provide both visual and vestibular stimulation to induce motion sickness. In this study, each participant used a controller to continuously log their level of motion sickness. By using independent component analysis (ICA) with time-frequency analysis and cross-correlation analysis, the authors were able to examine the EEG changes in brain-wave activity when induced by both visual and vestibular stimuli. They found a more complex interaction of power increases and decreases in different regions of the brain as the level of motion sickness changed. Another set of studies by Naqvi *et al.* [78, 79] recorded the EEG signals of participants viewing a movie on a 3D LCD TV in either 3D or 2D in order to determine if 3D movies cause a greater visual fatigue. Their study found a decrease in theta-power for the frontal regions of the brain in addition to a decrease in beta power in the temporal region. To the best of our knowledge, there has not yet been a systematic study that has used EEG to measure and quantify cybersickness for users in immersive virtual environments wearing head-mounted displays. Given the previous work quantifying motion sickness using EEG, we also believe EEG is appropriate for quantifying

cybersickness.

3.3 Materials and Methods

Our study evaluates the EEG dynamics of cybersickness from binocular visual stimuli in a virtual reality head-mounted display. To the best of our knowledge, this is the first study that uses EEG signals to continuously evaluate cybersickness in participants wearing head-mounted displays.

We used a 14-channel, 128 Hz, Emotiv Epoc EEG device, which has been successfully validated [80] and used for EEG research for a variety of research studies, including measuring the cognitive load [81], examining the relationship between the environment and happiness [82], and as a proof of concept for robust and mobile EEG recording in the outdoors [83]. We used the HTC Vive head-mounted display, which has a 110° field of view and a resolution of 1080×1200 per eye with a refresh rate of 90 Hz, for both eyes. In this user study, the participants were limited to rotational viewing with no translational movement permitted. The participants viewed a 3D stereo rendered scene in the head-mounted display that involved a fly-through of a virtual spaceport with twisting, turning, accelerating, and decelerating of the virtual camera. A screenshot of the scene is shown in Figure 3.1. In addition to the HMD and EEG devices, we used a Thrustmaster joystick device, which the participants used to manually record their current level of cybersickness during the camera fly-through. The participants were instructed to indicate, by tilting the joystick in any direction, the magnitude of their sickness. They were told that no

tilt indicated that they felt no sickness and that full tilt indicated extreme sickness. We have examined the correlation between the sickness reported by the participants and their EEG brain-wave recordings.

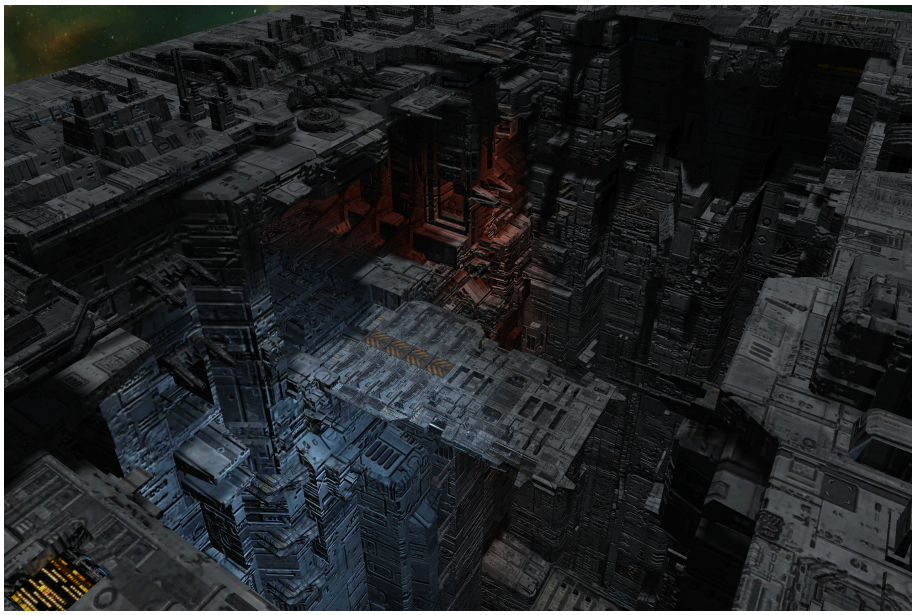


Figure 3.1: A still from the virtual spaceport flythrough used in our cybersickness study.

3.3.1 Participants

We recruited 44 participants from our university campus and surrounding community for the user study, of which 31 were male and 13 were female, with an average age of 27 and standard deviation of 8 years. Every participant had normal or corrected-to-normal vision (self-reported). The study session for each participant lasted around 30 minutes. Due to technical problems associated with the EEG recording interface, we had to discard one participant's data. We have

used the EEG data from the remaining 43 participants for our analysis. Stanney and Kennedy [84] note that 3040% of participants in flight-simulator studies do not experience simulator sickness.

3.3.2 Experimental Protocol

Each study participant was explained the entire procedure and how they would interact with the HMD and the input joystick mechanism. First, the EEG device was placed on the participant's head and manually configured until the EEG device showed that all electrodes had registered good contact with the head. Second, the participant donned the HMD and their interpupillary distance (IPD) was adjusted so that the participant could comfortably see the 3D stereo rendering in the HMD. Third, the participant was given the joystick in their hand and explained that they had to push the joystick based on how cybersick they felt. Finally, each participant was given 60 seconds to get used to and comfortable with the EEG, HMD, and the joystick. Throughout the entire study session, the participants were standing while wearing both the EEG and HMD as well as holding the joystick in their hands. After being acquainted, a baseline EEG reading was taken. Inside the HMD the location of the user was static, but they were allowed to rotate their view direction by turning their head. The participants were asked to make slow and deliberate head movements while wearing the HMD and EEG devices to minimize any risk of injury and electrode separation. After the baseline reading, the participant was re-instructed to use the joystick device to report their sickness levels and was then

virtually flown through the scene, which lasted approximately one minute. Our pre-study trials showed that if a participant was at all susceptible to cybersickness, they would most certainly feel sick in the one-minute virtual fly-through. Restricting the flythrough to one minute kept the exposure time minimal for participant safety but long enough to record a satisfactory amount of data. Throughout the flythrough, each participant continuously logged their self-reported level of cybersickness using the joystick, with no tilt corresponding to no reported sickness, and full tilt as severe sickness.

3.3.3 Signal Acquisition and Pre-Processing

As discussed earlier, we recorded the brain-wave activity using an Emotiv Epoc EEG with 14-channels sampling at 128 Hz. The name and locations of each of the nodes/channels The EEG headset uses a saline electrolyte solution on the contact heads. The raw data was acquired and saved to disk using the Emotiv Epoc C++ SDK which was integrated into our rendering program. This enabled the EEG recording and camera path to be synchronized for all participants. bf We used Matlab with the very commonly used EEGLAB for the EEG signal management and processing (https://sccn.ucsd.edu/eeglab/). The first part of signal processing involved importing each participant's raw Emotiv EEG data into Matlab and then into EEGlab for both the baseline and virtual flythrough recordings. Once the data is loaded, the mean power for each channel is calculated and subtracted from that channel's data, centering the signals. Next, a high-pass filter with a cut-off

frequency of 1 Hz was used in conjunction with a low-pass filter with a cut-off frequency of 50 Hz to remove any unwanted noise from the signals. Next, the filtered EEG signals are manually inspected for any recording anomalies, which can occur if a subject moves too abruptly or if an electrode temporarily loses contact. We manually remove those erroneous sections or reject the EEG sample entirely. In our study, we found EEG recording anomalies for one subject, and we removed its data from further processing. After pre-processing, we exported the data into an EEGlab study package.

While participants are viewing the fly-through of the space-port, they reported their level of cybersickness with the joystick, with more tilt indicating stronger sickness. The joystick was sampled at $90Hz$, the same as the frame-rate of the HMD. The cybersickness level is a score between zero and one, which is reported continuously in real-time without interrupting the experiment.

3.3.4 Independent Component Analysis

Similar to previous work on EEG-signal analysis [71,85], we decompose our filtered EEG signals, for each subject, into independent components, or brain sources, using Independent Component Analysis (ICA) [86] using EEGLab. The intuition behind the use of ICA is that the observed EEG signals are the result of a mixture of sources throughout the brain and scalp, which are assumed to be independent, such as eye-blinks, muscle movement, or other psycho-physiological stimuli, including cybersickness. In our study, we apply ICA to the EEG recordings for each

individual subject, resulting in 14 independent components per participant. From the calculated independent components, we cluster similar components using the built-in EEGLab K-means independent component clustering functionality. The idea is to cluster similar independent components so that similar underlying phenomena across participants are grouped together for further analysis and separates undesired phenomena from our target source. The resulting scalp maps from the clustered independent components are shown in Figure 3.2. From the 14 generated clusters, we found one to be most representative of cybersickness as it had the most statistically significant difference between the baseline EEG frequencies and the virtual flythrough EEG frequencies. These generated clusters, along with the one selected cluster, are shown in Figure 3.2. The selected cluster, which to EEGLab is cluster 12, from this point forward to refer to it as cluster A. Figure 3.3 shows the selected cluster with the EEG node labels in more detail. The Emotiv Epoc uses 14 electrodes, AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, and AF4.

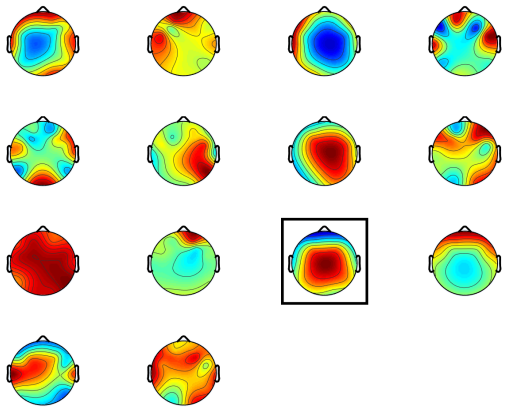


Figure 3.2: Averaged scalp maps of clustered independent components. The scalp map which correlated with cybersickness is shown in the black box.

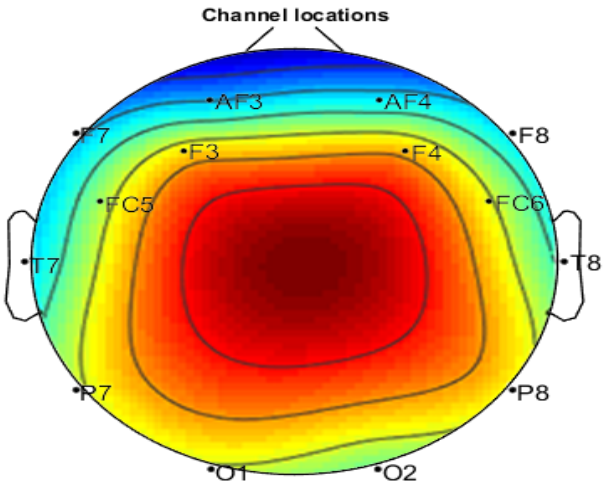
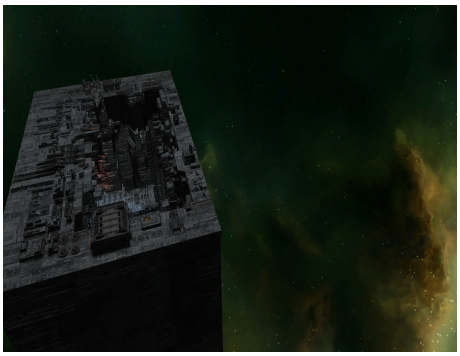


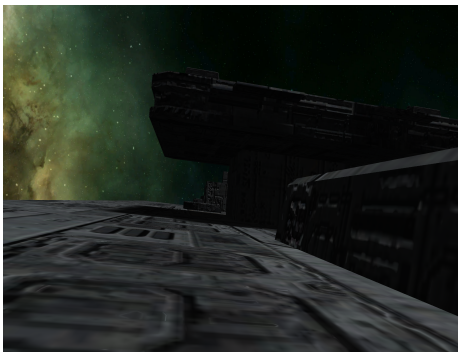
Figure 3.3: The names and locations of the 14 EEG electrodes in the Emotiv Epoc headset.

3.3.5 Time-Frequency Analysis

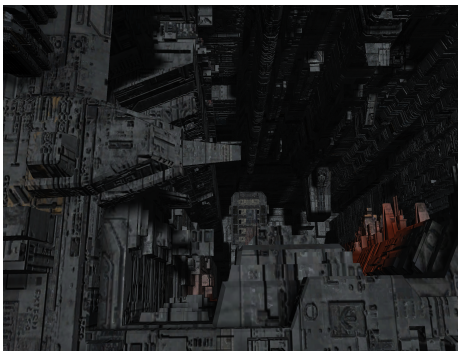
During the study, participants continuously logged their current feeling of cybersickness using a joystick while virtually flying through of the spaceport. We correlated the participants' self-reported cybersickness levels with the ICA cluster power spectra. As the reported level of cybersickness changes, we hypothesized that the ICA power should also change at different frequencies relative to the strength or weakness of reported cybersickness. To calculate the time-frequency spectra, we used the EEGlab ERSPs (Event Related Spectral Perturbation) function, resulting in an average of the power-spectrum density over time. The power-spectrum density is then converted into decibel power by the EEGLab.



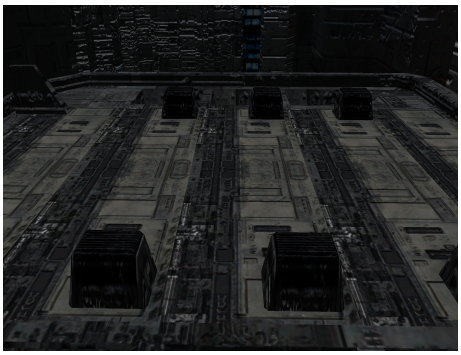
(a) The start of the fly through with the camera slowly moving towards the spaceport



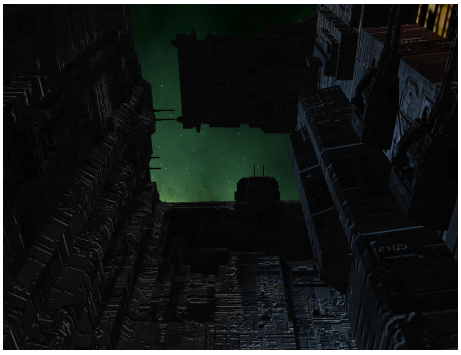
(b) The camera does a fast acceleration close to the surface of the spaceport at the 20-second mark.



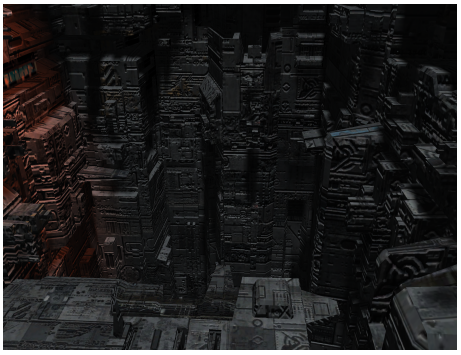
(c) A sudden and fast drop off the edge of the spaceport, around the 35-second mark.



(d) The free-fall suddenly decelerates near one of the landing platform, around the 43-second mark.



(e) The camera starts to accelerate directly upwards. This moment occurs at the 50-second mark.



(f) Finally, at the 60-second mark, the fly-through arrives at a ledge and shows the large depth of the spaceport.

Figure 3.4: The virtual camera flythrough of the spaceport that each participant in our study experienced. Note how the above correspond to the self-reported cybersickness levels in Figure 3.5.

3.4 Results

In this section, we review the results of our user study exploring cybersickness in virtual reality using EEG. First, we review the subjective sickness levels and symptoms as reported by each participant during and after the experiment. Second, we examine the results of the EEG analysis, showing a statistically significant difference between the averaged baseline and cybersickness EEG recordings. Third, we review the time-frequency spectral power graphs and compare them to the continuous self-reported sickness levels and show that there is a correlation between them.

3.4.1 Self-Reported Cybersickness

After the baseline EEG recording was taken, the EEG measurements during the virtual flythrough phase began. Each study participant was told that they would virtually fly-through the spaceport and if they felt any of the previously mentioned symptoms that they were to indicate their presence and strength by tilting the hand-held joystick device. We refer to this input as the participants' self-reported cybersickness levels, which are shown in Figure 3.5. In addition to the joystick information, each participant completed an SSQ form at the end of the study.

The highest peaks of the average of participants' self-reported cybersickness levels, shown in the bold black curve of Figure 3.5, can be attributed to specific events that occurred in the spaceport fly through (see Figure 3.4). The first peak corresponds to a sudden burst in camera acceleration in close proximity to the

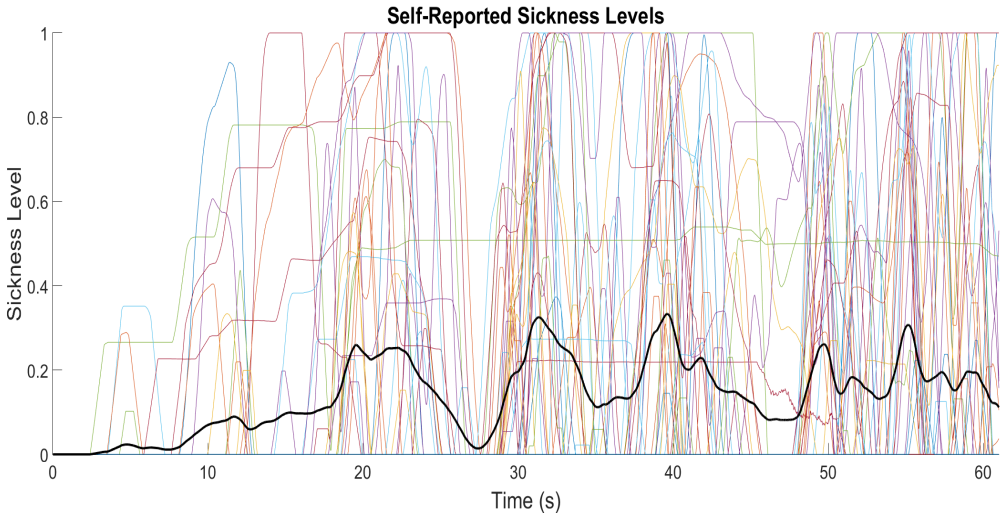


Figure 3.5: The self-reported cybersickness levels, using joystick, for each participant are shown in the thin colored curves. The bold black curve shows the average of all the participants' self-reported cybersickness levels.

surface of the spaceport. The second peak corresponds to a sudden free-fall off an edge of the spaceport. The third peak aligns with the sudden and hard pull-up of the camera after free-falling from the previous event. The fourth peak corresponds to the sudden acceleration upwards after the initial camera pull-up. The final peaks correspond to a sudden deceleration of the camera as it comes to rest on the landing platform of the spaceport.

In addition to the continuous self-reported information from the joystick, the Simulator Sickness Questionnaire (SSQ) scores from each participant were collected at the end of the session. The SSQ consists of 16 questions with 4 severity options, with values between 0 to 3, with 0 as none, 1 as slight, 2 as moderate, and 3 as Extreme. The average SSQ scores for each of the 16 symptoms and their variances are shown in Figure 3.6. Based on the information from the graph it can be concluded

that our participants primarily experienced varying levels of vertigo, dizziness, and general discomfort.

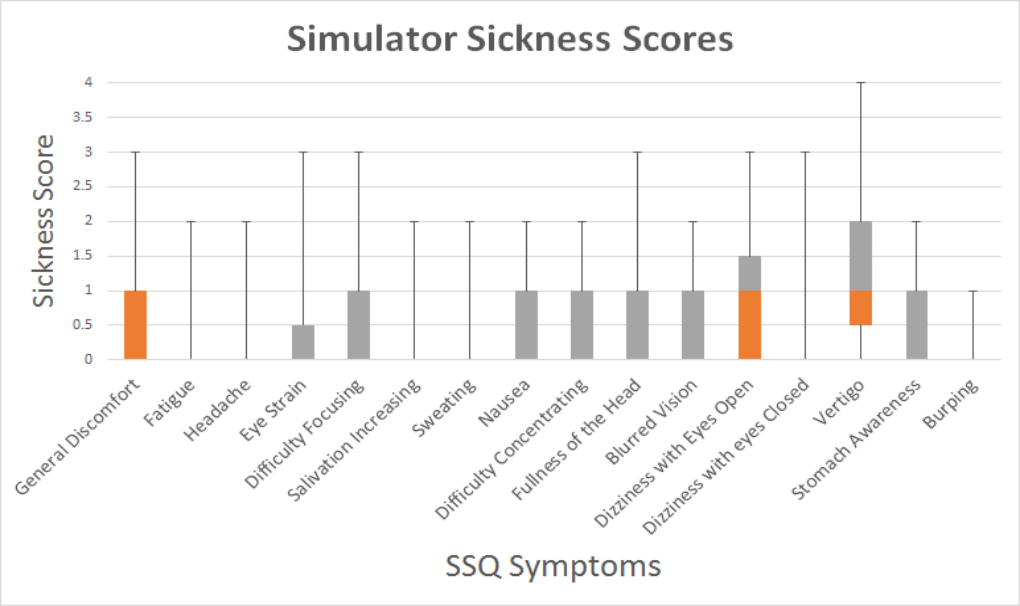


Figure 3.6: Participant Simulator Sickness Questionnaire (SSQ) scores after the experiment are shown here. The plot shows the median, first and third quartiles (orange and grey respectively), with the minimum and maximum shown as error bars.

Our study participants self-reported cybersickness through either the joystick or the SSQ survey. From the distribution of SSQ and joystick scores, we can see that the participants rated their level of cybersickness from mild to severe. This wide range of symptom intensities suggests that the brain-wave EEG data should be diverse and that not all users will be a part of the cybersickness revealing independent clusters.

In Figure 3.7 we show a comparison of the average level of self-reported cybersickness as reported through the joystick with the sum of reported SSQ sickness level

for each participant. The two distributions shown are correlated with a statistically significant score of 0.49 using a Pearson Correlation.

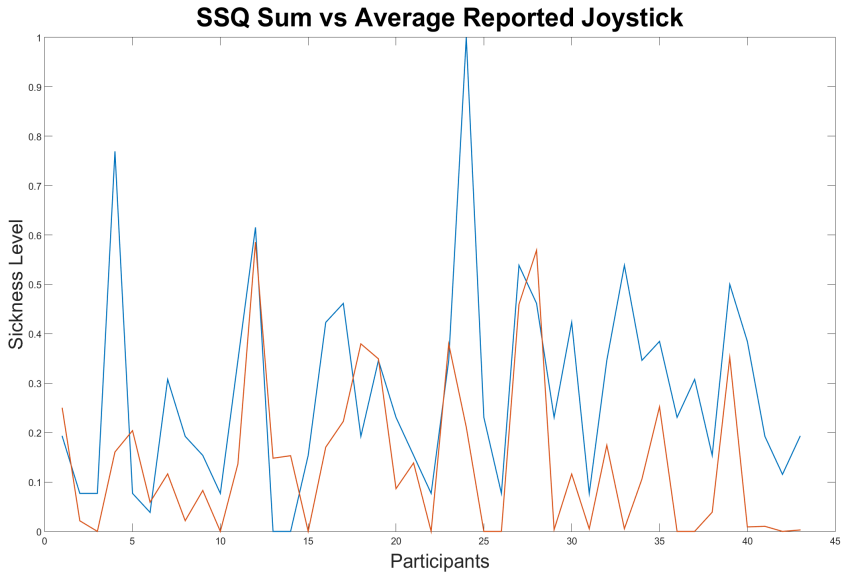


Figure 3.7: A comparison of the average score as reported by the joystick with the SSQ sum for each participant. The SSQ score and the self-reported cybersickness using the joystick have a Pearson Correlation r-value of 0.49.

3.4.2 Spectral Differences

In this section, we compare the differences between the spectral frequencies of the EEG recordings of the baseline (green curve) and the virtual flythrough (purple curve). From the frequency spectra of the 14 clusters shown in Figure 3.2, we have closely analyzed the one labeled as the 12^{th} cluster, hereafter referred as cluster A. The selected cluster had statistically significant differences between the baseline and the virtual flythrough EEG frequency spectra. Further, it represented a meaningful

fraction of the participants, composed of 24 out of the 43 total (55%) participants. We find it interesting that this is in general agreement with previous research by Stanney and Kennedy [84] in which they noted that 3040% of the participants in flight-simulator studies do not experience simulator sickness.

Figure 3.8 shows the mean component power spectra of the selected independent component cluster A for the baseline and virtual flythrough conditions. It is clear from the figure that there is a power increase across many frequencies for participants experiencing the virtual flythrough of the spaceport. In the component cluster spectra plot, we indicate where the EEG power changed significantly using paired t-tests. For the selected ICA cluster, we see that the difference between the baseline and virtual flythrough frequency spectra are statistically significant with ($p \leq 0.01$ for much of the frequency range), using EEGLab's built-in paired t-test with Bonferroni-correction statistical analysis. Similar to previous work which studied motion sickness, we also see a power increase across many frequency bands for the virtual flythrough scenario compared to the baseline.

The EEG spectral power differences, between the baseline EEG recording with the stationary scene and during the virtual flythrough, indicate that cybersickness can be detected using EEG. To be more specific, we have identified that an increase in spectral power, with respect to a baseline recording, is indicative of the onset of cybersickness. For both recording sessions, the participants used the EEG, the HMD, the joystick, and experienced the same environment while standing. The only difference was the camera motion during the virtual fly-through. We next look at the frequency spectra over time for the selected cluster, to examine when specifically

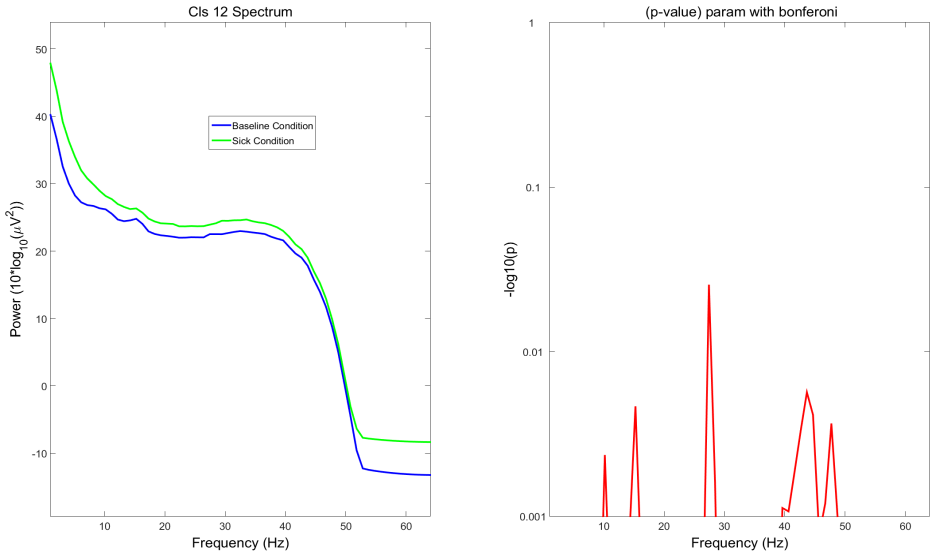


Figure 3.8: Comparison of the EEG power spectra between the baseline (blue) and virtual flythrough (green) for ICA cluster A. The paired t-test with Bonferroni-correction between the two spectra reveal $p < 0.001$ for much of the frequency ranges.

a participant experienced cybersickness and correlate them with their self-reported cybersickness levels.

