Human-AI Sensemaking with Semantic Interaction and Deep Learning

Yali Bian

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Christopher L. North, Chair
Hoda M. Eldardiry
Anuj Karpatne
Eric Krokos
Chandan K. Reddy

February 9, 2022
Blacksburg, Virginia

Keywords: Semantic Interaction, Interactive Deep Learning, Visual Analytics,
Sensemaking, Explainable AI, Human-in-the-loop Machine Learning
Copyright 2022, Yali Bian

Human-AI Sensemaking with Semantic Interaction and Deep Learning

Yali Bian

(ABSTRACT)

Human-AI interaction can improve overall performance, exceeding the performance that either humans or AI could achieve separately, thus producing a whole greater than the sum of the parts. Visual analytics enables collaboration between humans and AI through interactive visual interfaces. Semantic interaction is a design methodology to enhance visual analytics systems for sensemaking tasks. It is widely applied for sensemaking in high-stakes domains such as intelligence analysis and academic research. However, existing semantic interaction systems support collaboration between humans and traditional machine learning models only; they do not apply state-of-the-art deep learning techniques.

The contribution of this work is the effective integration of deep neural networks into visual analytics systems with semantic interaction. More specifically, I explore how to redesign the semantic interaction pipeline to enable collaboration between human and deep learning models for sensemaking tasks. First, I validate that semantic interaction systems with pre-trained deep learning better support sensemaking than existing semantic interaction systems with traditional machine learning. Second, I integrate interactive deep learning into the semantic interaction pipeline to enhance inference ability in capturing analysts' precise intents, thereby promoting sensemaking. Third, I add semantic explanation into the pipeline to interpret the interactively steered deep learning model. With a clear understanding of DL, analysts can make better decisions. Finally, I present a neural design of the semantic interaction pipeline to further boost collaboration between humans and deep learning for sensemaking.

Human-AI Sensemaking with Semantic Interaction and Deep Learning

Yali Bian

(GENERAL AUDIENCE ABSTRACT)

Human AI interaction can harness the separate strengths of human and machine intelligence to accomplish tasks neither can solve alone. Analysts are good at making high-level hypotheses and reasoning from their domain knowledge. AI models are better at data computation based on low-level input features. Successful human-AI interactions can perform real-world, high-stakes tasks, such as issuing medical diagnoses, making credit assessments, and determining cases of discrimination. Semantic interaction is a visual methodology providing intuitive communications between analysts and traditional machine learning models. It is commonly utilized to enhance visual analytics systems for sensemaking tasks, such as intelligence analysis and scientific research.

The contribution of this work is to explore how to use semantic interaction to achieve collaboration between humans and state-of-the-art deep learning models for complex sensemaking tasks. To do this, I first evaluate the straightforward solution of integrating the pretrained deep learning model into the traditional semantic interaction pipeline. Results show that the deep learning representation matches human cognition better than hand engineering features via semantic interaction. Next, I look at methods for supporting semantic interaction systems with interactive and interpretable deep learning. The new pipeline provides effective communication between human and deep learning models. Interactive deep learning enables the system to better capture users' intents. Interpretable deep learning lets users have a clear understanding of models. Finally, I improve the pipeline to better support collaboration using a neural design. I hope this work can contribute to future designs for the human-in-the-loop analysis with deep learning and visual analytics techniques.

Acknowledgments

The Ph.D. journey has been, if not the most compelling, definitely the most unforgettable phase of my life. I am lucky and so grateful that I got helps from a fantastic group of people who deserve acknowledgment.

First, I would like to thank my family. My mom and dad have been highly supportive of my pursuit of the Ph.D. Without their unconditional love and backing, I would never have had such excellent opportunities to venture out into the wider world. My girlfriend, Siyu Mi, has always believed in me and supported what I do. Without her encouragement and faith in me, I might not have had the courage to go where I have gone with this work.

Second, I would like to thank my advisor and committee members. Dr. Chris North is a knowledgeable and brilliant advisor as well as a genuine and sincere friend. I am lucky to have had an opportunity to work with someone that I can rely on for advice and guidance about anything. Thank you for being a great influence on my work and my life. I also would like to thank Drs. Hoda Eldardiry, Anuj Karpatne, Eric Krokos, and Chandan K. Reddy for serving on my committee. Dr. Krokos and Dr. Reddy have provided fantastic feedback and countless essential ideas for my research. Dr. Eldardiry and Dr. Karpatne have provided continuous support across each of my my milestones and offered suggestions that have considerably improved the quality of my thesis dissertation. I can't discount the feedback and support I have received from them. Thanks for being readers of my thesis and providing valuable feedback.

Third, many other professors at Virginia Tech helped me with my research, including but certainly not limited to Dr. Edward A. Fox, Dr. Scott McCrickard, Dr. Kurt Luther, Dr. Bert Huang, Dr. Nicholas Polys, and Dr. Doug A. Bowman. Their insightful comments have

tremendously reduced my efforts in uncovering the appropriate direction for my research. Dr. Alex Ender from Georgia Tech and Dr. Christopher Andrews from Middlebury College also generously provided their experimental results to support this dissertation. Thanks for being patient and making time to talk with me. In addition, I would also like to thank Chitra Phadke, my mentor at Bell Labs. She was so kind to introduce me to the world of industry research. I really enjoyed our vibrant discussions about ways to solve real-world problems with academic knowledge.

I would also like to thank my colleagues: Caleb Reach, Ji Wang, Nai-Ching Wang, Tianyi Li, Sid Holman, Mai Dahshan, and Moeti M. Masiane. They have helped me through different stages and contributed to some of the essential ideas in this dissertation. They also shared their invaluable career plans and lessons learned, which significantly broadened my horizons and influenced my career choices. My InfoVis Lab lab mates have also supported me in my doctoral program. They were constantly available with a sympathetic ear and clever research insights. Thanks for all your help.

Last but not least, this research was funded in part by NSF I/UCRC CNS-1822080 via the NSF Center for Space, High-performance, and Resilient Computing (SHREC).

Attributions

Chapter 3 is based on the paper titled "DeepVA: Bridging Cognition and Computation through Semantic Interaction and Deep Learning," which was presented in Proceedings of the IEEE VIS Workshop MLUI 2019: Machine Learning from User Interactions for Visualization and Analytics [14]. Coauthors of this work include John Wenskovitch and Chris North.

Chapter 4 is based on the paper titled "Evaluating Semantic Interaction on Word Embeddings via Simulation," which was presented at EValuation of Interactive VisuAl Machine Learning systems (EVIVA-ML), an IEEE VIS 2019 Workshop [13]. Coauthors of this work include Michelle Dowling and Chris North.

Chapter 5 is based on the paper titled "DeepSI: Interactive Deep Learning for Semantic Interaction," which was published through the Proceedings of the 26th International Conference on Intelligent User Interfaces [12]. This paper was coauthored by Chris North.

Chapter 6 is expanded from the paper titled "Semantic Explanation of Interactive Dimensionality Reduction," which was published through the 2021 IEEE Visualization Conference (VIS) [15]. Coauthors of this work include Chris North, Eric Krokos, and Sarah Joseph.

Chapter 7 is original work and has not been submitted for publication. Coauthors of this work include Wei Liu and Chris North.

Contents

List of Figures

List of Tables

List of Abbreviations

AE Auto-Encoder

AI Artificial Intelligence

BERT Bidirectional Encoder Representations from Transformers

BOVW Bag of Visual Words

CNN Convolutional Neural Networks

CV Computer Vision

DL Deep Learning

DNNs Deep Neural Networks

DR Dimension(ality) Reduction

HCI Human-Computer Interaction

HD High-Dimensional

kNN k Nearest Neighbour

LD Low-Dimensional

MDP Multidimensional Projection

MDS Multidimensional Scaling

ML Machine Learning

NLP Natural Language Processing

NNP Neural Network Projection

OLI Observation-Level Interaction

PCA Principal Component Analysis

PI Parametric Interaction

PrI Projection Interaction

SE Semantic Explanation

SI Semantic Interaction

SIFT Scale-Invariant Feature Transform

SSL Self-Supervised Learning

V2PI Visual to Parametric Interaction

VA Visual Analytics

WMDS Weighted Multidimensional Scaling

XAI Explainable Artificial Intelligence

Chapter 1

Introduction

Large and complex datasets [55] increasingly challenge both AI and analysts aiming at data exploration and decision making, especially in high-stakes domains such as emergency response planning and discrimination determination. Human-AI interaction can address this problem and accomplish these tasks in ways that exceed the separate individual capabilities of humans and AI [89]. However, previous work focuses mainly on the interaction between human and traditional machine learning models. There is still a lack of support for designing and building successful collaboration between human intelligence and state-of-the-art deep learning techniques [104].

1.1 Background: Semantic Interaction for Sensemaking

Visual analytics (VA) [38] enables human-AI collaboration via interactive visual interfaces. In VA systems, analysts and the underlying analytic model can communicate with each other via visualization. This communication allows analysts to explore and draw conclusions from large and complex datasets with the help of an AI partner. For example, VA systems can utilize dimensionality reduction (DR) methods to produce interactive data projections, allowing analysts to easily explore datasets in the low-dimensional visual space [23, 105, 124,

143].

Semantic interaction (SI) is a commonly utilized interaction methodology to enhance VA systems for sensemaking tasks. SI-enabled systems let the analyst directly manipulate interactive projections of data [148]. As shown in Fig. 1.1, analysts can manually and naturally interact with visual metaphors on the visual interface instead of directly and formally manipulating the model parameters (Step 1). The semantic meaning behind these projection interactions is the similarity the analyst wishes to find within the data during the sensemaking process [133]. Consequently, the SI system learns to infer users' intents (Step 2) and train the ML model to provide user-preferred visual feedback (Step 3). With these intuitive and natural interactions, the analyst can remain within the cognitive zone [65] while gaining assistance from the underlying model (Step 4). SI systems can thus enhance human-AI sensemaking in analytic tasks [178].

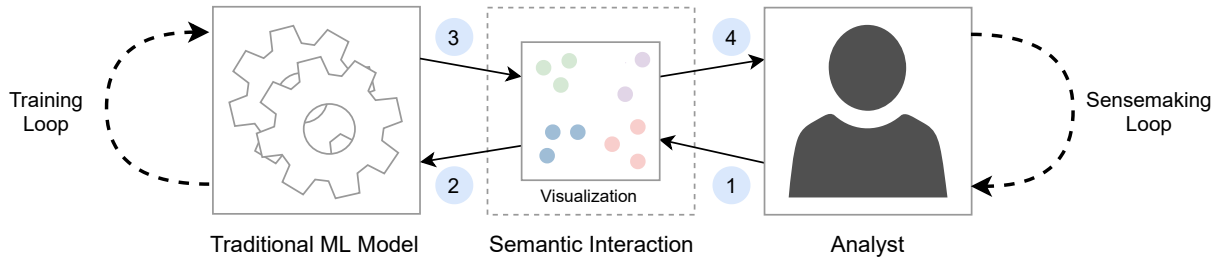


Figure 1.1: The semantic interaction system supports collaborations between analysts and traditional machine learning models with intuitive observation-level interactions on the visualization for sensemaking tasks.

Technically, current SI systems employ interactive DR [143, 178] as a bridge to connect AI and the analyst through a bidirectional process (steps 1-2 and 3-4) that can couple the training loop of the ML model and the sensemaking loop of the analyst. However, interactive DR supports the integration of only relatively simple machine learning models into the SI pipeline [80, 81, 99, 106, 117]. The limited computational power of traditional machine learning models restricts the sensemaking performance of SI systems.

1.2 Goal: Power Semantic Interaction with Deep Learning

DL [104] is a state-of-the-art representation learning method [10] that can automatically extract abstract and useful hierarchical representations from raw data [10]. These meaningful representations can effectively capture high-level concepts needed by the analyst [104]; deep learning outperforms traditional machine learning techniques in many fields. Convolutional neural networks (CNN) have reached the level of human performance in challenging visual tasks [73]. BERT and other transformer-based variations [42, 165] show breakthrough performance in handling a wide variety of natural language processing (NLP) tasks. This is because DL representations are closer to high-level concepts in the datasets than traditional engineered features.

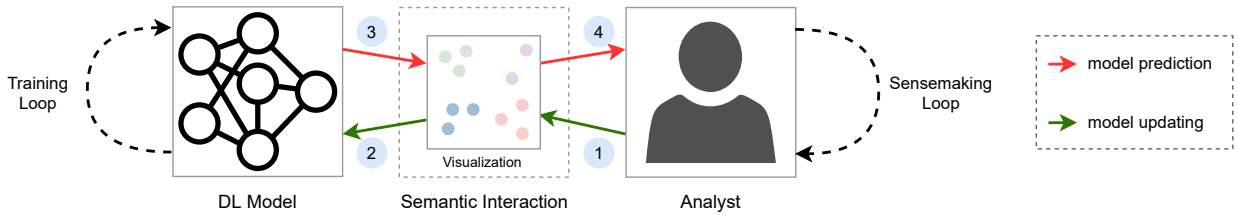


Figure 1.2: Outline of this thesis: human-DL interaction via semantic interaction for sensemaking tasks. The SI system utilizes the deep learning model to capture users' precise intents and provide accurate assistance.

Deep learning offers the new opportunity to power SI to capture the analyst's precise intent, thereby coupling human intelligence and AI for complicated sensemaking tasks. Furthermore, the vast quantity of learned knowledge inside the multilayer structure of deep neural networks offers an opportunity to provide a richer abstraction hierarchy to support incremental formalization and assist analysts in formalizing complex concepts.

This thesis aims to develop an SI framework for effective collaboration between human intel-

ligence and the deep learning model for sensemaking tasks. Specifically, it aims to integrate deep learning into the SI system to better capture the precise intents behind analysts' interactions and provide suggestions to promote effective decision-making during the interactive sensemaking process. Fig. 1.2 outlines the key process considered in this thesis.

1.3 Challenges: The Need for Interactive and Interpretable Deep Learning

Introducing deep learning within the semantic interaction pipeline poses new challenges in system design. First, deep neural networks consist of many parameters in different neural layers and neurons. A vast number of training examples are needed to learn accurate data representations. In the context of interactive machine learning [58], it is challenging to use a small number of semantic interactions with analysts to update the deep learning model to obtain user- and task-specific representations.

In addition, the multilayer nonlinear structure makes deep neural networks non-transparent, and their predictions are not traceable. It is difficult for analysts to understand deep learning feedback, build trust, and make decisions during the sensemaking process.

Furthermore, existing semantic interaction systems follow the standard VA framework [142], in which the analytic model and the visualization method are two separate components. During the co-learning process, these two components are executed individually in sequence. When using deep learning as the analytic model, it is time-consuming and unstable for the traditional DR method to transform the high-dimensional DL representations into 2D visual feedback[57, 67, 74].

To sum up, the complexity of the model structure and the training process makes DL chal-

lenging to integrate into the SI pipeline. There is still a lack of solutions to these challenges.

1.4 Research Questions

To tackle these challenges, I focus this dissertation on exploring the following major research questions:

- RQ 1: How do pretrained deep learning representations improve the SI system for sensemaking compared to traditional engineered features?
- RQ 2: How can interactive DL be integrated into the SI system to obtain user- and task-specific representations, thereby improving SI inference?
- RQ 3: How can DL interpretation be integrated into the SI system to generate semantic explanations and support the sensemaking loop for analysts?
- RQ 4: How can analytical and dimension-reduction models be merged into an end-to-end trainable deep neural network for visual analytics?

In the following subsections, I discuss each of these research questions in detail.

1.4.1 RQ 1: How do pretrained deep learning representations improve the SI system for sensemaking compared to traditional engineered features?

The fundamental hypothesis of this thesis is that human-DL collaboration outperforms traditional human-ML collaboration within SI systems. Validating this hypothesis is necessary before investigating and designing SI frameworks supporting human-DL sensemaking. SI

systems attempt to model the cognitive reasoning behind users' interactions with data instances. Thus, the detailed hypothesis is that DL representations contain meaningful high-level abstractions that can capture users' high-level cognitive intent. The prerequisite for powering SI with DL is to make sure deep learning representations do, in fact, better support SI inference for sensemaking tasks compared to engineered features. Therefore, I investigate RQ 1 with the following two sub-questions:

- RQ 1.1: How well do pretrained CNN representations support SI systems for visual concept learning tasks when compared to engineered features of images? (Chapter 3)

- RQ 1.2: How well do word embeddings support SI systems for visual text analysis when compared to engineered features of texts? (Chapter 4)

I use the pre-trained DL models as fixed feature extractors in the traditional SI pipeline to answer these two sub-questions. Specifically, the SI system uses DL representations as data features instead of hand-crafted features. To evaluate and compare SI models with different features (DL representations vs. engineered features), we design and implement an SI framework called DeepVA. DeepVA can use features at different levels of abstraction as the data input.

In Chapter 3, we perform visual concept learning tasks at three different difficulty levels, using semantic interaction systems with features at different abstraction levels.

In Chapter 4, we present how word embeddings, as an alternative to the traditional bag-of-words features, can support SI systems for visual text analytics. Results in both chapters show that SI systems with pretrained DL representations as data features perform better than systems with hand-crafted features [13, 14].

1.4.2 RQ 2: How can interactive DL be integrated into the SI system to obtain user- and task-specific representations, thereby improving SI inference?

While pretrained DL can improve system performance, integrating the interactive DL training loop into the SI pipeline is essential to utilize the full power of deep learning for sensemaking. Interactive DL can learn to provide user- and task-specific representations based on users' interactions, thereby improving SI inference (steps 1 and 2 in Fig. 1.2). RQ 2 explores the design of the SI pipeline powered with interactive DL. Central to this design goal are two research questions:

- RQ 2.1: How can semantic interactions be exploited to accurately adapt the pretrained DL representations to current analytic tasks?
- RQ 2.2: How can efficient adaptations be made so that a small number of semantic interactions suffice for analysts to express their intents?

In Chapter 5, I propose a novel DeepSI framework with the following two design rationales to address these two questions. First, I insert interactive DL training into the bidirectional structure of the original SI pipeline so that interactions trigger DL adaptation. In this way, new user- and task-specific representations are generated based on semantic interactions provided by analysts during their sensemaking process. Second, I employ the finetuning-based DL adaption approach and the MDS-based interactive DR model to minimize the number of parameters that require training in the underlying model, allowing DeepSI to tune the DL model efficiently basedon the analyst's interactions without information loss. Specifically, I use pretrained BERT [42], a state-of-the-art DL model for NLP tasks, as the DL model representative inside DeepSI for visual text analysis tasks.

1.4.3 RQ 3: How can DL interpretation be integrated into the SI system to generate semantic explanations and support the sensemaking loop for analysts?

As explored in RQ 2, SI applications with interactive DL can better capture analysts' precise intents and support complex decision-making tasks. However, the opaque nature of deep learning prevents incremental formalism and thereby inhibits the sensemaking process. Building the SI pipeline with interpretation methods is essential for mapping abstract DL representations to low-level features, thereby supporting human sensemaking (steps 3 and 4 in Fig. 1.2). RQ 3 focuses on the design of the SI pipeline with DL interpretation. Central to this design goal are two research questions:

- RQ 3.1: How can semantic explanations be generated for black-box models inside SI systems that support incremental formalization for sensemaking tasks?
- RQ 3.2: How can the generated model explanations be naturally contextualized into the projection representing the user's mental model?

To address these problems, we present a semantic explanation framework (SE) that generates and contextualizes counterfactual explanations [167] into SI designs. SE consists of two key components: counterfactual engine and counterfactual projection. Counterfactual engine interprets black-box models by identifying representative counterfactual explanations for each data instance. Counterfactual projection contextualizes counterfactual examples into the data projection.

I apply SE to the SI system with interactive DL. With SE, analysts can focus on the sensemaking task at hand while incrementally formalizing their mental model. We discuss this research question in detail in Chapter 6.

1.4.4 RQ 4: How can analytical and dimension-reduction models be merged into an end-to-end trainable deep neural network for visual analytics?

RQ 3 and 4 have explored the use of interactive and interpretable deep learning as the analytic model of SI systems for complex sensemaking tasks. These systems follow the standard visual analytics framework in which the analytic model and the visualization method are two separate components. During the co-learning process, these two components are executed individually in sequence. This results in the loss of desired properties, such as out-of-sample capability, stability, and inference speed.

To avoid this issue, I propose a neural design framework of visual analytics for SI called NeuralSI. In NeuralSI, the analytic model and the visualization method form one integrated end-to-end trainable deep neural network. The integrated deep neural network has two parts: a DL backbone model, which serves as the analytic model, and a projection head appended to the backbone, which functions as the visualization method. To understand what promotes the performance of the neural design, I investigate two detailed sub-questions:

- RQ 4.1: How do the three design aspects (projection head architecture, projection parameter initialization, and loss function for visual interactions) affect the NeuralSI framework?
- RQ 4.2: How does the performance of the best NeuralSI compare with the state-of-the-art SI system?

When choosing the best design options and combinations, we can implement a NeuralSI system with comparable performance to state-of-the-art standard systems with interactive deep learning. We present this research question and detail the study results in Chapter 7.

1.5 Structure of this Dissertation

Chapter 1 introduces the background of and motivation for this work, lists the key challenges and goals, and presents the research questions. This section concludes Chapter 1.

Chapter 2 presents a literature review focusing on semantic interaction, deep learning, visual analytics & deep learning, explainable artificial intelligence, and multidimensional projection with neural networks.

Chapters 3 and 4 address Research Question 1 by integrating two different pretrained distributed representation methods (CNN, word embedding) into the DeepVA system for two different analytics tasks (visual concept analysis, text analytics). Both chapters show that SI systems with pretrained DL representations as data features perform better than systems with hand-crafted features.

Chapter 5 addresses Research Question 2 by designing and implementing the DeepSI system, which effectively and efficiently integrates interactive DL training into the SI pipeline. DeepSI can tune the DL model efficiently to capture precise intents from the analyst's interactions without information loss.

Chapter 6 addresses Research Question 3 by designing and implementing the DeepSE system, which complements the DeepSI system with semantic explanations to interpret the DL model for incremental formalization.

Chapter 7 addresses Research Question 4 by designing and implementing the NeuralSI system, which replaces the visualization component of SI systems with a projection neural network. The neural design enables the SI system with desired properties, such as out-of-sample capability, stability, and inference speed.

Chapter 8 summarizes the main contributions of the dissertation and speculates about future

research directions in human-AI sensemaking.

1.5.1 Complete List of Research Questions

- RQ 1: How do pretrained deep learning representations improve the SI system for sensemaking compared to traditional engineered features?
  - RQ 1.1: How do pretrained CNN representations support SI systems for visual concept learning tasks when compared to engineered features of images? (Chapter 3)
  - RQ 1.2: How do word embeddings support SI systems for visual text analysis when compared to engineered features of texts? (Chapter 4)
- RQ 2: How can interactive DL be integrated into the SI system to obtain user- and task-specific representations, thereby improving SI inference? (Chapter 5)
  - RQ 2.1: How can semantic interactions be exploited to accurately adapt the pretrained DL representations to current analytic tasks?
  - RQ 2.2: How can efficient adaptations be made so that a small number of semantic interactions suffice for analysts to express their intents?
- RQ 3: How can DL interpretation be integrated into the SI system to generate semantic explanations and support the sensemaking loop for analysts? (Chapter 6)
  - RQ 3.1: How can semantic explanations be generated for black-box models inside SI systems that support incremental formalization for sensemaking tasks?
  - RQ 3.2: How can the generated model explanations be naturally contextualized into the projection representing the user's mental model?

- RQ 4: How can analytical and dimension-reduction models be merged into an end-to-end trainable deep neural network for visual analytics? (Chapter 7)
  - RQ 4.1: How do the three design aspects (projection head architecture, projection parameter initialization, and loss function for visual interactions) affect the NeuralSI framework?
  - RQ 4.2: How does the performance of the best NeuralSI compare with the state-of-the-art SI system?

Chapter 2

Review of Literature

Five related components support this research: semantic interaction, deep learning, visual analytics and deep learning, explainable artificial intelligence, multidimensional projection with neural networks.

2.1 Semantic Interaction

Semantic interaction (SI) [49, 53] is an interaction methodology that is commonly utilized to enhance visual analytics (VA) for sensemaking tasks. I illustrate semantic interaction in the following four perspectives: semantic interaction for sensemaking, semantic interaction pipeline, interactive dimension reduction, and semantic interaction applications.

2.1.1 Semantic Interaction for Sensemaking

Parametric Interaction (PI) enables users to adjust parameters (such as feature weights) to incorporate human knowledge and questions into the model during the sensemaking process [51]. Therefore, users can explore and analyze high-dimensional data based on their own needs and domain-specific problems. However, parameter tuning usually needs particular mathematical knowledge, which is a daunting burden for analysts with cognitively demanding tasks [91]. They must pause the analysis of the data to determine how to adjust model

parameters to formally externalize their intents for the next exploration [48].

To solve this bottleneck, semantic interaction was proposed [48]. SI-enabled systems let the analyst directly manipulate interactive projections of data [148]. The semantic meaning behind these projection interactions is the similarity relationships the analyst wishes to find within the data during the sensemaking process [133]. Through these intuitive and natural interactions, the analyst can remain within the cognitive zone [65], and the system thereby enhances the analyst's efficiency in performing analytic tasks [178]. The cognitive connection between the analyst and the spatial layout, through the visual *proximity* $\approx$ *similarity* metaphor, makes SI an effective interaction methodology for analysts to express their intent, as it does not require analysts to formalize their analytic reasoning [114].

To sum up, using cases (informal relationships) rather than formalized feature descriptions, the analyst can express high-level semantics on the fly [1].

Figure 2.1: SI pipeline that follows the standard VA framework (adapted from [15, 53, 149]). The interactive DR component serves as the visualization method responsible for capturing the analyst's intent and updating the projection in response. The relevance model, also known as the metric learning method [20], is the analytic model responsible for inferring the intents from interactive DR and providing the updated relevance as feedback.

2.1.2 Semantic Interaction Pipeline

Existing SI systems follow the standard VA framework, including data, the analytic model, and the visualization. As shown in Figure 7.1, the analytic model utilizes a distance metric learning method to infer the intent behind the analyst's interactions [20]. The visualization component is an interactive dimensionality reduction (DR) method [143, 164] which supports the bidirectional transformation between the high-dimensional feature space and the low-

dimensional visual space [15, 147].

The analytic model and the visualization method can be combined into one parametric interactive DR for both concept learning and visualization generation. It is the interactive DR's responsibility to learn new model parameters that capture the analyst's intent behind the modified projection and, in response, use the learned parameters to update the projection. As shown in the SI pipeline (Fig. 2.2), analysts gain insights from the DR model projection (visualization) and express their preferences by repositioning data points in the projection (human projection). Consequently, the interactive DR learns to capture the analyst's intent behind the modified projection and convert it back to high-dimensional space. Next, the tuned DR uses the preferred high-dimensional representations to update the projection (model projection).

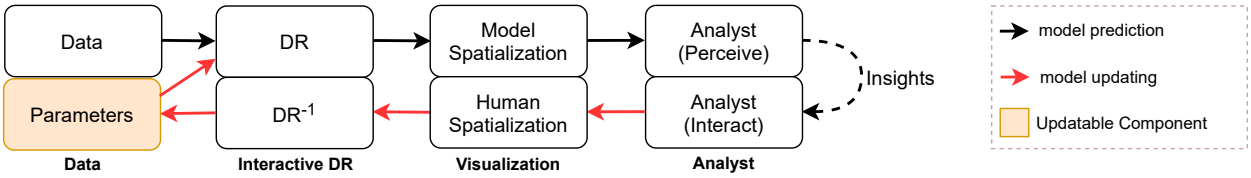


Figure 2.2: SI pipeline showing the communication between the analyst and VA system, adapted from [53, 149]. The interactive DR component is responsible for capturing the analyst's intent from the human modified projection (denoted as human spatialization) and, consequently, updating the projection in response (denoted as model spatialization).

2.1.3 Semantic Interaction Implementation

Several machine learning models have been explored to implement the bi-directional SI pipeline. SI systems are mainly implemented with an interactive dimensionality reduction (*DR*) model. The DR model provides a spatial layout as an interaction medium by which analysts naturally externalize their preferences. OLI (Observation-Level Interaction) [51] and it generalized frameworks, V2PI (Visual to Parametric Interaction) [106]

and Bayesian visual analytics (BaVA) [80], show how popular linear dimension reduction models, such as *Weighted Multidimensional Scaling* (WMDS) [146] and principal component analysis (PCA) [182], can be inverted to allow the direct manipulation of the points within the projection.

In addition, increasingly powerful DR models have been adapted in a semi-supervised manner to improve SI inference. To support more complex tasks and interactions, multiple models were chained together as a single interactive DR model, which is called multi-model SI [19, 47, 175]. Recently, to adapt more powerful but complex non-invertible DR algorithms, such as t-SNE [113] and UMAP [118], Zexplorer [117] used the invertible neural encoder [57] to emulate these models as the interactive DR model in SI applications. To steer nonlinear data models for users' complex, high-level domain knowledge, AxiSketcher [99] introduces drawing as an interaction to express their complex intents, and nonlinear axis mapping methods to model the intents.

2.1.4 Semantic Interaction Applications

SI has been commonly utilized to enhance VA systems for a variety of sensemaking tasks, including visual text analytics [53], multivariate data analysis [147], and visual concept explorations [14]. It has been applied to text analytics in ForceSPIRE [53], where users can make sense of text data for intelligence analysis. StarSPIRE [19, 47] extends ForceSPIRE to unify the sensemaking loop by coupling synthesis with foraging during the text analytics. Cosmos [47] suppots SI with topic modeling [16] for sensemaking with big text. It has also been applied to high-dimensional quantitative data in Andromeda [149]. Semantic interaction was also applied on images in ACTIVECANVAS [76], to add extra dimensions or attributes to the data based on the domain expertise of the user expressed via user

interactions.

2.2 Deep Learning

The ability to extract abstract but meaningful features makes deep learning an attractive approach for complex problems like visual object recognition and natural language understanding. I illustrate deep learning in the following three perspectives: deep learning representation, BERT, pretrained DL adaptation.

2.2.1 Deep Learning Representation

In contrast to conventional machine learning techniques, deep learning can extract useful and meaningful representations in complex data automatically through the usage of deep artificial neural networks. As stated by LeCun et al., "Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction" [104]. In deep neural networks, initial layers are used to learn lower-level concepts, such as local features in images. Representations in later layers are computed based on the previous layers and contain higher-level concepts that result from the combination of these earlier layers. For instance, when using CNN on images, local combinations of edges form motifs, which then assemble into parts, which assemble into objects [97].

Deep learning representations are typically used internally to the learning process to support the final layer for analytics tasks such as classification. However, as high-level abstract features, deep learning representations are powerful and could be used outside of deep learning processes to solve other problems. Razavian et al. [136] conducted experiments for different

recognition tasks based on traditional classification methods and CNN representations as input features. Their results show that features obtained from deep learning with convolutional nets should be the primary candidate in most visual recognition tasks. Athiwaratkun et al. [7] showed that even pre-trained CNN is much more useful as generic feature extractors for computer vision tasks, than commonly used engineered features, such as *Scale-Invariant Feature Transform* (SIFT) [112]. Amir et al. [194] provides a fully computational approach for modeling the structure of space of visual tasks for images through transfer learning techniques. Been et al. [94] introduces Concept Activation Vectors (CAVs), which provide an interpretation of a neural net's internal state in terms of human-friendly concepts.

2.2.2 BERT

BERT (Bidirectional Encoder Representations from Transformers) [42] is a DL language representation model. BERT is first pretrained on raw text data to learn general language representations. The pretrained BERT then can be easily adapted to downstream NLP tasks such as sentiment analysis and semantic textual similarity [24]. The adapted BERT model is able to provide task-specific representations and shows state-of-the-art performance in these downstream tasks. Technically, BERT is a Transformer encoder [165], containing a stack of transformer layers. The transformer layer learns token-level representations. For input token sequences, the transformer layer learns a new vector for each token based on all other tokens, using the self-attention mechanism [8]. Through the stack of transfer layers, BERT can convert a sequence of tokens into deep representations.

2.2.3 Pretrained DL Adaptation

There are two main paradigms to adapt the pretrained DL representation model to downstream tasks: feature extraction and fine-tuning [132]. The feature extraction approach uses a task-specific architecture to adapt the pretrained representations to downstream tasks (e.g. ELMo [131]). In this approach, parameters inside the task-specific architecture are trained on the downstream tasks. In contrast, instead of using a new architecture, the fine-tuning method appends one additional output layer to the pretrained DL and tunes the whole pretrained model with the downstream tasks. The fine-tuning approach requires relatively less training data because it introduces minimal task-specific parameters and does not need to learn randomly initialized task-specific parameters from scratch.

2.3 Visual Analytics and Deep Learning

Research on the combination of visual analytics and deep learning can be categorized into two areas: deep learning for visual analytics, visual analytics for deep learning.

2.3.1 Deep Learning for Visual Analytics

This section highlights previous relevant work on DL for VA: supporting visual analytics systems with DL for complex sensemaking tasks. Hsueh-Chien et. al. [34] used CNN techniques to assist users in volume visualization designing through facilitating user interaction with high-dimensional DL features. In RetainVis [100], an interactive and interpretable RNN model was designed for electronic medical records analysis and patient risk predictions, which can be steered interactively by domain experts. Gehrmann et al. [60] proposed a framework of collaborative semantic inference that enables the visual collaboration between humans

and DL algorithms. Sharkzor [134] is an interactive deep learning system for image sorting and summary, based on users' semantic interactions. Hyesook et al. [159] have proposed a visualization system to analyze deep learning models with air pollution data according to the characteristics of spatiotemporal data. Deep motif dashboard [101] provides a series of visualization strategies to sequence patterns extracted from three DL models (convolutional, recurrent, and convolutional-recurrent networks) to identify meaningful DNA sequence signals.

2.3.2 Visual Analytics for Deep Learning

Deep learning representations and structures are difficult to interpret because of the innate complexity and nonlinear structure of deep neural networks, leading to the need for interpretable or explainable methods. Fred et. al. [78] summarizes thoroughly the state-of-the-art of visual analytics in deep learning research for model explanation, interpretation, debugging, and improvement. Jaegul et. al. [35], reviews existing visual analytics techniques in making deep learning interpretable and controllable by humans from the perspective of visual analytics, information visualization, and machine learning. To help researchers comprehend the abstract representations or complicated structures of deep learning, visual analytics systems have been developed [108, 154, 185].

Beyond structures, other related work that helps researchers understand the dimensions of deep learning representations has been well described [62, 144, 195]. For a more detailed illustration of VA for DL, please refer to the exhaustive surveys [35, 78].

2.4 Explainable Artificial Intelligence

2.4.1 Dimensionality Reduction Explanation

Various explainable visualizations techniques have been proposed to assist analysts in reasoning data projection results [124]. I describe them in the following three categories.

Generic DR Explanation

Here, I focus on work proposed generic explanation methods for DR projections. The statistical chart is the most commonly used explanation method that automatically derives statistical descriptions and visualizes statistical charts for projection patterns, such as clusters [90]. The statistical chart is only useful for interpreting linear projections. Analysts are commonly confused about the inconsistent explanations between input features and the nonlinear projection results.

Nonlinear Projection Explanation

For nonlinear projection, Stahnke et al. [160] proposed the probing projections, which uses a grayscale heatmap as the projection background to indicate the selected feature value distribution for actual samples. DimReader [59] is a model-agnostic application that uses user-designed perturbations to create generalized axes for nonlinear projections. These methods explain a single feature at a time. The analyst must manually select and inspect all features in turn to find the most relevant one. Praxis [23] is a general framework for interacting with data points in both low- and high-dimensional spaces. Praxis uses prolines to show how the input change could influence the model output.

In addition, several visual explantion methods have been proposed to interpret the projection results of nonlinear methods, that can be categorized as: model-specific [26, 160], model-agnostic [23, 59, 61]. These methods offer detailed and valuable guidance in determining the meaning of the data projection.

Interactive DR Explanation

Interactive DR systems with SI usually adapt linear DR methods [19, 20, 80, 149], because of the easy interpretation property in supporting analysts' process of incremental formalism [48, 152]. Analysts can naturally associate their externalized concepts with the change of projection, which is linearly associated with updates of input feature importance. For example, Andromeda [149] provides the global feature importance of the model through slider bars. StarSPIRE [19] supports incremental formalization in text analysis by revealing upweighted keywords that support the projection.

Recently, several SI systems have employed expressive nonlinear dimensionality reduction to improve the inference ability [12, 117]. However, exiting visual explanations used in SI systems, such as the global feature importance through slider bars [149] and the node-link diagram connected by shared entities [19], only work for linear models.

2.4.2 Counterfactual Explanation and Visualization

Counterfactual explanations describes the causal relationships between the change to data features and the change to model results, for what-if analysis [167]. Recently, counterfactual explanations have been widely applied to interpret machine learning, particularly classification models [115, 167]. A variety of VA techniques have been developed for counterfactual explanations, including ViCE [179], What-If Tool [33], and DECE [33].

Counterfactual Explanation

Counterfactual explanations [167] describes the causal relationships between the change to data features and the change to model results, for what-if analysis. Counterfactuals have been widely applied to the real world because they can elicit causal reasoning in humans. Recently, counterfactual explanations have been commonly used in interpreting machine learning, particularly classification models. Wachter et al. [167] first proposed the unconditional counterfactual explanation.