3.4.3 Time-Frequency with User input signals

During the virtual flythrough of the spaceport, the study participants continuously recorded their current levels of cybersickness through a joystick device. The self-reported cybersickness levels for each participant are shown in Figure 3.5 along with the average sickness level shown in black. We used time-frequency analysis to evaluate the EEG spectra changes across all participants with the self-reported

sickness levels. The values for all frequencies, averaged over all users, for cluster A is shown in Figure 3.9.

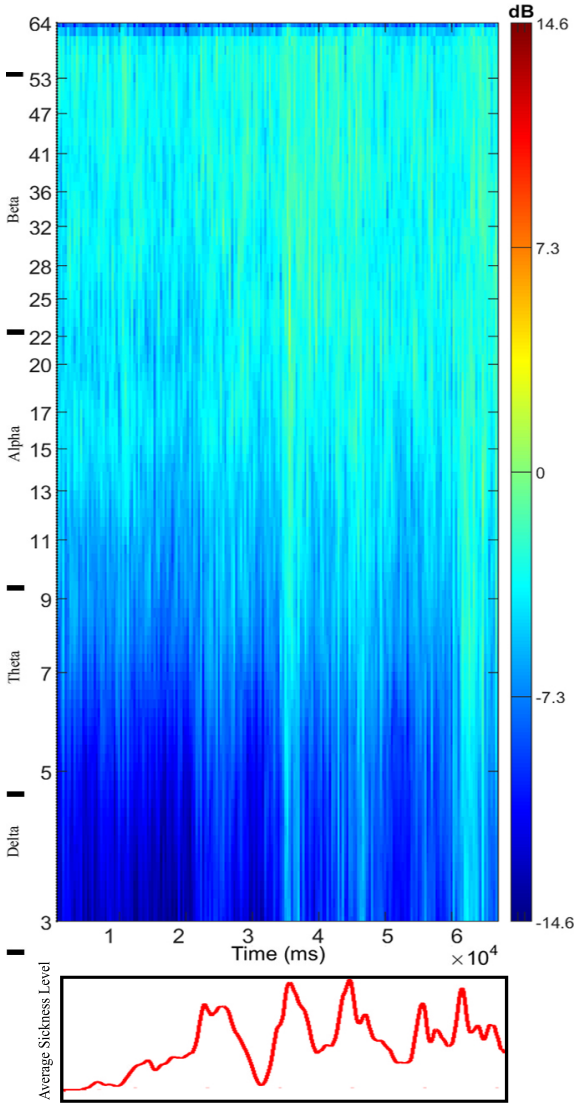


Figure 3.9: Time-Frequency visualization of cluster A. The average self-reported cybersickness levels are shown below in red.

The average self-reported cybersickness is shown below the time-frequency visualizations in red. We observe a correlation between the spectral power changes shown in the time-frequency plots, especially for the lower frequency bands, and the

self-reported cybersickness levels from the participants. To assess the degree of correlation, we computed the correlation between the time-frequency band values and the average self-reported cybersickness information. The Pearson correlation r-value scores for each of the four frequency bands are presented in Table 3.1. Figure 3.10 compares each of the frequency bands with the average self-reported cybersickness levels over time.

Table 3.1: Correlations (Pearson R-values) between average ERSP values for the four frequency bands and the self-reported cybersickness levels. All the correlations are statistically significant ($p < 0.001$). The graphs of the various frequency bands for clusters A can be seen in Figure 3.10.

Frequency Band	Cluster A
Delta Band $(1.0 - 4.0$ Hz$)$	0.642
Theta Band $(4.0 - 7.0$ Hz$)$	0.589
Alpha Band $(7.0 - 13.0$ Hz$)$	0.476
Beta Band $(13.0 - 25.0$ Hz$)$	0.465

Our analysis shows that a statistically significant and high correlation exists for Delta, Theta, and Alpha bands for cluster A with the self-reported cybersickness information from the participants. This is perhaps best illustrated in the Time-Frequency plot with the self-reported cybersickness through joystick input for one of the participants, as shown in Figure 3.11.

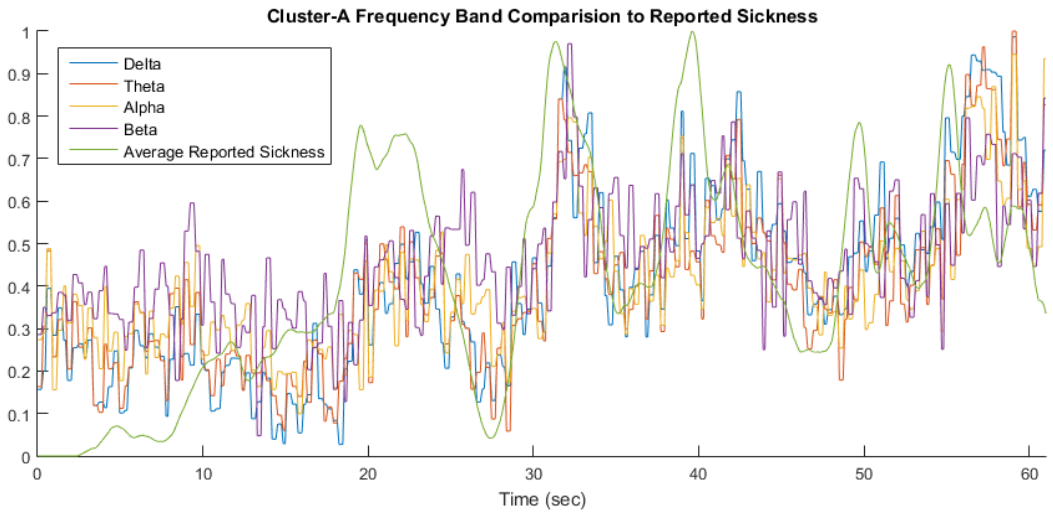


Figure 3.10: Average over four frequency Bands for Cluster A compared with the average self-reported cybersickness (in green).

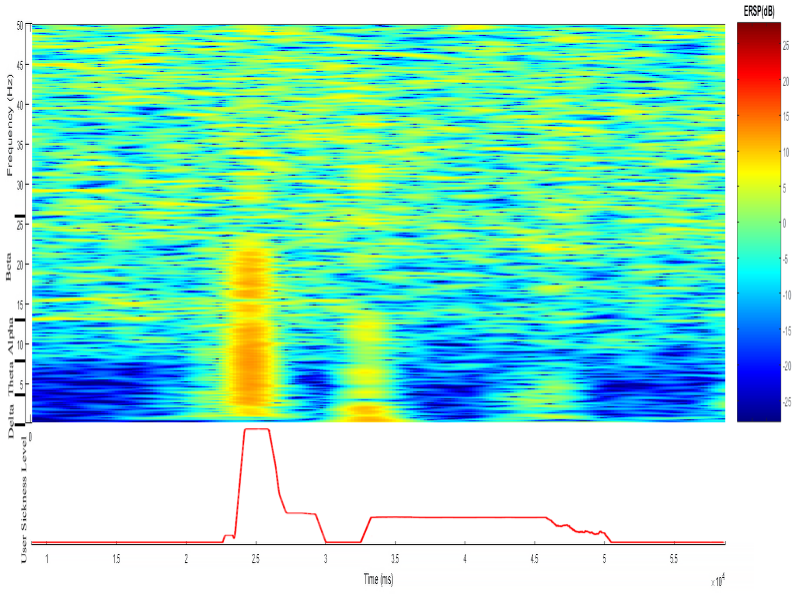


Figure 3.11: Visualization of the ERSP from a cluster A participant with self-reported cybersickness levels. Note how the changes in ERSP values, especially for Delta and Theta bands, align with the participant's self-reported cybersickness.

3.4.4 External Factors

We note that the high-correlations between the averaged time-frequency signals of the different EEG bands and the average joystick signal may be due to a confounding effect of increased cybersickness and the actual movement of the joystick. However, the joysticks movements were very sparse and were not sustained over long periods of time. The participants were instructed to tilt the joystick only when they felt their level of cybersickness changed. It has been shown in previous studies [71, 87, 88] that changes in spectral power as a result of finger and hand movements for sustained attention tasks diminished quickly, within the order of a few seconds. The effect would also result in a spectral change and rebound effect within that short period of time, which we do not see in our EEG signals. In addition, during the baseline recording, our participants held and moved the joystick (randomly) to simulate the same effect and these do not appear in the EEG signals either. Therefore, any changes of the joystick would not have influenced the overall spectral frequency and power differences of our analysis.

Another external factor for consideration is head movement. Participants were instructed to freely look around the same environment that they would be placed in during the fly-through (the spaceport) in the baseline recording session. During the fly-through, the participants could also freely look around their environment as the camera flew through the scene. Therefore, any significant spectral differences between the baseline and sick condition are unlikely to be due to head movement.

3.5 Conclusions and Future Work

Throughout the course of the study, we witnessed a wide range of reactions to the rendered stimuli. Some participants experienced minor discomfort, while others experienced moderate to high levels of cybersickness. Each participant was asked to briefly report what aspect of the experience made them the most cybersick. They reported that the sudden changes in direction and velocity of movement made them feel ill compared to when the motion was more smooth. In addition, they reported that the anticipation of where the camera was going to move heightened their reaction. Lastly, they expressed that if they were in control of the camera, as opposed to the camera being automatically moved, they might have felt less sick due to prior knowledge and mental preparation of what was about to happen. One observation that the test administrator made was that approximately 70% of participants would lean their bodies, with varying (in some cases, almost alarming) degrees of tilt, based on the motion of the camera. Approximately 32% of participants had previous experience with a head-mounted display.

In this chapter, we have presented our findings of a user study with the goal of continuously measuring and quantifying cybersickness. In our study, the participants wore both an HMD and an EEG recording device, while being presented visual stimuli of a virtual flythrough in a spaceport. The recorded EEG data was decomposed using ICA to separate the underlying sources of the brainwave activity and eliminate noise. The independent components were then clustered across users for the purposes of comparing the EEG of those grouped users. Through independent

component analysis and time-frequency spectral analysis, our findings suggest that a spectral power increase in the Delta, Theta, and Alpha frequency bands, relative to a baseline, strongly correlates to the presence of cybersickness.

Our findings in this chapter are just a first step to the many opportunities that present themselves in using EEG to study cybersickness in virtual environments. Some of the more important amongst these include a better understanding of the sources of cybersickness, the relationship of the duration of immersion to cybersickness, and the effect of age and gender on cybersickness. A number of cybersickness mitigation strategies have been studied over the last decade, but their evaluation has been largely based on questionnaires at the end of the immersive experience. An exciting direction of future work is in continuous evaluation of the effectiveness of cybersickness mitigation strategies, while the user is immersed in the virtual world. In our study, the participants were not asked to perform a task. It would be interesting to explore what effect if any, task performance has on cybersickness. Finally, it will be highly desirable, if at all possible, to move towards standards of assessing cybersickness and to use them to rate hardware (headsets, trackers, and displays) as well as the content (games, performances, and other immersive experiences).

Chapter 4: Enhancing Deep Learning with Visual Interactions

4.1 Introduction

Computer-based semantic learning systems have made impressive strides in the last few years, but there remains a striking disparity between the abilities of humans and machines. Current-generation deep-learning systems require thousands of finely-labeled images to train. If a child needed a thick stack of images of cats to learn what a cat looks like, we would be in deep trouble [89]. The goal of this chapter is to take the first steps to bridge this gap, by using visual interactions to help enhance the performance of deep learning. Furthering the capabilities of deep learning through interactions can help it emerge as a powerful engine for new discoveries in high-dimensional data.

A challenge with deep learning, as previously mentioned, is that it requires large amounts of precisely annotated and labeled data for training purposes. Recent advances in data collection and annotation have allowed for the creation of large high-dimensional datasets, consisting of millions of points, which are often used to train deep-learning models. However, in many real-world datasets, the labels and annotations provided may be incomplete or not capture all the distinctions within the data, known and unknown. Precisely annotating a high-dimensional dataset

that contains many labels is expensive, time consuming, and error-prone, caused possibly by mislabeling data-points, the unintended omission of precise labeling, or the subjectivity of the annotator(s).

Semi-supervised learning aims to improve classification performance by using labeled and unlabeled examples. In this work we extend semi-supervised-learning to include coarse-labeling that a user can refine based on visual feedback. Since it is often easier and faster to carry out coarse labeling, we expect our approach to be broadly applicable to many more datasets where coming across large amounts of precisely labeled data is difficult.

In this chapter, we present our approach to facilitate visually-driven deep learning that enables the refinement of a coarsely-labeled dataset through intuitive interactions by leveraging the latent structures present in high-dimensional datasets. Through the combined efforts of human analysts and deep learning, we hope to not only facilitate discovery of hidden communities and structures within these high-dimensional datasets, but also start to bridge the gap in our understanding of deep learning.

This chapter makes the following contributions to interaction-assisted deep-learning for high-dimensional semantic labeling:

- We use permutation-invariant deep neural networks to generate 2D point distributions such that similar points are proximal.
- We iteratively and manually refine the distribution and labeling of data points using visual feedback, which are in-turn used to refine the deep neural net-

works.

- We facilitate the discovery of detailed latent groups or labels from datasets that only consist of a few high-level or no labels.

4.2 Related Work

A new focus on bringing human judgment and intuition back into the data exploration process has recently gained popularity. Individually, machines and people are generally good at solving different problems, but the union of their strengths will lead to new and better insights. Our system builds upon the ideas by Turkay *et al.* [90], who have shown that placing the human into the data analytics process, particularly for high-dimensional data, is beneficial especially when the temporal, perceptual, and cognitive abilities of the user can be leveraged. Recent results also show the benefits of increased immersion the on the recall of spatially organized data [91].

We drew inspiration from the work by Endert *et al.* [92], whose insight was to use a model that is continuously updated with information from user interactions, and of Sacha *et al.* [93], who have shown that although there exist many tools and techniques for dimensionality reduction, there is a lack of general-purpose tools for human-in-the-loop interactive dimensionality reduction and exploration.

We next present a review of related research, touching on dimensionality reduction, high-dimensional visualization, interactive analysis of high-dimensional data, label generation, active learning, along with comparisons between traditional semi-

supervised learning and our goal, label discovery and refinement.

4.2.1 Dimensionality Reduction

The idea behind dimensionality reduction is to capture the dominant trends in the data and project them to a lower dimension. We use dimensionality-reduction techniques to produce a human comprehensible representation of an input dataset to allow analysts to visually detect clusters and refine the data labeling. Since the previous work on dimensionality reduction is extensive, we only refer to a small representative set here.

One set of dimensionality reduction techniques is based on Taylor-series expansions. Roweis and Saul [94] introduced and compared locally linear embedding (LLE) projections of data to those produced by principal component analysis (PCA) and multi-dimensional scaling (MDS) [95]. Belkin and Niyogi [96] introduced Laplacian eigenmaps that generalize the idea of LLE by computing a low-dimensional manifold representation of a high-dimensional data set such that local neighborhood information is preserved in a least-squares sense. The dimensionality reduction by this algorithm produces an approximation that reflects the intrinsic geometric structure of the high-dimensional dataset. The idea of spectral dimensionality reduction has also been used in visual depiction of graph relationships [97].

A popular approach, introduced by Maaten and Hinton [98], is a technique called t-Distributed Stochastic Neighbor Embedding (t-SNE), which is often used to generate visualizations of high-dimensional data in two dimensions. This is done

using a modified algorithm based on Stochastic Neighbor Embedding by Taylor *et al.* [99]. Their approach generates a view that reveals the structure of low-dimensional manifolds within the high-dimensional datasets and visualizes those manifolds using a collection of two-dimensional scatterplots. The algorithm is non-linear, performing different transformations on different regions of the feature space, looking for similarities in these smaller regions. The need to search through and check many different configurations of hyper-parameters just to generate the visualization, such as step and perplexity, can lead to a lot of time being spent simply searching for meaningful and revealing information, while also potentially leading to incorrect interpretations of the underlying data. According to Wattenberg *et al.* [100], t-SNE does not always produce meaningful representations. For our work, it is crucial that the generated visualizations facilitate intuitive and accurate interpretations of the data.

We next review deep-learning-based dimensionality-reduction techniques. Hinton *et al.* [101] popularized the idea of using a multilayer neural network to perform dimensionality reduction. Hadsell *et al.* [102] developed a convolution-based encoder using deep learning to perform dimensionality reduction to create a globally coherent nonlinear function that maps the data evenly on an output space. The advantage of these techniques is that the learning relies only on point-neighborhood relationships and does not require any distance measurements in the high-dimensional input space. More recently, Chen *et al.* [103] developed a system using deep learning Autoencoders to perform dimensionality reduction on hyper-spectral data. In their approach they combine neighboring pixels to generate spatial feature sets which

produce output class labeling after autoencoding and logistic regression. We use a modified version of Hadsell *et al.* [102] method to enable handling of coarsely labeled data.

Traditional dimensionality reduction and machine learning approaches, such as the ones reviewed above, require fully and accurately labeled data. Further, in contrast to our approach, they are single-shot techniques, with no mechanism to leverage the analyst's domain knowledge or pattern recognition ability to guide, shape, or influence the clustering of the data.

4.2.2 High-Dimensional Community Visualization

Visualizing high-dimensional data is a core element of our system since our goal is to reveal hidden patterns and communities within. In this section we review a selection of previous recent work on high-dimensional data visualization. A common technique for high-dimensional visualization is multidimensional scaling (MDS), pioneered by Kruskal [95]. The goal of MDS is to visually group data objects such that similar objects are spatially close to each other and dissimilar data objects are far away, as determined by a similarity function. The idea of plotting similar objects in close spatial proximity is fundamental to our own work.

One example of using spatial proximity for conveying information is the work by Amir *et al.* [104] who have created a 2D visualization tool for the analysis of high-dimensional biological-cell data. Each individual cell is represented as a point in a scatter plot, with the positioning of each point calculated using the t-SNE [98]

algorithm.

One of the most common and easiest-to-interpret visualization techniques is the scatterplot. Dang and Wilkinson [105] have improved the classical scatter-plots with scagnostics (scatter-plot diagnostics). One of the main problems in using scatter-plots is that as the dimensionality of the dataset increases, the number of generated scatter-plots also increases rapidly. To get around this problem, Dang and Wilkinson developed techniques to enable discovery of hidden structures within a subset of scatter-plots through various transformations.

The work we present here takes inspiration from the previous visualization works mentioned. Our goal is to build upon the foundations of that work and expand the capability of visualization-based investigation by incorporating and interleaving an intelligent system (deep learning) into the exploration process, rather than make an advance in the field of visualization itself.

4.2.3 Interactive Analysis of High-Dimensional Data

The ability to directly interact with high-dimensional data in order to search for trends or other interesting information is crucial. For our system, we expect a human analyst to understand, interact, and interpret results based on the output of a dimensionality reduction algorithm. In this section we review some of the previous works whose goal was to provide a method to interact in an iterative manner with high-dimensional datasets.

Ip *et al.* [106] developed a tool that facilitated the visual exploration of hid-

den spatial structures in volumetric datasets. The tool used a 2D intensity-gradient histogram to enable a user to iteratively search for interesting regions in the 3D volume. The interaction involved manipulating and selecting regions of the generated histogram through normalized cuts. We were inspired by the work done by Ip *et al.* and have also included the usage of normalized cuts to aid in our own segmentation and clustering process.

Chen *et al.* [107,108] developed a system which leverages deep learning to guide user exploration of interesting high-dimensional and temporally changing structures within volumetric cell data. Rather than using the raw features provided by the data, deep learning is able to transform those features into compact and semantically meaningful representations, which better capture and distinguish biological properties, such as boundaries and other components. Using this new feature space, a quantifiable metric of similarity between these deep-learning constructed features may be calculated, simplifying the transfer function used for volume rendering, giving rise to the interesting structures within the original biological data.

Liu *et al.* [109] developed an application that allows for the interaction and exploration of high-dimensional data presented in low-dimensional space. The primary contribution in their work is the use of distortion-guided manipulations, where a user can select a data-element and then move or delete it, causing point-wise distortion measures to be re-calculated and visualized. As a data-element is moved in the 2D space, a global structural change occurs on the fly, which provides information regarding the relationship between different parts of the 2D projection. Liu *et al.* [110] further expanded upon the previous work with the inclusion of a subspace

view navigation graph. This graph allows for animated transitions between different subspace views to facilitate easy comparison between those views. Our work shares a similar goal to Liu *et al.*, exploring high-dimensional data, but their tool is designed for understanding how different variables are related in low-dimensional space, by changing the configurations of the projection, rather than finding new subsets/categories in low-dimensional space. In addition, there is no mechanism in place for influencing the projections based on other input data, such as a newly discovered group or label information.

4.2.4 Label Generation

For many applications, it is useful to be able to generate labels for various content. For example, these labels could be hashtags for Twitter [111], keywords for image or scene search [112] or for music labeling [113]. Traditional labeling classification consists of learning from a dataset where each data example is associated with a known single label. For example, if there are only two distinct labels in the dataset, the problem is known as binary classification [114]. In multi-label classification each data example is associated with more than a single label. A common application for multi-label classification is image labeling, online search, and machine learning [115] [116]. However, the problem this chapter tackles is often considered a challenge that many multi-label classification approaches suffer from, which is the lack of large, precisely labeled datasets to train on. Rather, many datasets have missing, or over simplified labels. This is the challenge we tackle in

this chapter, identifying, differentiating, and re-labeling hidden labels and groups within high-dimensional data.

One way to get around this problem is to pre-train the classification network (in the case of deep learning) in an unsupervised manner [117], without the usage of labels. Erhan *et al.* [117] have shown that pre-training the classification network allows the layers to focus on and capture the variation and nuances of the data itself, which allows for better regularization and generalization of the network. An example of this is the work done by Hinton and Salakhutdinov [101], who use an autoencoder to pre-train a deep learning network to learn a low-dimensional embedding of high-dimensional data. The primary purpose of the previous work is to increase the performance of trained models by seeding the weights in the network through pre-training. Instead of modifying the methods behind training networks, we are improving model performance by modifying the training data itself to be more precise and accurate, resulting in a better model. The benefit of our approach is that the result of our technique, an improved training set, can be leveraged by all forms of machine learning which use training data, not just neural networks, as well as other applications where labeled datasets are required.

Lee [118] developed a technique that uses labeled and unlabeled data to train a deep neural network in a semi-supervised fashion. For the data with unknown labels, a pseudo-label is assigned with the maximum predicted probability after every update cycle, which is then used as if it were the true label. Testing their approach using the MNIST dataset, they were able to develop a deep neural network capable of state-of-the-art classification performance. Similar to the previous work,

our system is designed to help discover the true label of the data-point based on the given labeling present in the dataset. As the data points are relabeled, they are placed back into the dataset and are used for re-training the network. However, our technique benefits from the ability of an analyst to use their judgment and domain knowledge to help drive the re-labeling of data, while also benefiting from the suggestions and output of a model. If the model should make a mistake in labeling, a human analyst can step into the loop and correct the mistake, before it propagates and influences the future labeling of data. Lastly, the previously mentioned approach can only assign data points with a label that already exists in the dataset, and cannot generate new labels or groups, which may be necessary if a large enough portion of the data is without a label.

There has not been much work on including human-in-the-loop interactions to enhance labeling accuracy. Much of the prior work relies on statistical models that are limited by the domain knowledge of the dataset or expert availability. One example of rectifying crowd sourced labeling with expert review is the work by Hung *et al.* [119]. In their work, they use statistical methods to generate a few meaningful questions about the data, previously labeled by crowd-sourcing, to minimize the amount of expert time needed to correct that labeling. The output of the crowd sourcing and expert refinement of the labeled data may then be used for machine learning. The authors found that they were able to reduce the amount of time needed from domain experts to refine and correct the labeling to achieve near perfect classification. While the impact and usefulness of the previous work cannot be understated, the weakness of the previous method is that there remains a large

reliance of crowd sourced information to provide the majority of labels and handle the burden of labeling the data. This may not always be possible, as the domain of the data to be labeled may not be easily understood by the population in general (such as hyper-spectral imagery or computer-network data). The datasets typically targeted for crowd sourcing, and those used in the previous work, are image tagging and sentiment analysis. As the ability of a crowd sourced audience to provide labels reduces, the burden on the expert increases dramatically, and the effectiveness of the previous work diminishes. Our tool is designed to handle high-dimensional and abstract data, that at face value would be daunting to label without additional meta information (such as maps or domain knowledge), and to present that data in an easy-to-interpret-and-manipulate format regardless of the type of data.

4.2.5 Deep Learning Semi-Supervised Classification

In this section, we cover some of the recent related works which use deep learning for classification tasks. A new approach by Rasmus *et al.* [120] uses a ladder-network for a classification task, where unsupervised and semi-supervised training methods are combined and used simultaneously to improve the overall training and performance of a model. A ladder network utilizes an autoencoder model, but with additional "skip" connections between the encoder and decoder to transfer details which would otherwise be lost. In their work, the ladder network is treated as both a noisy encoder and a noise decoder, known as a denoising autoencoder (dAE), which also functions as a hierarchical latent variable model. In the dAE,

an autoencoder is trained to reconstruct the original observation from a corrupted version, which is compared to a clean version of the original observation run through the encoder. Using this setup, they are able to perform state of the art semi-supervised classification on the MNIST and CIFAR-10 datasets using only a small subset of the data, using only 100 examples and 1000 examples for two different tests, but such that every unique label (type) is present and no single label group is over-represented in the training set. Using a ladder network, the authors demonstrate that they can achieve low test error percentages (high classification accuracy) as compared to previous semi-supervised classification results using extremely small training labeled sets.

Pezeshki *et al.* [121] conducted an investigation into how the different aspects of a ladder network function, and the influence of different amounts of training data on those components. In their experiments, they removed or reconfigured individual components to learn about their relative importance in the operation of the model. One particularly interesting result reported is that the skip connections between layers become less important as the number of training examples increases, with more emphasis placed on the injection of noise in each layer. The experimental setup included evaluations using a permutation-invariant MNIST dataset, evaluated on 100 and 1000 training examples, such that every label is equally represented and present. The authors found that using their own configuration of the ladder network, they were able to achieve state of the art classification accuracy.

Our work differs from the previous work in a few ways. The goal of our work is label discovery, to refine the initial coarse labeling provided with a dataset to a more

precise labeling, and adding labels which did not previously exist in the dataset. The power of the previous works is their ability to leverage the unsupervised power of deep learning in conjunction with traditional semi-supervised classification. The previous works are able to generate impressive classification accuracy given a small number of training examples, but still require knowledge and examples for each of the unique categories, where our approach does not have this requirement.