They treated counterfactual explanation as an optimization problem. The optimization objective includes two parts: first, minimize the distance between the counterfactual example and the original data instance, denoted as *proximity*; second, the output of the model on the counterfactual example meets the desired results, denoted as *validity*. For example, to explain document classification, Martens et al. [115] define the counterfactual explanation as a minimal set of words, such that removing words within this set from the document changes the predicted class from the class of interest, which is called *evidence counterfactual*.

Counterfactual Visualization

A variety of VA techniques have been developed for counterfactual explanations. ViCE [63] proposed an instance-level visualization of counterfactual explanation of the underlying SVM classifier to provide end-users with personalized, actionable insights. What-If Tool [179] adopted a perturbation-based method to help users interactively probe machine learning models, which offers the functionality of finding the nearest counterfactual data points with different labels. DECE [33] is an interactive visualization system to help users understand a model's decision on both instance and data subsets level counterfactual explanations for tabular data. The above techniques aim to augment the counterfactual explanation explorations

with visualization and interaction.

2.5 Multidimensional Projection with Neural Networks

Multidimensional projection (MDP) [124, 181] is a commonly used data visualization technique for exploratory analysis of high-dimensional data. MDP transforms multidimensional data into scatter plots to reflect the data patterns in the original high-dimensional space. A large amount of MDP techniques (such as PCA [182], MDS [146], and T-SNE [113]) have been integrated into VA systems for different data exploration tasks, such as generating maps and exploring dimensions [124, 143]. However, these traditional MDP methods have several drawbacks in real-world usage [164].

A series of neural networks including autoencoder [74], DrLIM [67], and neural network projections (NNP) [57], have been proposed as replacements for traditional dimension reduction models such as T-SNE and MDS [164]. These neural versions of visualization show substantial advantages for VA systems, such as out-of-sample capability, stability, and inference speed [56, 67, 117]. The neural MBP has been used as the visualization method in VA systems for these desired properties. For example, Zexplorer [117] used a pretrained NNP [57] as the interactive DR in SI systems.

Chapter 3

DeepVA: Enhance SI with Pretrained CNN Representations

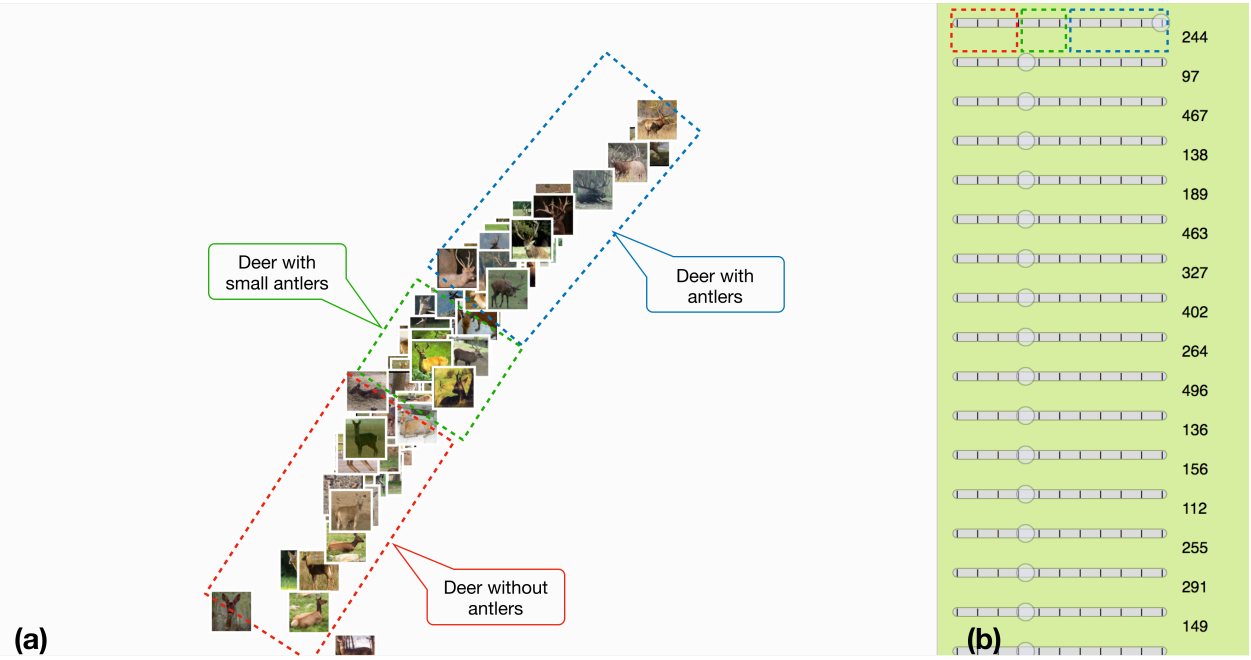


Figure 3.1: DeepVA projection of deer images based on the concept "antler." (a) Images are projected using features in their deep learning representations. (b) The weights of features used in the projection. DeepVA couples cognition and computation through the use of semantic interaction on high-level deep learning features. In this case, DeepVA captures a user's organizing intent about the high-level visual concept "deer with antlers," and maps it to the deep learning feature d_{244} that contains relevant semantic information.

This chapter examines how *deep learning* (DL) representations, in contrast to traditional engineered features, can support *semantic interaction* (SI) in visual analytics **(RQ 1-1)**. SI

attempts to model user's cognitive reasoning via their interaction with data items, based on the data features. We hypothesize that DL representations contain meaningful high-level abstractions that can better capture users' high-level cognitive intent. To bridge the gap between cognition and computation in visual analytics, we propose *DeepVA* (Deep Visual Analytics), which uses high-level deep learning representations for semantic interaction instead of low-level hand-crafted data features. To evaluate DeepVA and compare to SI models with lower-level features, we design and implement a system that extends a traditional SI pipeline with features at three different levels of abstraction. To test the relationship between task abstraction and feature abstraction in SI, we perform visual concept learning tasks at three different task abstraction levels, using semantic interaction with three different feature abstraction levels. DeepVA effectively hastened interactive convergence between cognitive understanding and computational modeling of the data, especially in high abstraction tasks.

3.1 Introduction

Coupling cognition and computation through interactivity is a challenging and important research topic of visual analytics (VA) [38]. *Semantic Interaction (SI)* [49] seeks to enable coupling during sensemaking [133] tasks. To synthesize information, users can manipulate and organize data items to express their reasoning. Analytic methods can then help the synthesis process by systematically observing these user interactions and learning a synthesis model. Systems can respond to the user's cognitive reasoning process by mapping the interactive intents to underlying model parameters on data features [51].

However, a major problem is that it can be difficult to capture users' semantic intents in complex sensemaking tasks because of the gap between high-level cognition and low-level computation. For example, Endert's study [48] revealed a distinction between high-level

cognitive features and low-level data features. Since it typically requires significant cognitive effort to reduce high-level concepts to low-level features through the course of incremental formalization [152], SI models attempt to relieve this requirement by indirectly learning the mapping. Analytic models must transform users' high-level cognitive concepts or topics into low-level data features. But, the low-level data features might not support a clear mapping, making it very difficult to efficiently capture subtle high-level intents.

Simultaneously, recent research on deep learning presents promising solutions to address the challenge of bridging the semantic gap. Deep learning can automatically learn meaningful data features (representations) from a large amount of data, and it outperforms many traditional machine learning techniques in many fields [104]. For example, *Convolutional Neural Networks* (CNN) have been very successful for image recognition tasks [97]. The DL approach has several benefits: it relieves the need to manually engineer good features for the underlying machine learning model; the meaningful representations can better capture high-level concepts needed by the analyst [94]; and, the trained DL model can often be transferred to other applications using similar datasets [136].

Therefore, to overcome the semantic gap challenge, this paper explores the potential use of deep learning features for SI to support visual analytics for interactive data synthesis tasks. We hypothesize that deep learning representations can capture the meaning of users subtle interactive intents in SI. Thus it enables more complex and effective interactive syntheses of the data. **Specifically, in this paper, we investigate how deep learning representations can improve learning from users' *observation-level interactions (OLI)* [51], a particular type of semantic interaction that involves organizing data points in a dimension-reduced projection.** To generalize the effect, we hypothesize that higher abstraction-level features will better support higher abstraction-level tasks with SI.

To explore this hypothesis, we design a system that focuses on providing analysts with OLI

interactions for image sorting and spatial organizing. We extend the SI pipeline model in our designed system by introducing data features at different abstraction levels into the SI process. We examine image feature sets at three different abstraction levels: color histogram (low-level), SIFT features (mid-level), and deep learning representations (high-level). To make a systematic assessment, we perform a 3×3 case study. SI with high-level DL features (SI_{high}, also denoted as DeepVA) is compared to SI with the lower-level engineered feature sets (SI_{low} and SI_{mid}). For each, a synthesis task is performed at three different abstraction levels of visual concepts: "ground" (low-level), "deer" (mid-level), and "deer with antler" (high-level).

The results demonstrate initial evidence for the validity of our hypotheses. SI with DL features (DeepVA) effectively and efficiently supported synthesis tasks at all three task abstraction levels. Because DL representations contain relatively high-level concepts, users' complex high-level intents were more directly mapped to these features with less interactive cost compared to the lower-level feature sets. In other words, DeepVA better optimized the coupling process between cognition and computation.

The contributions of this chapter are:

1. A new SI model, DeepVA, that uses deep learning techniques to enhance VA systems with SI in solving more difficult synthesis tasks.

2. An extended SI system designed for image analysis that provides the evaluation and comparison between SI models with features in different abstraction levels.

3. A systematic evaluation of nine case studies is performed to examine DeepVA's ability to address exploratory synthesis tasks, in comparison to the use of lower-level data features.

3.2 System Design and Implementation

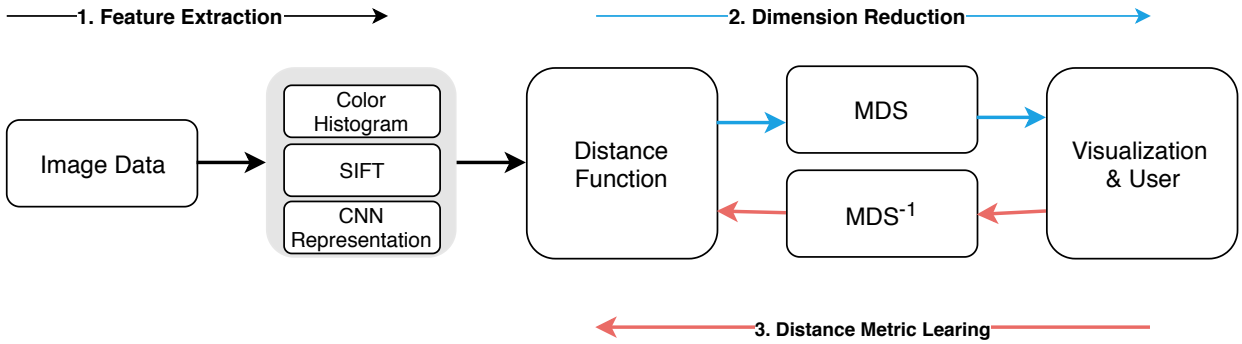


Figure 3.2: The system pipeline is composed of three important components. 1. Feature extraction: the system extends the SI pipeline with features at three different abstraction levels. For image data: F_{low} is the color histogram, F_{mid} is the SIFT feature, and F_{high} is the DL representation; 2. Dimension reduction: projects the data based on the learned distance function; 3. Distance metric learning: updates the distance function based on the analyst's interactions in the visualization.

To explore and evaluate the effectiveness of different abstraction levels of data features in semantic interactions (OLI in particular) on capturing user cognition, we design a system prototype (Fig. 3.2) based on Andromeda [149]. We focus on the process of feature extraction, as well as the other two important components in OLI models (dimension reduction and distance metric learning).

In this paper, the system is designed and implemented for image analysis tasks. However, the model generalizes to tasks, such as document analysis, or numerical data analysis. OLI systems are designed to enable users to create customized projections of high-dimensional data that represent their own expertise. Users express hypothesized similarities by directly manipulating a subset of points in the display. Semi-supervised metric learning then learns a new weighted distance function based on the data features, and updates the projection accordingly. Typically, the user's goal is to create a spatial projection that captures a desired cognitive concept. In the case of image data, that means organizing images according to

important content within the images. For example, an analyst might spatially organize images of animals according to similarities in a biological taxonomy, the physical layout of a zoo, or their own preconceptions about animal similarities.

The main challenge of OLI systems is whether the metric learning can adequately capture the user's intended structure given the available data features. Our hypothesis is that higher-level features, such as deep learning representations, will be better able to capture user's high-level OLI intents.

3.2.1 Feature Extraction

To examine the influence of features at different abstraction levels on OLI models, we focus on exploratory analysis of image data with three commonly used features: color histogram ($\mathbf{F_{low}}$), the Scale-Invariant Feature Transform (SIFT) ($\mathbf{F_{mid}}$), and CNN representation ($\mathbf{F_{high}}$). The first two represent traditional low-level engineered feature sets, while the latter represents high-level learned features. As shown in Fig. 3.2, the system with the three different levels of images features are equivalently used as input features in the SI system. To improve the modeling efficiency and remove unwanted variation, the number of features is limited to 512 in all three feature extraction methods.

$\mathbf{F_{low}}$ - Color Histogram: The color histogram [68] is a representation of the distribution of colors in an image. Color histograms contain only basic low-level pixel information, without any semantic information. Images with an RGB color model were transferred to a color histogram that contains 255×3 dimensions. To reduce the feature size to *512D*, a *Principal Component Analysis* (PCA) [182] dimension reeducation model was performed on the histogram.

$\mathbf{F_{mid}}$ - SIFT: SIFT has been the most widely-used hand-crafted feature generator for

content-based image retrieval in the past decade. Unlike CNN, SIFT only describes local patterns as features [112] in images and only local semantic information is embedded in each dimension. We use OpenCV [86], a widely-used computer vision library, to extract SIFT keypoints and descriptors for each image. However, the features extracted through SIFT are variable size. To fix the number of features to *512D*, *Bag of Visual Words* (BOVW) [189] is used to select the most representative 512 SIFT features for the images.

$\mathbf{F_{high}}$ - CNN Representation: Due to the training requirements of deep neural networks, a huge amount of labels are needed, which is difficult for traditional VA models to supply. Also, deep learning representations are mysterious, analysts could not directly map their intents to representation space. To address these challenges, we applied basic transfer learning in the feature extraction process for CNN representations (Fig. 3.2). We use a pre-trained deep learning model as a fixed feature extractor. We freeze the weights for the entire network except for the final fully connected layer. This last fully connected layer is replaced with a new one with random weights and only this layer is trained by the default statistic model used in OLI models. We use the pre-trained ResNet (ResNet-18 model) [73] on ImageNet [97] (removing the last fully-connected layer) as a feature extractor, a CNN model in the torchvision package, which contains commonly used datasets and model architectures for computer vision, on top of the deep learning platform PyTorch [126]. Other deep learning frameworks, including TensorFlow [2], Caffe [88], and Theano [11], also contain pre-trained deep learning models and could be used to extract DL representations. The high-dimensional representations were compressed to $512D$, using representation transformations in the torchvision package.

3.2.2 Dimension Reduction

OLI systems use dimension reduction models to provide analysts an interactive space and visual feedback about the learned concepts.

Distance function

Similar to previous OLI systems, we use a weighted distance function to capture and reflect analysts' preference and understanding of the similarities between images, based on the extracted features, in a 2D visual projection. Initially, the uniformly weighted Euclidean distance function is used for the default projection:

$$D_w(x_i, x_j) = \sqrt{\sum_{k \in w} w_k * (x_{i,k} - x_{j,k})^2} \tag{3.1}$$

This distance, described in Equation 3.1, is the Euclidean distance between the normalized features of image x_i and x_j, including a weight w applied to each feature that denotes the importance of that feature to the current projection. At system initialization, each of the weights associated with the features in the dataset are set to 1, indicating that each feature has equal contribution to the pairwise distances between images. These weights are updated in response to user interaction in the process of learning a new distance function to capture the users' cognitive intents, detailed in the next subsections.

WMDS

As in each interactive loop, the updated distances between images are mapped to a 2D visual projection. The mapping from representation to visualization helps users make sense of the highly-weighted features, verifying their concepts through the updated visualization layout.

We use multi-dimensional scaling (MDS) together with the weighted distance function to create a weighted-MDS (WMDS) projection of the weighted high-dimensional features into the 2D space [149]. In the projection, MDS seeks to minimize the mean squared error between the 2D and nD pairwise distances.

Visualization

With the dimension reduction model, the similarity relationships between images are shown in the scatter plot visualization Workspace view. The learned weights of features are also visualized as sliders in the Feature view. The Workspace View provides a space to perform image movement interactions (OLI), affording the user with the ability to express semantic concepts. The Feature View displays weights that have been learned for data features.

Workspace view As shown in Fig. 3.1a, the Workspace View is a scatterplot of images. The distance between images reflects their relative similarity as weighted by the underlying analytic models. At first, images scatter in the Workspace based on an equally-weighted dimension reduction of the high-dimensional features. If two images are positioned close to each other in this initial projection, it implies that these two images are likely similar based on all data features. There are three interactions provided in the Workspace for users to explore and view the images. Users can drag images to modify the projection and trigger the learning of new weights. When a user hovers on an image, the values for each feature dimension will be displayed in the Feature View with a yellow circle glyph, and the hovered image will be highlighted with a yellow border. Clicking a single image or drawing a box around multiple images, causes the images to be highlighted with blue borders, and also displays the associated feature values with blue circles on the Feature view sliders.

Feature View This view (Fig. 3.1b) displays the weighted data features. For DeepVA, DL

representations extracted from transfer learning are used as input features. These features are visualized by interactive sliders. The weight applied to a feature is mapped to the knob position of its slider. The weights can be directly modified by dragging the knobs. Updated weights will be automatically applied to update the image projection in the Workspace View. Additionally, the weights displayed for each feature will update after the user drags images to update the projection via OLI. The purpose of interacting with the sliders is primarily for hypothesis testing, and making sense of abstract features. If users formulate a hypothesis about one feature, they can increase the weight on that feature to test if the new weight creates a layout in the Workspace View that matches the desired outcome of the hypothesis. The updated image layout in the Workspace View can reveal some conceptual meaning about the up-weighted feature. For example, as shown in Fig. 3.1, the feature d_{244} of the DL representation is linked to the semantic concept "antlers," because when d_{244} was increased in weight, the updated Workspace shows that images of deer are linearly organized based on how much antlers the deer have.