4.2.6 Interactive Intelligent Systems and Active Learning

There have been other systems that combine the abilities of humans and machines to achieve a goal that would be difficult by either alone. The process of continuously updating a model through human interaction, which then produces results for a human to judge and operate with, often called semi-supervised learning, is also known as active learning. The advantage of active learning is that both the machine and human are able to make better decisions as well as continuously update and refine their decisions based on the output from the other. An example of this iterative updating process is the work by Ware *et al.* [122] who have developed a system for interactively training a machine learning classifier by leveraging the background and domain knowledge of the users. The system used a 2D scatterplot visualization where two of the many feature attributes selected define the plane axes. By iterating over the different pairs of attributes, the authors found that users were able to create classifiers on par with those of the state of the art by visually partitioning the data and drawing decision boundaries.

A system called CueT by Amershi *et al.* [123] uses human analysts to train and refine a recommender system for network system triage (linking associated error messages into a common problem). In their work, they build a similarity matrix to group and classify incoming alarms and tickets, which is refined through interactions of the analyst. These recommendations are then sorted and presented at the top of a ticket list and color-coded based on severity. Soto *et al.* [124] developed a system called ViTA-SSD for presenting and identifying patterns among semi-structured documents for text mining and analysis. This is done by leveraging a learned corpus of important words from the documents along with meta-data to generate document clusters. These 2D clusters are generated using a dimensionality reduction based on a combination of t-Distributed Stochastic Neighbor Embedding (t-SNE) and K-means clustering, which can be refined using a user adjustable distance metric for measuring similarity between documents. These clusters are then visualized using a scatter plot, and can be refined through the selection of particular keywords or documents which an analyst may find useful, to allow further refinement of the documents and clusters relating to the user input to be shown in the following iterations. The ultimate goal is to allow for analysts to discover correlations, trends, and similarities between different sets of documents and topics, such as similarities of documents across different topic domains. The primary limitation of the previously mentioned systems is that they have been engineered to handle a specific or discrete type of data, or to try to help solve a specific kind of problem. In addition, the previous systems are designed to help find correlations or differences between data elements, or to improve the speed with which a user can interact with large amounts

of a specific kind of data. While also finding correlations and patterns in high-dimensional data, our system allows for the actual modification and improvement of a given dataset for future use. Not only does our approach improve the model used within our own tool, it allows for the improvement of any future models trained using the same data.

4.3 Our Approach

In this chapter, we present our approach to facilitate the discovery of latent groups and labels within high-dimensional datasets through spatially meaningful visualizations generated using deep learning. In addition, we also present our technique that allows an analyst to iteratively define and refine the labeling of a dataset to reveal interesting trends and sub-communities. We illustrate our approach with several datasets.

A general overview of our technique as compared to traditional approaches can be seen in Figure 4.1. The traditional learning paradigm consists of first labeling a large dataset through intensive and tedious work, and then using that constructed dataset to train a model. As stated earlier, this approach can suffer from labeling errors and omissions, which can lead to errors and gaps in classification. In our approach, we take the same datasets along with a simple labeling scheme to train a deep neural network and produce a visually meaningful representation of the dataset and the labels associated with that data. Using that visual representation, an analyst can visually refine the labeling of the data based on patterns and spatial

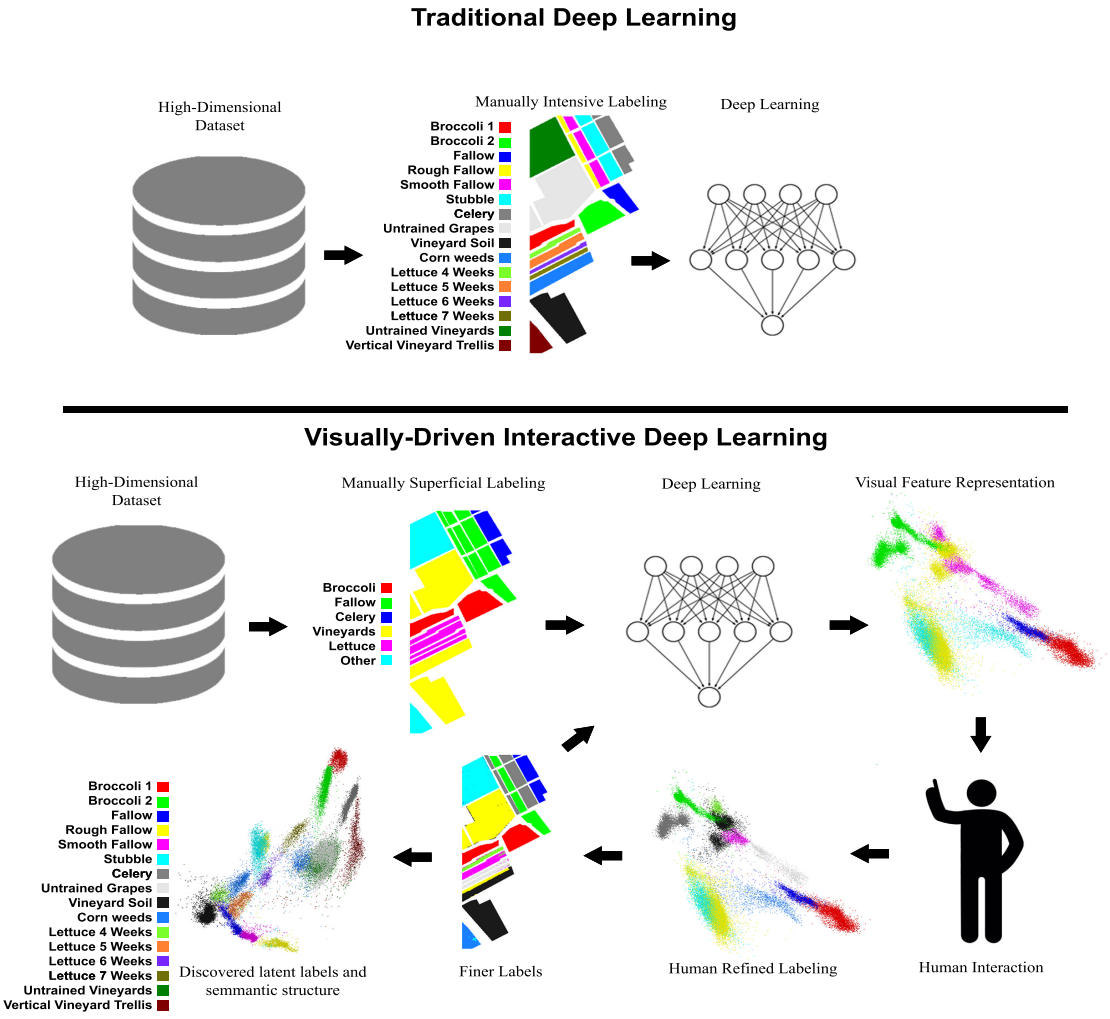


Figure 4.1: A brief illustration of the difference between traditional deep learning techniques and our approach. deep learning traditionally requires a large, time-consuming, and precisely labeled dataset for training. For many different reasons, such datasets may be inappropriately labeled. In our approach, we start with coarse labels (that are typically far easier to construct) and then refine them through an iterative process, involving visual interactions and deep learning.

distinctions, which are fed back into the model to generate a new spatial representation. After a few iterations, our approach can accurately refine the labeling and

discover hidden groups within high-dimensional datasets. The end of the iterative process depends on the structure of the data. As the user makes changes to the labeling of various data points, the structure of the presented data changes. As the iterative process continues, the structure changes less and less and eventually converges. Once the structure has converged, the user may conclude the iterative process.

4.3.1 Point-Distribution Generation

Our goal is to generate a spatially meaningful 2D representation of the data such that data elements with similar features are located close together. To generate the representation, we used Matlab and Python with the deep learning libraries TensorFlow [125] and Keras [126]. Therefore the deep-learning performance reported here is not particularly optimized for time but for flexibility. We developed a Siamese deep neural network inspired by Hadsell *et al.* [102], which is built on-top of a Variational Autoencoder as inspired by Hinton and Salakhutdinov [101]. Our visual knowledge discovery program uses 2D and 3D rendering techniques with GPU-accelerated OpenGL and C++. Data processing was performed on a computer with an Intel Xenon 2.6 GHz CPU and a NVIDIA GTX 1080 GPU.

The deep neural network used for dimensionality reduction and revealing hidden clusters is composed of two systems, a Variational Autoencoder, and a Siamese network. For a given dataset, whose labeling may be coarse or absent, the autoencoder first learns a 2D representation based solely on the data. The idea is to let

the data itself drive the dimensionality reduction, and to have any emerging groups that come out of the process be based on the features within the data. From this pre-trained representation, the Siamese network then steps in, using the same network structure and weights. Its' goal is to refine that network, by pushing and pulling on these clusters in 2D space, such that similarly labeled groups of points are pulled together, and dissimilar groups are pushed apart. It is this joint process, of leveraging both the labels and features, that is used to generate the 2D distributions. Figure 4.3 shows the effect of the Siamese network on a distribution of points generated by the Variational Autoencoder.

A visual representation of the network structure and procedure can be seen in Figure 4.4. The deep neural network structure starts with an input layer with the size of the dimensionality of the input data. From that input layer, that data is passed into a set of three one-dimensional convolutional layers, a 128 filter, 9 feature wide layer, a 64 filter, 6 feature wide layer, and a 64 filter, 3 feature wide layer. All convolutional layers have rectified linear activation functions [127], also known as Relu, as defined as $\max(0, x)$. The last convolutional layer is then passed into a Flatten layer, which brings the internal data structure back to a 1D representation. From this flattened layer, the data is passed into a Dense layer with two nodes, using the identity activation function $f(x) = x$, which is used to generate the (x, y) coordinates of a given datapoint. This network structure is shared between both the Variational Autoencoder and the Siamese networks.

To update the Variational Autoencoder, reconstruction loss (binary cross-entropy [101]) and Kullback-Leibler (KL) Divergence [128] are used to refine the

network. The Variational Autoencoder is completely unsupervised, relying entirely on the features within the data. The goal is to generate a 2D representation of the data which captures as many of the intrinsic properties of the high-dimensional data, such that from the 2D representation, the original high-dimensional data can be reconstructed accurately. For the Siamese network, two data points are fed into two networks with the same structure and weights, with the weights of the networks updated identically across iterations based on the contrastive loss function.

$$Distance = \sqrt{\sum (x_a - x_b)^2}$$

$$ContrastiveLoss = y * Distance + (1 - y) * \max(0, 1 - Distance)$$

Figure 4.2: Equations used in calculating the Constrastive Loss for the Siamese Network

To refine the Siamese network, we use a contrastive loss function using Euclidean distance (as defined in Figure 4.2), which compares the two output 2D points and checks to see if they belong to the same label, in conjunction with ADADELTA [129] during training. If the points to the same label, then the loss is the Euclidean distance between them. Otherwise, we adjust the loss to be high (one minus the distance), so that the points are pushed apart. This creates a 2D distribution of points such that points belonging to the same label are spatially distinct to points belonging to different label. We use a batch size of $20,000$ and train for a total of 10 iterations for both the Variational Autoencoder and Siamese

networks, values which were chosen as a balance to maintain real-time usability and data distribution integrity. The same network and weights are used and shared across iteration stages. Once the network is finished training, the network is used to generate the 2D distribution of points. After the 2D points are generated, the network weights are saved, to be re-loaded and re-used in the next iteration. For each of the spatial scatter-plot representations, we process, visualize, and allow the user to manipulate the entire dataset.

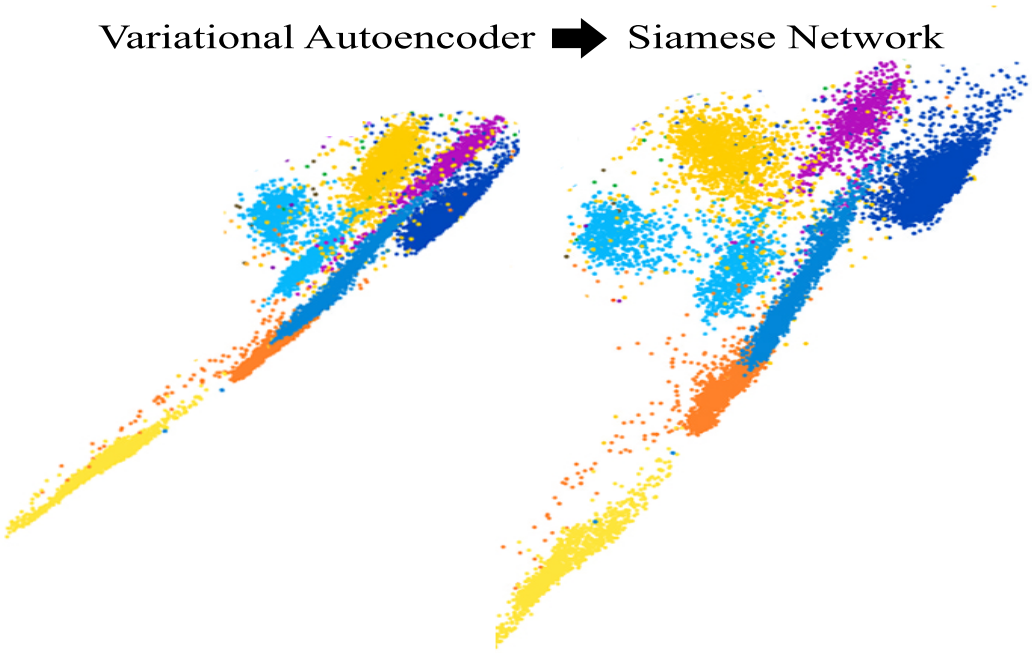


Figure 4.3: An example of the result of running the Variational Autoencoder and then refining that result using the Siamese Network. By running the Siamese network after autoencoder, the generated clusters tend to be tighter with more space in-between, making the individual clusters easier to identify. The user has the ability to adjust the number of iterations the Siamese network runs which affects the tightness of the clusters.

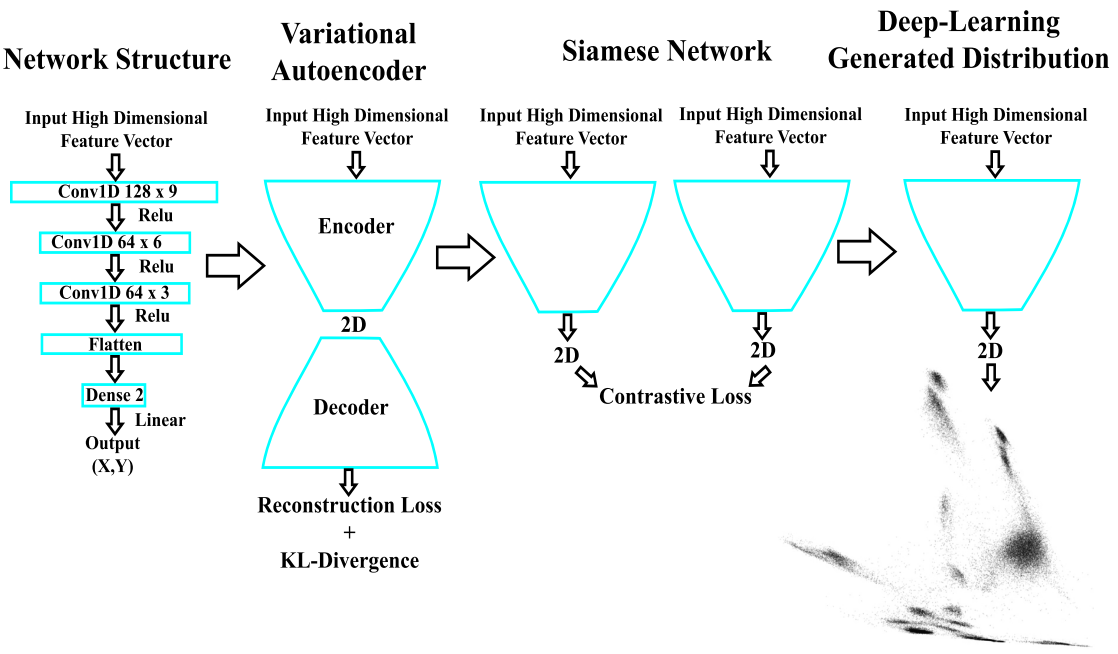


Figure 4.4: A visual representation of the network structure used in both the Variational Autoencoder and the Siamese network. The same network structure and weights are shared across the networks. First the Variational Autoencoder runs, and the Siamese network continues using the network weights generated by the autoencoder. After the Siamese network has run, the resulting network is used to generated the 2D distribution of points. The network weights are also saved, reused, and refined in the following iterations.

4.3.2 Cluster Visualization and Manipulation

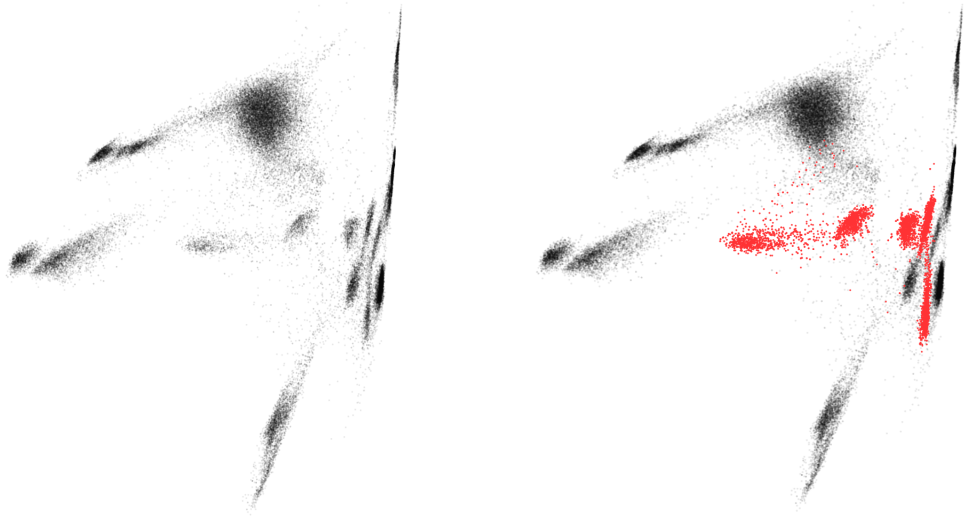
Our approach utilizes a 2D scatterplot to visualize the output from the deep learning network. Each data-point in the dataset is assigned a 2D position from the deep learning network and colored based on the group or label it currently belongs

to. The first distribution of points is generated using the initial coarse labeling from the dataset. As points are removed and added to new or existing groups, the distribution of the 2D datapoints changes to reflect the changing relationships between them.

We next discuss how the user interacts with the system to modify and reorganize the labeling of the 2D points. Initially, the entire distribution is presented to the user, but such that every point shares the same color (gray), which helps reveal the density of points, as shown in Figure 4.5(a). When a user clicks on a point, all the points which share the same group as the selected point, become activated (Figure 4.5(b)), as indicated by becoming colored. Multiple groups of points can be activated in this way. When the user clicks on the white space between points, all selected points/groups become deactivated, and return to the gray color. This scheme allows for an almost unlimited number of groups/labels to be handled by our system, especially since only a relatively few groups will be interacted with at any given time. When a user identifies a potential sub-cluster, they first "activate" the set of points by clicking on one of the points in the region of interest. After clicking, a menu appears, prompting the user with several different options. An example of this menu can be seen in Figure 4.6. The most basic interaction is "Select Points", which allows for a manual selection of points. When this button is clicked, the cursor changes to a square glyph, 5 pixels in size, to be used like a paint brush select points, as can be seen in Figure 4.7(b). When points are selected, then turn brighter. For larger selections, the control key may be held, which allows for a drag-and-select circle to appear, which automatically selects all the points that

fall within its boundary that belong to the selected group (Figure 4.7(c)). To turn these selected points into a new group, the user would then click the "Cut Group" button. These controls allow for manual interaction with the points, but can be tedious to use, and should only be used for fine-tuned selection. For larger changes to the points, we have implemented an automatic partitioning tool, inspired by how humans would naturally segment the points. The "Cut Group" menu option starts a normalized cut operation [130] on the clicked/activated group (if no points are currently selected). From the points belonging to the activated group, a similarity matrix is calculated by comparing the distances between all pairs of points, and then replacing those distance values x with $1.0/x$ to form a correlation matrix. Using this matrix, a diagonal and Laplacian matrix [131] are computed. The Laplacian matrix is then passed into an eigensolver, which generates the two smallest eigenvalued-eigenvectors. Ignoring the Fiedler vector, the vector whose eigenvalue is zero, the elements of the correlation matrix are reordered using the values of the second-smallest-eigenvalued eigenvector. Next, we compute an appropriate cutting location by minimizing the NCut function as described in the formulas in Figure 4.8, with cut(A,B) as the total weight of the edges connecting A and B, assoc(A,V) as the total weight of the edges of A in V, and with NCut(A,B) as the cost to cut A and B, normalized to favor roughly equally sized segments within a tolerance. The index which is found to have the minimum NCut value is used as the splitting point, and the group of points which was under the cursor at the time of first click are assigned to a new group. An example of the output from the NCut algorithm can be seen in Figure 4.9, with the left, smaller group of points being selected and cut

from the larger group. The second button, "Select Group" is used to select an entire set of points belonging to a group. This is useful for merging groups or merging a group with a selection of points from another group. To merge a group, the "Merge Group" button is used. This will merge together all selected points into a single group (label). The final button, "Deep Learning", initializes a call to run the deep learning algorithm which takes the current labeling distribution and retrains the network to generate a new distribution of points.



(a) An initial view of the Salinas Valley dataset, with none of the points or groups selected

(b) Same set of points with one group activated.

Figure 4.5: Initial view of points, with all points given the same color, which are only assigned a color once selected or activated by the user.

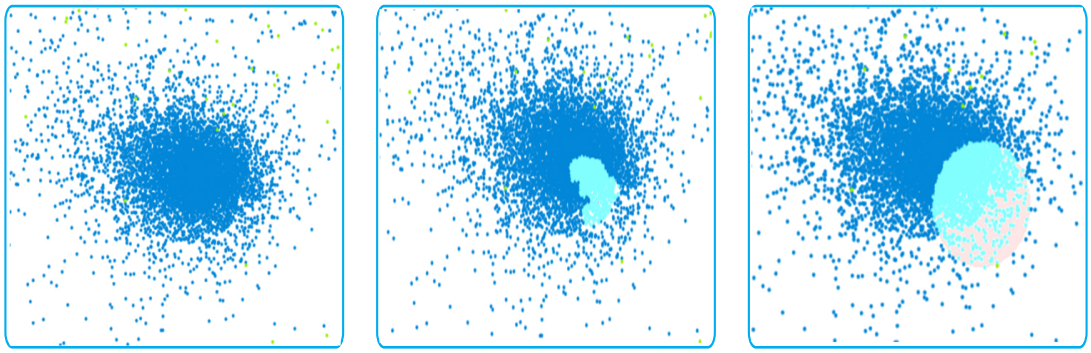
We have identified a few techniques that often enable an analyst to locate latent structures. The most common indication of a latent structure arises as visually distinct cluster or sets of clusters in the 2D visualization. The distinction could



Figure 4.6: The menu interface presented when a user selects a point/group.

appear as geometric or color separation, or a combination of the two.

An example of such an interaction can be seen in Figure 4.10(a). A slightly more complicated and common scenario is where the previous example exists, but the second cluster overlaps with a different cluster of another label. This can indicate that either the secondary group belongs to a separate group, such as in Figure 4.10(b), or that the two clusters belong to the same group and should be merged, as in Figure 4.10(c). This often indicates that the two groups share a common feature subset. A merge may become necessary when trying to extract a hidden cluster, and instead of one clear cluster being extracted, two smaller clusters overlap, indicating that the clusters belong to the same group. Another common scenario is when a bulge exists somewhere on a large cluster, such as in Figure 4.10(d). Finally,



(a) A set of points before selection

(b) Same set of points with the top section being selected using a "paint brush"

(c) Same set of points selected using a circle selection interface

Figure 4.7: Selection of points through manual paint-brush and circle selection interaction.

$$SimilarityMatrix(x, y) = \frac{\sqrt{(x-y)^2}}{2 * \sigma^2}$$

$$CorrelationMatrix = \frac{1.0}{SimilarityMatrix}$$

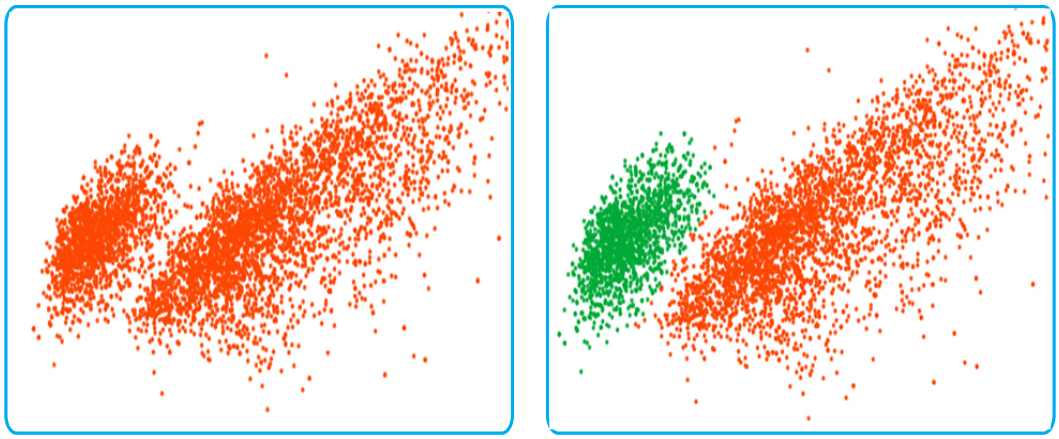
$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

$$assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$$

$$NCut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

Figure 4.8: Equations used to compute normalized cut.

a more complex and rare scenario exists when there are two sub-clusters within one common parent and one child group surrounds the other child group. An example



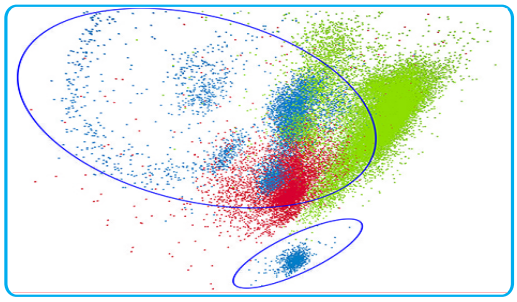
(a) A set of points before Ncut segmentation

(b) Same set of points after Ncut segmentation

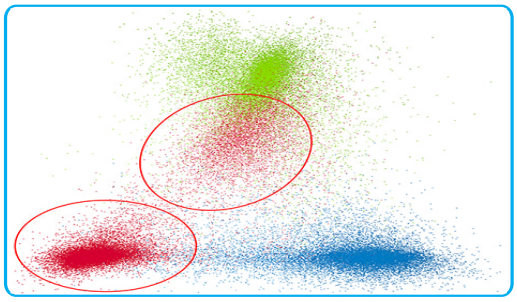
Figure 4.9: Segmentation of a set of points using normalized cut (Ncut).

of this can be see in Figure 4.10(e). This is often represented as a dense set of points surrounded by many disparate points surrounding the central denser group.