3.2.3 Distance Metric Learning

With OLI, users' concepts are expressed by dragging elements closer or farther based on desired similarity. As discussed previously, OLI enables analysts to synthesize concepts by organizing data elements in a 2D visual interface. To capture users' cognition, the underlying model should have the ability to automatically capture, infer and associate users' intents with data features through these OLIs.

$WMDS^{-1}$: Inverse WMDS

Commonly, a distance metric learning method is used to update the distance function based on OLI interactions with 2D position information. The underlying distance metric learning method, as described in Andromeda [149], is executed to infer the user's organizational intent and to recalculate the layout of all images based on the new spatial positions of the dragged images. An appropriate metric-learning method for WMDS projections is WMDS^{-1}, also know as Inverse WMDS. WMDS^{-1} uses a similar optimization as WMDS, but instead re-trains the distance function by updating the feature weights w according to the new distances expressed by the user.

Through the interactive loop, the analyst iteratively provides more feedback to incrementally update the distance function. With more feedback, the distance function will more accurately reflect the analyst's intent.

3.3 Case Study Design

3.3.1 Research Questions

We investigate the following research question: Given OLI, how well do different data feature abstraction levels support OLI in learning different user concept abstraction levels? Specifically, does OLI with high abstract DL representations SI_{high} (DeepVA) improve performance at interactively learning users' higher-level concepts?

We hypothesize that SI_{high} (our proposed method, DeepVA), will facilitate higher-level synthesis tasks better than SI with low-level features (SI with SIFT features SI_{mid}, SI with color histogram SI_{low}), since it offers a better match between high-level cognitive concepts

and high-level computational features. Thus, the high-level features should be able to more easily capture the high-level concepts expressed by the user. SI with low-level features has been well evaluated for different data types [19, 48, 149]. However, there is no study of SI with more abstract learned features. To assess this hypothesis, we compare SI_{high} used in DeepVA with two other variants of SI using lower-level feature sets, SI_{mid} and SI_{low}.

3.3.2 Design Rationale

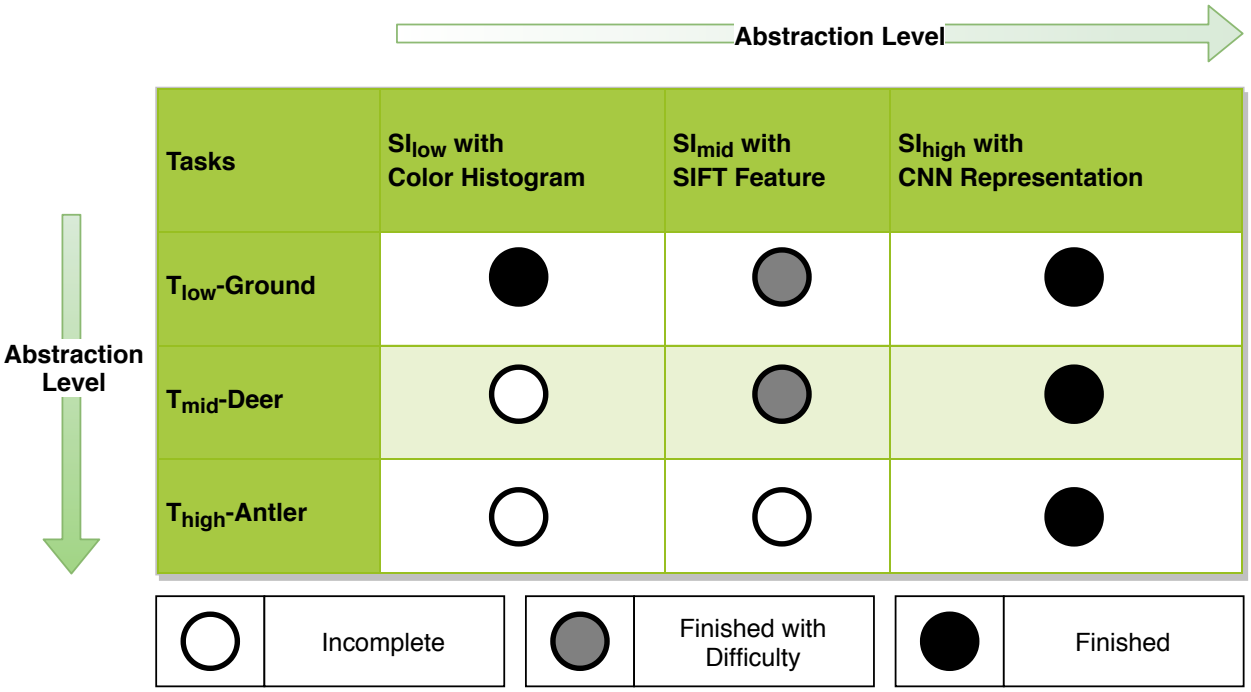


Figure 3.3: A summary of our study design and results: concept learning tasks at three different abstraction levels, and OLI SI methods with features at three different abstraction levels. The results show that SI with higher-level features can accomplish higher-level user tasks (as well as lower-level tasks).

For a systematic evaluation, we perform 9 case studies on visual concept learning tasks at three different conceptual abstraction levels, using OLI SI methods with three different feature abstraction levels. As shown in Fig. 3.3, this 3×3 factorial study helps illuminate

our research questions.

Visual concept learning tasks have been well explored in the field of computer vision. There are three commonly used types of image features with different levels of semantic information: Color Histogram (low abstraction feature); SIFT features (mid abstraction feature); and CNN Representation (high abstraction features). Likewise, visual concepts can be categorized and ranked according to their level of abstraction. For example, the concept "deer with antlers" is more high-level than "deer," because "deer with antlers" can be defined only when "deer" is defined, and requires a more advanced structure of deer images. Whereas, "ground" is conceptually more low-level and more easily maps directly to low-level features like "green". We adapt the visual concept learning task to VA, as it is well representative of many types of sensemaking tasks in which VA users sort and organize images (or other types of information) to synthesize high-level visual concepts.

3.3.3 Data

Studies were conducted on the STL10 dataset [36], which contains 10 classes of objects such as animals and cars. We choose STL10 over other commonly used datasets, such as CIFAR-10 or MNIST, for two reasons: images in STL10 have higher resolution (96x96 pixels), that can help users and DL representations to detect subtle visual concepts, such as antlers; images in STL10 have a variety of simple scenes, such as "blue sky" and "green ground," in addition to complex objects.

3.3.4 Task Design

As shown in Fig. 3.3, we designed visual concept learning tasks at three different difficulty levels. For each task, with selected images loaded in our system, we attempted to express the

desired visual concept by using OLI to drag a subset of the images to organize a representative 2D structure. For each, we tried interacting with subsets of different sizes, until the underlying model was able to effectively learn the desired concept, which we could recognize by how well the rest of the images were organized by the model and whether it matched the desired concept. A better SI feature set should require fewer interactions by the analyst to express the desired concept to the underlying model. In general, SI users want to effectively train their models with as few interactions as possible.

TASK_{low}-Ground: "Ground" is simple and basic concept that should only need local features to capture well, since it can be directly described by basic color features. 100 images are loaded into visual interface, including 50 images about green ground, and 50 images about blue sky or sea.

TASK_{mid}-Deer: "Deer" is a complex object-level concept that requires many local features to be combined. 100 images are used, including 50 images about deer, and 50 images about birds. We used images of deer and birds since they have similar backgrounds containing trees, thus not falling into a trap like $wolf \approx snow$ [139].

$\text{TASK}_{\text{high}}$-Antler: To design an even more complex task, we choose the visual concept "antler" that contains more subtle concept refinements, since it has similar shapes as background trees, and also must be connected on the head of a "deer." In this task, 100 images are used, including 50 images about deer with antlers, and 50 without antlers.

3.4 Qualitative Results

We describe in detail three of the case studies about TASK_{mid}-Deer, using SI_{high}, SI_{mid}, and SI_{low}. The other 6 studies about TASK_{low} and TASK_{high} are included in the quantitative

evaluation results in Fig. 3.3 and 3.7.

3.4.1 $\mathbf{TASK}_{mid}$: $\mathbf{SI}_{high}$ with CNN representation

In this method, the underlying analytic model used CNN representations to capture and infer the analyst's goal, "deer", based on OLI interactions.

Initial Layout: In the initial default projection 3.4a, we can identify that there is already a clear boundary between deer and birds within the projection, even though they have similar backgrounds such as trees or green ground. We conclude that many of the learned features, when equally weighted, work together to classify deer from birds.

Interactive Image Movement: To better express the concept "deer," we interactively group three images of deer to the top right region of the projection, indicated by the blue arrows in Figure 3.4a, and three images of other concepts far away from the deer cluster, indicated by the red arrows. Then we click the "Update" button to update the image layout and representation feature weights based on the moved images.

Updated Layout: After the layout updates, we identify that there are two clusters in Fig. 3.4b. The bottom left cluster circled in red contains images of "birds," while the right cluster circled in blue contains images of "deer." The Representation View shows that there are at least three features of the CNN representation (d_{10}, d_{391}, and d_{511}) which have increased in weight. This indicates that the visual concept "deer" and these three representation features are highly related.

Manipulating Representation Weights: To further test this hypothesis, we increased and decreased the weight on the top three dimensions: d_{10}, d_{391}, d_{511}. We found that d_{10} could classify deer from other images, and so we maximized the weight of d_{10} to further update the layout and examine the relationship between this dimension and the concept

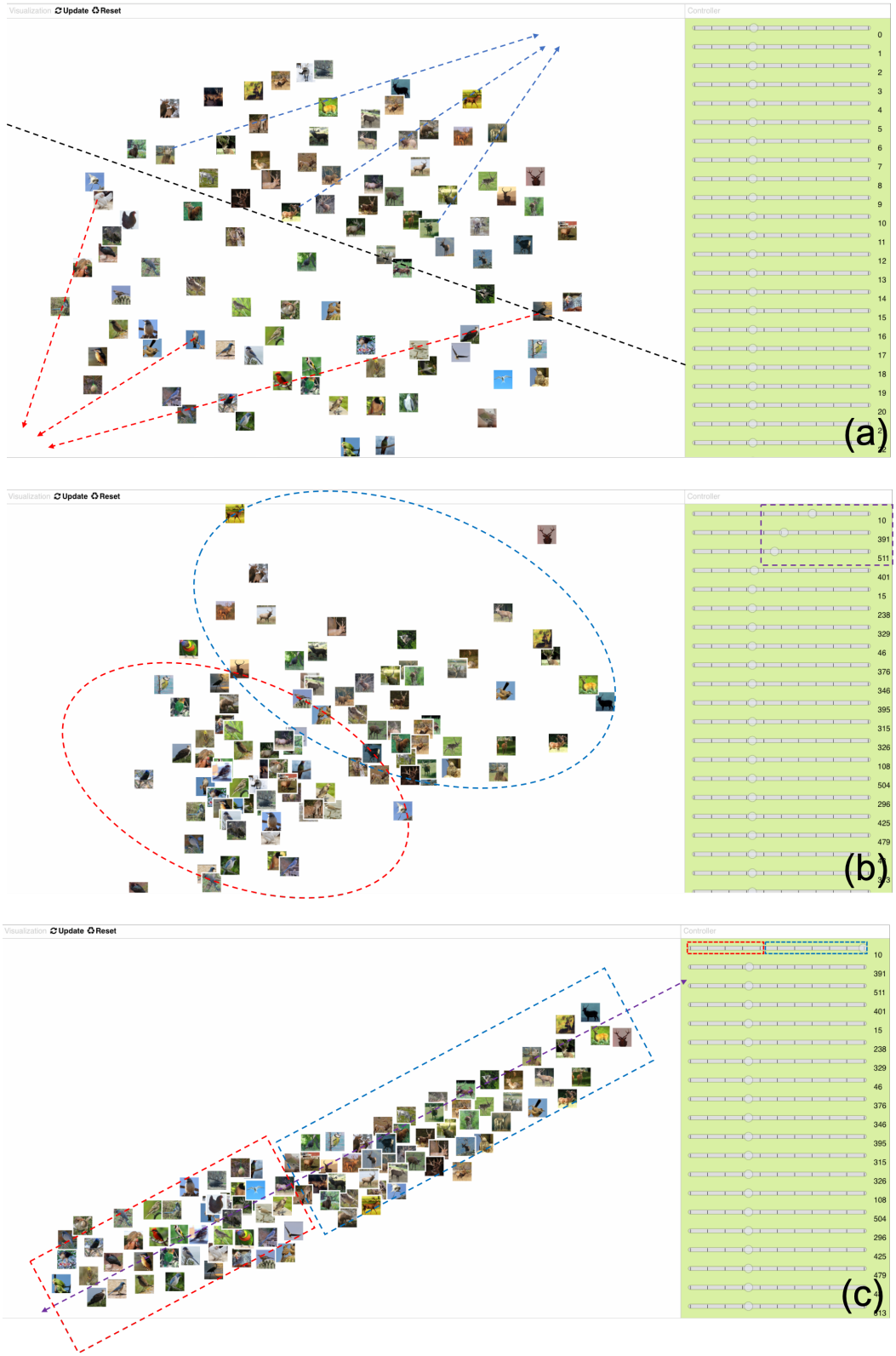


Figure 3.4: Several screenshots during the synthesis of the visual concept "Deer" using SI_{high}.

"deer."

Conclusion: As shown in Fig. 3.4c, we found that there is a linear relationship between representation dimension d_{10} and the visual concept "deer." This shows DeepVA (SI_{high}) could easily and effectively capture this user concept, and relate it to CNN representations.

3.4.2 TASK_{mid} : SI_{mid} with SIFT features

Initial Layout: Initially, the default projection (Fig. 3.5a) shows all deer and birds are scattered in the Workspace, there is no clear boundary separating deer from others. It makes sense because SIFT features contain local patterns, which cannot directly distinguish two object-level concepts.

Interactive Image Movement: Similarly, we drag three images of deer to the top right region on the view, indicated by the blue arrows in Fig. 3.5a, and three images without deer to the bottom left region, indicated by the red arrows, to update the model and layout.

Updated Layout: After the layout updates, we find that all deer images are grouped in the center of the view, however, mixed with some bird images (Fig. 3.5b). We then drag six images in the cluster, to better distinguish the deer in the center cluster from others, then update layout again.

Manipulating Feature Weights: The updated layout (3.5c) shows a better cluster of deer. Though, there are still a small number of images about birds, especially birds on trees, mixed in the deer group. Then we test if the up-weighted features could be associated with the deer concept, and found that several features combined helped to group deer, but not any one particular Sift feature. Then more interactions are performed until a clear boundary is shown between "deer" and others (shown in Fig. 3.7).

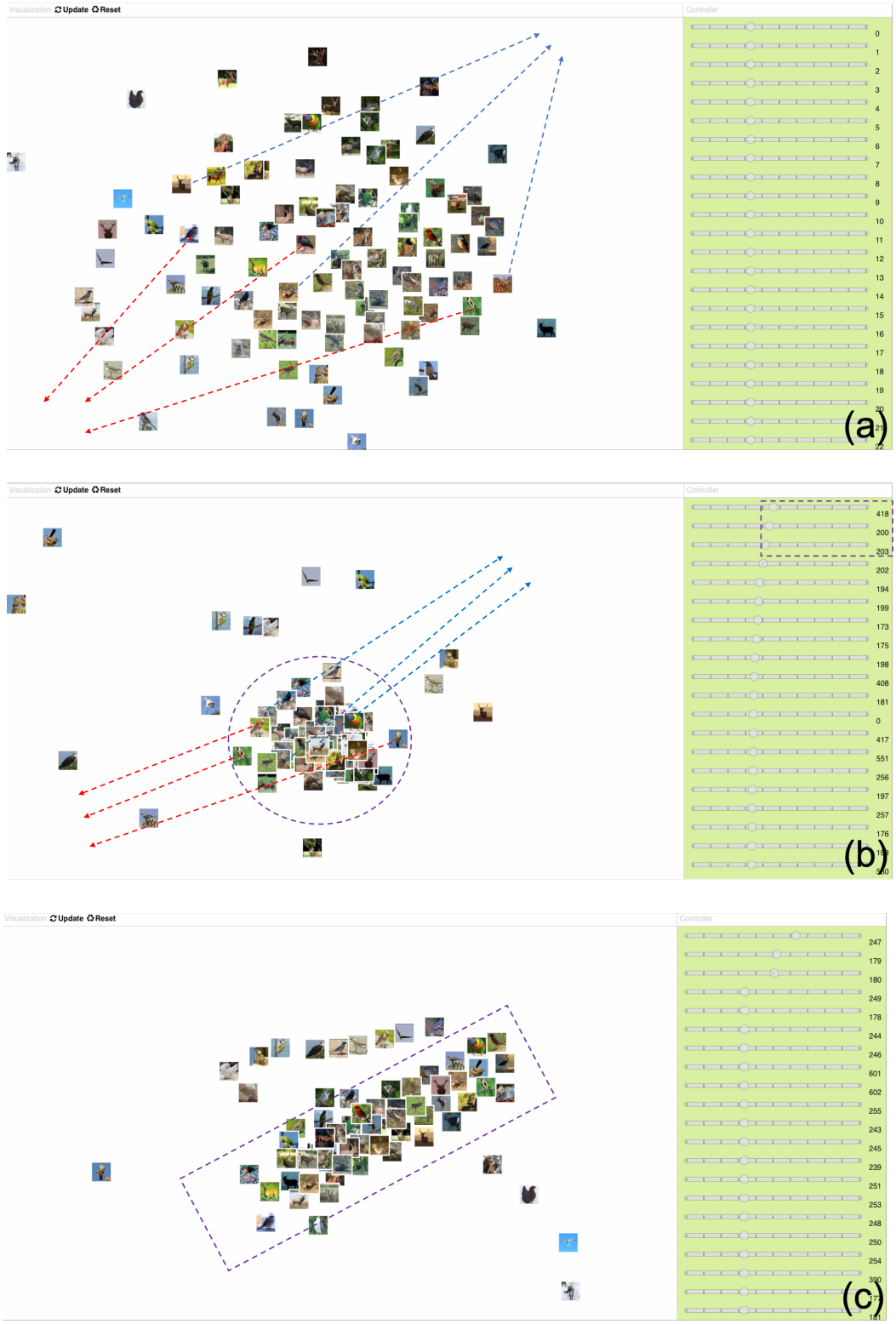


Figure 3.5: Several screenshots during the synthesis of the visual concept "Deer" using SI_{mid}.

Conclusion: We conclude that with SIFT features, the local features "trees" and "antlers" are similar, which leads to the mix of "birds in trees" and "deer." To better distinguish, more interactions and complex combinations of features are needed.

3.4.3 $\mathbf{TASK}_{mid}$: $\mathbf{SI}_{low}$ with Color Histogram

Initial Layout: As in Fig. 3.6a, images are projected by default based on colors instead of objects.

Interactive Image Movement: We selected and moved 6 images: three deer images to top right, indicated by the blue arrows in Fig. 3.6a, and 3 other images to bottom left, indicated by the red arrows.

Updated Layout: The updated layout (Fig. 3.6b) shows that images with similar colors are still closer to each other regardless of whether it contains deer. Furthermore, all images gather together in the center, making it difficult to visually classify deer from others. To better express the concept, we drag 6 more images and update the layout (Fig. 3.6c). The projection is worse, almost all images cluster together. Even with many more interactions performed, until giving up, there is still no evidence of a better structure about the concept "deer."

Conclusion: This evidence suggests that synthesizing complex concepts is difficult or impossible with only the basic data features in color histograms.

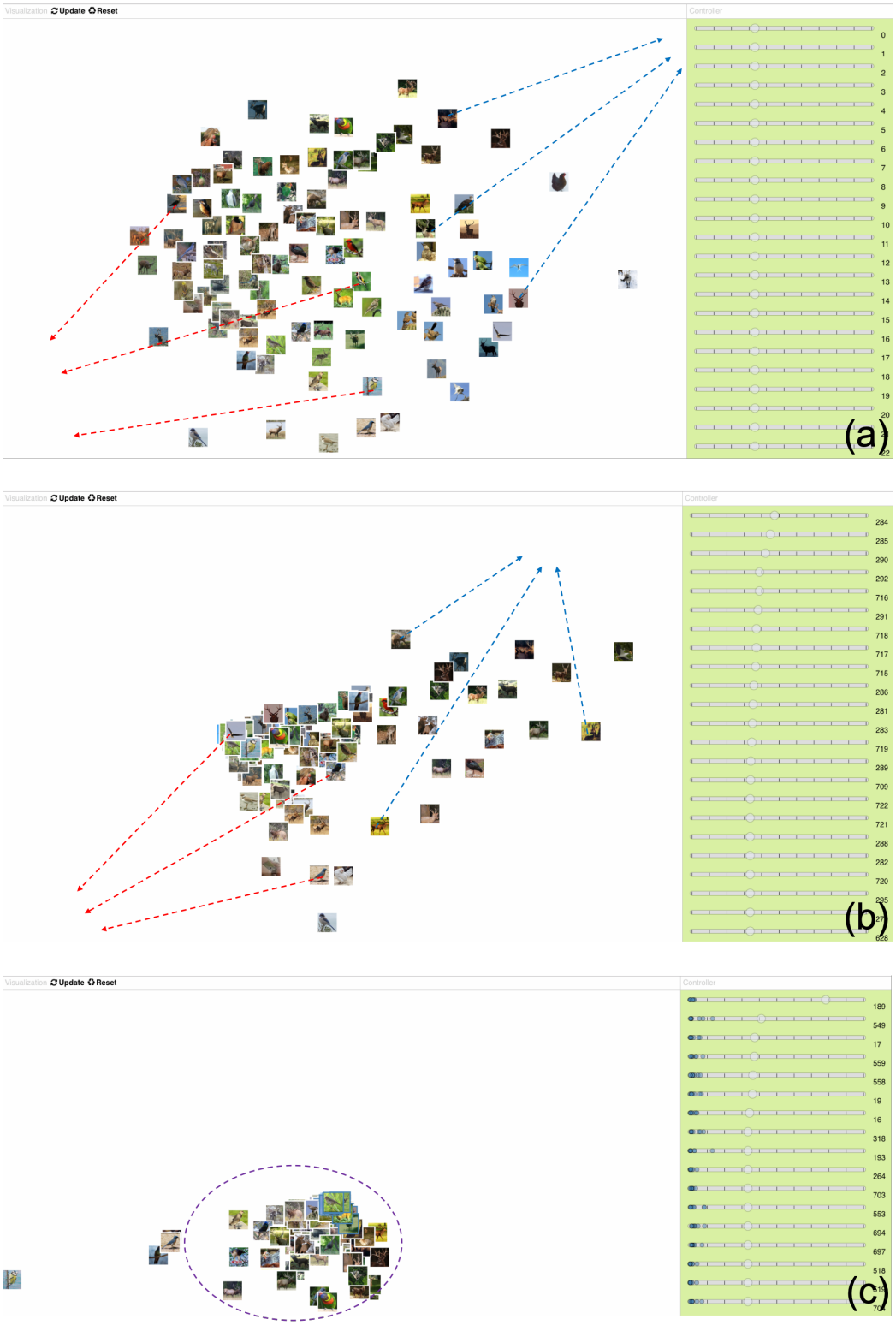


Figure 3.6: Several screenshots during the synthesis of the visual concept "Deer" using SI_{Low}.

3.5 Quantitative Results

3.5.1 Measures and Metrics

In addition to the 3 case studies detailed above, another 6 studies were also carried out for Task_{low} and Task_{high}. We evaluate the effectiveness of DeepVA (SI_{high}) through the 9 studies, compared with the other two methods in terms of two aspects: completeness, and interactive cost.

Completeness

As shown in Fig. 3.3, for a specified task with a specified SI method, a circle is used to represent the task completeness. A white circle means we could not finish the task through the given method. A gray circle means we could finish the assigned task to some degree of precision, but the concepts in the image projection were still not completely delineated. A black circle means we could readily complete the task with the given SI method.

DeepVA (SI_{high}) helped complete tasks across all three difficulty levels. SI_{mid} mostly solved the two easier tasks (Task_{low} and Task_{mid}) with less complex concepts, but more interactive effort was required. SI_{low} only completed Task_{low} with a simple concept.

Interactive Cost

We recorded the number of user interactions (both image and feature movement) performed to complete each synthesis task as shown in Fig. 3.7. If we gave up before successfully completing the task, interaction cost is marked as **infinity** ∞. The results indicate that DeepVA helped complete tasks with fewer interactions, in comparison to the other two

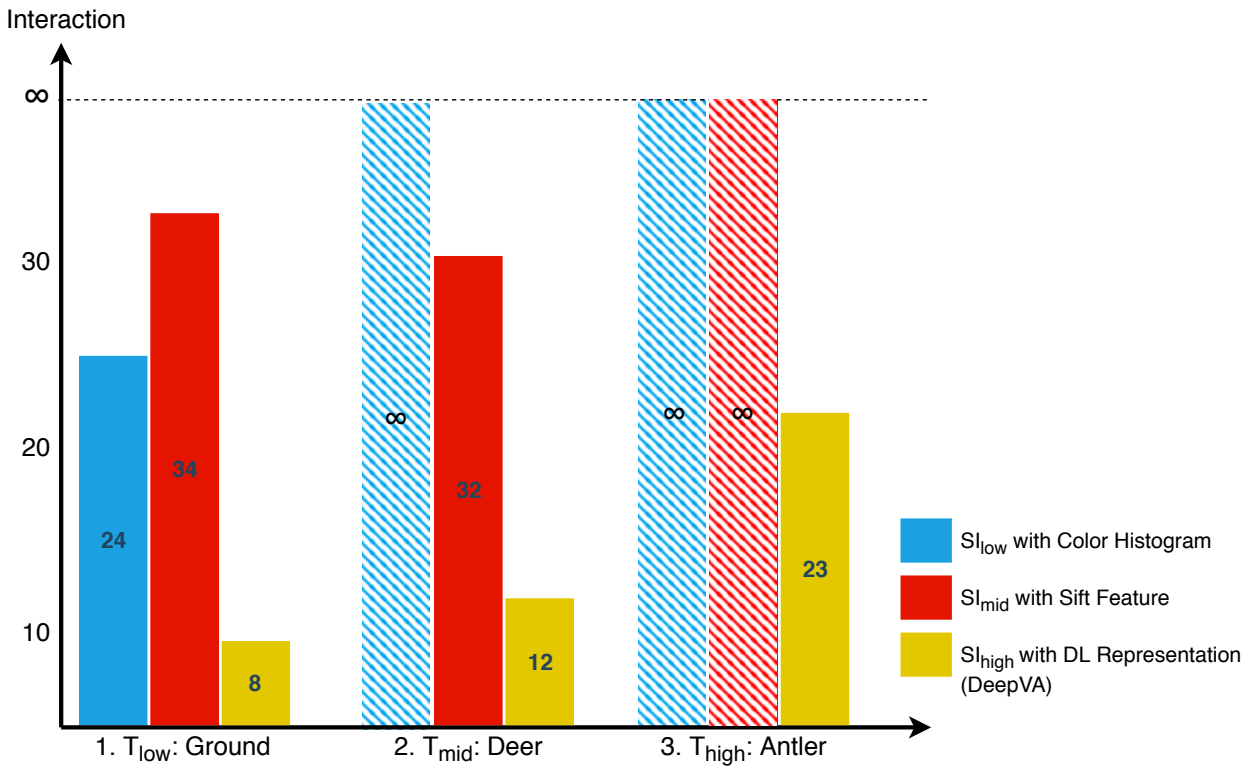


Figure 3.7: Interactive cost: number of interactions required to complete the task with the assigned method.

methods. For a more complex task, more interactions were needed. For SI_{mid}, in both $Task_{low}$ and $Task_{mid}$, more than 30 interactions were needed. It is interesting that for $Task_{low}$, SI_{mid} required more interactions than SI_{low}.

3.5.2 Summary of Results

DeepVA (SI_{high}) is more effective

The result in Fig. 3.3 shows an upper triangular structure that well-matches our hypothesis regarding the match-up of task and feature abstraction levels. DeepVA is able to effectively complete all three tasks. This indicates that higher-level features can enable users to efficiently synthesize higher-level concepts (as well as simpler low-level concepts) using OLI.

Even for Task$_{high}$, with the most complex concept "antler," DeepVA managed to map it to one representation feature (d_{244} as shown in Fig. 3.1 and Fig. 3.8). This can greatly reduce users' interaction effort during sensemaking, and narrow the gap between complex cognitive concepts and high-dimensional computational features. In contrast to DeepVA, SI$_{mid}$ and SI$_{low}$ cannot adequately capture the complex concepts expressed in Task$_{high}$ with lower-level features. Even in Task$_{low}$ and Task$_{mid}$, several features together were needed to define the visual concept and more interactive training was required. Analysts may need to carefully tweak the combination of features to work for the desired concept. This clearly answered our research question that DeepVA (SI$_{high}$) can improve SI performance at synthesizing concepts at all levels.

Figure 3.8: The most and least similar pictures of 'antler' concept over d_{244}. The upper row of nine images are deer with the largest values of d_{244}, while the lower row contains the nine deer images with the smallest values of d_{244}. We can conclude that large values of d_{244} imply the presence of antlers.

DeepVA (SI$_{high}$) is more efficient

As shown in Fig. 3.7, in each case study, DeepVA required fewer interactions than other methods. OLI with higher-level features, is more efficient in capturing users' synthesized concepts. The number of interactions used in Task$_{high}$ with DeepVA is less than the other two methods used in simpler Task$_{low}$ and Task$_{mid}$. Furthermore, the relationship between interactive cost and the task complexity in DeepVA is more consistent than the other meth-

ods. That is, the more complex a task is, the more interactive cost is needed. The other two methods seem to be more tuned to a specific task. SI_{low} performs best on Task_{low}, while SI_{mid} performs best on Task_{mid}.

3.6 Discussion

Through these studies, we find that OLI with higher-level features can capture more complex concepts more efficiently. OLI with deep learning representations (DeepVA) can effectively and efficiently help users to synthesize complex concepts. This also means transfer learning does extract meaningful and appropriate representations from the given image dataset, using the pre-trained deep learning model. In addition to these research results, we also find the following insights.

3.6.1 Learned Features as Representative of Cognition

Through these case studies, we found that all three visual concepts of different complexity can be directly mapped to one feature of DL representations. In Task_{low}, the concept "ground" can be defined by d_{184}. In Task_{mid}, the concept "deer " is directly mapped to d_{10} (Fig. 3.4c). In Task_{high}, the concept "antler" is linked to d_{244} (Fig. 3.1). Cognitive intents could be directly mapped to deep learning features. This suggests a potential match-up between cognitive and computational DL features.

It was somewhat surprising that DeepVA effectively captured cognitive concepts at all three levels of complexity, not just the high level. The match-up not only explains the effectiveness of DeepVA, but also illustrates the internal learning process of DL to discover data representations at multiple levels of abstraction. With a huge amount of data labelled for specific

tasks such as feature detection or classification, deep learning techniques can automatically discover the best representations for the tasks. Through this process, DL compresses and distributes data from input layers into later layers. In the ImageNet dataset, many kinds of images are labelled in many kinds of categories as training data, spanning many kinds of cognitive tasks at many different levels. Thus, the learned DL captures all relevant concepts across different complexity levels, such as "ground", "deer", and "antlers".

3.6.2 Coupling cognition and computation with DeepVA

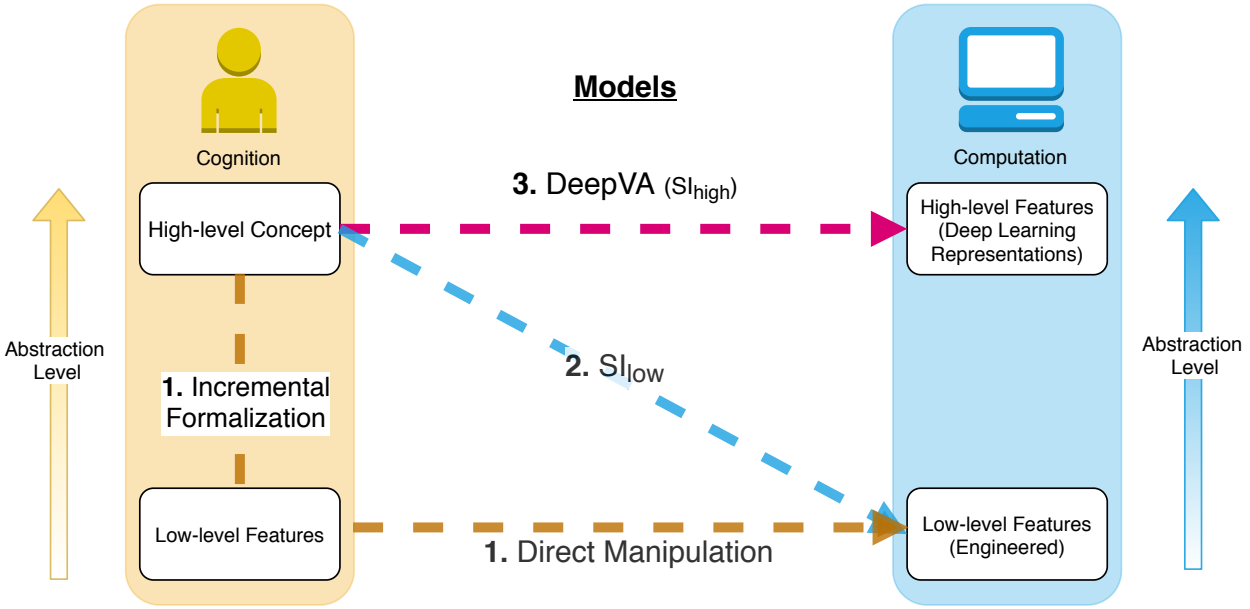


Figure 3.9: Coupling cognition and computation through SI methods: (Method 1) Users internally map high-level concepts to low-level features, then directly manipulate those engineered features. (Method 2) SI_{low} maps user's high-level concepts to engineered low-level features, via various SI methods. (Method 3) SI_{high} (DeepVA) maps user's high-level concepts directly to high-level learned features, via various SI methods.

Deep learning techniques could greatly enhance SI systems. We analyze the advantage of SIs with deep learning from the perspective of bridging the gap between cognition and computation for complex tasks. As shown in Fig. 3.9, users' cognitive reasoning can be represented

through a range of abstraction levels. Through the process of incremental formalization, users can gradually internally transform their higher-level abstract concepts into specific lower-level features such as keywords or image colors. A similar abstraction hierarchy exists on the computational side, ranging from traditional low-level engineered features, like pixels, to higher-level abstractions like deep learning representations. A goal of VA is to bridge the chasm between cognition and computation via various models.

Traditionally, **direct manipulation** forces users to first formalize their concepts into low-level features, so that they can then directly manipulate the corresponding engineered features. For example, Google requires users to first identify specific keywords to enter, and IN-SPIRE [184] enables users to change the weighting of specific keywords when computing the corpus projection. This overhead is especially problematic when users are already devoting their full cognitive effort to sensemaking tasks. At early stages in the sensemaking process, high-level cognitive concepts can be difficult for users to express and formalize in terms of unfamiliar low-level features.

Semantic interactions such as observation-level interaction help the process by extrapolating from the user's synthesis-related interactions to map the user's high-level cognitive reasoning concepts into the low-level computational feature space. An advantage of this strategy is that it can help users perform incremental formalism [152] by automatically mapping to the low-level features. Endert et al. [48] showed the distinction between user's high-level concepts and computational low-level features in their user study. However, SI is thus inherently limited by the chosen feature set. To date, SI with low-level engineered feature sets (method 2 in Fig. 3.9), have been used to perform mappings from simple linear models (e.g. ForceSPIRE [53]) to more complex non-linear models (e.g. AxiSketcher [99], iGTM [69]). However, regardless of the complexity of the mapping method, the use of low-level feature sets will always limit the inference capability. Like other machine learning

techniques, without enough knowledge pre-loaded into the features, it is difficult to infer users' intents based on visual interactions.

Different from the above two methods, **DeepVA** involves identifying and overcoming this limitation. DeepVA improves the cognitive-computational coupling by using deep learning representations as high-level features, which update the model with higher-level conceptual semantics (method 3 in Fig. 3.9). Unlike lower-level engineered features, deep learning representations could contain more high-level concepts. Thus, even basic linear mappings between DL features and human cognition is valid and enough for SI models.

3.6.3 Interpretable Deep Learning Representations

We mainly explored how VA powered by deep learning can facilitate analysts in complex synthesis tasks. Beyond its usage in VA, this combination can also assist DL researchers in interpreting DL representations by adding human-centered tasks as constraints. DeepVA can help understand the semantic meanings behind the otherwise mysterious DL features. As in the case study, mappings users' interactions onto DL features, and the use of the features to help users explore more about specific semantic concepts, can both help DL researchers. Different image projections can express concepts, and reveal insights about the meaning of the associated up-weighted DL features. Likewise, through DeepVA, users can make sense of a single feature by up-weighting it, and then examining the resulting projection. Semantic meanings of the up-weighted feature could be expressed and understood through the updated Workspace view. For example, to understand the feature d_{244}, we maximized its weight by dragging its slider to the right. Through the updated image layout in Fig. 3.1, we could conclude that the feature d_{244} is related to the visual concept "antler."

3.6.4 Limitations and Future Work

In its current state, DeepVA might not help users with incremental formalism [152], because both the cognitive and computational activity are occurring at higher-levels of abstraction. An open question is to investigate how the high-level DL features can be automatically mapped back to low-level features to assist the user in completing the formalization loop.