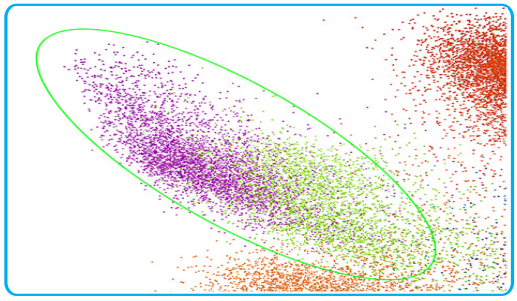
One common problem across all visualizations which perform dimensionality reduction is overlapping points. In high-dimensional space, points may be far apart, but when projected down into 2 or 3 dimensions, they can overlap and obscure one another. Our tool uses two different techniques to handle overlap. First, all points are rendered using transparency, such that the points below can still be seen. Secondly our visualization is designed around the idea that only a few clusters, or sets of points, will be interacted with at any given time. Therefore, when a group is selected, all the points belonging to that group "pop" to the front of the visualization and are given a distinct color, while all inactivated groups retain a high-transparency



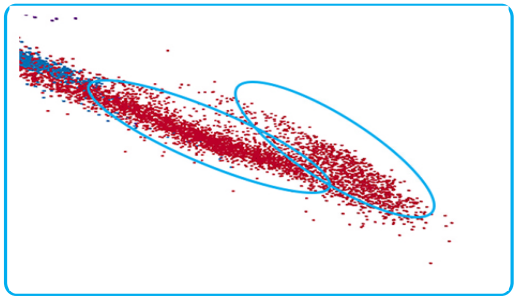
(a) An example of clear cluster separation.



(b) An example of two clusters which are distinct, but the separation boundary is not clearly defined

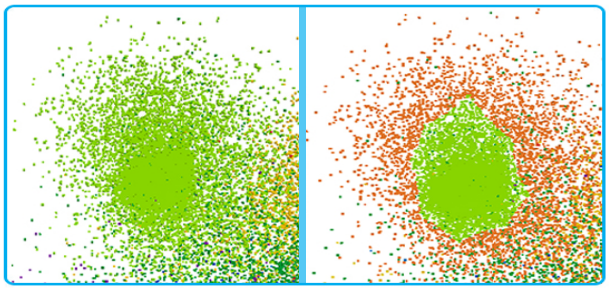


(c) A merge example, where the green and purple clusters should be combined. This can occur when a larger green cluster from a previous iteration is partially split.



(d) An example of a hidden cluster adjoining to another cluster, typically identified as a bulge protruding off another dense cluster of the same color.

value and gray color. This allows points of the given group to become easily visible compared to inactive points or points in the background. Figure 4.11 demonstrates a simple example, where the purple and yellow points are highly overlapped, and when

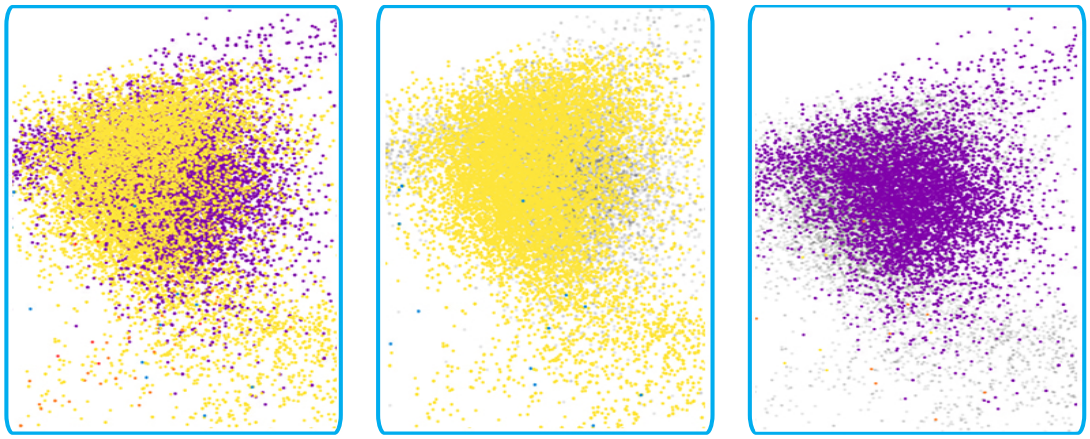


(e) An example of a cluster existing within another cluster. These are typically identified as points of the same label radiating from a dense center.

Figure 4.10: Common Interaction Techniques

selected, the other group fades away. Using our tool, we are able to individually select and visualize the independent clusters to see their full extent.

Throughout this process, it is possible for a user to make an error in labeling, such as mislabeling a few points. This is an inevitable part of the discovery process. If a new group is accidentally created, such that two groups now exist where one group should have only existed, it is easy enough to simply select both groups and merge them into a new group. Second, if a few points are mislabeled, it is very likely that those mislabeled points will appear spatially within the group they were meant to belong to after re-running the point-distribution algorithm, and contrast visually with those points. This contrast of points would then lead the user to merge those two groups together. In addition, this mechanism is often leveraged to tease apart tight groups and to test if a hidden group may exist over multiple iterations (an example of which can be seen in Figure 4.12). We present an example in Figure 4.13,



(a) Two clusters, purple and yellow, are obscuring one another

(b) The yellow set of points are selected, and the purple points are greyed-out

(c) The purple set of points are selected, and the yellow points are greyed-out

Figure 4.11: Handling of overlapping sets of points.

where there exist two independent clusters, and a user has accidentally re-assigned half the points of one cluster into the domain of another. Figure 4.13 shows that our algorithm is resilient against such mistakes, and that while deep learning is taking into account the labels assigned to each data point, the underlying features of the data are also drive the distribution of points, and that the mistakenly labeled points are still co-located with points sharing their true labeling.

There are three main components for each iteration. The first is training the neural network with the current labeling of the dataset to generate the 2D scatterplot. The second component is the creation, modification, and deletion of

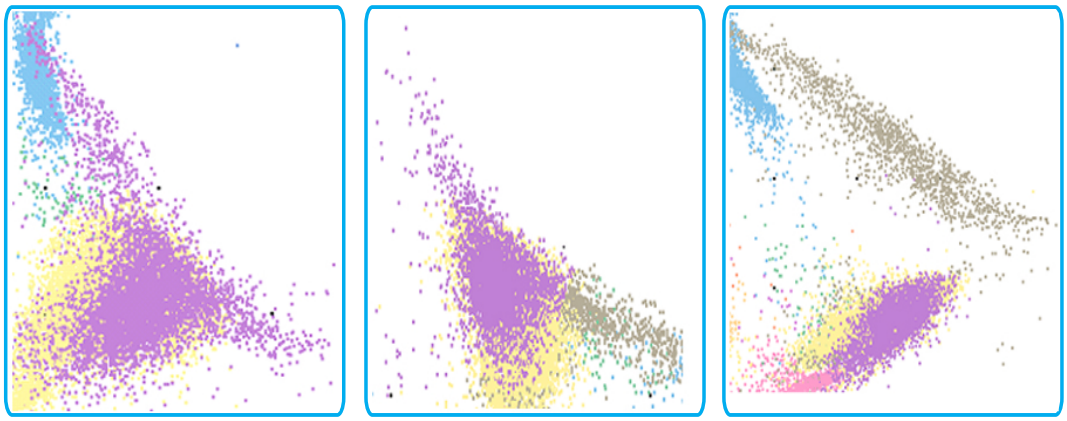


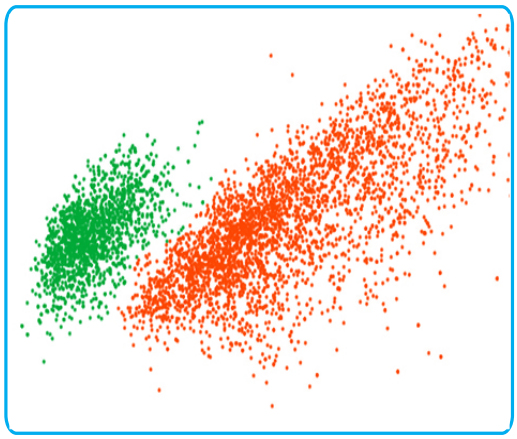
Figure 4.12: An example of where a few points are selected from a parent cluster, and over time the points are separated away from the parent cluster to form their own cluster.

groups using the visual interface and human interaction. Lastly, the modified labels are fed back into the neural network, where the weights of the network are updated from the previous iterations using the new labeling, and then a new distribution of points are generated.

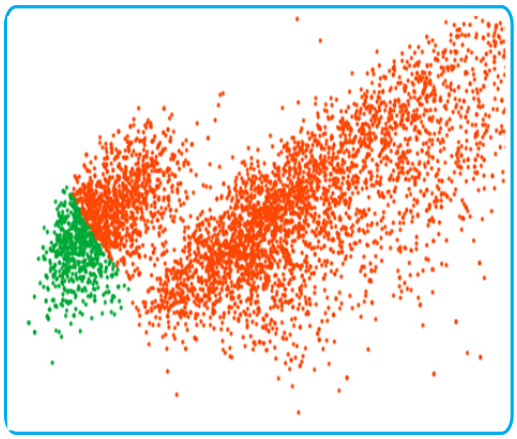
4.4 Results

In this section, we present the results of our technique on three different high-dimensional datasets. Our goal was to ease the burden of human labeling and iteratively use deep learning to discover the latent groups and labels within these high-dimensional datasets.

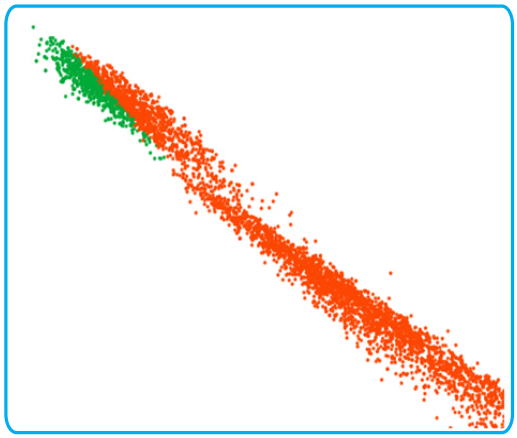
In this section, we will review two of the datasets tested in our system, a hyper-spectral image of Pavia University [8], and a hyper-spectral scan of Salinas Valley [9]. While the datasets we have selected and used for our study are spatial



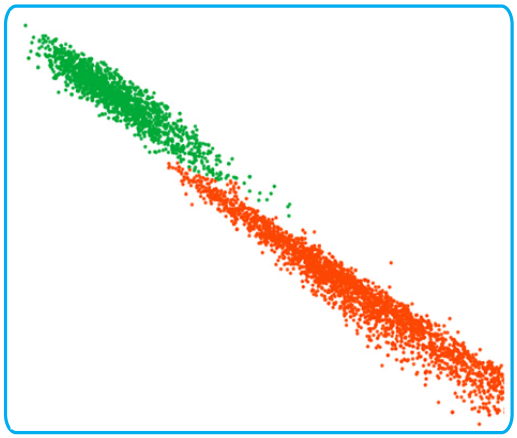
(a) Initial labeling distribution



(b) A labeling mistake where more than half of the points are mis-assigned to a nearby group



(c) The result of regenerating the point distribution using the erroneous labeling



(d) The labeling after correction

Figure 4.13: The resilience of the algorithm to labeling mistakes.

by nature, we disregard the spatial and neighborhood aspects of these datasets and consider each data-point individually during our discovery trials. Many real-world datasets have no natural or obvious spatial component. Therefore, our goal is to generate meaningful spatial representations using our technique that enables the discovery of latent communities for non-spatial datasets. However, it is still possible to use spatial datasets in our approach, but the spatial aspect will act as meta information that is not directly utilized by our technique at present. Throughout this chapter, we use the innate spatial distributions of our selected datasets as an easily understandable visual metaphor to help analyze how well we are able to recover the hidden labels, but these visual representations are never used or seen during the iterative discovery process. The only visualizations presented to the user during the iterative discovery process are the constructed scatter-plot spatial representations.

For each dataset, we present a series of figures showing the initial coarse labeling, the feature-space distribution generated through iterative deep learning guided by visual interactions, and the true distribution of labels. Throughout the process, the ground-truth labeling was not used to aid the discovery and refinement process, but is being presented here merely as an illustrative tool. Below we discuss the results for each of the previously presented datasets.

4.4.1 Pavia University Dataset

The hyperspectral Pavia University dataset consists of a 610 × 340 pixel image, where each pixel is represented by a 103-dimensional feature vector containing intensity values at different spectral bands. Each pixel has been labeled based on the identity of the surface. There are 9 discrete labels: asphalt, meadows, gravel, trees, metal sheet, soil, bitumen, brick, and shadow. We decided to group those labels into three coarse categories natural surfaces, roads, and buildings, which can be seen in Figure 4.14. In Figure 4.14, we show the initial spatial representation of the Pavia University dataset.

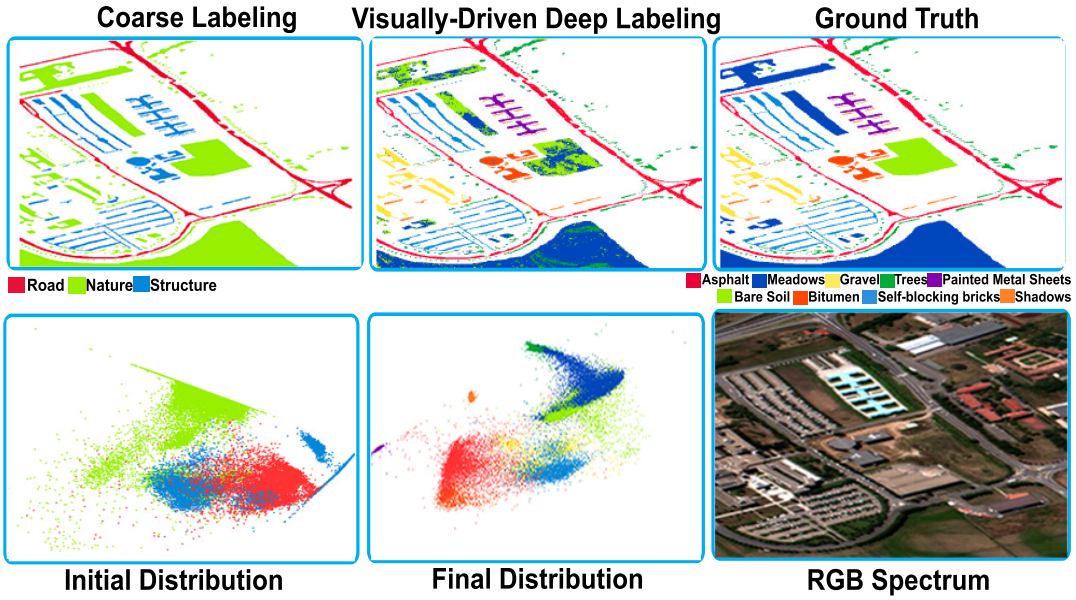


Figure 4.14: A comparison of the initial coarse labeling, to the generated refined labels, and the precise (true) labels of the Pavia university dataset after 3 iterations. Starting from the three initial categories: natural surfaces, roads, and buildings, we were able to reconstruct the distribution of the 9 labels with an accuracy of 88.2%.

Starting from this initial representation, we manually select and re-label points into new clusters, and then generate new point distributions in an iterative manner. Within 3 iterations we were able to generate 9 labels from the initial three coarse labels and improve the labeling accuracy from 67.7% to an accuracy of 88.2%. To test the robustness of our technique, a hold-out test evaluating 10% of the data was conducted, to determine if the hold-out test set could be accurately classified/grouped into the proper cluster when trained on the other 90% of the dataset. For the 10% holdout test-set, 95.3% of those points were correctly assigned to their proper group, confirming the generalization and stability of the technique. There is a lot of overlap between the different "brush" categories; trees, dirt, and meadows. We were unfortunately unable to differentiate these categories perfectly, but based on the satellite RGB imagery, we believe that these categories are not strictly distinct and therefore, there exist a lot of overlap between those categories. An example of this is the irregular patch of "meadows" in the center of campus that many participants in our user study, presented in more detail later, labeled as an independent group, as seen in Figure 4.15. The proximity of this cluster in the 2D space indicates that, while this group of points is similar to "meadows" group, there is a clear distinction in their features, as supported by the aerial view. The reconstructed labeling we generated for this dataset can be seen in Figure 4.14 along with the ground truth. The average time required to run each iteration of the deep neural network was 9 seconds.

Based on the final data-point distribution as shown in Figure 4.14, it is possible to further refine the cluster segmentation and labeling, which we discuss more

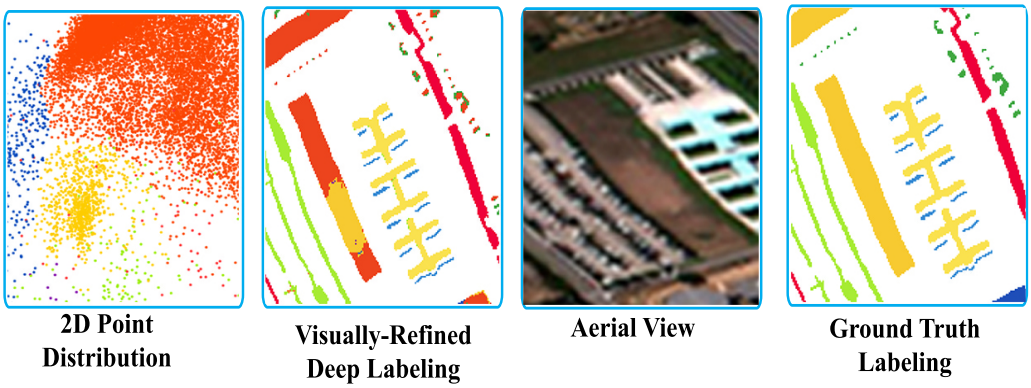


Figure 4.15: A comparison between the labeling generated by our system and the ground truth labeling. The labeling generated by our system is driven by the clearly distinct group off the main body of points above it. This difference is supported by a visual difference in the aerial view.

in a later section. The next dataset we present is more complex, involving more dimensions and more hidden labels.

4.4.2 Salinas Valley

Here we review our results for another hyperspectral dataset, a (224-band) scan of Salinas Valley, California. The dataset consists of a 512×217 image, with each pixel represented by a 224 (204 after removing the water absorption bands) feature vector. The valley consists of fallows, broccoli weeds, stubble, celery, grapes, various vineyards, corn weeds, bare soils, and various stages of lettuce growth for a total of 16 different labels. For our purposes, we clustered those 16 labels into 6 groups, where each group contains similar labels. For example, there were four

labels consisting of lettuce at various stages of growth. These were clustered into a single group, which would constitute a reasonable assignment if the growth stages were not known at the time of initial coarse labeling. The 6 groups we created are named broccoli, fallow, celery, lettuce, vineyards, and other. This can be seen in Figure 4.16. In the bottom left of Figure 4.16 we show the initial distribution of data points from the Salinas Valley dataset. After three iterations of re-labeling and re-generating the point position distribution, starting from the 6 initial coarse labels and an accuracy of 58.61%, we reconstruct the 16 precise labels with an accuracy of 97.4%. The state-of-the-art technique using the Salinas Valley dataset achieves 97.11% classification accuracy [132]. The average amount of processing time required by each iteration of the deep neural network was 10 seconds. Another hold-out test using 10% of the data was conducted using the Salinas Valley dataset, to determine if the test set could be accurately classified/grouped into the proper cluster. For the 10% holdout test-set, trained on the other remaining 90% of data, 100% of those points were correctly assigned to their proper group, also confirming the generalization and stability of the technique.

4.4.3 User Study

To demonstrate the usability and capability of our approach and tool, we conducted a small user study, inviting participants to iteratively investigate the two datasets presented earlier in this chapter. The participants were informed that their goal was to try to determine if there were any undiscovered groups within the pre-

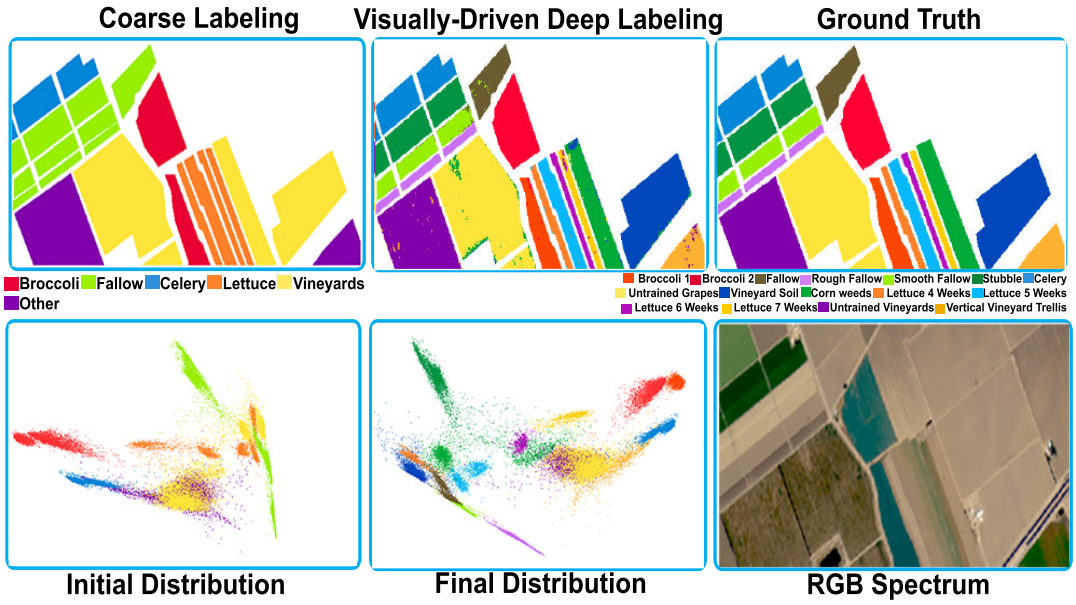


Figure 4.16: A comparison of the initial coarse labeling, the generated refined labels, and the precise (true) distribution of labels, of the Salinas valley dataset after five iterations. Starting with 6 initial coarse labels, we were able to reconstruct the distribution of the 16 labels with an accuracy of 97.4%.

sented visualization, and were briefly shown how to interface with the tool. No information on the number of true clusters, the nature of the data, or how to determine what makes a cluster distinct were provided. Our goal was to simply provide the bare-minimum needed for the participant to use the tool. All participants found the tool to be very intuitive and easy to use. Each participant started from the exact same layout of points, network structure, and initial network weights, and were instructed to continue refining their labeling until they verbally stated that they were satisfied with their groupings. After the study, the participants were explained the true nature of the task, and were astonished with the ease they were able to accomplish this task. The results of the user study are summarized in Figures 4.17 and

4.18. Note that the scores for the primary author have been also been added to these plots, showing that the performance between experienced and inexperienced users are comparable. For each dataset, there is a clear iterative upwards improvement in the labeling accuracy of the datasets, suggesting that even users who have had no prior experience working with dimensionality reduction and data analytics were able to not only use our tool, but achieve positive or spectacular results. The solid-line plots show the accuracy of the data-labels with respect to the true, hidden labeling, for each iterative step, showing that the recovered labeling increases in accuracy with each subsequent iteration. The dashed-line shows the amount of time spent on manual interaction during each iteration of the reconstruction process, revealing a general downward trend. This suggests that in the beginning, large changes are being made in labeling, which changes to fine refinements as the iterative labeling process continues. It is important to note that, although we use accuracy as a measure to quantify the success of our system, it is not exactly a proper measure. The goal of our system is hidden label/group discovery, and throughout the rediscovery task, the primary author and participants have found a few sub-groups or hidden groups within the existing datasets, which are clearly differentiated in our visualization (covered further in the next section). The participants, unaware of the underlying metric of accuracy, have over-labeled the datasets, driving down their otherwise high accuracy scores, but in doing so, have been successful in their stated goals and the goals of the presented system.

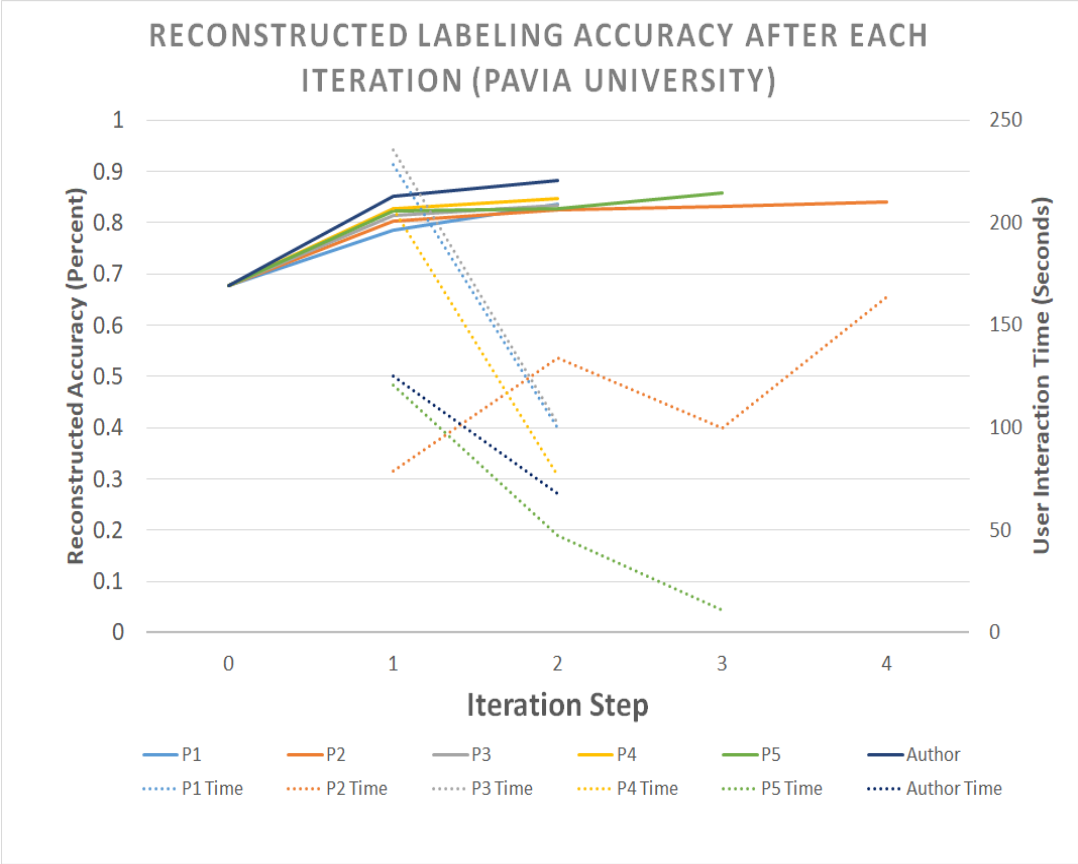


Figure 4.17: Participants accuracy and timings for the Pavia University dataset.

4.4.4 Interpreting Discovered Labels

For our testing, we used datasets where the precise labeling of data was complete, and then grouped those precise labels to create coarse labels. Our goal was to reconstruct and rediscover the provided distribution and labeling of the data. Because the full precise labeling (the ground truth) is known, we are able to discern which of the newly discovered labels are associated with each pre-existing label. However, our technique is designed to be used for hidden label/group discovery. A user could start with a group whose label is known, and discover it contains a sub

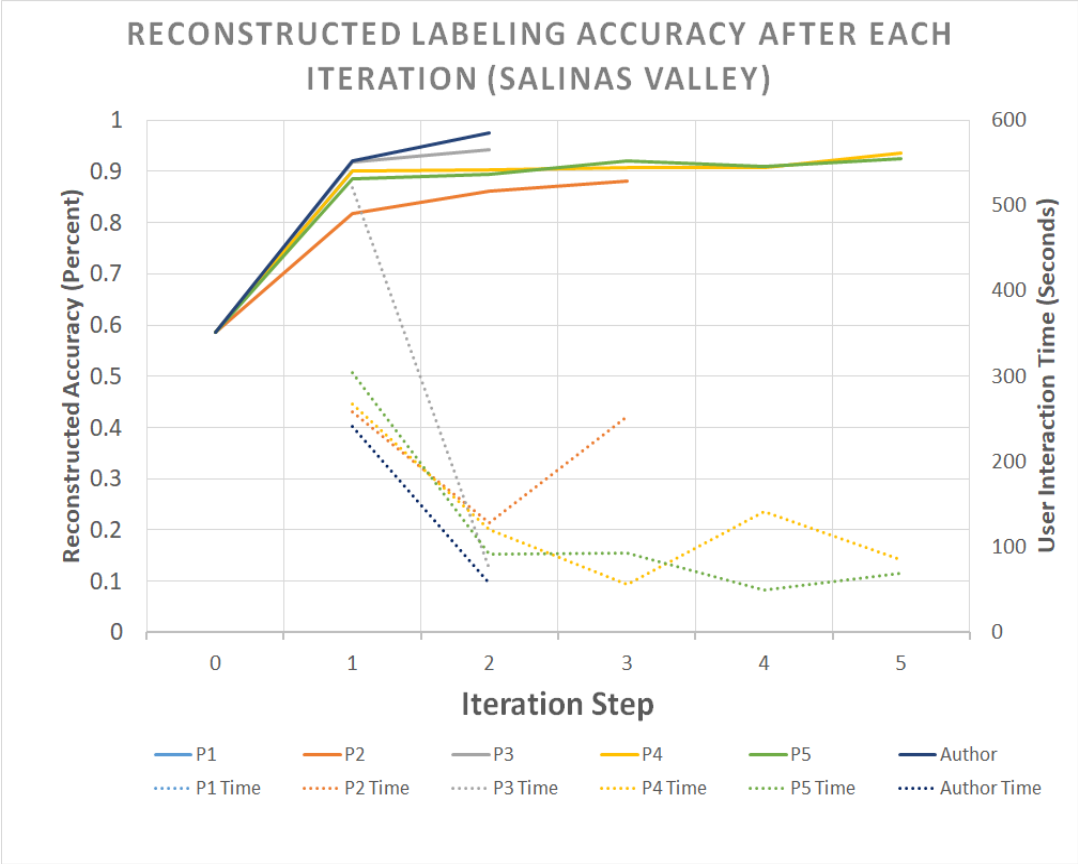


Figure 4.18: Participants accuracy and timings for the Salinas Valley dataset.

group or is composed of a few sub groups, which could be then found and re-labeled using our technique. It may then be possible to identify what the newly discovered labels correspond to or mean based on additional meta information. For example, we identified through our visualization that the coarse lettuce category from the Salinas Valley dataset had four sub-clusters within it, but it would be very difficult to identify that the reason they are different is due to age, based only on the hyperspectral imagery, and without the additional meta information.

In our iterative discovery process on the two previous datasets, it was possible to continue the iterative process to generate more labels, beyond the given ground-

truth labeling. For example, in the Pavia University dataset (shown in Figure 4.14), in addition to the findings presented in Figure 4.15, we discovered another sub-cluster within one of the larger groups for the painted metal sheets. In Figure 4.19, we show the result and spatial representation of sub-dividing the purple cluster. By using the additional meta-information (in this case the spatial information) we can see there is a consistent spatial coherence for the newly discovered group. The newly discovered sub-cluster runs along the vertical portions of the roof of these buildings. We may speculate as to why there is a distinct difference here, but unfortunately the precise labeling and aerial view given with the dataset does not reveal any clues as to why there is a consistent feature difference within the group. Another set of interesting discoveries made were in the Salinas Valley dataset (as shown in Figure 4.16). In Figure 4.20, we compare the aerial view of the Salinas valley with the labeling generated using our technique and the ground-truth labeling as provided by the dataset. Looking at the aerial view, a discoloration in the field is visible, which clearly does not match the rest of the field, indicating a different surface material is present. The ground truth labeling does not take into account this discoloration, and techniques which use shape or structure to partition or label an image may miss this entirely. Our technique reveals that this surface material was different, and correlates with the aerial view. The absence of these hidden groups within the datasets may be for one of the earlier presented reasons: lack of knowledge of the existence, a simple mistake, or labeling simplification by relying only on the satellite RGB, or overall shape information.

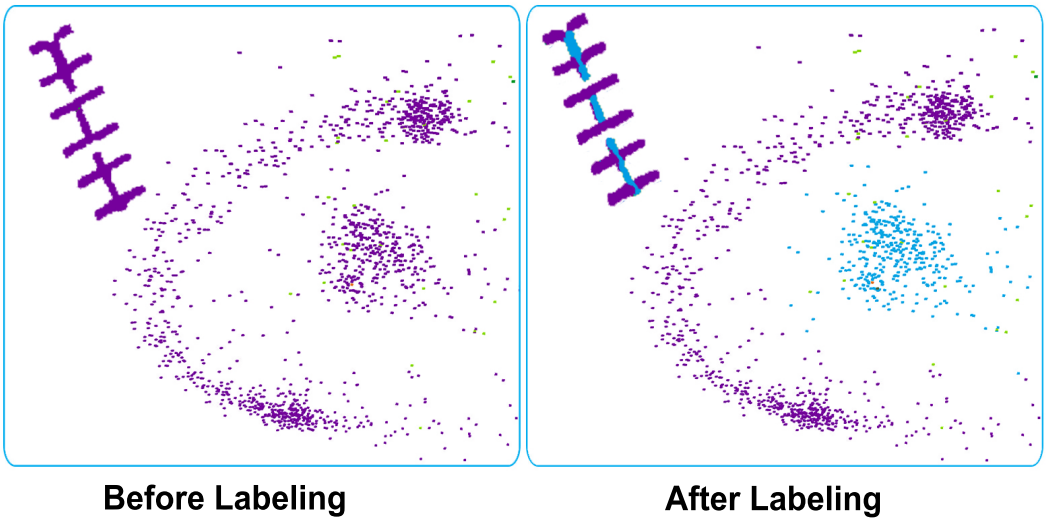


Figure 4.19: A potential newly discovered sub-group within the painted metal sheets label in the Pavia University dataset. On the left shows the point distribution and spatial representation of the labeling as generated by our technique. On the right shows another iteration showing if we had sub-divided the current points and the resulting spatial representation.

4.4.5 DNS Query Dataset

One of the goals of the presented system is to find hidden groups within real-world, large, high-dimensional datasets where there is little to no meta information, such as spatial cues, to aid in that process. In this section, was present the results of our system on a real-world dataset from the University of Maryland D-Root DNS server, one of 13 root DNS services which handle DNS requests for the entire internet. DNS stands for Domain Name Service, which is responsible for answering queries, converting human understandable URLs into IP-addresses that the com-

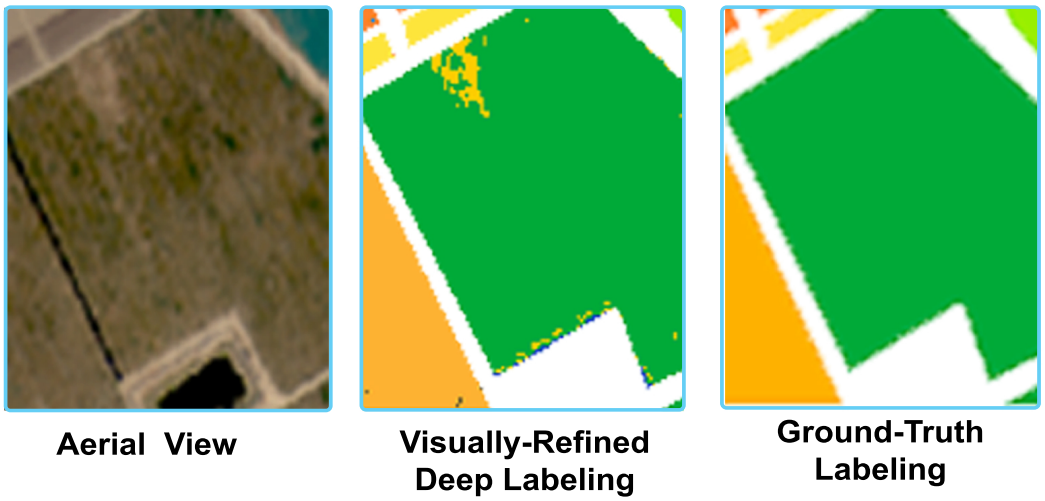


Figure 4.20: A potential newly discovered sub-group within the untrained grapes label in the Salinas Valley dataset. The left figure shows the meta information used, the aerial view of the valley. The middle image shows the labeling as generated using our approach. The right image shows the labeling as given by the dataset. The yellow labeling portion in the middle image matches with a discoloring in the aerial view, which suggests that there may be a different material there.

puter can understand. The D-Root DNS service receives billions of such requests per day. As part of their analyses, understanding the nature and distribution of the traffic they receive is crucial. Using this data, our goal was to uncover clusters of queries, to determine if there were any trends in the huge volume of queries the D-Root DNS server receives. To convert the queries into a format that can be passed to deep learning, each query string is converted to a vector of TF-IDF (Term Frequency, Inverse Document Frequency) values, where each value of the feature vector corresponds to a particular character. For our purposes, we process an hour of traffic, which is 7.5 gigabytes for nearly 7 million queries (data points). In Figure 4.21,

we show the evolution of the generated query space throughout the discovery and labeling process, as performed by the primary author. To start, there were no labels (as nothing was known about the dataset), and after 8 iterations, 42 groups/labels were generated (although this process could continue to further refine the groups). To verify the success of our process, we have manually investigated the discovered clusters and have provided each group with a human-understandable label in addition to presenting any trends that exist within a group or across groups. The result of this labeling is shown in Figure 4.22. Many different groups were discovered, the most salient being those regarding erroneous IP address queries. Using our tool, we were able to identify different configurations of queried IP addresses based on the error of the request. Other interesting discoveries were sets of code fragments and commands being issued over DNS, which is common for command and control of botnets. Lastly, we were able to find different configurations of alphabetic-based queries, ranging from differently sized sets of random characters, to more formed queries with specific domains. Our collaborators at the University of Maryland D-Root server agree that not only have we confirmed the various types of traffic they see, but we have also revealed other types of traffic, that they were unaware of but are now interested in.

4.5 Discussion

In recent years there has been an explosion of large high-dimensional datasets. Extracting meaningful information hidden within these high-dimensional datasets is

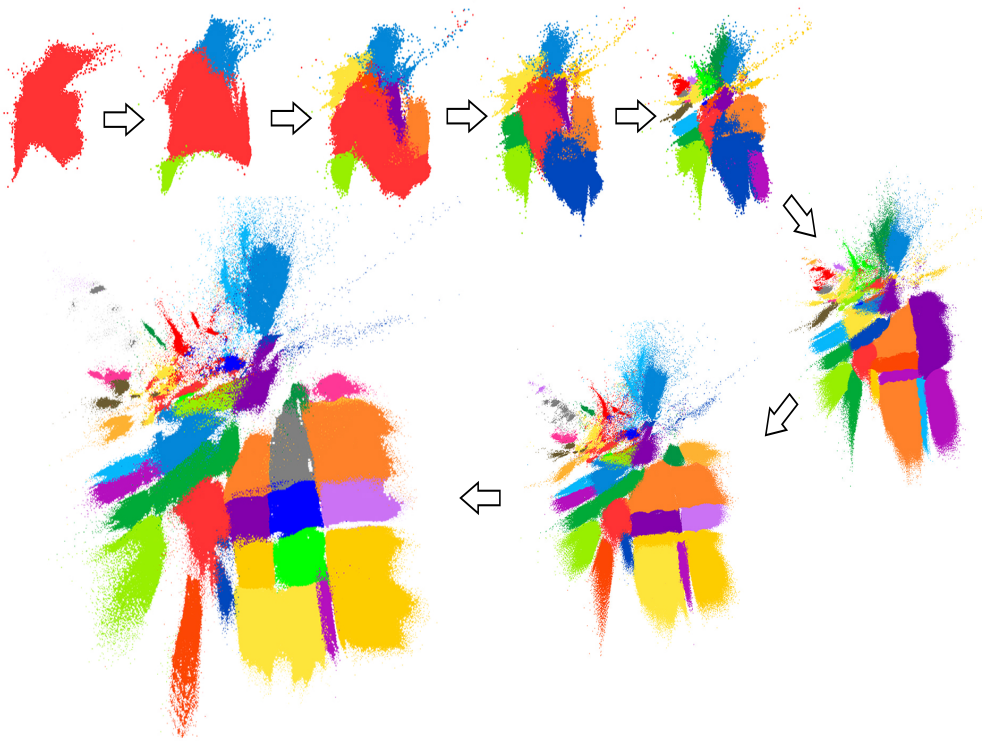


Figure 4.21: The evolution of the query-space representation over eight iterations, showing the influence of the iterative labeling. Note that for some clusters certain colors have been re-used due to the high-number of groups.

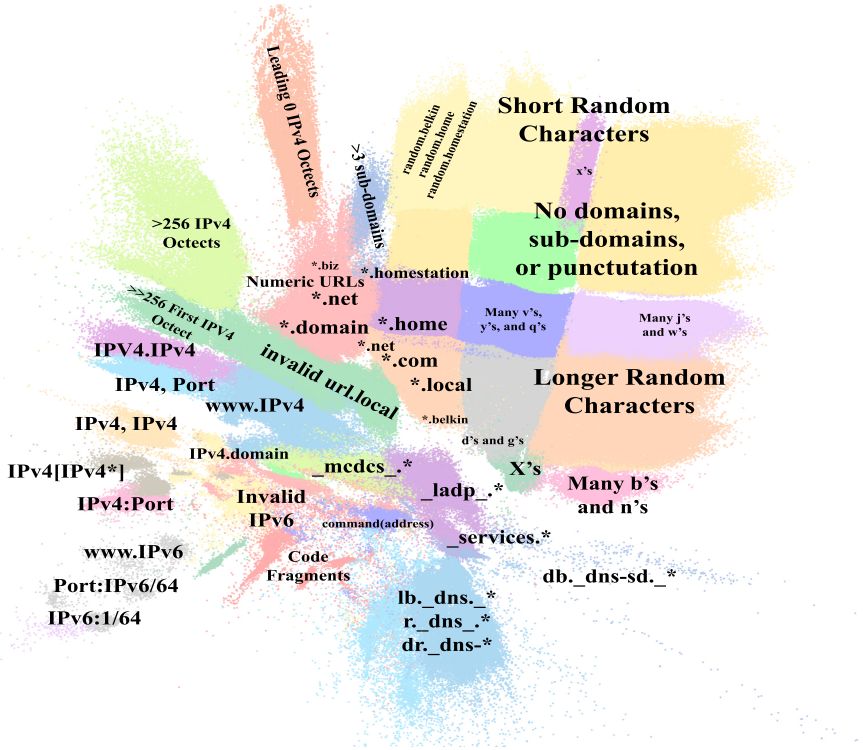


Figure 4.22: The output of the DNS query investigation coarsely labeled to identify the meaning of the individual clusters and their relation to each other. Note that for some clusters certain colors have been re-used due to the high-number of groups.

not trivial, and is made more difficult by erroneous and high-level (coarse) labeling. These errors may be caused by subjectivity, lack of time, or misunderstanding of the data. One way of revealing hidden trends or structures within a high-dimensional dataset is to group similar points based on their features. Identifying similar high-dimensional data-points is difficult in part due to the Curse of Dimensionality, which states that the distance between all pairs of high-dimensional points converge, meaning that traditional distance techniques reveal little information. Another difficult aspect of high-dimensional information visualization is deciding on how to show distinctions and similarities between high-dimensional points in an intuitive and insightful way.

In our approach, finer labeling and classification of the data requires more iterations. To reduce the amount of human effort required we have added the point-and-click normalized cut segmentation. This is particularly effective for segmenting large numbers of smaller clusters quickly. Introducing more automated tools such as this to reduce the effort required by a human, but still keeping a human in the loop and giving them the final say, would be an interesting direction of future research.

4.6 Conclusions

The current advances and future potential of deep learning is without question. The objective of this chapter is to provide a first step to advance our capability and understanding of deep learning. In this chapter we presented our approach to facilitate the discovery of latent structures and communities/labels within high-

dimensional non-spatial datasets using deep learning. Given a coarsely, or broadly labeled, or unlabeled high-dimensional dataset, we generate a 2D distribution of the data based on the idea that similarly attributed and labeled points should be in close proximity to each other. Through an iterative process, an analyst can select points and assign them to new or existing communities, which is then used in conjunction with deep learning, to refine the 2D spatial distribution of the points, revealing more new information. Supplementing the instinct and ability of analysts to identify patterns with deep learning, we have shown on three different datasets that our technique is able to reconstruct hidden structures and communities. In many previous works, deep learning has been used as a black-box classifier, with little or no human interaction. Our technique, presented here, enables both deep learning and the human analyst to support, refine, and enhance each other through visualization.

Chapter 5: Visual Analytics for Root DNS Data

5.1 Introduction

There are two schools of thought on how to deal with cyber-attacks, automatic detection methods, and human-driven investigation. Automatic detection methods work by modeling normal and abnormal behavior, through prior knowledge of the behavior of malware and since by definition a cyber-attack is abnormal. However, humans and the machines are also capable of abnormal behavior, causing these automatic detection algorithms to throw many false alarms. In addition, these models take months to create, are created using attack data that occurred on average at least six months prior, and generally only detect the presence of old cyber-attacks, and are therefore unprepared for ever changing and newer attacks. In addition, these methods generally only report the presence of an attack, but give no details on specifics. In contrast, and in part in response to automated mechanisms, many cyber-security analysts prefer manual investigation and analysis of attacks. There is an overall mistrust of automated systems by those who perform cybersecurity analysis [133]. Generally, analysts tools consist of listing packets and related information, whereby data is inspected line by line. In contrast, many tools often only provide very high-level abstractions of the data, typically in the form of histograms,

where the histogram consists of the number of received packets.Few tools fill the gap between very high and low-level analysis, as well as provide distinct informative views of the underlying data.

One major target of cyber-attacks is the Domain Name System (DNS) infrastructure, responsible for converting human-understandable URL queries into machine-understandable IP-addresses.The ubiquity and central importance of DNS make it a tempting target for attack and exploitation. If these domain name systems go down, it would create unprecedented chaos and instability on the internet as IP addresses change, caches expire, and queries remain unresolved. Finding, characterizing, understanding, and the mitigation of these attacks on DNS is of utmost importance.

One core aspect of maintaining and defending the DNS is providing DNS analysts with a method of monitoring the queries received. Those DNS analysts that are able to easily comprehend the variety and scope of the queries that pass through their system will be better able to characterize attacks, anomalies, and normal behavior. The challenge is the sheer amount of root DNS traffic which ranges from 100 to 300 GB per server, per DNS letter, per day. The primary systems in use today typically focus on packet counts and origination (source IPs). Modern packet analyzers generally present every aspect of every packet in tabular lists, with query information buried deep in expandable subsections in those lists, or as a single column among many. These techniques, while providing unparalleled detail, do not leverage our innate ability to process spatially organized data to find patterns and anomalies. In addition, they have little emphasis on the aspect

of DNS that makes it so important, the queries themselves. By presenting DNS queries and IP-activity in a spatially and temporally coherent manner, with cross-visualization interactivity, enabling high and low-level investigation, DNS analysts will be able to more effectively process and organize that data. In this chapter, we present our visualization which has been designed to take the vast amounts of root DNS queries, organize them in a spatially comprehensible manner, and facilitate easy investigation to not only answer existing questions, but to help DNS analysts discover new questions. We validate our presented approach on data from one of these 13 root DNS providers, namely the D-Root.

In summary, this chapter makes the following contributions to immersive analytics and visualization for network security:

- We have designed a dual-interactive-visualization system for DNS query and IP data which leverages 3D graphics techniques to convey that data in a novel representation.

- We visualize an order of magnitude more DNS data than previous systems, providing analysts with high-level situational awareness, while preserving low-level details and nuance, without the need for switching between multiple different applications.

- We organize abstract DNS queries in an easy to interpret spatial layout using a deep learning variational autoencoder, such that co-located queries are semantically similar. We also leverage volume rendering to provide analysts with a high-level spatiotemporal understanding of how the distribution of DNS

queries change.

- We identify and characterize distinct DNS anomalies and attacks through an informal empirical evaluation and discussion of discovered trends and clusters with industry experts.

In the following section, we present a review of the challenges, standard practices, as well as present and characterize new techniques.

5.2 Background

Originating in the days of ARPANET, the DNS can be considered as a simple list of host names with their mappings of to and from addresses, maintained in a frequently-updated host table. However, the open nature of DNS makes it vulnerable to a wide variety of attacks and abuses.

The constant attack by malicious sources has necessitated the need for automated intrusion detection systems (IDS).However, many industry operators have observed that modern IDS, although useful, are not optimal or trustworthy [134], in-part due to the large presence of false positives and inability to detect the latest threats [133].Often these systems require a human-in-the-loop to review these detection alerts, and to contextualize the alerts with additional information [135], often manually with separate visualization tools from the IDS [136]. Therefore, visualizations that can provide summary and precise representations of the data is of utmost importance [137]. In the remainder of this section, we review techniques with visualization as a core-component of the analytic and investigation process.

5.2.1 Traditional 2D Network and DNS Visualization

Traditional techniques for visualizing network data include charts (histograms), line-plots (including parallel coordinate views), graphs (including node-link diagrams), among others. The challenge is the enormous and always increasing amount of data to portray. Visualizing all aspects of the data at once is untenable. Tools such as Excel, NetStat [138] and Wireshark [139] (Figure 5.2), outline every aspect of every packet. This gives analysts an unprecedented level of detail, but hinders finding trends, correlations, and anomalies over time [140]. Traditionally, an analyst will write queries to explore their data, leveraging their background knowledge of the dataset. This process is extremely tedious and labor intensive, and generally requires a known starting point [141]. Generally, DNS analysts are interested in monitoring the health of their system, the flows of traffic, patterns, and anomalies.

Histograms are very commonly used for quick analysis of overall trends, such as direct comparisons between adjacent periods of time [142], the count of a particular feature, such as the number and type of alerts [143], and portray counts of packets [142], query types [144], and severity [145]. Histograms are arranged in 1D, by stacking elements to simultaneously show different properties [142], or with curved and circular representations [143], and in 3D [146]where the direction and orientation of the histogram along the z-axis provides additional information.

Similar in function to histograms, line-plots can convey counts over time [145]. One common implementation is parallel-coordinates, used to find botnets [147]and anomalies [146]in DNS traffic by plotting packet attributes along each axis such

as IP-address, time, and attribute counts. Circular representations, such as those used for network intrusion detection [148], can reveal patterns providing what happened, where, and when. Theme rivers, akin to stacked histograms, have been used to visualize changes and anomalies in DNS query traffic [149]. One problem with parallel-coordinate visualizations, including traditional line-plots, are intentional and unintentional obfuscation (Windshield Attacks) [150]. Similarly, as the number of axes and data-points grow, the data elements can self-occlude and hide lingering patterns.

Network graphs, representing IPs, AS, domains, machines, queries, or users, connected via edges (shared traffic, association, or other connections) [133] have been used to visualize communities of hosts in DNS traffic [151], changes in DNS routing and look-up behaviors [152], and anomalous behavior in failed DNS queries [153].While network graphs are useful, previous research [154] found that their effectiveness decreases dramatically if the graph exceeds approximately twenty vertices, limiting their effectiveness for fine-level network analysis.

Many new visualizations leverage TreeMaps [155], which color-code packet counts and anomalies in IP-address bins. Other visualizations correlate geospatial aspects of DNS traffic and overlay packet counts [156,157]. More creative visualizations, such as glTail (http://www.fudgie.org/) and Logstalgia [158] use interactive graphics to render a log file as dynamic 2D simulations.

The previously mentioned visualizations generally trade scalability for fine-level detail, and focus either on high-level summary overviews for large amounts of data, or detailed views for small amounts of data. Therefore many analysts use

multiple tools to gain a complete picture, but this creates an unnecessary context switch and overhead. Additionally, many approaches layout their information with a focus on aesthetic qualities such as maintaining symmetry with uniform glyph positions, potentially compromising latent global and local data structures [159]. In our visualization, we preserve and show both precise and high-level representations for vast amounts of DNS data. Previously, the focus has been on the evolution of source IP packet counts over time, with little to no emphasis on the messages in the packets. The DNS system exists to handle queries, so enabling analysts to explore the changing distributions of queries is of critical importance. Such a visualization would be infeasible using traditional visualizations due to the arbitrary and high dimensionality of the queries, in addition to the irregular behavior of their transmission. Our work portrays a spatiotemporal distribution of packets and queries over time, revealing patterns and anomalies difficult to identify through earlier means.

5.2.2 3D Network Visualization

While 2D visualizations are regarded as easier to create and understand (in terms of time required for comprehension), recent research has shown there are many benefits to 3D visualizations over 2D for abstract data visualization [3–5], including clearer spatial separation, reduced over-plotting, and enabling faster construction and deeper mental models. In addition, many visualizations rely heavily on spatialization, encoding information in the location of data-elements. The addition of a

third dimension makes available more insightful relative positioning [160].

One of the earliest uses of 3D for cyber-security visualization was by Stephen Lau [161]. The visualization uses a 3D scatter-plot to reveal patterns associated with vulnerability scan attacks. To minimize clutter from 2D parallel coordinate visualizations, many are expanding into the third dimension [162]. P3D, a 3D parallel coordinate network security visualization [150] creates multiple 3D planes, each with a set of either IPs, packet counts, ports, or other information along the x and y-axes, with lines connecting these planes representing connections or FTP transfers, to detect port scans while preventing the occlusion attacks such as Port Source Confusion and Windshield Wiper attacks [163]. Another example is Daedalus-Viz [164], which consists of several circular rings, corresponding to various monitored organizations, in orbit around a central sphere representing the complete IPv4 space, with connecting lines indicating the transfer of packets.

One main drawback of the previous systems is the relatively small amount of data they can visualize. We expand upon these ideas by combining elements of scatterplots and parallel coordinate visualizations. Rather than just plotting one element per cell, we interleave multiple data points within a given spatiotemporal cell using 3D transparency, enabling more information to be presented, as well as a direct comparison between similar elements. Lastly, plotting many discrete points temporally increases the overall visual complexity. Instead, we have clumped together spatially coherent groups of points into mesh surfaces to minimize the visual clutter, to reveal structural patterns and changes within the original query point cloud.