DeepVA is highly dependent on pre-trained DL networks through transfer learning techniques. If the dataset for analysis is too different from the big dataset that was used to train the deep learning models, the extracted representations might not contain meaningful concepts that users are interested in. Additional research is needed to enable interactive re-training of the deep learning models that would shift it towards users' concepts of interest. Furthermore, significantly greater scalability is needed. To provide a VA application, supported fully by deep learning models, would require thousands of features for representations. This problem can potentially be alleviated by selecting the most relevant features and layers of DL representations based on the targeted analysis tasks during the transfer learning process. An interesting question for future work is how the selection of DL layers could adapt to human incremental formalization of concepts. For example, lower layers might be selected as the user's concepts become more formalized.

3.7 Conclusion

In this paper, we explored SI powered by high-level DL representations, in comparison to traditional lower-level engineered data features. The revised SI model, called DeepVA, consists of two methods: (1) transfer learning to integrate deep learning into VA, (2) observation-level interaction with high-level DL features to capture and model users' cognitive reasoning

process. To evaluate DeepVA, specifically how SI with higher-level data features overcomes the limitations of SI with low-level features, we implemented an SI system for visual image analysis that enables multiple data feature sets. We then performed 9 case studies on interactive synthesis tasks at three different conceptual abstraction levels, using SI methods with three different feature abstraction levels. Overall, DL features improved SI performance. More generally, higher-level data features better support SI users in constructing higher-level concepts. Results demonstrated that SI with higher-level DL features can better capture users' complex cognitive syntheses with less interactive cost than traditional lower-level data features. As a result, DeepVA is more effective and efficient in supporting interactive sensemaking tasks via SI. Furthermore, DeepVA offers new insight into effectively bridging the gap between human cognition and computational models.

Chapter 4

DeepVA: Enhance SI with Word Embeddings

Semantic interaction (*SI*) attempts to learn the user's cognitive intents as they directly manipulate data projections during sensemaking activity. For text analysis, prior implementations of SI have used common data features, such as bag-of-words representations, for machine learning from user interactions. Instead, we hypothesize that features derived from deep learning word embeddings will enable SI to better capture the user's subtle intents **(RQ 1-2)**. However, evaluating these effects is difficult. SI systems are usually evaluated by a human-centred qualitative approach, by observing the utility and effectiveness of the application for end-users. This approach has drawbacks in terms of replicability, scalability, and objectiveness, which makes it hard to perform convincing contrast experiments between different SI models. To tackle this problem, we explore a quantitative algorithm-centered analysis as a complementary evaluation approach, by simulating users' interactions and calculating the accuracy of the learned model. We use these methods to compare word-embeddings to bag-of-words features for SI.

4.1 Introduction

Semantic interaction (SI) [48, 54] is an interaction technique for non-experts to interact with the underlying algorithms of visual analytics (*VA*) systems [38]. With SI, analysts can focus on reasoning and manipulating data in the 2D spatialization instead of directly interacting with the underlying machine learning (*ML*) models [4]. It is, therefore, the system's responsibility to tune the ML models by capturing users' interactions and inferring the analyst's intent [148]. These intents are then translated to updated model parameters via semi-supervised machine learning techniques (ML) [54], such as metric learning [20]. Hence, analysts can remain concentrated on their sensemaking activities [133].

In this paper, we investigate two questions about SI: (1) Can word embeddings help SI learn user's interactive intents better than traditional features? (2) How can we comparatively evaluate alternative SI models, such as in question (1)?

The human-centred approach [18] is the primary evaluation method for SI systems. The SI system is measured by the utility and effectiveness of the application for end-users through studies of human subjects. For example, ForceSPIRE [52] is a visual text analysis system powered by SI, and its effectiveness was measured by analysts' performance on an intelligence analysis task. This approach is highly dependent on human feedback, which leads to several challenges when comparing the effectiveness of different SI models because: user interactions are difficult to precisely replicate; SI interactions are incremental and iteratively build; does not scale well to compare many model alternatives as needed in ML; and usage differences can mask subtle model performance differences.

To cope with these problems, we design and perform an additional quantitative algorithm-centred analysis that simulates human interactions as a complement to human-centered evaluation. In the evaluation, we use the labelled text datasets as the ground truth, since

there is no ground truth for the user's complex intents and concepts. Then the ground truth is used to simulate user's interactions, and we calculate and compare the accuracy of trained models.

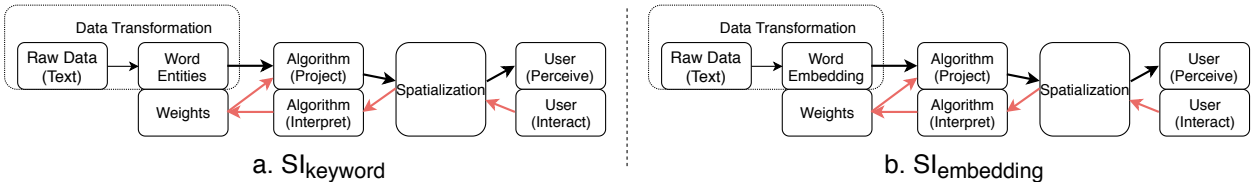


Figure 4.1: In the SI pipeline, distance metric learning interprets users' interactions on the projection. (a) In $SI_{keyword}$, the extracted features of text data are keywords; (b) In $SI_{embedding}$, the features are embedding vectors.

In this paper, we first describe two alternative SI models for visual text analysis using different data features as inputs (embedding vectors, and keywords vectors). We then implement a generalized visual text analysis prototype with SI that can take both embedding vectors and keywords vectors as features. We then demonstrate how the two kinds of evaluation methods together provide a complementary evaluation and comparison of these two SI models.

4.2 Semantic Interaction with Word Embeddings

We investigate the opportunity of using word embeddings to better support SI. SI's ability to infer the analyst's reasoning process is fundamentally limited by the feature space of the underlying machine learning. Text analytics systems with SI, such as ForceSPIRE [19, 52] and Cosmos [47], typically use keywords (such as text terms and phrases) as data features ($SI_{keyword}$ in Fig. 4.1a), known as bag-of-words [198]. Recently, word embedding techniques [37, 120, 131, 192], also known as deep learning representations [7, 104, 150, 191], have shown significant advantages in numerous tasks in natural language processing and information retrieval [98, 103, 156] over bag-of-words features. Therefore, the combination

of SI with word embedding ($SI_{embedding}$ in Fig. 4.1) might enable better learning ability than $SI_{keyword}$ to update the underlying machine learning models [119]. The intuition is that word embeddings could represent more abstract concepts that are closer to modeling human cognitive reasoning. To test the hypothesis, we focus on the comparisons of $SI_{embedding}$ with $SI_{keyword}$ through the same SI pipeline but with different document features as input:

- **Keyword features used in $SI_{keyword}$:** We use TF-IDF values as the keyword features, and word hashing to compress the large number of words from the document collection into 300 dimensions.
- **Embedding features used in $SI_{embedding}$**: We use the pre-trained GloVe model [129] with 300 dimensions to extract embedding features, and use the "basic averaged word embeddings" method to compute average word embedding for all words in a document [103].

4.2.1 Application Prototyping

As shown in Fig. 4.1, $SI_{embedding}$ has a similar structure with $SI_{keyword}$ as they both use numerical vectors to represent documents (embedding vectors, and TF-IDF vectors). We are able to switch between $SI_{keyword}$ and $SI_{embedding}$ in the same SI prototype. We build the prototype upon the foundation of Andromeda [149], a visual analytics tool for exploring high-dimensional data projections. The prototype can use either bag-of-words or embedding vectors as features and update the feature weights to capture analysts' intents. For bag-of-words, the system will up-weight the shared words in the dragged documents in response to the analyst dragging two or more documents closer together. For embedding vectors, the system will up-weight the dimensions of the embedding vectors with similar patterns in the dragged documents.

Model	$\mathbf{T}_{rec}$	$\mathbf{T}_{religion}$	$\mathbf{T}_{sys}$	$\mathbf{T}_{vis}$
$SI_{embedding}$	(**0.921**, **0.812**)	(**0.829**, **0.773**)	(**0.895**, **0.774**)	(0.958, **0.809**)
$SI_{keyword}$	(0.497, 0.570)	(0.576, 0.581)	(0.511, 0.584)	(**0.961**, 0.793)

Table 4.1: Accuracies of $SI_{embedding}$ and $SI_{keyword}$ on each of the four tasks (T_{rec}, $T_{religion}$, T_{sys}, and T_{vis}). Two kinds of accuracies are measured: the average accuracy of the trained model after the last interaction and the average accuracy of the models over all interactions.

As shown in Figure 4.2-1, the visualization is a projection of documents. The distance between documents reflects their relative similarity according to a weighted distance metric over their features. At first, documents scatter in the workspace based on their features and an equally-weighted dimension reduction of the high-dimensional representation. If two documents are positioned close to each other in this initial projection, it implies that these two documents are similar based on all vector features. Semantic interaction enables analysts to directly manipulate the projection by moving the documents to express their own domain knowledge about desired similarities. With this interaction, analysts can express the semantic relationships between documents, which thereby informs the underlying weighted distance model and updates the projection. Through this, the projection can be customized according to the learned intent of the analyst.

4.3 Evaluating SI Embedding

Traditionally, the human-centred approach is a good method to test if the SI model is an effective application for analysts to perform sensemaking tasks. However, comparing two SI models ($SI_{keyword}$ and $SI_{embedding}$) through this approach is inadequate. Thus, we also design a replicable quantitative study to validate $SI_{embedding}$ in comparison to $SI_{keyword}$ in 4 simulated text analysis tasks of different difficulty levels.

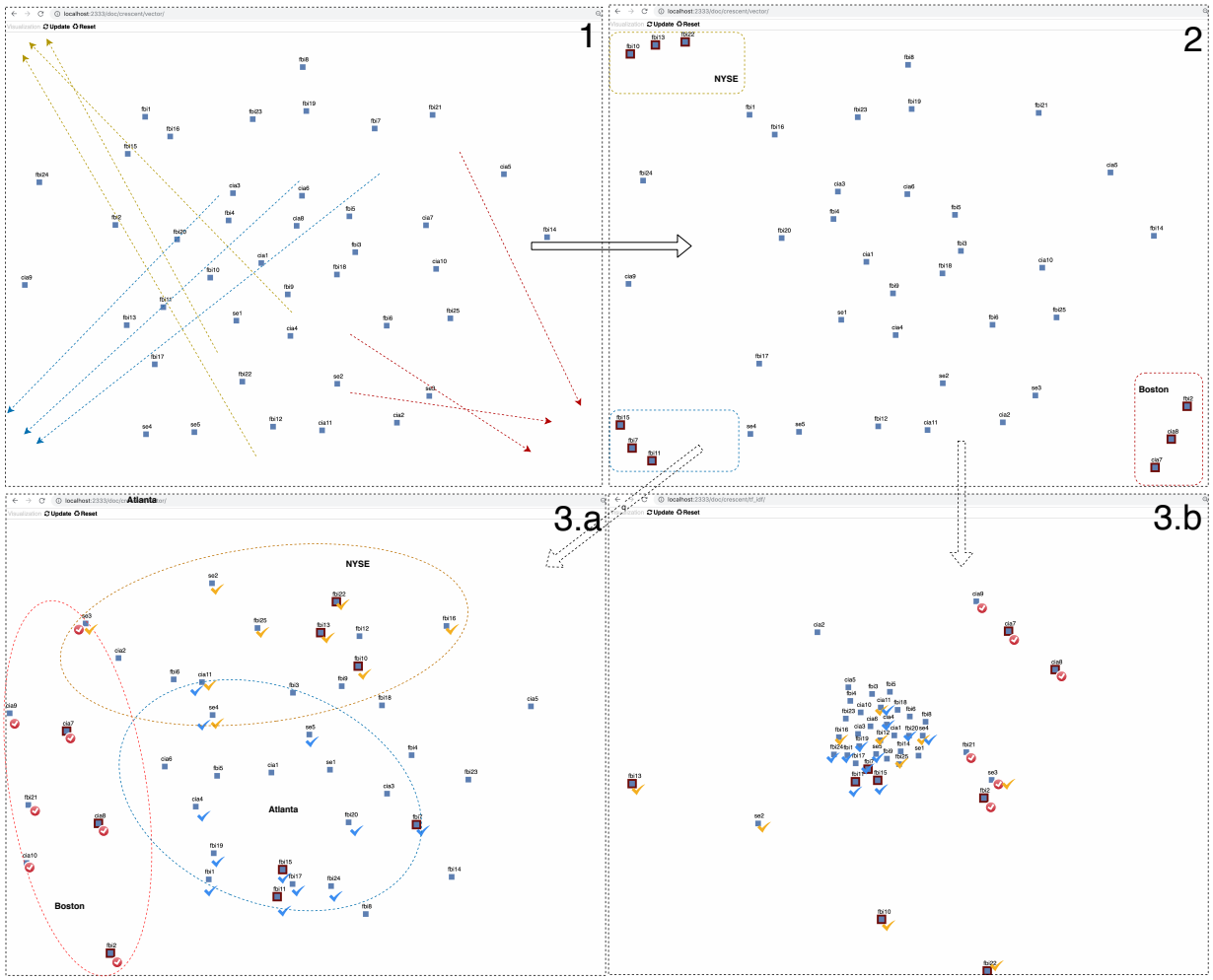


Figure 4.2: Several screenshots during the two case studies, as discussed in Section 5.4.1: **Frame 1 and 2** show the similar initial steps performed by analysts in both case studies. **Frame 3.a** shows the resulting projection based on analysts' interactions, in the case study using $\text{SI}_{embedding}$. **Frame 3.b** shows the resulting projection based on analysts' interactions in the case study using $\text{SI}_{keyword}$.

4.3.1 User-Centered Qualitative Analysis

In this section, we present a practical use case about intelligence analysis as our qualitative study to compare $\text{SI}_{keyword}$ against $\text{SI}_{embedding}$. For this analysis, we engaged an expert in intelligence analysis to review the visual result generated from both $\text{SI}_{embedding}$ and $\text{SI}_{keyword}$ to provide qualitative feedback, and provide grounding for the quantitative analysis.

Crescent Dataset

The crescent dataset [84] has 42 fictional intelligence reports regarding a coordinated terrorist plot in Boston, New York, and Atlanta. Only 24 reports are relevant to these plots. The task for our qualitative analysis is to identify these three terrorist threats by using $\text{SI}_{embedding}$ or $\text{SI}_{keyword}$.

By moving documents in the visualization according to their perceived similarity, analysts express their reasoning process to the system. The ground truth of this task is as follows, which we use to measure the accuracy of the models: **Boston:** cia7, cia8, cia9, cia10, cia11, fbi1, fbi2, fbi21, se3, se4; **New York:** cia11, fbi1, fbi10, fbi13, fbi16, fbi22, fbi25, se2, se3, se4; **Atlanta:** cia4, cia11, fbi1, fbi7, fbi11, fbi15, fbi17, fbi19, fbi20, fbi24, se4, se5; **Irrelevant:** fbi3, fbi4, fbi5, fbi6, fbi8, fbi9, fbi14, fbi18, fbi23, cia1, cia2, cia3, cia5, cia6, se1.

Case study with $\text{SI}_{embedding}$

In this first case study, the analyst performs the task by using the prototype system with embedding as the document features. As shown in Figure 4.2-1, the analyst updates the layout to reflect the perceived similarities between the documents, grouping three documents about "New York" to the top left region of the projection, indicated by the yellow arrows in Figure 4.2-2, three documents about "Atlanta" to the bottom left region indicated by blue arrows, and three documents about "Boston" to the bottom right part indicated by red arrows.

After the layout updates, some semantic relationships are revealed (Figure 4.2-3.a). The left cluster in red circle contains documents about "Boston," the top right cluster in yellow circle contains documents about "NYSE," and the bottom right cluster in blue circles

contains documents about "Atlanta." We found that there are semantic mappings between this updated layout and the ground truth; all relevant reports that belong to single plot are well placed into each cluster. Reports about the coordination between two or more plots are between clusters. For example, the report "se3" contains information regarding both the "Boston" and "NYSE" plots, so the document is located between these two clusters. Thus, $\text{SI}_{embedding}$ can capture analysts' intents and update the model features accordingly. The updated layout then reflects the semantic meanings behind the intents.

Case Study Using $\text{SI}_{keyword}$

Mirroring our case study with $\text{SI}_{embedding}$, the analyst performs similar interactions with the prototype. As shown in Figure 4.2-3.b, from the updated layout, there are no clear boundaries between different plots. Even after continued interactions, the model is unable to properly differentiate between the three terrorist plots. For example, documents about "Atlanta" and "NYSE" are mixed together. Furthermore, the documents that are pulled out of the big central cluster in panel 4 are primarily the user interacted documents from panel 2. This might indicate that the model is over-fitting based on some specific unimportant keywords.

Expert Review

Besides comparing with the ground truth as we mentioned in Section 5.4.1, we also asked an expert familiar with the dataset to evaluate the updated layout generated in both $\text{SI}_{embedding}$ and $\text{SI}_{keyword}$ (as shown in Figure 4.2-3.a and Figure 4.2-3.b). The expert noted that the $\text{SI}_{embedding}$ layout provides more meaningful information about there plots than $\text{SI}_{keyword}$ layout because the documents from different plots are grouped in different regions. In con-

trast, the layout generated by $SI_{keyword}$ makes it difficult to clearly distinguish between these plots. For example, documents in the the $SI_{embedding}$ layout regarding coordination between the plots are placed between the relevant groups. This is exemplified by how the report "se3" is between the "NYSE" group and the "Boston" group since it discusses how terrorists involved in both plots communicated. In the layout produced by $SI_{keyword}$, this document lies at the bottom right of the projection, which does not have such an immediately apparent and relevant semantic meaning.

Additionally, several irrelevant reports were distinguished from the groups in the layout produced by $SI_{embedding}$, such as "cia5", "fbi23" and "fbi14," even though they share many keywords with other relevant reports. In contrast, the layout produced by $SI_{keyword}$ only causes one irrelevant report ("cia2") to be pulled away from other reports.

4.3.2 Algorithm-centered Quantitative Analysis

In this subsection, we describe our quantitative comparison between $SI_{keyword}$ and $SI_{embedding}$. We begin with the experimental design, followed by a description of the datasets we choose for the experiment. Finally, we show the evaluation results and discuss the SI inference abilities of the two models.