5.3 Problem and Solution

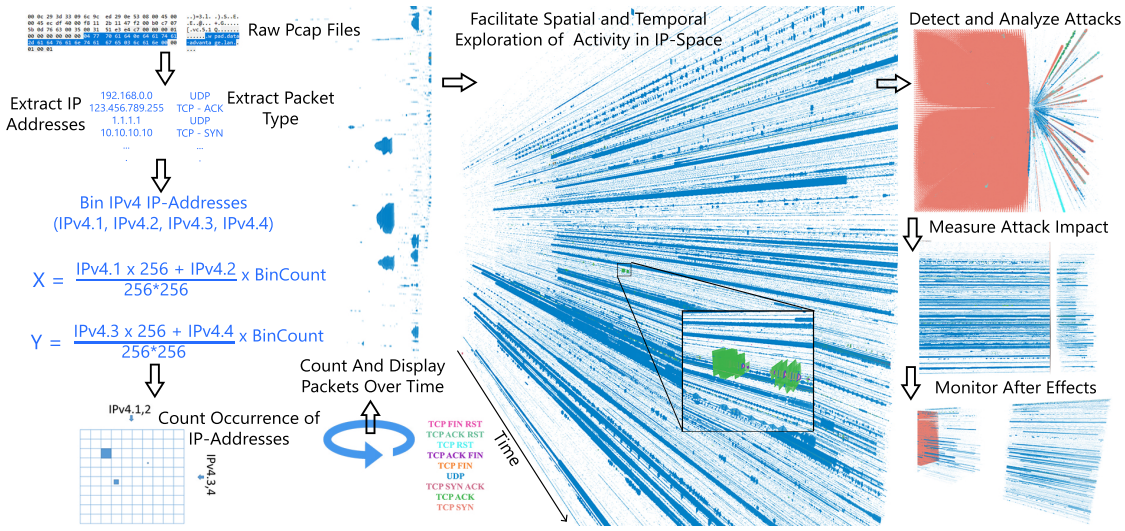


Figure 5.1: Overview of the process from raw pcap files to The Flow-Map IP-Space visualization. Starting from a binary pcap file, we extract and count the occurrence of each IPv4 IP-Address and type of packet. Next, the IPs are converted from a 4D to a 2D grid representation, with glyphs scaled and colored based on the number and type of packets. This process repeats for each time slice, with slices stacked along the z-axis. The result is then visualized using 3D accelerated rendering, which allows for high-level structure and low-level analysis, to help analysts establish a sense of normalcy (central blue image), identify outliers (green TCP burst), classify and characterize attacks (top right), measure attack impacts (middle right), and monitor after effects (lower right).

5.3.1 The Challenge

As part of our development process, we interviewed DNS analyst experts from the University of Maryland D-Root. One of the challenges they face is the scope and enormity of the data they manage. Over the course of an average day at just one of their 131 global facilities, they process over 100 GB of traffic, with a peak traffic around 300 GB. When under attack by a typical DDOS, one server can process roughly 600 GB in one day.

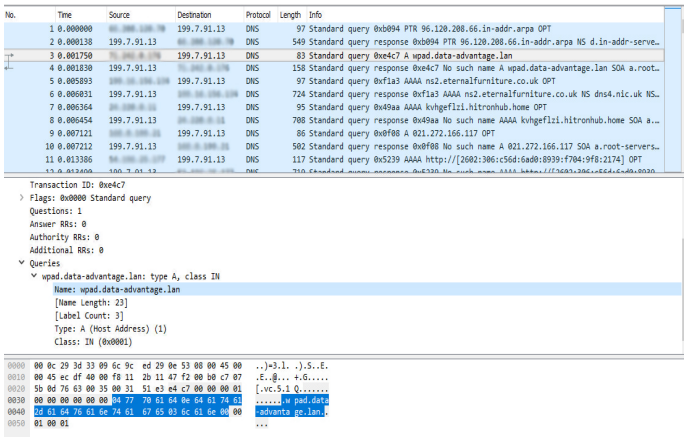


Figure 5.2: An example analysis in Wireshark, a widely used pcap analyzer.

Traditional query visualization tools are very limiting and typically omit the queries and contents of the packets as part of the investigation, emphasizing packet counts and the distribution in the IP-space. As stated earlier, most `pcap` (packet capture) tools present packets line-by-line. An example of queries presented in a commonly used tool (Wireshark [139]) is presented in Figure 5.2. While this level of detail can be very useful, it limits the ability of an analyst to generalize and discover trends due to the lack of a global, visual, and temporal coherence as well as eliciting

a sense of information overload.

There have been many root DNS attacks, historically lasting three [165] and five hours [166]. DDOS attacks outside the realm of the root-DNS on average last less than twenty hours [167].To ensure we can cope with the largest of attacks, we visualize 24 hours of traffic in our case study. However, our visualization is capable of showing larger durations of time.

For the purposes of this chapter, we explore one recent root DNS attack. On June 25th 2016, a moderately sized DDOS attacked all root DNS authorities in a coordinated attack. A report published by the root DNS authorities on this specific attack can be found here (http://root-servers.org/news/events-of-20160625.txt).According to the official report, all DNS root name servers received a high rate of TCP SYN packets in a SYN flood attack for nearly four hours. The source addresses appeared to be randomized and uniformly distributed throughout the IPv4 address space. The observed traffic volume due was up to approximately 10 million packets per second (approximately 17 GB/s), per DNS root name server letter. Our goal is to provide analysts with a sense of normalcy over the course of a day, contextualize attacks when and if they occur, to aid in subsequent investigation and mitigation strategies, and help develop a characterization of current and future attacks for comparison.

Previous network visualizations have visualized up to approximately 350 million packets [152, 168, 169].In our presented visualization, we visualize, in real-time using 3D accelerated rendering techniques, over 2.4 billion packets, consisting of over 487 million unique queries, spanning 24 hours, from the McLean Virginia D-Root

DNS site.

5.3.2 Approach Overview

There were three design considerations driving our development, to display an entire day of query traffic from a Root DNS server, show high-level structures and patterns in an intuitive manner with interactions enabling finer investigation, and display time as a spatial dimension. In this chapter, we present two complementary visualizations of the activity in both the IP and query domains. An overview of the IP-space and query-space construction processes from raw packets to visualization are presented in Figure 5.1 and Figure 5.3.

5.3.3 Flow-Map IP-Space Visualization

The IP-space visualization uses what we call a Flow-Map, spatially presenting information regarding packet counts and types over time. The IP-space consists of 4-octets, resulting in over 4 billion unique values/indexes. To properly visualize such a space would require a four-dimensional cube, or a very tightly indexed 1D histogram. The compromise reached with our DNS experts, to maintain a fine-level of detail without overwhelming the user, as summarized in Figure 5.1, is to reduce the IP space from over 4 billion to $10,000$ values, by combining the first and second octet pairs into a single value. These values are used to bucket (bin) the IP-addresses of the packets into a 2D IP-space. Our current scheme does not take into account autonomous domains. Therefore, some IP addresses that belong to very different

autonomous domains will be sent to the same bucket while IP addresses that belong to the same corporation could be sent to different buckets. It should be possible in future, to add an Autonomous Domain layer above it, showing only those IP-bins belonging to a particular autonomous domain, or by reorganizing the IP-space into an autonomous domain space. Due to irregular packet arrival times, packets are temporally binned into 5 second chunks. Our discussions with DNS experts revealed that knowing the precise IP and arrival time of particular packet was insignificant and that gaining a general understanding of the distribution of packet sources is preferable. A novel characteristic of our visualization is that each bin/cell contains multiple glyphs, seamlessly linked together, sized based on the number of received packets (within a given time it represents), and colored transparently based on the type of packet received (UDP, TCP SYN, etc.). The transparency and sizes of the glyphs can be adjusted to aid in minimizing occlusion for certain views, revealing hidden information, and to emphasize different bins with certain counts of packets. The advantage of this Flow-Map representation is that time is represented as a spatial component, removing temporal animations or scrolling through individual time slices, thereby providing a globally and temporally coherent model for the analyst. Within this constructed visualization, analysts may freely move and rotate their view to get a high-level overview of the space, zoom in close for an in-depth analysis, and change the current transparency and glyph scaling levels using the keyboard and mouse. Hovering the cursor over any given element presents additional information, among which are the range of the IP bin, the number of packets, and the time.

5.3.4 IP-Space Observations

Three general observations can be made using the Flow-Map representation as shown in Figure 5.1. First, most of the space is empty, suggesting that most queries fall into relatively few IP-bins. Second, for most filled bins, the glyphs are relatively small, suggesting that most of the queries received are singletons (customers send just one or a few packets). Third, there are a few persistent high-packet count buckets that send out thousands of queries in just a few seconds. From this IP-space representation, an analyst can grasp the nature of the changing volume of traffic.

Within the first few hours of data, we found a small selection of interesting patterns. First is the anomaly in highlighted in the middle of Figure 5.1 which shows an instance of high TCP-based packet activity, as indicated by the green color. This was a large burst of packets, as indicated by the large size of the glyphs, and was distinct in that no TCP activity preceded or followed this period lasting roughly a minute. In the left Figure 5.4, we have extracted three IP bins that have regularly repeating, self-similar, internal patterns of traffic. This subset of data can be seen near the top-right of the blue Flow-Map in Figure 5.1. From our discussion with D-Root experts, this traffic might be from external monitoring sources, who periodically query the root DNS, resulting in this regular pattern. After a closer look at the queries from these IP-bins in the query-visualization, we found that the received queries were generally of the form `*.trafficmanager.net`. The top-right of Figure 5.1 shows the distribution of IPs used in the TCP-SYN flood attack (half

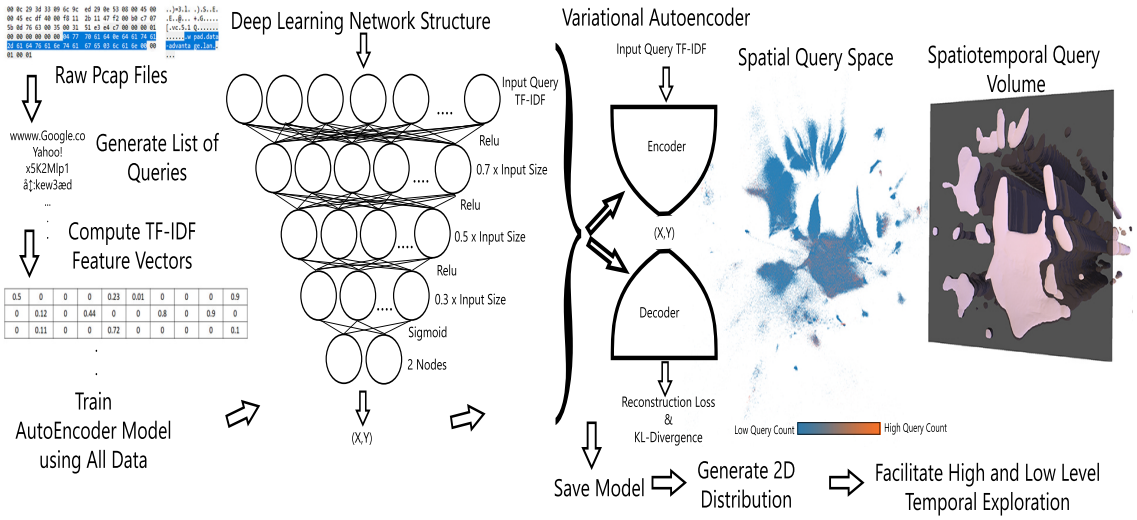


Figure 5.3: A overview of the process from raw pcap files to Query-Space visualization. Starting from a binary pcap file, we extract and count each query. Next, each query is converted to a TF-IDF (Term Frequency, Inverse Document Frequency) character-level feature vector. A deep learning autoencoder is trained using all queries, to generate a visually coherent spatial distribution of queries when projected into a 2D space. This distribution of queries is then visualized using 3D accelerated rendering, which allows for high-level temporal structure (top-right) and low-level query analysis (bottom right).

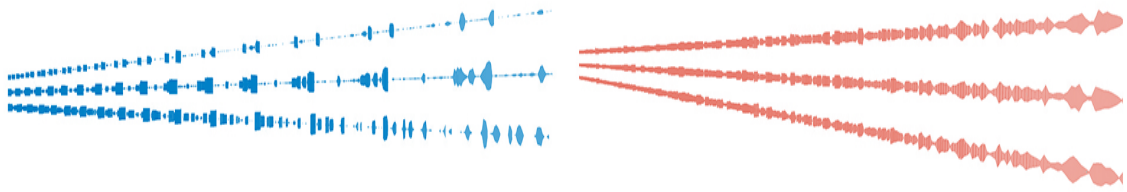


Figure 5.4: Interesting self-similar patterns of intra-IP-bin queries and across-IP-bin traffic (from the TCP-Syn DDOS) over time in D-Root traffic.

of the first IP Octet, and the entirety of the other octets, contrary to the official report indicating the totality of the IPv4 space was spoofed) , the decreased volume of traffic from lost customers, and the resulting hard-drive failure as a result. The right of Figure 5.4 shows a high inter-bin temporal similarity found across all IP-bins involved in the TCP-Syn flood. It is possible that all these characteristics could serve as an attack signature. Thanks to the preservation of the low-level of detail, which would otherwise be summarized or abstracted away by other tools, these interesting patterns and anomalies were identified, driving further investigation.

5.3.5 Deep Learning Driven Query Space Visualization

Our query-space visualization provides analysts with a deeper understanding of the distribution of received queries. We organize the non-spatial queries into a spatial representation, enabling easy detection of patterns and structure from the large amount of data using deep learning. Each query is visualized as a 3D sphere, positioned near similar queries, and colored to indicate the number of times it was received. Using this visualization, an analyst can see the high-level distribution and volume of queries, then drill down to discover the precise queries received and draw observations, panning and zooming the camera with the keyboard or mouse, and obtain details for a specific data-element by hovering over it with the cursor. The goal is to provide high-level information, in the form of natural and easy to interpret geometric categorical structures as generated by deep learning and to facilitate low-level investigation by providing an analyst to get close and personal with the raw

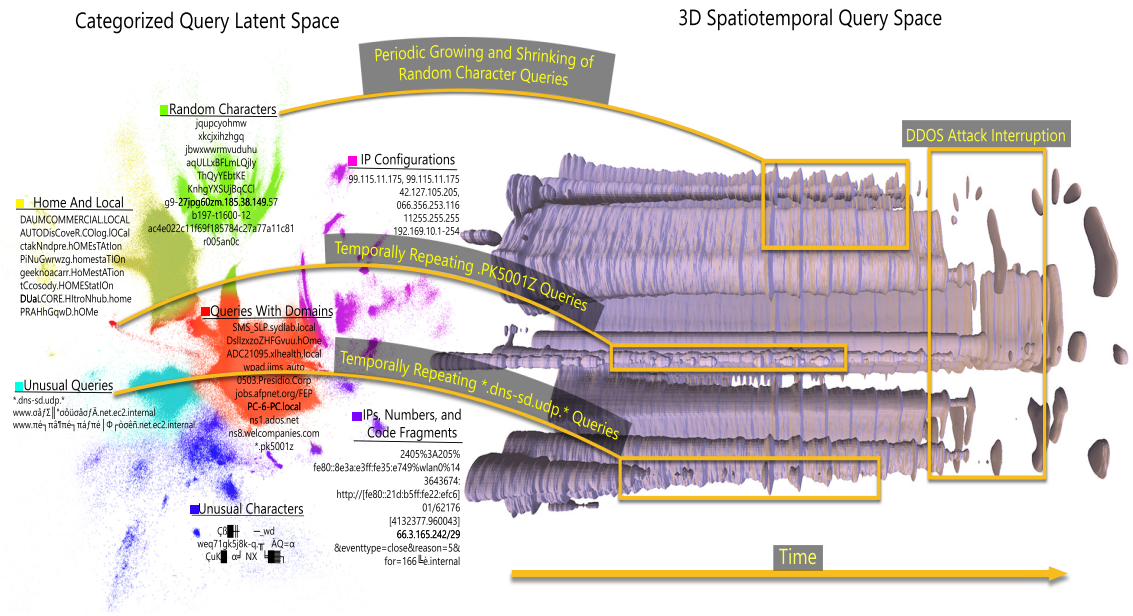


Figure 5.5: In the lower-left, the 2D query-space clustered into semantic query categories. There is a general trend with alphabetic-based queries towards the left, and numeric-based queries the right, along with a general trend to normal characters at the top, and unusual characters at the bottom. The lower-right shows the temporal query visualization, portraying a high-level temporal overview of the distribution of queries. The top reveals a selection of interesting observations, such as, rapidly diminishing groups of queries at the start, temporally repeating groups of queries, the large reduction in queries during the attack, and an overall decrease in the number of queries over time.

query data.

To generate the distribution of queries, we use deep learning to find patterns within the set of queries, and then project those queries into a 2D plane. Previous techniques, such as spectral clustering and PCA were unable to generate adequate representations, less cope with the huge quantities to be projected, due to the re-

quirement of a full similarity matrix between all pairs of elements. In addition, the lack of a ground truth or implicit notion of similarity inhibits the use of other traditional methods. Therefore, we use deep learning, which is able to handle the data in chunks, maintain a global conception of the data-space, and learn its own metric of similarity from the data rather than have one imposed. Each query string is converted to a vector of TF-IDF (Term Frequency, Inverse Document Frequency) values, resulting in a 256-length character feature vector. TF-IDF is commonly used in natural language processing algorithms for converting text into a machine-understandable format by converting each word/letter into a value based on its importance to the local sentence/word relative and the entire corpus. All query feature vectors are used to train an unsupervised variational autoencoder, which learns to project the queries into a latent space, with the constraint that it must then accurately recover the original input feature vector from that projection, resulting in similarly structured queries as spatially co-located. An example of the queries projected into a semantic latent space is presented in the lower-left of Figure 5.5.

The deep learning network uses four decreasingly sized dense layers, to capture interesting co-occurrences of features, each using a Rectified Linear Unit activation function [127] (Relu), defined as $\max(0, x)$, except the last output layer, which uses a Sigmoid, defined as $\frac{1}{1+exp(-x)}$. This network structure is embedded into a Variational Autoencoder, which refines the internal weights. A visual overview of the model is presented in Figure 5.3. The model is trained using all queries as a pre-processing stage before the visualization starts. The libraries TensorFlow [125] and Keras [126] are used to build, train, and process the deep learning network.

A coarse labeling of the query-space visualization with examples of queries within each group are presented in the left of Figure 5.5. There is a general trend with alphabetic-based queries appearing on the left, and numeric-based queries appearing on the right, along with a general usage of normal characters on the top, and unusual characters on the bottom. Through our investigation, we have identified several interesting general groups of queries. The most common form of queries are those primarily composed of random configurations of alphabetical characters. Other discovered groups include configurations of IP addresses and numbers, invalid queries which include distinct domains such as `.com` and `.org`, device names with local and home domains, and more strangely, queries consisting of unusual (non-alphanumeric) characters (most likely binary and code fragments).

5.3.5.1 3D Query Flow Visualization

The enormous scale of the DNS data pushes the limits of dimensionality projection. Earlier solutions would, in general, independently project down segments of the overall high-dimensional data, creating separate local models, resulting in projections that could not be directly compared due to differences in projection axes, or result in a gradual change in axes as more data is projected. Both solutions are undesirable due to the increased cognitive burden of constantly updating one's understanding of the local axes and interpretation of the projection. With deep learning, we are able to remove this effect by training a globally consistent model, providing analysts with a cohesive and consistent representation. In addi-

tion, provided a set of projected points, displaying a time-varying scatter-plot is a challenging problem. A common solution is to use animation to convey time. However, research has shown animation incurs higher cognitive loads, can be often difficult to comprehend, and is generally less effective in communication than a static visualization [170]. Therefore we opted for a static approach for visualizing temporal information similar to the IP-space. However, unlike the IP-space which consists of regular interval IP-addresses, the queries have arbitrary distances between each other, requiring a different visualization method. Another solution is to stack individual 2D scatter-plot time-slices, but for large datasets, this leads to an overwhelming amount of information, as well as a large amount of visual clutter. Therefore, we use a volume-rendered marching cubes approach using CUDA and OpenGL. This view is designed, in contrast to the earlier blanket temporal 2D view, to provide analysts with a structural high-level overview of how the query-space changes over time, naturally revealing temporally repeating, absent, and anomalous queries. Similar to the IP-space visualization, the camera in the temporal query-space visualization may be translated, rotated, and zoomed for closer inspection using the keyboard and mouse. An example of this temporal query visualization is presented in Figure 5.5.

Among many observations, one unexpected finding is that as time continues, the overall number/volume of queries, particularly on the periphery involving unusual queries, decreases. Starting from the beginning of the day, many groups of queries have high volume and variety, in particular, those belonging to the `.pk5001z` group. However, as the day continues, this cluster of queries decreases, with a reg-

ularly repeating increase and decrease in volume. Other groups, such as those consisting of invalid IP-address fragments, start with a high volume of traffic, but quickly drop-off, as shown in the top-left of Figure 5.5.

Another emergent observation is the large quantity of temporally repeating queries, which fade in and out, growing and shrinking in volume and diversity. A large group of these, consisting of random character queries, have temporally aligned high and low points, as shown in the middle of Figure 5.5. Others, such as the aforementioned `.pk5001z` group, along with many groups of unusual character queries, such as the group of queries of the form `*_dns-sd.udp_*`, also share an intermittent and oscillatory emission. The discovery of such patterns would have been almost impossible without such a visualization. Due to the oscillatory nature of these queries, it is therefore likely they originate from an erroneous program.

Lastly, the TCP-Syn flood DDOS, as visually portrayed distinctively from the norm in the IP-space visualization as a large increase in activity in TCP-Syn packets, is measured as an absence of queries in the 3D spatiotemporal visualization, notably as the reduction in traffic before the large empty gap in the right of Figure 5.5. Although we see that some queries are processed, the majority of traffic, particularly on the periphery, has ceased. The large gap, similar to its portrayal in the IP-space, corresponds to a hard-ware failure. In the IP-space visualization, we learned that there was an overall reduction in the number of queries after the attack. In this view, we can also see that only some of the traffic has returned, but now we are informed that primarily those queries consisting of domains are processed, and unexpectedly, those groups of queries on the periphery are mostly absent.

5.3.6 Dual IP-Query Visualization Interaction

In contrast to previous cyber-visualizations, which focus primarily on the counts of packets, we present a visualization capable of providing analysts with a well-rounded and complete representation of the DNS data. This involves a dual representation, namely an IP and query space. In our system, the IP-space view is presented on the left of the screen, with the query-space shown on the right. Users simply move their cursor from one view to the other to direct their input focus. Until now we have focused on the construction, interaction, and discoveries made with these visualizations independently. In this section we review the interactions and discoveries made when analyzing the D-Root DNS data in a dual-representation.

Suppose the analyst desires to know from which IPs and times a particular query originated. In the query-space visualization, an analyst can double-click on the particular query, to highlight the corresponding bins and times in the IP-space view. We are not filtering out the response packets from the D-Root DNS, so most queries will have at least two IP-space occurrences. To select multiple queries, an analyst holds control+click to brush-select the queries. One selection of queries, presented in the top-left of Figure 5.6 originate from a wide range of IP-bins and time periods. The selected queries are of the form `*.pk5001z`, whose occurrence in the query-space visualization surprised our DNS experts. Initially, they thought these queries were from one or a few IPs (old hardware), but in the dual interaction, we see that these packets come a wide range of IP addresses for a long duration of time. At the bottom-left of Figure 5.6, the selected group of queries arrives in

three IP-bins, one corresponding to the D-Root, and the others corresponding to sources, indicating that these queries are an anomaly due to the small number of sources. Interestingly, one source is consistent, with the other intermittent. One query was "211.67.67.217.", with the remainder of a similar structure. Using our visualization and interaction methodology, analysts may also to do the reverse, and select individual IP-Time-Bins and visualize the corresponding queries, or select an entire duration by double-clicking the IP bin, as shown in the bottom-right of Figure 5.6, revealing this temporally oscillating IP-bin (the middle-left from Figure 5.4 primarily consists of `*.trafficmanager.net` queries. The developed visualizations along with the dynamic interactions enable analysts to visually identify new behavior, develop hypotheses, and overall gain a deeper understanding of the network flow.

5.4 Empirical Validation

The primary motivation of our visualization is to help analysts understand the distribution of queries they receive, how it changes over time, identify anomalous behavior, and help them explore new questions. Throughout our investigation, we found and reviewed several trends and anomalies using our visualization. We present a few insights that were discovered by three DNS experts, who will be named A, B, and C. The discoveries made would have been difficult, if not impossible, to make through traditional analysis tools that lack organizing DNS information in an intuitive spatial and temporal representation. The discoveries presented here

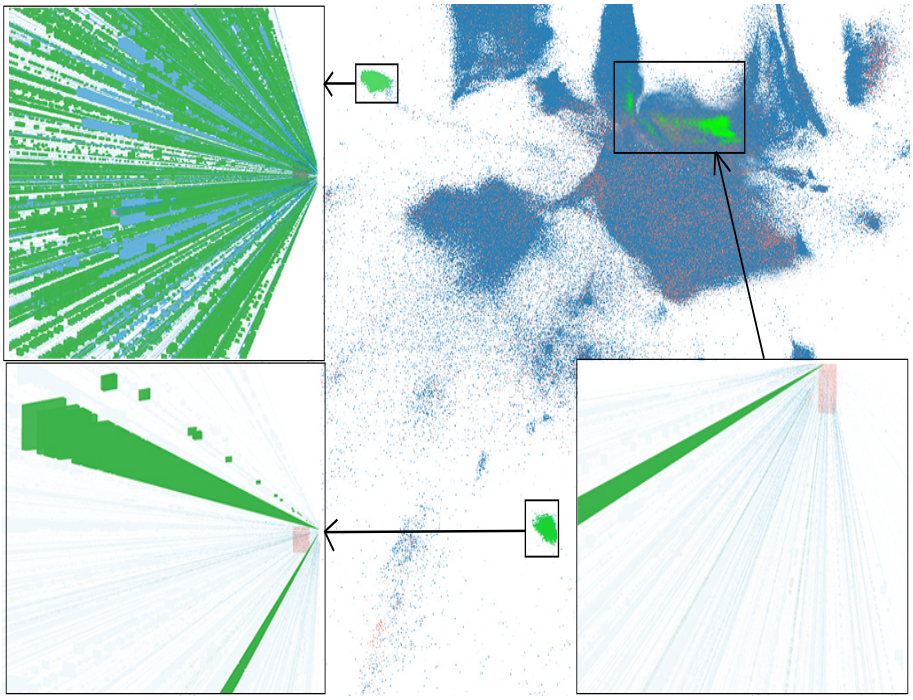


Figure 5.6: Two selected regions of queries. The top selection indicates the queries originate from a wide range of IPs, while the bottom selection indicates those queries came from very few IPs. Non-included IPs may be set transparent (bottom) or left un-transparent(top). The bottom-right image shows the resulting queries included from an entire IP-bin.

originate from multiple joint discussions and sessions with both the authors and DNS experts engaging with the visualizations.