Experiment Setup

We need ground truth to evaluate the underlying algorithms used in $SI_{keyword}$ and $SI_{embedding}$ quantitatively. There is no ground truth about users' complex intents and interactions, such as a labeled intelligence hypothesis in a 2D projection. However, there are sufficient text datasets labelled for classification, such as 20newsgroup (section 4.3.2). The labels can be used to simulate different positions in the projection: for example, documents with negative

label can be located at the top left part of the spatial layout, and documents with positive label can be located at the bottom right part. Therefore, we can use classification datasets mapped in the 2D projection as the ground truth, and we only evaluate the inference ability to capture the intents embedded in classification tasks. Then, subsets of the documents from different positions in the 2D projection will be picked and used as simulated semantic interactions from analysts.

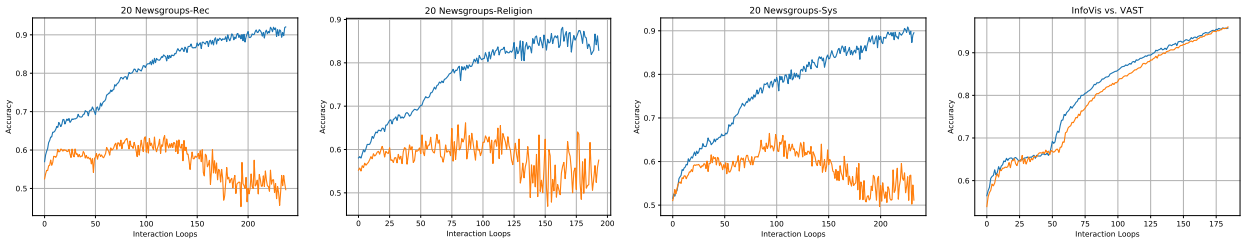
Furthermore, since *incremental formalism* [151] is an important aspect of semantic interaction, we must test the models' ability to learn incrementally over the course of many interactions. To simulate and evaluate the *incremental inference* ability of these models, the selected interactions will be iteratively passed into the SI model. Through these iterative SIs, the underlying models incrementally update to display better results to the analyst.

For example, in the task T_{vis} (sec. 4.3.2), we simulate that the analyst wants to organize documents by separating the two conferences (InfoVis and VAST) in the 2D projection as ground truth: InfoVis documents placed at top left of the projection, and VAST documents placed at bottom right. Then in each loop, five documents from InfoVis collection, and five documents from VAST collection will be picked and simulated as semantic interactions: moving five documents from InfoVis to $(0, 0)$ position and moving another five documents from VAST to $(1, 1)$ position. In each iteration, the SI model ($SI_{embedding}$ or $SI_{keyword}$) is executed to infer updated model parameters from the simulated semantic interactions. Then the inference ability of each model can be calculated by the quality of the underlying model in classifying InfoVis documents from VAST based on the ground truth.

Datasets and Tasks

To perform this quantitative analysis, we use two datasets: 20News dataset [102] and Vispubdata dataset [85]. The 20 Newsgroup dataset is a collection of newsgroups posts on 20 topics. Based on this dataset, we create three tasks to simulate users' intents. Each of the 3 tasks are to separate the documents into 2 groups based on 2 pre-determined topics: $\mathbf{T}_{rec}$: 594 documents from "rec.autos" and 600 documents "rec.motorcycles"; $\mathbf{T}_{sys}$: 578 documents from "comp.sys.mac.hardware" and 592 documents from "comp.sys.ibm.pc.hardware"; $\mathbf{T}_{religion}$: 379 documents from "talk.religion.misc" and 599 documents from "soc.religion.christian".

The Vispubdata dataset contains information on IEEE Visualization (IEEE VIS) publications. We select the abstracts of the academic papers published in two conferences from the dataset: InfoVis (IEEE Information Visualization), and VAST (IEEE Visual Analytics Science and Technology). The documents from different conferences are used to represent different concepts (users' intents), We create one task based on these abstracts to separate InfoVis papers from VAST papers, defined as T_{vis}, including 397 papers from InfoVis and 531 papers from VAST.



(a) T_{rec} (b) $T_{religion}$ (c) T_{sys} (d) T_{vis}

Figure 4.3: The accuracies of both $\mathrm{SI}_{embedding}$ (blue) and $\mathrm{SI}_{keyword}$ (orange) over each interaction across the four tasks (T_{rec}, $\mathrm{T}_{religion}$, T_{sys}, and T_{vis}).

Results

In this experiment, we have run the $SI_{embedding}$ and $SI_{keywrod}$ model multiple times on the four tasks with ground truth. After each interaction (involving five document movements), we calculate the model accuracy using the k-nearest-neighbour (kNN) classifier [39] (as done in Dis-Function [20]): using the cross-validation over the data and set k to 3. The average accuracy of the two models over all iterations for fours tasks is shown in Figure 4.3.

Table 4.1 shows the average accuracy of the last interaction (performed multiple times) and average accuracy of $SI_{embedding}$ and $SI_{keyword}$ across the four tasks. $SI_{embedding}$ achieved the highest scores in the first three tasks, indicating that $SI_{embedding}$ provides more accurate representations of the documents after the interactions, indicating its higher inference ability over $SI_{keyword}$. In T_{vis}, $SI_{embedding}$ and $SI_{keyword}$ have similar accurracies, meaning their inference abilities were roughly equal in this task.

As shown in the Figure 4.3, we further analyzed the incremental updates from the two models by evaluating each model's accuracy after every iteration, thereby simulating the incremental analysis process. In the first three tasks, it is shown that using $SI_{embedding}$ can get better performance than $SI_{keyword}$ over all the iteration loops. The accuracy of $SI_{embedding}$ generally increases with each interaction, showing incremental changes to the model's representation of the user's intent over time. In contrast to the first three tasks, however, in T_{vis}, both $SI_{emebedding}$ and $SI_{keyword}$ have similar accuracy over the interactions. Finally, the desired accuracy should be as close to 1 as possible. T_{vis} is a relatively "easy" task, as reflected by the final accuracies of both models close to 1.0. These results indicate that $SI_{keyword}$ performs well in relatively easy tasks.

4.3.3 Discussion

Our experimental results substantiate our hypothesis that $\text{SI}_{embedding}$ is more effective than $\text{SI}_{keyword}$ for modeling user intent, and better supports incremental formalism based on users iterative interactions. The human-centered qualitatively evaluation method shows $\text{SI}_{embedding}$ is more effective than $\text{SI}_{keyword}$ in real world analysis tasks. This provides a direct evaluation of SI models by the observed utility and effectiveness for end-users. The quantitative analysis offers a complementary approach, and provides replicable and scalable evaluations for SI models. It provides more stable and detailed feedback about the SI model performance in tasks of different difficulty levels. Thus, we confirmed that word embedding can better support SI by better capturing the users' high-level interactive intents.

4.4 Conclusion

In this work, we presented $\text{SI}_{embedding}$ as an alternative to the traditional bag-of-words model ($\text{SI}_{keyword}$) often used in visual text analytics systems. To make a complete and convincing comparison between $\text{SI}_{embedding}$ and $\text{SI}_{keyword}$, we performed a quantitative evaluation by simulating analysts' interactions and calculating the accuracy of the underlying trained ML models, as a complement to the traditional user-centered qualitative evaluation. Results indicate that deep learning distributed representations, such as word embedding, can be exploited to improve interactive visual analytics methods such as semantic interaction.

Chapter 5

DeepSI: Interactive Deep Learning for Semantic Interaction

In this chapter, we explore the **RQ 2** and design novel interactive deep learning methods to improve semantic interactions in visual analytics applications. The ability of semantic interaction to infer analysts' precise intents during sensemaking is dependent on the quality of the underlying data representation. We propose the $\text{DeepSI}_{\text{finetune}}$ framework that integrates deep learning into the human-in-the-loop interactive sensemaking pipeline, with two important properties. First, deep learning extracts meaningful representations from raw data, which improves semantic interaction inference. Second, semantic interactions are exploited to fine-tune the deep learning representations, which then further improves semantic interaction inference. This feedback loop between human interaction and deep learning enables efficient learning of user- and task-specific representations. To evaluate the advantage of embedding the deep learning within the semantic interaction loop, we compare $\text{DeepSI}_{\text{finetune}}$ against a state-of-the-art but more basic use of deep learning as only a feature extractor pre-processed outside of the interactive loop. Results of two complementary studies, a human-centered qualitative case study and an algorithm-centered simulation-based quantitative experiment, show that $\text{DeepSI}_{\text{finetune}}$ more accurately captures users' complex mental models with fewer interactions.

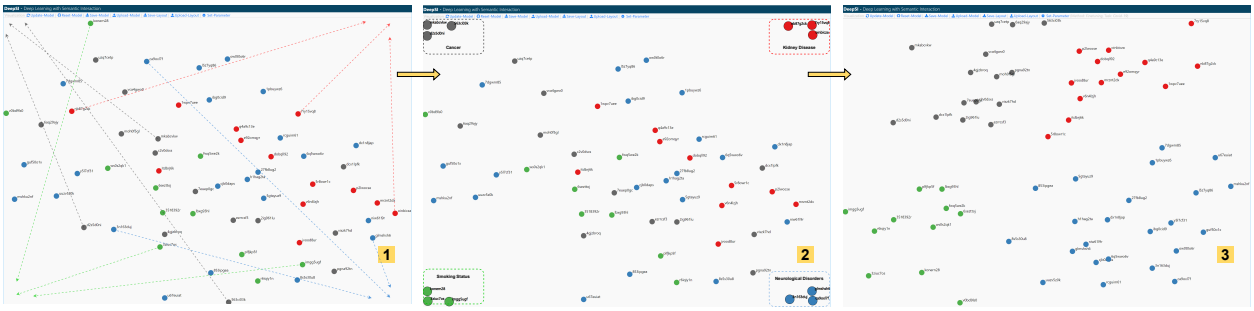


Figure 5.1: Screenshots during the analysis of COVID-19 research articles about four risk factors (depicted in different colors) using our proposed model DeepSI$_{\text{finetune}}$: (1) the initial layout of all articles projected from pretrained BERT representations of the raw text data; (2) the analyst performs semantic interactions to provide visual feedback regarding articles about different risk factors; these interactions are then exploited to tune the underlying DL model BERT; (3) the resulting projection updated by the tuned BERT.

5.1 Introduction

Semantic interaction (SI) [49, 53] is an interaction methodology that is commonly utilized to enhance visual analytics (VA) systems. SI-enabled systems let the analyst directly manipulate interactive projections of data [148]. The semantic meaning behind these projection interactions is the similarity relationships the analyst wishes to find within the data during the sensemaking process [133]. As shown in Fig. 5.1-2, the analyst drags 12 COVID-19 article points into four clusters to provide the visual feedback of grouping articles based on their perceived relevant risk factors. With these intuitive and natural interactions, the analyst can remain within the cognitive zone [65], thereby enhancing the analyst's efficiency in performing analytic tasks [178]. In the system, an interactive dimensionality reduction (DR) component [143, 178] plays a key role in capturing the analyst's intent behind these interactions by learning a new projection layout (Fig. 5.1-3). To determine the analyst's precise intent, increasingly powerful interactive DR models [178] have been proposed, from linear [80, 81, 106] to non-linear models [99, 117], and from single-model to multi-model approaches [19, 46, 47, 175].

However, the ability of semantic interaction to infer analysts' precise intents during sensemaking is dependent on the quality of the underlying data representation. Deep learning (DL) [104] is a state-of-the-art representation learning method [10], which can automatically extract abstract and useful hierarchical representations from raw data [10]. This offers the new opportunity to power SI in capturing the analyst's intent. We denote the DL-enhanced SI system as **DeepSI**. Previous researches have shown that even the usage of the pretrained DL representations as fixed data features have better performance than hand-crafted features in SI-enabled VA systems [13, 14]. We denote this straightforward DeepSI design with the basic use the pretrained DL as only a feature extractor in SI pipeline as $\text{DeepSI}_{\text{vanilla}}$.

In this paper, we aim to further improve semantic interaction inference by fine-tuning the model to obtain user- and task-specific representations from the pretrained DL model. Central to this design goal are two research questions:

- *How to exploit semantic interactions to accurately adapt the pretrained representations to current analytic tasks?*
- *How to make efficient adaptations, so that a small number of semantic interactions are enough for analysts to express their intents?*

To address these two questions, we propose a novel DeepSI framework, $\text{DeepSI}_{\text{finetune}}$, with the following two design goals. First, we insert the interactive DL training into the bidirectional structure of the semantic interaction pipeline, so that interactions trigger the DL adaptation. Thereby, new user- and task-specific representations are generated based on semantic interactions provided by analysts during their sensemaking process. Second, we employ the fine-tuning based DL adaption approach and the MDS-based interactive DR model to minimize the number of parameters that require training in the underlying model.

Therefore, $\text{DeepSI}_{\text{finetune}}$ can tune the DL model efficiently from the analyst's interactions without information loss. Specifically, we use the pretrained BERT [42], a state-of-the-art DL model for NLP tasks, as the DL model representative inside $\text{DeepSI}_{\text{finetune}}$ for visual text analysis tasks.

To assess how well $\text{DeepSI}_{\text{finetune}}$ addresses these questions by integrating DL into the semantic interaction loop, we compare it with the well-evaluated baseline model $\text{DeepSI}_{\text{vanilla}}$ [13, 14], which uses DL outside of the interactive loop, in two complementary experiments: a human-centered qualitative case study about COVID-19 academic articles; and an algorithm-centered simulation-based quantitative analysis of three commonly used text corpora: Stanford Sentiment Treebank (SST), Vispubdata, and 20 Newsgroups. The results of both experiments show that $\text{DeepSI}_{\text{finetune}}$ not only captures the analyst's precise intent more accurately, but also requires fewer interactions from the analyst.

Specifically, we claim the following contributions:

1. The $\text{DeepSI}_{\text{finetune}}$ framework that integrates DL into the human-in-the-loop iterative sensemaking pipeline to improve semantic interaction inference.

2. Two complementary studies, a user-centered qualitative case study and an algorithm-centered simulation-based quantitative experiment, that measure the performance of our method and reveal improvements.

5.2 Background

In order to frame our discussion of our model $\text{DeepSI}_{\text{finetune}}$, this section briefly describes DeepVA, the state-of-the-art SI model with pretrained DL [14]. For the purpose of comparison, we implement a specific version of DeepVA that uses BERT, which we denote as

Table 5.1: A list of variables used throughout this paper and their descriptions.

Variable	Description
d	A set of documents for analysis
N, M	Number of samples in d, number of dimensions of d
n	Number of samples moved by the analyst, $n \ll N$
x	High-dimensional feature of d. DL representations (768 dimension-size BERT embeddings)
y	Coordinates in the 2D visual spatialization of d. Set either by analysts' interactions on the visualization, or by the underlying SI model, which maps x to y
$w_{\text{dimension}}$	Parameters of dimension weights of x. (a 768 dimension-size vector)
w_{BERT}	Internal parameters of the pretrained BERT model. BERT_{base} is used in this paper, which contains 110 million parameters
$dist$	Euclidean distance between data samples in d. Weighted Euclidean distance is used if $w_{\text{dimension}}$ applied. $dist_H$ defines the high-dimensional similarity. $dist_L$ defines the low-dimensional similarity

$\text{DeepSI}_{\text{vanilla}}$. For reference, Table 5.1 describes frequently used variables throughout this paper. We use the pretrained BERT model as a representative DL model in DeepSI system designs. Note that WMDS is used as the default interactive DR in DeepSI frameworks for three reasons. First, the WMDS is a simple linear DR algorithm, so that we can focus on assessing the effects of data representations on the model performance. Second, WMDS is agnostic to the choice of the weighted distance function. Third, WMDS enables analysts to express their synthesis process by manipulating data point proximities to reflect their perceived similarity [148].

$\text{DeepSI}_{\text{vanilla}}$ uses the DL model as only a feature extractor in the SI pipeline. As shown in Fig. 5.2, the pretrained parameters inside the BERT model are frozen. Thereby, for an input, BERT provides a fixed general-purpose representation, which is then used as the data features in the interactive training loop. The BERT model is outside of the interactive loop. Therefore, the interactive DR model, WMDS, is responsible for updating dimension weights

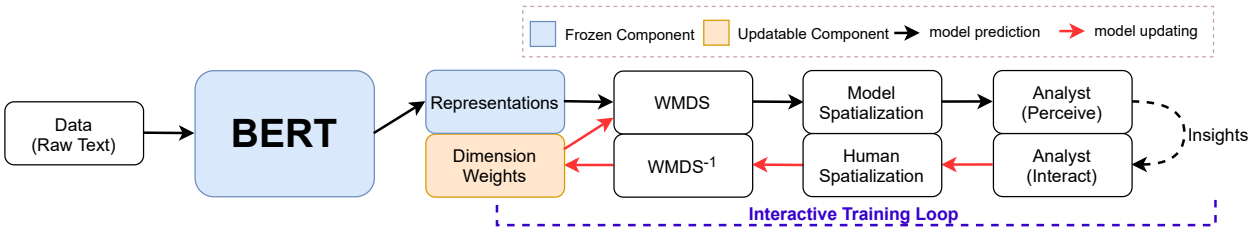


Figure 5.2: DeepSI$_{\text{vanilla}}$ pipeline, adapted from [13]: using the pretrained BERT as only a feature extractor pre-processed outside of the interactive loop in SI pipeline. All parameters inside the pretrained BERT model are frozen and the output data representations are fixed. WMDS is the interactive DR, which is responsible to tune the dimension weights $\boldsymbol{w}_{\text{dimension}}$ to capture the analyst's intent.

$\boldsymbol{w}_{\text{dimension}}$ to capture the analyst's intent as a weighting of the BERT features. The complete process of the pipeline is as follows.

Before entering the interactive training loop, the data representations are initialized by the BERT model with the pretrained parameters $\boldsymbol{w}_{\text{BERT}}$:

$$\boldsymbol{x} = \text{BERT}(d, w_{\text{BERT}}) \tag{5.1}$$

In the forward model-prediction direction, WMDS is performed to project high-dimensional data points ($\boldsymbol{x}$) into the two-dimensional spatialization ($\boldsymbol{y}$), with current dimension weights $w_{\text{dimension}}$ (initially, all weights are equal). This provides a new projection for the analyst to perceive and interact.

$$\boldsymbol{y} = \arg\min_{y} \sum_{i<j\leq N} \Big(dist_L(y_i, y_j) - dist_H(x_i, x_j, w_{\text{dimension}}) \Big)^2 \tag{5.2}$$

In the backward model-updating direction, the analyst provides visual feedback by repositioning n data points within the projection. WMDS^{-1} uses the low-dimensional pairwise distances between the moved n data points as input, to learn new dimension weights $\boldsymbol{w}_{\text{dimension}}$ to make sure these moved data points have similar relationships in the high-dimensional

space, based on the following optimization criterion:

$$\boldsymbol{w}_{\text{dimension}} = \arg\min_{w} \sum_{i<j\leq n} \Big(dist_L(y_i, y_j) - dist_H(x_i, x_j, w_{\text{dimension}}) \Big)^2 \tag{5.3}$$

Therefore, through this loop, the dimension weights $\boldsymbol{w}_{\text{dimension}}$ are trained interactively and incrementally based on analysts' interactions to capture their intents.