5.4.1 DNS Expert A

DNS expert A stated that they would often monitor the overall health with a `pcap` analysis program, and look at a small random selection of packets to see how those queries looked. As expert A was interacting with our visualization, he

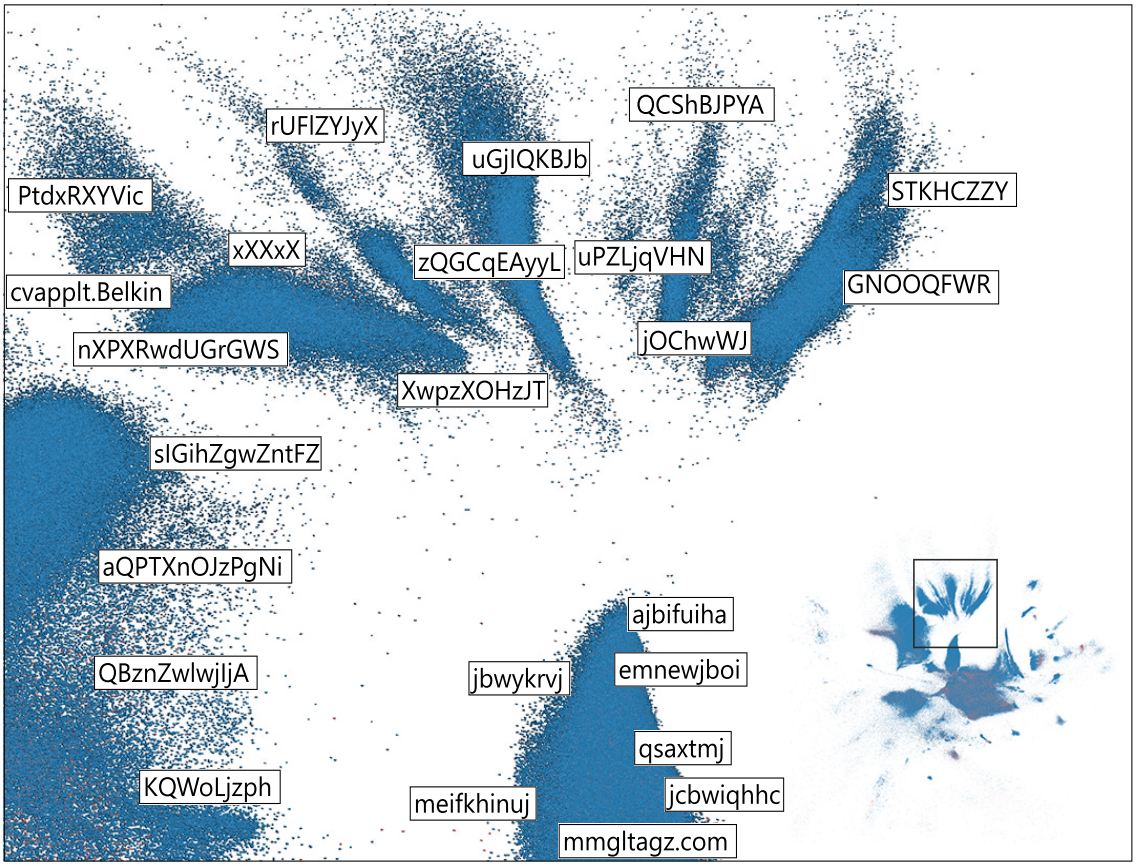


Figure 5.7: The region of the query-space consisting of different distributions of random characters.

zoomed into a cluster of points and noted that there was a high volume of lower-case random character queries. He pointed out it was interesting that our visualization could cluster such queries which could come from the Chrome internet browser. Upon further discussion, we learned that when Chrome starts, it tries to learn the nature of the DNS it sits behind by issuing multiple random queries, as ISPs tend to wildcard DNS servers to catch all domains and load advertisements. If the result of the random queries is a valid response, then Chrome knows something is playing strange with the DNS. For those who do not sit behind one of these particular

ISPs, these random queries end up at the root to be resolved. While we cannot attribute all of the random lower-case queries to Chrome, it is likely responsible for a large majority. In our visualization, there is a large chunk of the distribution space dedicated solely to sets of random characters, with small differences between them (typically the frequency and capitalization of individual letters), as can be seen in Figure 5.7.

In addition, DNS expert A was able to learn that many random character queries contained valid domains. When the root encounters such queries, it forwards those queries to the authority domain listed as part of the query. This could lead to a kind of DDOS attack from re-directed queries. With this new information and our visualization, it may be possible to establish filters to mitigate the effects of such queries.

Lastly, DNS expert A noticed a large collection of queries from different routers and modems. Human error or malfunctioning machines often result in erroneous queries. One example of this unusual behavior was the presence of a large number of queries of the form `*.pk5001z`. This initially struck our DNS experts as very unusual, and after some investigation on their end, they found that these types of requests are typically associated with a particular model of modems, namely the PK5001Z flavor of modem. The presence of these queries at the root indicates that someone somewhere has a miss-configured or infected modem sending erroneous queries.In addition, there was an entire distinct cluster dedicated to queries of the form `*.Home`, `*.Belkin`, and `*.local`, indicating erroneous configurations of home routers and devices. The presence of router and modem based queries, while known

to our analysts, surprised them by their variety and the age of the originating hardware. In particular, DNS expert A found a set of queries belonging to a 20 year old version of OS VXworks. Using our visualization tool, our DNS experts have been informed on the scope of this problem, and that this traffic can lead to intense bursts when an outdated system desperately searches for a valid DNS response.

5.4.2 DNS Expert B

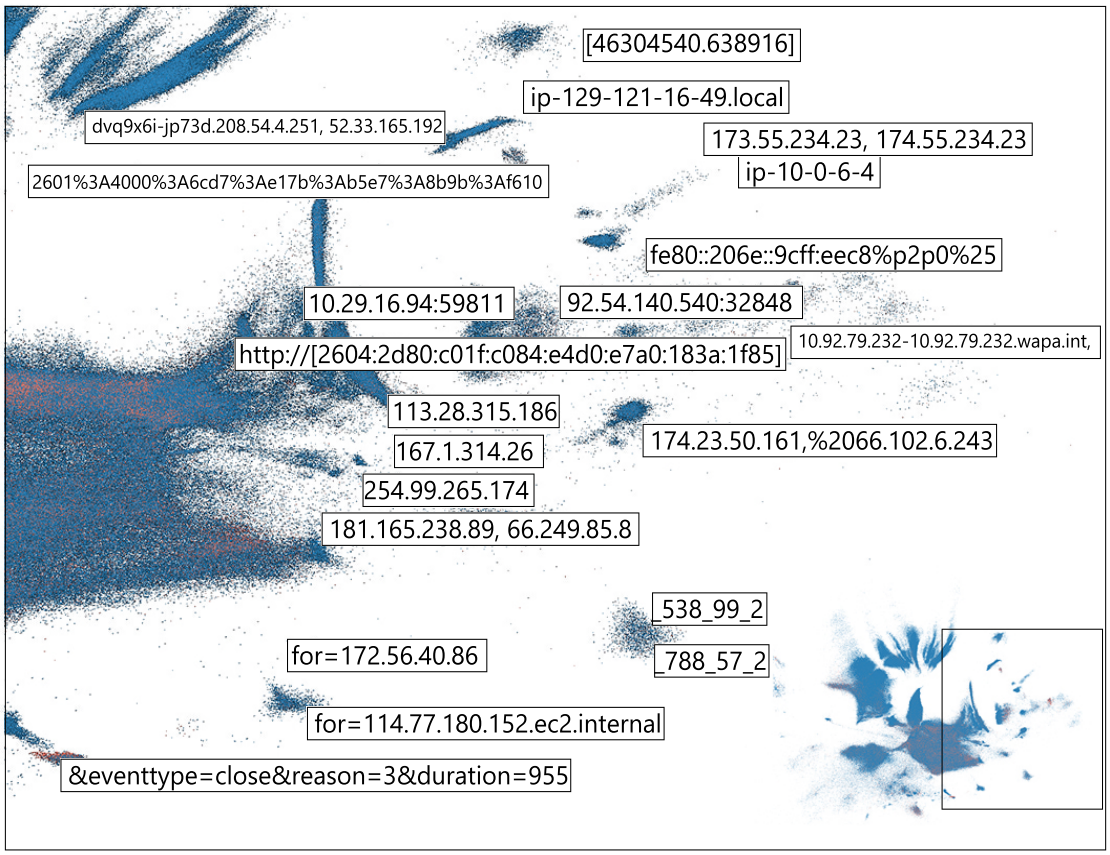


Figure 5.8: The region of the query-space consisting of different distributions of IP addresses, fragments, and expressions.

Rarely, people directly enter an IP address into the web-browser to directly

connect to a specific IP-enabled device. However, expert B noted queries containing IP addresses, which often contain different mistakes. Our visualization is able to identify and cluster these mistakes. Previously, IP-based investigations primarily use the source IP in the packet header, rather than look at IPs in the query itself. Using our visualization, we can see the distribution of queried IPs and the mistakes made when querying them. DNS expert B found that the most common mistake is the usage of an invalid IP-octet (> 255), or an incorrect port address, often using brackets, dashes, or parenthesis to delineate port. More elaborate mistakes include entering too few or too many octets, surrounding the IPv4 address with brackets and other formatting, or enter two or more IP addresses at once, separated in many different ways. Other errors include partial URLs followed or preceded by IP addresses, erroneous bit masks, IP addresses which replace different numbers with letters (perhaps in an attempt to use IPv6), strange hybrid combinations of URLs with IPv6 IPs, generally in the form of `http://`, and IPs containing many percent symbols, perhaps in an attempt to use a regular expression, or as a fragment from a `printf` statement. An instance could be programs erroneously copying code or URL fragments into a browser DNS query packet. As a result, many of these queries reach the root DNS. In our query-space visualization, there are a few clusters dedicated to these types of queries, as shown in Figure 5.8.

In addition, we also often find command and instruction segments or simple statements, such as a large occurrence of `for=` statements, boolean expressions, and variable assignments. For other queries, we find many instances of queries structured as `www.` followed by a random collection hexadecimal and unusual characters.

We believe these are instances of broken applications going through random permutations of URLs trying to resolve to a valid response.In addition, sending commands through DNS is a common way to control bot-nets. Learning of the occurrence and distribution of these queries have reinforced their belief that the majority of traffic they receive is machine rather than human generated. Just as with the random characters, knowing the types of queries containing IP addresses, and knowing that they cannot be resolved, would allow automatic filtering of such traffic earlier.

5.4.3 DNS Expert C

Expert C noted that there were a large number of queries that contain unusual characters as shown in Figure 5.9. These characters are those that cannot be interpreted by normal ASCII.Therefore, for our parsing purposes, we relied on the ISO/IEC 8859-1:1998 (also known as Latin-1) character encoding to properly decode and display the queries. The very existence of these queries is unusual, as people do not generally perform queries using such characters. Expert C has theorized that these queries are data and code binary fragments, likely erroneously copied from an invalid buffer. Another theory is they consist of exfiltrated data exploiting the DNS system (DNS tunneling). Possible examples of this were the large occurrence of long queries containing sequences of `prodID=` and other delineated information. Another source of these unusual characters could be bad Unicode translation in software applications. In our IP and temporal query-space visualization, these particular sets of queries tend to fluctuate depending on the hour of the day (with many only being

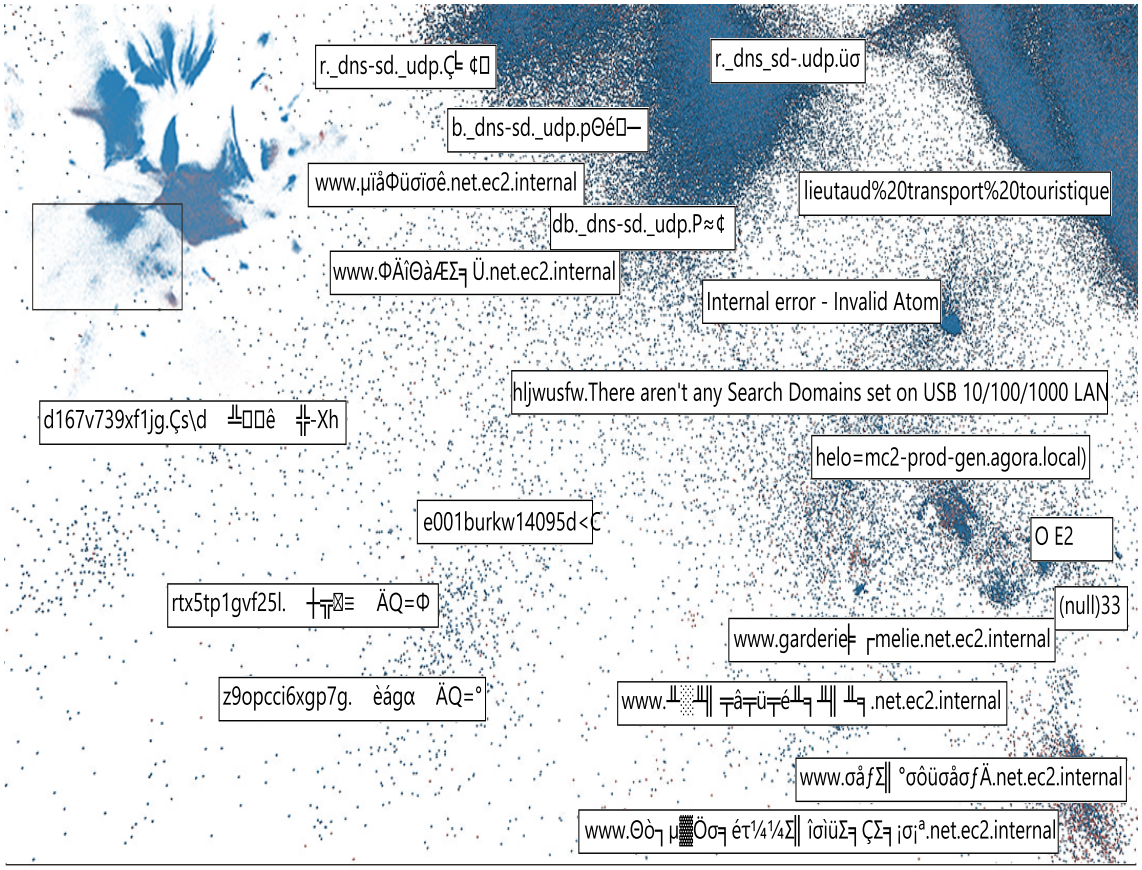


Figure 5.9: The region of the query-space consisting of unusual characters and queries.

issued in the early morning and late at night as indicated by our temporal query visualization), suggesting that a machine is likely the initiator. Expert C noted that these observations would have been very difficult to make without the usage of our visualization.

5.5 Conclusions

The goal of our visualization was to provide a natural and easy to use interface for working with large amounts of real-world DNS IP and query data, for providing

analysts with a general overview of the distribution of the packet traffic and queries, while also allowing them to investigate small temporal events, individual queries, and find correlations between the IP and query spaces. We have shown that using deep learning to generate spatial representation of non-spatial queries is a very effective method of presenting such data. By working closely with real-world root DNS experts, we have been able to find new and interesting anomalies, groups, and patterns that were previously unknown, and have led to further investigation. As the internet of things is set to grow exponentially, the number of erroneous, malformed, and junk queries is set to explode, as well as increasing complexity and scale of future attacks. Having knowledge of the different types of queries and packet behaviors, what they tend to look like, and how they change over time, may allow for DNS analysts to start automatically filtering these packets as the traffic gradually increases, to keep operations functioning normally. Prior to our visualization, the DNS experts would often only look at a handful of queries at a time, not fully grasping the variety and dynamics of the queries flowing through their network. With this new knowledge and capability, closer inspections of their vast quantities of DNS data may now be conducted, and a greater preparedness for the future may now begin with greater confidence.

Chapter 6: Conclusions and Future Work

6.1 Conclusions

In this dissertation, we have presented our research towards enabling visual analytics within virtual environments. We found that the use of virtual memory palaces with head-mounted displays improves recall accuracy as compared to traditional desktops. We had 40 participants memorize and recall faces on two display-interaction modalities for two virtual memory palaces, with two different sets of faces. The HMD condition was found to have a statistically significant 8.8% improvement in recall accuracy compared to the desktop condition. Given the results of our user study, we believe that virtual memory palaces offer us a fascinating insight into how we may be able to organize and structure large information spaces and navigate them in ways that assist in superior recall.

We presented our findings of a user study with the goal of continuously measuring and quantifying cybersickness. Using an EEG, the recorded participant data was decomposed using ICA to separate the underlying sources of the brainwave activity and eliminate noise. The independent components were then clustered across users for the purposes of comparing the EEG of those grouped users. Through independent component analysis and time-frequency spectral analysis, our findings

suggest that a spectral power increase in the Delta, Theta, and Alpha frequency bands, relative to a baseline, strongly correlates to the presence of cybersickness.

We presented an approach to facilitate the discovery of latent structures and communities/labels within high-dimensional non-spatial datasets using deep learning. Through an iterative process, an analyst can select and assign points to new or existing communities, which is then used in conjunction with deep learning, to refine the 2D spatial distribution of the points, revealing more new information. We have shown on three different datasets that our technique is able to reconstruct hidden structures and communities, enabling both deep learning and the human analyst to support, refine, and enhance each other through visualization.

We developed a visualization for working with large amounts of real-world DNS IP and query data. Our goal was to provide analysts with a general overview of the distribution of the packet traffic and queries, while also allowing them to investigate large and small temporal events, individual queries, and find correlations between the IP and query spaces. We have shown that using deep learning to generate spatial representation of non-spatial queries is a very effective method of presenting such data. By working closely with real-world root DNS experts, we have been able to find new and interesting anomalies, groups, and patterns that were previously unknown, and have led to further investigation.

6.2 Future Work

Virtual reality has recently taken center stage in the graphics and entertainment world. The potential possibilities for virtual reality are endless, including applications spanning entertainment, engineering, medicine, communication, and others. The DNS visualization research presented in this dissertation is just one of many real-world applications where analysts struggle to handle the vast and ever-increasing amounts of data. The amount of data each of us creates, processes, and interacts with everyday is immense. Visualizing such large amounts of data and internet traffic all at once, providing analysts with a complete situational awareness, would traditionally call for a large display. Traditionally, this would mean using more monitors or using larger displays with more screen real-estate. A radical shift away from increasing the display dimensions for traditional visualizations is necessary. Instead of expanding the sizes of our displays for visualizing large amounts of information, it would be interesting if smaller displays, such as head-mounted displays and virtual reality, could be used to visualize the same or more information. Our hope is that the work presented here is a first step towards realizing this goal.

6.3 Peer Reviewed Publications

- Eric Krokos, Kirsten Whitley, and Amitabh Varshney, "Visual Analytics for Root DNS Data", *IEEE Symposium on Visualization for Cyber Security (VizSec 2018)*, Berlin, Germany, October 2018, (accepted for publication, September 2018).

- Eric Krokos and Amitabh Varshney, "Interactive Characterization of Cybersickness in Virtual Environments using EEG", *Virtual Reality*, (submitted September 2018).

- Eric Krokos, Hsueh-Chien Chen, Jessica Chang, Celeste Lyn Paul, Bohdan Nebesh, Kirsten Whitley, and Amitabh Varshney, "Enhancing Deep Learning with Visual Interactions", *ACM Transactions on Interactive Intelligent Systems*, (accepted for publication, July 2018).

- Eric Krokos, Catherine Plaisant, and Amitabh Varshney, "Virtual memory palaces: immersion aids recall", *Virtual Reality* (May 2018): 1-15.

- Eric Krokos, Catherine Plaisant, and Amitabh Varshney, "Spatial Mnemonics using Virtual Reality", *In Proceedings of the 10th International Conference on Computer and Automation Engineering (ICCAE 2018)*, pp 17-30, presented at Brisbane, Australia, February 24, 2018.

- Hsueh-Chien Cheng, Antonio Cardone, Somay Jain, Eric Krokos, Kedar Narayan, Sriram Subramaniam, and Amitabh Varshney, "Deep-learning-assisted Vol-

ume Visualization", *IEEE Transactions on Visualization and Computer Graphics* (accepted for publication, January 2018).

- Hsueh-Chien Cheng, Antonio Cardone, Eric Krokos, Bogdan Stoica, Alan Faden, and Amitabh Varshney, "Deep-learning-assisted visualization for live-cell images", *In Proceedings of the IEEE International Conference on Image Processing (ICIP 2017)*, September 2017, pp 1377-1381, IEEE.

- Eric Krokos, Hanan Samet, and Jagan Sankaranarayanan, "A look into Twitter hashtag discovery and generation", *Proceedings of the 7th ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, November 2014, pp 49-56, ACM.

Table 1: Trend Scores for each face for Face Set 1 from Google Trends, with the average trend score of 30.5 and standard deviation of 21.86. The data was collected from April, May, June, and July of 2015.

FaceSet1	APR	MAY	JUN	JUL	AVG
Martin Luther King	14	14	10	8	11.5
Bill Gates	48	50	48	47	48.25
Mahatma Gandhi	57	55	57	54	55.75
Donald Duck	66	71	71	65	68.25
Buzz Lightyear	37	38	36	41	38
George Washington	21	21	15	15	18
George Bush	2	2	2	2	2
Oprah Winfrey	13	12	10	12	11.75
Taylor Swift	59	79	69	68	68.75
Steve Jobs	2	3	2	3	2.5
Michael Jackson	3	3	4	3	3.25
Harry Potter	6	7	8	11	8
Stephen Hawking	43	36	31	30	35
Mona Lisa	38	38	31	29	34
Shrek	9	10	9	9	9.25
Frodo Baggins	19	18	19	17	18.25
Albert Einstein	44	43	39	34	40
Vladimir Putin	36	31	27	22	29
Galileo Galilei	34	35	32	35	34
King Louis XVI	65	73	60	56	63.5
Napoleon Bonaparte	42	44	46	34	41.5

Table 2: Trend Scores for each face for Face Set 2 from Google Trends, with the average trend score of 29.83 and standard deviation of 18.32. The data was collected from April, May, June, and July of 2015.

FaceSet2	APR	MAY	JUN	JUL	AVG
Abraham Lincoln	39	35	26	25	31.25
Katy Perry	37	37	35	34	35.75
Hillary Clinton	32	11	12	13	17
Arnold Schwarzenegger	25	25	34	39	30.75
Tom Cruise	17	16	15	28	19
Batman	27	27	29	37	30
Mickey Mouse	76	75	73	78	75.5
Marilyn Monroe	49	56	64	45	53.5
Testudo	2	2	2	3	2.25
Winston Churchill	48	50	38	36	43
Barbie	42	42	44	45	43.25
Mark Zuckerberg	21	20	18	19	19.5
Robin Williams	2	1	1	2	1.5
Dalai Lama	26	26	32	36	30
Kim Jong-un	20	30	17	16	20.75
Harrison Ford	21	15	12	15	15.75
Bill Clinton	22	18	14	14	17
Michelle Obama	8	6	8	9	7.75
Queen Victoria	48	55	42	40	46.25
Cleopatra	56	52	50	51	52.25
Nikola Tesla	33	36	32	37	34.5

Table 3: Angular Resolution of Faces in the Town and Palace scenes, with the average and standard deviation of the angular resolutions of the set of faces for each scene. The difference in angular resolutions between the two scenes was not statistically significant with ($p = 0.44 > 0.05$).

Face Number	Town	Palace
1	6.61	7.86
2	7.29	7.86
3	5.72	7.81
4	6.92	7.81
5	7.29	8.02
6	5.83	8.02
7	7.81	4.73
8	7.81	5.46
9	7.60	5.46
10	5.98	4.73
11	6.04	6.51
12	7.44	6.40
13	4.94	4.94
14	4.68	4.94
15	7.44	6.09
16	4.37	6.09
17	5.26	5.46
18	4.53	5.36
19	7.81	7.55
20	5.72	6.35
21	7.81	6.35
Average	6.42	6.37
Standard Dev	1.17	1.16



Figure 1: Face Set 1, containing 21 faces.



Figure 2: Face Set 2, containing 21 faces.

Bibliography

[1] Eugenia M Kolasinski. Simulator sickness in virtual environments. Technical report, DTIC Document, 1995.

[2] Sue VG Cobb, Sarah Nichols, Amanda Ramsey, and John R Wilson. Virtual reality-induced symptoms and effects (VRISE). *Presence*, 8(2):169–186, 1999.

[3] Jorge Poco, Ronak Etemadpour, Fernando Vieira Paulovich, TV Long, Paul Rosenthal, Maria Cristina Ferreira de Oliveira, Lars Linsen, and Rosane Minghim. A framework for exploring multidimensional data with 3d projections. In *Computer Graphics Forum*, volume 30, pages 1111–1120. Wiley Online Library, 2011.

[4] Monica Tavanti and Mats Lind. 2d vs 3d, implications on spatial memory. In *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on*, pages 139–145. IEEE, 2001.

[5] Antonio Gracia, Santiago González, Víctor Robles, Ernestina Menasalvas, and Tatiana Von Landesberger. New insights into the suitability of the third dimension for visualizing multivariate/multidimensional data: A study based on loss of quality quantification. *Information Visualization*, 15(1):3–30, 2016.

[6] Joseph J LaViola Jr. A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin*, 32(1):47–56, 2000.

[7] Sharon R Holmes and Michael J Griffin. Correlation between heart rate and the severity of motion sickness caused by optokinetic stimulation. *Journal of Psychophysiology*, 15(1):35, 2001.

[8] Fabio Dell'Acqua, Paolo Gamba, and Alessio Ferrari. Exploiting spectral and spatial information for classifying hyperspectral data in urban areas. In *Geoscience and Remote Sensing Symposium, 2003. IGARSS'03. Proceedings. 2003 IEEE International*, volume 1, pages 464–466. IEEE, 2003.

[9] David A Landgrebe. *Signal theory methods in multispectral remote sensing*, volume 29. John Wiley & Sons, 2005.

[10] Jaynes Julian. The origin of consciousness in the breakdown of the bicameral mind. *gli Adelphi*, 1976.

[11] Henry L Roediger. Implicit and explicit memory models. *Bulletin of the Psychonomic Society*, 13(6):339–342, 1979.

[12] Markus Knauff. *Space to reason: A spatial theory of human thought*. MIT Press, 2013.

[13] Frances Amelia Yates. *The art of memory*, volume 64. Random House, 1992.

[14] Joshua Harman. Creating a memory palace using a computer. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, pages 407–408, 2001.

[15] Robert Godwin-Jones. Emerging technologies from memory palaces to spacing algorithms: approaches to second-language vocabulary learning. *Language, Learning & Technology*, 14(2):4, 2010.

[16] John D Mayer, Peter Salovey, David R Caruso, and Gill Sitarenios. Emotional intelligence as a standard intelligence. *Emotion*, 1(3):232–242, 2001.

[17] Howard Gardner. *Multiple intelligences: New horizons*. New York: Basic books, 2006. ISBN: 978-0465047680.

[18] Duncan R Godden and Alan D Baddeley. Context-dependent memory in two natural environments: On land and underwater. *British Journal of psychology*, 66(3):325–331, 1975.

[19] Richard Skarbez, Frederick P. Brooks, Jr., and Mary C. Whitton. A survey of presence and related concepts. *ACM Comput. Surv.*, 50(6):96:1–96:39, November 2017.

[20] Mel Slater. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 364(1535):3549–3557, 2009.

[21] Claudia Repetto, Silvia Serino, Manuela Macedonia, and Giuseppe Riva. Virtual reality as an embodied tool to enhance episodic memory in elderly. *Frontiers in Psychology*, 7:1839:1–1839:4, 2016.

[22] Lawrence W Barsalou. Grounded cognition. *Annu. Rev. Psychol.*, 59:617–645, 2008.

[23] Lawrence Shapiro. *Embodied cognition*. Routledge, 2010. ISBN: 978-0415773423.

[24] Tamas Madl, Ke Chen, Daniela Montaldi, and Robert Trappl. Computational cognitive models of spatial memory in navigation space: A review. *Neural Networks*, 65:18–43, 2015.

[25] Stefan Leutgeb, Jill K Leutgeb, May-Britt Moser, and Edvard I Moser. Place cells, spatial maps and the population code for memory. *Current opinion in neurobiology*, 15(6):738–746, 2005.

[26] György Buzsáki and Edvard I Moser. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nature neuroscience*, 16(2):130–138, 2013.

[27] Edvard I Moser, Emilio Kropff, and May-Britt Moser. Place cells, grid cells, and the brain's spatial representation system. *Annual review of neuroscience*, 31, 2008.

[28] Neil Burgess. Spatial cognition and the brain. *Annals of the New York Academy of Sciences*, 1124(1):77–97, 2008.

[29] Oliver Baumann and Jason B Mattingley. Medial parietal cortex encodes perceived heading direction in humans. *Journal of Neuroscience*, 30(39):12897–12901, 2010.

[30] Colin Lever, Stephen Burton, Ali Jeewajee, John O'Keefe, and Neil Burgess. Boundary vector cells in the subiculum of the hippocampal formation. *Journal of Neuroscience*, 29(31):9771–9777, 2009.

[31] Arne D Ekstrom, Michael J Kahana, Jeremy B Caplan, Tony A Fields, Eve A Isham, Ehren L Newman, and Itzhak Fried. Cellular networks underlying human spatial navigation. *Nature*, 425(6954):184–188, 2003.

[32] Tom Hartley, Colin Lever, Neil Burgess, and John O'Keefe. Space in the brain: how the hippocampal formation supports spatial cognition. *Phil. Trans. R. Soc. B*, 369(1635):20120510, 2014.

[33] Caswell Barry, Colin Lever, Robin Hayman, Tom Hartley, Stephen Burton, John O'Keefe, Kate Jeffery, and N Burgess. The boundary vector cell model of place cell firing and spatial memory. *Reviews in the Neurosciences*, 17(1-2):71–98, 2006.

[34] Jangjin Kim, Sébastien Delcasso, and Inah Lee. Neural correlates of object-in-place learning in hippocampus and prefrontal cortex. *Journal of Neuroscience*, 31(47):16991–17006, 2011.

[35] Malcolm W Brown and John P Aggleton. Recognition memory: what are the roles of the perirhinal cortex and hippocampus? *Nature Reviews Neuroscience*, 2(1):51–61, 2001.

[36] V Hok, E Save, PP Lenck-Santini, and B Poucet. Coding for spatial goals in the prelimbic/infralimbic area of the rat frontal cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 102(12):4602–4607, 2005.

[37] Eric LG Legge, Christopher R Madan, Enoch T Ng, and Jeremy B Caplan. Building a memory palace in minutes: Equivalent memory performance using virtual versus conventional environments with the method of loci. *Acta psychologica*, 141(3):380–390, 2012.

[38] E Fassbender and W Heiden. The virtual memory palace. *Journal of Computational Information Systems*, 2(1):457–464, 2006.

[39] Doug A Bowman and Ryan P McMahan. Virtual reality: how much immersion is enough? *Computer*, 40(7):36–43, 2007.

[40] Ajith Sowndararajan, Rongrong Wang, and Doug A. Bowman. Quantifying the benefits of immersion for procedural training. In *Proceedings of the 2008 Workshop on Immersive Projection Technologies/Emerging Display Technologiges*, IPT/EDT '08, pages 2:1–2:4, 2008.

[41] Eric D Ragan, Ajith Sowndararajan, Regis Kopper, and Doug A Bowman. The effects of higher levels of immersion on procedure memorization performance and implications for educational virtual environments. *Presence: Teleoperators and Virtual Environments*, 19(6):527–543, 2010.

[42] Randy Pausch, Dennis Proffitt, and George Williams. Quantifying immersion in virtual reality. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 13–18, 1997.

[43] Roy A Ruddle, Stephen J Payne, and Dylan M Jones. Navigating large-scale virtual environments: what differences occur between helmet-mounted and desktop displays? *Presence: Teleoperators and Virtual Environments*, 8(2):157–168, 1999.

[44] Katerina Mania, Tom Troscianko, Rycharde Hawkes, and Alan Chalmers. Fidelity metrics for virtual environment simulations based on spatial memory awareness states. *Presence: Teleoperators and Virtual Environments*, 12(3):296–310, 2003.

[45] Joel Harman, Ross Brown, and Daniel Johnson. Improved memory elicitation in virtual reality: New experimental results and insights. In *IFIP Conference on Human-Computer Interaction*, pages 128–146. Springer, 2017.

[46] Frederick P Brooks Jr, John Airey, John Alspaugh, Andrew Bell, Randolph Brown, Curtis Hill, Uwe Nimscheck, Penny Rheingans, John Rohlf, Dana Smith, Douglass Turner, Amitabh Varshney, Yulan Wang, Hans Weber, and Xialin Yuan. Six generations of building walkthrough: Final technical report to

the National Science Foundation. 1992. TR92-026, Department of Computer Science, University of North Carolina at Chapel Hill.

[47] Barbara M Brooks. The specificity of memory enhancement during interaction with a virtual environment. *Memory*, 7(1):65–78, 1999.

[48] Anthony E Richardson, Daniel R Montello, and Mary Hegarty. Spatial knowledge acquisition from maps and from navigation in real and virtual environments. *Memory & Cognition*, 27(4):741–750, 1999.

[49] Maryjane Wraga, Sarah H Creem-Regehr, and Dennis R Proffitt. Spatial updating of virtual displays. *Memory & Cognition*, 32(3):399–415, 2004.

[50] Simon T. Perrault, Eric Lecolinet, Yoann Pascal Bourse, Shengdong Zhao, and Yves Guiard. Physical loci: Leveraging spatial, object and semantic memory for command selection. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 299–308, 2015.

[51] 3DMarko. Medieval town 01 dubrovnik, 2011.

[52] 3DMarko. Palace interior 02, 2014.

[53] John E Harris. Memory aids people use: Two interview studies. *Memory & Cognition*, 8(1):31–38, 1980.

[54] Jennifer A McCabe. Location, location, location! Demonstrating the mnemonic benefit of the method of loci. *Teaching of Psychology*, 42(2):169 – 173, 2015.

[55] George A Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[56] Alan D Baddeley and Graham Hitch. Working memory. *Psychology of Learning and Motivation*, 8:47–89, 1974.

[57] Richard C Atkinson and Richard M Shiffrin. Human memory: A proposed system and its control processes. *Psychology of learning and motivation*, 2:89–195, 1968.

[58] Katerina Mania and Alan Chalmers. The effects of levels of immersion on memory and presence in virtual environments: A reality centered approach. *CyberPsychology & Behavior*, 4(2):247–264, 2001.

[59] Jack M Loomis, James J Blascovich, and Andrew C Beall. Immersive virtual environment technology as a basic research tool in psychology. *Behavior research methods, instruments, & computers*, 31(4):557–564, 1999.

[60] Thomas D Parsons. Virtual reality for enhanced ecological validity and experimental control in the clinical, affective and social neurosciences. *Frontiers in human neuroscience*, 9:660, 2015.

[61] Maria V Sanchez-Vives and Mel Slater. From presence to consciousness through virtual reality. *Nature Reviews Neuroscience*, 6(4):332–339, 2005.

[62] Youngmin Kim, Amitabh Varshney, David W Jacobs, and François Guimbretière. Mesh saliency and human eye fixations. *ACM Transactions on Applied Perception (TAP)*, 7(2):12, 2010.

[63] Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.

[64] Taro Maeda, Hideyuki Ando, and Maki Sugimoto. Virtual acceleration with galvanic vestibular stimulation in a virtual reality environment. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, pages 289–290. IEEE, 2005.

[65] Bernhard E Riecke, Jörg Schulte-Pelkum, Franck Caniard, and Heinrich H Bulthoff. Towards lean and elegant self-motion simulation in virtual reality. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, pages 131–138. IEEE, 2005.

[66] Lisa Rebenitsch and Charles Owen. Review on cybersickness in applications and visual displays. *Virtual Reality*, 20(2):101–125, 2016.

[67] JJ-W Lin, Henry Been-Lirn Duh, Donald E Parker, Habib Abi-Rached, and Thomas A Furness. Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. In *Virtual Reality, 2002. Proceedings. IEEE*, pages 164–171. IEEE, 2002.

[68] Ajoy S Fernandes and Steven K Feiner. Combating VR sickness through subtle dynamic field-of-view modification. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 201–210. IEEE, 2016.

[69] Simon Davis, Keith Nesbitt, and Eugene Nalivaiko. A systematic review of cybersickness. In *Proceedings of the 2014 Conference on Interactive Entertainment*, pages 1–9. ACM, 2014.

[70] Patricia S Cowings, Steve Suter, William B Toscano, Joe Kamiya, and Karen Naifeh. General autonomic components of motion sickness. *Psychophysiology*, 23(5):542–551, 1986.

[71] Yu-Chieh Chen, Jeng-Ren Duann, Shang-Wen Chuang, Chun-Ling Lin, Li-Wei Ko, Tzyy-Ping Jung, and Chin-Teng Lin. Spatial and temporal EEG dynamics of motion sickness. *NeuroImage*, 49(3):2862–2870, 2010.

[72] Senqi Hu, Kathleen A McChesney, Kathryn A Player, Amy M Bahl, Jessica B Buchanan, and Jason E Scozzafava. Systematic investigation of physiological correlates of motion sickness induced by viewing an optokinetic rotating drum. *Aviation, space, and environmental medicine*, 1999.

[73] Li-Wei Ko, Chun-Shu Wei, Shi-An Chen, and Chin-Teng Lin. EEG-based motion sickness estimation using principal component regression. In *Neural Information Processing*, pages 717–724. Springer, 2011.

[74] By Chin-Teng Lin, Li-Wei Ko, Jin-Chern Chiou, Jeng-Ren Duann, Ruey-Song Huang, Sheng-Fu Liang, Tzai-Wen Chiu, and Tzyy-Ping Jung. Noninvasive neural prostheses using mobile and wireless EEG. *Proceedings of the IEEE*, 96(7):1167–1183, 2008.

[75] Chin-Teng Lin, Shang-Wen Chuang, Yu-Chieh Chen, Li-Wei Ko, Sheng-Fu Liang, and Tzyy-Ping Jung. EEG effects of motion sickness induced in a dynamic virtual reality environment. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 3872–3875. IEEE, 2007.

[76] Young Youn Kim, Hyun Ju Kim, Eun Nam Kim, Hee Dong Ko, and Hyun Taek Kim. Characteristic changes in the physiological components of cybersickness. *Psychophysiology*, 42(5):616–625, 2005.

[77] Byung-Chan Min, Soon-Cheol Chung, Yoon-Ki Min, and Kazuyoshi Sakamoto. Psychophysiological evaluation of simulator sickness evoked by a graphic simulator. *Applied ergonomics*, 35(6):549–556, 2004.

[78] Syed Ali Arsalan Naqvi, Nasreen Badruddin, Munsif Ali Jatoi, Aamir Saeed Malik, Wan Hazabbah, and Baharudin Abdullah. EEG based time and frequency dynamics analysis of visually induced motion sickness (vims). *Australasian Physical & Engineering Sciences in Medicine*, 38(4):721–729, 2015.

[79] Syed Ali Arsalan Naqvi, Nasreen Badruddin, Aamir S Malik, Wan Hazabbah, and Baharudin Abdullah. EEG alpha power: An indicator of visual fatigue. In *Intelligent and Advanced Systems (ICIAS), 2014 5th International Conference on*, pages 1–5. IEEE, 2014.

[80] Hiran Ekanayake. P300 and emotiv epoc: Does emotiv epoc capture real eeg? *Web publication http://neurofeedback. visaduma. info/emotivresearch. htm*, 2010.

[81] Erik W Anderson, Kristin C Potter, Laura E Matzen, Jason F Shepherd, Gilbert A Preston, and Cláudio T Silva. A user study of visualization effectiveness using EEG and cognitive load. In *Computer Graphics Forum*, volume 30, pages 791–800. Wiley Online Library, 2011.

[82] Peter Aspinall, Panagiotis Mavros, Richard Coyne, and Jenny Roe. The urban brain: analysing outdoor physical activity with mobile EEG. *British journal of sports medicine*, pages bjsports–2012, 2013.

[83] Stefan Debener, Falk Minow, Reiner Emkes, Katharina Gandras, and Maarten Vos. How about taking a low-cost, small, and wireless EEG for a walk? *Psychophysiology*, 49(11):1617–1621, 2012.

[84] Kay M Stanney and Robert S Kennedy. The psychometrics of cybersickness. *Communications of the ACM*, 40(8):66–68, 1997.

[85] Arnaud Delorme, Terrence Sejnowski, and Scott Makeig. Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis. *Neuroimage*, 34(4):1443–1449, 2007.

[86] Scott Makeig, Anthony J Bell, Tzyy-Ping Jung, Terrence J Sejnowski, et al. Independent component analysis of electroencephalographic data. *Advances in neural information processing systems*, pages 145–151, 1996.

[87] Ruey-Song Huang, Tzyy-Ping Jung, and Scott Makeig. Event-related brain dynamics in continuous sustained-attention tasks. *Foundations of augmented cognition*, pages 65–74, 2007.

[88] Ruey-Song Huang, Tzyy-Ping Jung, Arnaud Delorme, and Scott Makeig. Tonic and phasic electroencephalographic dynamics during continuous compensatory tracking. *NeuroImage*, 39(4):1896–1909, 2008.

[89] Jason Matheny. Intelligence advanced research projects activity. 3rd Annual BRAIN Initiative Investigators Meeting, North Bethesda, Maryland, 2016.

[90] Cagatay Turkay, Erdem Kaya, Selim Balcisoy, and Helwig Hauser. Designing progressive and interactive analytics processes for high-dimensional data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):131–140, 2017.

[91] Eric Krokos, Catherine Plaisant, and Amitabh Varshney. Virtual memory palaces: immersion aids recall. *Virtual Reality*, pages 1–15, 2018.

[92] Alex Endert, Patrick Fiaux, and Chris North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 473–482. ACM, 2012.

[93] Dominik Sacha, Leishi Zhang, Michael Sedlmair, John A Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C North, and Daniel A Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):241–250, 2017.

[94] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[95] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[96] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[97] Yehuda Koren. On spectral graph drawing. In Tandy Warnow and Binhai Zhu, editors, *Computing and Combinatorics*, volume 2697 of *Lecture Notes in Computer Science*, pages 496–508. Springer Berlin Heidelberg, 2003.

[98] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[99] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. Modeling human motion using binary latent variables. *Advances in neural information processing systems*, 19:1345, 2007.

[100] Martin Wattenberg, Fernanda Vigas, and Ian Johnson. How to use t-sne effectively. *Distill*, 2016.

[101] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[102] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006.

[103] Yushi Chen, Zhouhan Lin, Xing Zhao, Gang Wang, and Yanfeng Gu. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected topics in applied earth observations and remote sensing*, 7(6):2094–2107, 2014.

[104] El-ad David Amir, Kara L Davis, Michelle D Tadmor, Erin F Simonds, Jacob H Levine, Sean C Bendall, Daniel K Shenfeld, Smita Krishnaswamy, Garry P Nolan, and Dana Pe'er. visne enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature biotechnology*, 31(6):545–552, 2013.

[105] Tuan Nhon Dang and Leland Wilkinson. Transforming scagnostics to reveal hidden features. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1624–1632, 2014.

[106] Cheuk Yiu Ip, Amitabh Varshney, and Joseph JaJa. Hierarchical exploration of volumes using multilevel segmentation of the intensity-gradient histograms. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2355–2363, 2012.

[107] Hsueh-Chien Cheng, Antonio Cardone, Eric Krokos, Bogdan Stoica, Alan Faden, and Amitabh Varshney. Deep-learning-assisted visualization for live-cell images. In *Proceedings of 2017 IEEE International Conference on Image Processing*, ICIP. IEEE, September 2017.

[108] Hsueh-Chien Cheng, Antonio Cardone, Somay Jain, Eric Krokos, Kedar Narayan, Sriram Subramaniam, and Amitabh Varshney. Deep-learning-assisted volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–14, January 2018.

[109] Shusen Liu, Bei Wang, P-T Bremer, and Valerio Pascucci. Distortion-guided structure-driven interactive exploration of high-dimensional data. In *Computer Graphics Forum*, volume 33, pages 101–110. Wiley Online Library, 2014.

[110] Shusen Liu, Bei Wang, Jayaraman J Thiagarajan, P-T Bremer, and Valerio Pascucci. Visual exploration of high-dimensional data through subspace analysis and dynamic projections. In *Computer Graphics Forum*, volume 34, pages 271–280. Wiley Online Library, 2015.

[111] Eric Krokos and Hanan Samet. A look into twitter hashtag discovery and generation. In *Proceedings of the 7th ACM SIGSPATIAL Workshop on Location-Based Social Networks (LBSN14), Dallas, TX, Nov*, 2014.

[112] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408. ACM, 2003.

[113] Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris, and Ioannis P Vlahavas. Multi-label classification of music into emotions. In *The International Society of Music Information Retrieval*, volume 8, pages 325–330, 2008.

[114] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[115] Naonori Ueda and Kazumi Saito. Parametric mixture models for multi-labeled text. *Advances in neural information processing systems*, pages 737–744, 2003.

[116] Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30. Springer, 2004.

[117] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

[118] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.

[119] Nguyen Quoc Viet Hung, Duong Chi Thang, Matthias Weidlich, and Karl Aberer. Minimizing efforts in validating crowd answers. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 999–1014. ACM, 2015.

[120] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[121] Mohammad Pezeshki, Linxi Fan, Philemon Brakel, Aaron Courville, and Yoshua Bengio. Deconstructing the ladder network architecture. In *International Conference on Machine Learning*, pages 2368–2376, 2016.

[122] Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H Witten. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292, 2001.

[123] Saleema Amershi, Bongshin Lee, Ashish Kapoor, Ratul Mahajan, and Blaine Christian. Cuet: human-guided fast and accurate network alarm triage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 157–166. ACM, 2011.

[124] Axel J Soto, Ryan Kiros, Vlado Kešelj, and Evangelos Milios. Exploratory visual analysis and interactive pattern extraction from semi-structured data. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(3):16, 2015.

[125] Martín Abadi et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[126] François Chollet et al. Keras: Deep learning library for Theano and Tensorflow. *URL: https://keras. io/k*, 2015.

[127] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[128] Shun-ichi Amari, Andrzej Cichocki, and Howard Hua Yang. A new learning algorithm for blind signal separation. In *Advances in neural information processing systems*, pages 757–763, 1996.

[129] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[130] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[131] William N Anderson Jr and Thomas D Morley. Eigenvalues of the laplacian of a graph. *Linear and multilinear algebra*, 18(2):141–145, 1985.

[132] Xudong Kang, Shutao Li, and Jon Atli Benediktsson. Spectral–spatial hyperspectral image classification with edge-preserving filtering. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5):2666–2677, 2014.

[133] Hadi Shiravi, Ali Shiravi, and Ali A Ghorbani. A survey of visualization systems for network security. *IEEE Transactions on visualization and computer graphics*, 18(8):1313–1329, 2012.

[134] Stephen G Eick. Engineering perceptually effective visualizations for abstract data. In *In Scientific Visualization Overviews, Methodologies and Techniques, IEEE Computer Science*. Citeseer, 1995.

[135] Alissa Torres. Building a world-class security operations center: A roadmap. *SANS Institute, May*, 2015.

[136] John Goodall, Wayne Lutters, and Anita Komlodi. The work of intrusion detection: rethinking the role of security analysts. *AMCIS 2004 Proceedings*, page 179, 2004.

[137] Vinicius Tavares Guimaraes, Carla Maria Dal Sasso Freitas, Ramin Sadre, Liane Margarida Rockenbach Tarouco, and Lisandro Zambenedetti Granville. A survey on information visualization for network and service management. *IEEE Communications Surveys & Tutorials*, 18(1):285–323, 2016.

[138] Giovanni Vigna and Richard A Kemmerer. Netstat: A network-based intrusion detection approach. In *Computer Security Applications Conference, 1998. Proceedings. 14th Annual*, pages 25–34. IEEE, 1998.

[139] Angela Orebaugh, Gilbert Ramirez, and Jay Beale. *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.

[140] Glenn A Fink, Christopher L North, Alex Endert, and Stuart Rose. Visualizing cyber security: Usable workspaces. In *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*, pages 45–56. IEEE, 2009.

[141] G.A. Fink, C.L. North, A. Endert, and S. Rose. Visualizing cyber security: Usable workspaces. In *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*, pages 45–56, Oct 2009.

[142] Kulsoom Abdullah, Chris Lee, Gregory Conti, and John A Copeland. Visualizing network data for intrusion detection. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pages 100–108. IEEE, 2005.

[143] Ying Zhao, FangFang Zhou, XiaoPing Fan, Xing Liang, and YongGang Liu. IDSRadar: a real-time visualization framework for IDS alerts. *Science China Information Sciences*, 56(8):1–12, 2013.

[144] Bin Yu, Les Smith, and Mark Threefoot. Semi-supervised time series modeling for real-time flux domain detection on passive DNS traffic. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 258–271. Springer, 2014.

[145] Anatoly Yelizarov and Dennis Gamayunov. Visualization of complex attacks and state of attacked network. In *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*, pages 1–9. IEEE, 2009.

[146] Troy Nunnally, Kulsoom Abdullah, A Selcuk Uluagac, John A Copeland, and Raheem Beyah. Navsec: A recommender system for 3D network security visualizations. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security*, pages 41–48. ACM, 2013.

[147] Inhwan Kim, Hyunsang Choi, and Heejo Lee. BotXrayer: Exposing botnets by visualizing DNS traffic. In *KSII the first International Conference on Internet*, 2009.

[148] Yarden Livnat, Jim Agutter, Shaun Moon, Robert F Erbacher, and Stefano Foresti. A visualization paradigm for network intrusion detection. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pages 92–99. IEEE, 2005.

[149] Guihua Shan, Yang Wang, Maojin Xie, Haopu Lv, and Xuebin Chi. Visual detection of anomalies in DNS query log data. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pages 258–261. IEEE, 2014.

[150] Troy Nunnally, Penyen Chi, Kulsoom Abdullah, A Selcuk Uluagac, John A Copeland, and Raheem Beyah. P3D: A parallel 3D coordinate visualization for advanced network scans. In *Communications (ICC), 2013 IEEE International Conference on*, pages 2052–2057. IEEE, 2013.

[151] Marios Iliofotou, Prashanth Pappu, Michalis Faloutsos, Michael Mitzenmacher, Sumeet Singh, and George Varghese. Network monitoring using traffic dispersion graphs (tdgs). In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 315–320. ACM, 2007.

[152] Qingnan Lai, Changling Zhou, Hao Ma, Zhen Wu, and Shiyang Chen. Visualizing and characterizing DNS lookup behaviors via log-mining. *Neurocomputing*, 169:100–109, 2015.

[153] N. Jiang, J. Cao, Y. Jin, L. E. Li, and Z. Zhang. Identifying suspicious activities through dns failure graph analysis. In *The 18th IEEE International Conference on Network Protocols*, pages 144–153, Oct 2010.

[154] Mohammad Ghoniem, J-D Fekete, and Philippe Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 17–24. Ieee, 2004.

[155] Rosa Romero-Gomez, Yacin Nadji, and Manos Antonakakis. Towards designing effective visualizations for DNS-based network threat analysis. In *Visualization for Cyber Security (VizSec), 2017 IEEE Symposium on*, pages 1–8. IEEE, 2017.

[156] Sean McKenna, Diane Staheli, Cody Fulcher, and Miriah Meyer. Bubblenet: A cyber security dashboard for visualizing patterns. In *Computer Graphics Forum*, volume 35, pages 281–290. Wiley Online Library, 2016.

[157] Iman Sharafaldin, Amirhossein Gharib, Arash Habibi Lashkari, and Ali A Ghorbani. Botviz: A memory forensic-based botnet detection and visualization approach. In *Security Technology (ICCST), 2017 International Carnahan Conference on*, pages 1–8. IEEE, 2017.

[158] Andrew Caudwell. *Logstalgia*, 2014. http://logstalgia.io/.

[159] Tatiana Von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J van Wijk, J-D Fekete, and Dieter W Fellner. Visual analysis of large graphs: state-of-the-art and future research challenges. In *Computer graphics forum*, volume 30, pages 1719–1749. Wiley Online Library, 2011.

[160] Hans-Jorg Schulz, Steffen Hadlak, and Heidrun Schumann. The design space of implicit hierarchy visualization: A survey. *IEEE transactions on visualization and computer graphics*, 17(4):393–411, 2011.

[161] Stephen Lau. The spinning cube of potential doom. *Communications of the ACM*, 47(6):25–26, 2004.

[162] Colin Ware. *Information visualization: perception for design*. Elsevier, 2012.

[163] Gregory Conti, Mustaque Ahamad, and John Stasko. Attacking information visualization system usability overloading and deceiving the human. In *Proceedings of the 2005 symposium on Usable privacy and security*, pages 89–100. ACM, 2005.

[164] Daisuke Inoue, Masashi Eto, Koei Suzuki, Mio Suzuki, and Koji Nakao. Daedalus-viz: novel real-time 3D visualization for darknet monitoring-based alert system. In *Proceedings of the ninth international symposium on visualization for cyber security*, pages 72–79. ACM, 2012.

[165] Giovane Moura, Ricardo de O Schmidt, John Heidemann, Wouter B de Vries, Moritz Muller, Lan Wei, and Cristian Hesselman. Anycast vs. DDoS: Evaluating the november 2015 root DNS event. In *Proceedings of the 2016 Internet Measurement Conference*, pages 255–270. ACM, 2016.

[166] ICANN. Factsheet - root server attack on 6 february 2007, 2007. https://www.icann.org/en/system/files/files/factsheet-dns-attack-08mar07-en.pdf.

[167] Steve Mansfield-Devine. The growth and evolution of DDoS. *Network Security*, 2015(10):13–20, 2015.

[168] Christopher Amin, Massimo Candela, Daniel Karrenberg, Robert Kisteleki, and Andreas Strikos. Visualization and monitoring for the identification and analysis of DNS issues. In *Proceedings of the Tenth International Conference on Internet Monitoring and Protection*, 2015.

[169] Michael Aupetit, Yury Zhauniarovich, Giorgos Vasiliadis, Marc Dacier, and Yazan Boshmaf. Visualization of actionable knowledge to mitigate DRDoS attacks. In *Visualization for Cyber Security (VizSec), 2016 IEEE Symposium on*, pages 1–8. IEEE, 2016.

[170] Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. Animation: can it facilitate? *International journal of human-computer studies*, 57(4):247–262, 2002.