$\text{DeepSI}_{\text{vanilla}}$ has been well-evaluated previously. DeepVA [14] used ResNet [73] as an image data feature extractor in the SI system that assists users performing visual concepts analysis using DL representations. In [13], Bian et al. compared SI systems that use embedding vectors as features and those that use bag-of-words as features in visual text analysis tasks. Experiments in both works show that even the general-purpose representations of pretrained DL models can enable SI to better capture the analyst's intent than hand-crafted features. However, using the general-purpose pretrained representations still restricts SI inference. In the next section, we propose $\text{DeepSI}_{\text{finetune}}$, which exploits fine-tuned representations to further improve SI inference. As the best-performing model from previous studies, $\text{DeepSI}_{\text{vanilla}}$ is the baseline model for comparison.

5.3 Model Description

This section outlines the main design, model pipeline, and implementation details of $\text{DeepSI}_{\text{finetune}}$.

5.3.1 Model Design

We propose two main design goals to address the two research questions discussed in Sec. 7.1.

Design goal 1 - Integrating DL into the human-in-the-loop interactive sensemak-

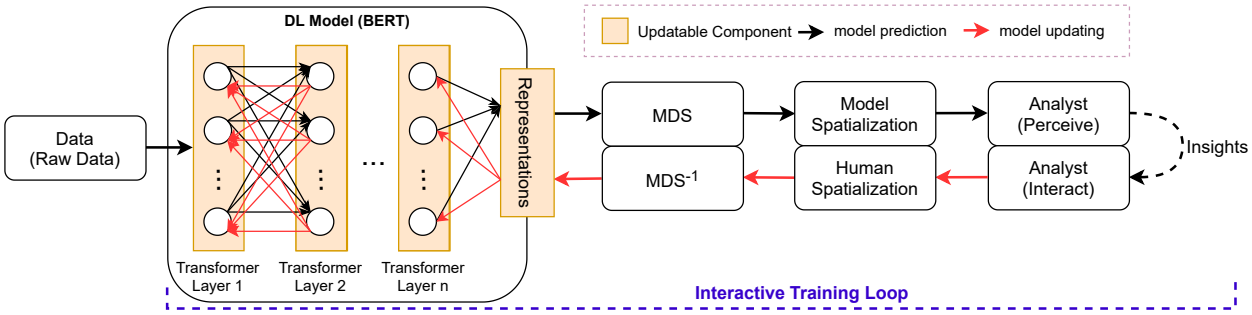


Figure 5.3: DeepSI$_{\text{finetune}}$ pipeline: embedding BERT within the SI loop. Semantic interactions are exploited to fine-tune BERT interactively through backpropagation. The tuned BERT is responsible for generating new representations, so as to capture the analyst's intent. Thereby, no external parameters are needed.

ing pipeline. To get user- and task-specific representations, it is necessary to iteratively train the DL model with semantic interactions during the human-in-the-loop process. Inspired by multi-model SI systems [19, 47, 175], we inserted the DL model update and prediction process into the bidirectional semantic interaction loop, as shown in Fig. 5.3. The DL model update and prediction process occurs before the interactive DR model. The interactive DR model passes the analyst's visual feedback from the human spatialization to the DL model. The visual feedback is then used to update the parameters inside the DL model ($\boldsymbol{w}_{\text{BERT}}$) through the DL backpropagation [141] (red arrows inside the DL component). With updated parameters $\boldsymbol{w}_{\text{BERT}}$, the BERT model calculates new representations for input data by the forward propagation [196] through the internal transformer layers (black arrows inside the DL component). Through the interactive sensemaking process, the DL model is trained by semantic interactions in an interactive machine learning setting [5, 58]. Thereby, improved representations are generated to accurately capture the analyst's intent.

Design goal 2 - Introducing minimal parameters into the interactive DL training pipeline. To solve analytic tasks efficiently, the analyst prefers to perform fewer interactions in each sensemaking loop. However, DL model training typically needs a relatively large amount of training data. To reduce the number of interactions needed for training, we

should introduce minimal parameters into the pipeline while integrating the DL model training. For this design goal, we made specific modifications to both the DL and the interactive DR components. First, we used the fine-tuning approach to adapt the pretrained BERT model with semantic interactions. Unlike the feature-based method, fine-tuning approach introduced minimal task-specific paramters [67]. This drastically reduced the required training data. Further, we used MDS/MDS^{-1} as the interactive DR component. During the interactive BERT model training, representations are updated to capture analysts' intents. It is unnecessary to tune extra parameters for the same purpose in the interactive DR model. Therefore, we used MDS/MDS^{-1} without dimension weights $\boldsymbol{w}_{\text{dimension}}$ as the interactive DR component. There are no parameters to tune in this component. Therefore, users' interactions can be passed directly to DL training without information loss.

5.3.2 Model Pipeline

We illustrate the DeepSI$_{\text{finetune}}$ pipeline (Fig. 5.3) in detail through the human-in-the-loop sensemaking process. In the forward model-prediction direction, new representations are generated for the dataset d through the forward propagation calculation of the BERT model, with current BERT parameters ($\boldsymbol{w}_{\text{BERT}}$):

$$\boldsymbol{x} = \text{BERT}(d, w_{\text{BERT}}) \tag{5.4}$$

The high-dimensional DL representations x are then projected to the 2D spatialization (model spatialization) by MDS through the following equation:

$$\boldsymbol{y} = \arg\min_{y} \sum_{i<j\leq N} \Big(dist_L(y_i, y_j) - dist_H(x_i, x_j) \Big)^2 \tag{5.5}$$

In contrast to Eq. 5.2, the high-dimensional distance function is not explicitly weighted. Instead, the updates to $\boldsymbol{y}$ in each loop are captured by the fine-tuned representation x itself. The analyst perceives the updated spatialization and gains insight.

In the backward model-updating direction, the analyst modifies the visual layout (human spatialization) by repositioning some samples to express the preferred similarities between them. Then, MDS^{-1} uses the human-defined similarities between n moved data points, $dist_L(y_i, y_j))$, to steer the BERT model parameters to generate better high-dimensional representations x, such that the similarity of the representations reflects the proximity of the points in the modified projection, as follows:

$$w_{\text{BERT}} = \arg\min_{w_{\text{BERT}}} \sum_{i<j\leq n} \Big(dist_L(y_i, y_j) - dist_H(\text{BERT}(d_i, w_{\text{BERT}}), \text{BERT}(d_j, w_{\text{BERT}})) \Big)^2 \quad (5.6)$$

The optimization objective is to fine-tune BERT weights $\boldsymbol{w}_{\text{BERT}}$ to minimize the difference between low-dimensional and high-dimensional distances of n moved data points through backpropagation. All internal parameters of the BERT model ($\boldsymbol{w}_{\text{BERT}}$) are updated in order, from last transformer layers to previous layers, by a gradient descent optimization algorithm [140]. After the backpropagation, the updated $\boldsymbol{w}_{\text{BERT}}$ is used in the forward propagation to calculate new representations $\boldsymbol{x}$ in Eq. 5.4.

Through this human-in-the-loop interactive DL process, the BERT model is tuned properly to generate user- and task-specific representations, so as to capture analysts' precise intents: samples that should be closer to each other in the visualization obtain similar features, while more distant samples gain differing features.

5.3.3 Prototyping Detail

Here, we describe the implementation details of DeepSI prototypes used in our experiments, including model settings and visualization design. These implementations are applicable for both $\text{DeepSI}_{\text{finetune}}$ and $\text{DeepSI}_{\text{vanilla}}$.

Model Settings

We use Pytorch [127], a well-known Python DL framework, to implement the DeepSI system. For the forward DR component, MDS is adapted from Scikit-Learn [128]. The MDS^{-1} is implemented in Pytorch as a neural network layer. The pretrained BERT model is adapted from the publicly available Python library, Transformers [183]. Transformers provides two sizes of pretrained BERT models: $\text{BERT}_{\text{BASE}}$, and $\text{BERT}_{\text{LARGE}}$. We used the small BERT model ($BERT_{BASE}$) (bert-base-uncased, 12-layers, 768-hidden, 12-heads, 110M parameters), because it is more stable on small datasets. For a document containing a list of tokens, $\text{BERT}_{\text{BASE}}$ can convert each of the tokens into a 768-dimensional vector. To generate fixed-length encoding vectors from documents of different lengths, we appended a MEAN pooling layer to the last transformer layer of the BERT model, such that the output representation for a document was a 768-dimensional vector. Therefore, the $w_{\text{dimension}}$ used in $\text{DeepSI}_{\text{vanilla}}$ is also a 768 dimension vector. We also tested other pooling strategies, such as MAX pooling and CLS pooling [138]. However, there was no obvious performance difference, and the MEAN pooling showed slightly better performance. In addition, we used the Adam optimizer [95] to optimize the DeepSI model parameters in the model-updating direction. We also explored other optimizers provided by PyTorch. Across all our experiments, we found that Adam optimizer performed the best. Further, we found that the suggested learning rate ($3e^{-5}$) for finetuning BERT models in [42] led to optimal $\text{DeepSI}_{\text{finetune}}$ performance in experiments.

Visualization Design

We drew inspiration for visualization design from SI-enabled VA applications, including Andromeda [148], ForSPIRE [53], and Dis-Function [20]. As shown in Fig. 5.1, the visual interface mainly uses a scatterplot as the projection layout. This scatterplot not only displays the relationships between data updated by the underlying projection model, but also allows the analyst to intervene and modify the layout. Specifically, in the forward model-prediction direction, the positions between data points on the scatterplot reflect the points' relative similarity learned by underlying models, either by the projection method or by the fine-tuned BERT model, shown in Fig. 5.1-1 (model spatialization). In the backward model-updating direction, the user can drag several data points to new positions to modify similarities between points based on their preference, shown in Fig. 5.1-2 (human spatialization). Having both the underlying models and analysts work on the same visualization provides direct and effective communication between humans and computation. In addition to the scatterplot view, the prototype also provides a sidebar view to help analysts review the content of a selected document when exploring in the scatterplot view. In this paper, we intentionally focus on the scatterplot view in the screenshots, to focus on the analysis of the model performance.

5.4 Experiments

To evaluate $\text{DeepSI}_{\text{finetune}}$, we conducted the following experiments. To examine how well $\text{DeepSI}_{\text{finetune}}$ addresses the goals, we measured its performance in two respects:

- **Accuracy**: How accurately can $\text{DeepSI}_{\text{finetune}}$ capture the analyst's intent?
- **Efficiency**: How many interactions does $\text{DeepSI}_{\text{finetune}}$ need to capture the analyst's

intent properly?

We use $\text{DeepSI}_{\text{vanilla}}$, described in Sec. 5.2, as the baseline model to evaluate the advantage of $\text{DeepSI}_{\text{finetune}}$'s task-specific, instead of general-purpose, representations. Boukhelifa et al. [18] proposed the complementary evaluation of interactive machine learning systems by using both algorithm-centered and human-centered evaluation methods. We perform both evaluation methods in our experiments: the case study in Sec. 5.4.1 is the human-centered qualitative analysis, and the simulation-based evaluation method in Sec. 5.4.2 is the algorithm-centered quantitative analysis.

5.4.1 Case Study: COVID-19

Recently, COVID-19 [162] has become a global pandemic. It is essential that medical researchers quickly find relevant documents about a specific research question, given the extensive coronavirus literature. We used an analysis task on academic articles related to COVID-19 in this case study to examine our proposed $\text{DeepSI}_{\text{finetune}}$, compared with the baseline model $\text{DeepSI}_{\text{vanilla}}$. In this study, we performed the same task with the help of both DeepSI prototypes and then measure the model performance in the following two perspectives:

- **Accuracy**: the quality of the projection updated by the underlying model given the task's ground truth.

- **Efficiency**: how many interactions are needed for the underlying model to provide a useful projection.

Dataset and Task

The COVID-19 Open Research Dataset (CORD-19) [1] contains a collection of more than 200,000 academic articles about COVID-19. CORD-19 also proposes a series of tasks in the form of important research questions about the coronavirus. One of the research tasks focuses on identifying COVID-19 risk factors [2]. In this case study, we selected a task that requires identifying articles related to specific risk factors for COVID-19. We asked an expert to choose as many research papers as possible about risk factors from CORD-19. We found four main risk factors: cancer (15 articles), chronic kidney disease (13 articles), neurological disorders (23 articles), and smoking status (11 articles). We used these four risk factors as the ground truth for the test task, and loaded all these 62 articles into our DeepSI prototypes. Therefore, the test task was to organize these 62 articles into four clusters with our DeepSI prototype such that each cluster represented articles of a specific risk factor.

This particular ground truth is just one possible way an analyst might want to organize this group of documents. So our goal is to see if this particular set of expert knowledge can be easily injected using SI to help re-organize the documents in this particular way. To help judge the quality of the visual layout in organizing this particular ground truth, we color the dots according to the ground-truth risk factors in the visualization: **cancer** (black dot •), **chronic kidney disease** (red dot •), **neurological disorders** (blue dot •), **Smoking status** (green dot •). It should be noted that the underlying model was not provided with the ground truth or color information. The ground truth is only injected via semantic interaction from the human in the form of partial groupings of only a few of the documents.

[1] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge
[2] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge/tasks?taskId=558

Study Procedure

To compare the projection layouts updated by two models based on the same input interactions, semantic interactions based on the ground truth are performed in the shared visual projection and then applied to the two models separately. Fig. 5.1 and Fig. 5.4 show the process of interactions applied separately to $\text{DeepSI}_{\text{finetune}}$ and $\text{DeepSI}_{\text{vanilla}}$ prototypes. In both figures, frame 1 and frame 2 are from the shared visual projection. Frame 1 in both figures (Fig. 5.1-1 and Fig. 5.4-1) shows the same initial layout updated by the default pretrained BERT model. In the initial projection layout, all the articles are combined. This means that the pretrained BERT model cannot distinguish these articles by their related risk factors. Interactions were performed within the projection based on the ground truth to reflect the perceived connections between articles: grouping three articles about cancer to the top-left region of the projection, indicated by the black arrows; three articles about chronic kidney disease to the top-right region indicated by red arrows; three articles about smoking status to the bottom-left part indicated by green arrows; and three articles about about neurological disorders to the bottom-right part indicated by blue arrows.

Frame 2 (Fig. 5.1-2 and Fig. 5.4-2) is the same human spatialization and shows four clusters created by the ground truth. After we clicked the 'model update' button on the menu bar to start the model training process. Then, the same human spatialization was used to train both models. After the two models had been updated, the updated projections of these two models showed the performance difference in frame 3 (Fig. 5.1-3 and Fig. 5.4-3). Subsequently, the performance of the model could be assessed based on how reasonable the layout was in comparison with the ground truth.

$\text{DeepSI}_{\text{finetune}}$ spatialization: The projection updated by $\text{DeepSI}_{\text{finetune}}$ is shown in Fig. 5.1-3. There are four clear clusters, and all articles are clearly grouped into the correct clusters.

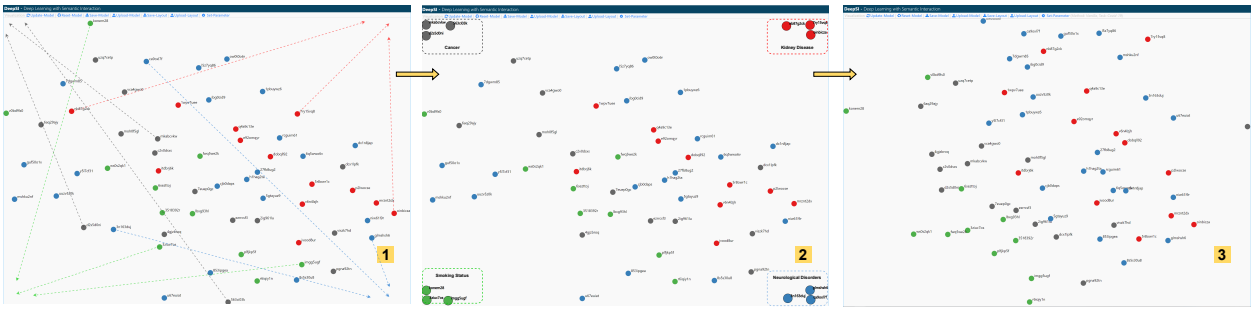


Figure 5.4: Screenshots during the case study using DeepSI$_{\text{vanilla}}$: Frame 1 and 2 show the similar initial steps performed by the analyst in Fig. 5.1. Frame 3 shows the resulting projection updated by DeepSI$_{\text{vanilla}}$.

The top left cluster contains all the articles about cancer (●), the top right cluster contains articles about kidney disease (●), the bottom left contains articles about smoking status (●), and the bottom right contains articles about neurological disorders (●). This means the new representations generated by the fine-tuned BERT model are able to accurately capture the semantic meanings behind users' interactions.

DeepSI$_{\text{vanilla}}$ spatialization: With the same interactions as input, the updated DeepSI$_{\text{vanilla}}$ shows a different layout. As shown in Fig. 5.4-3, there are no clear clusters in the updated layout compared with Fig. 5.1-3. Articles about different risk factors still overlap. Even after continued interactions based on the ground truth, DeepSI$_{\text{vanilla}}$ is unable to properly capture the user's semantic intent and differentiate these articles.

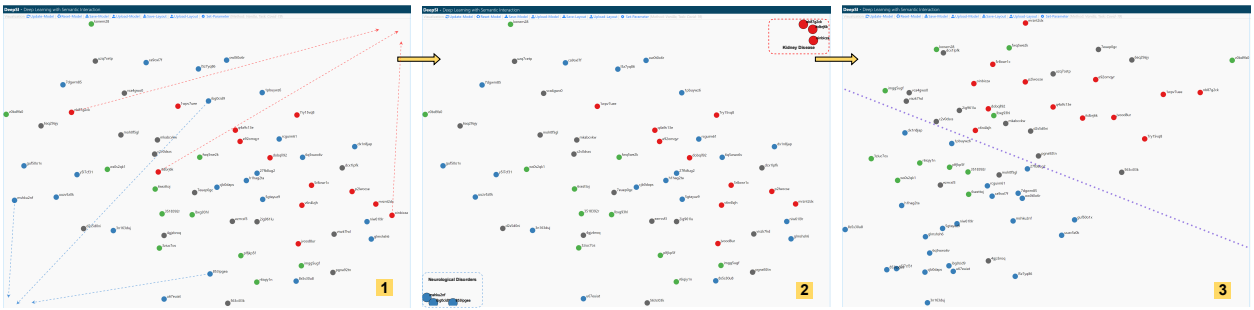


Figure 5.5: Further case study using DeepSI$_{\text{vanilla}}$ in grouping two clusters: Frame 1 is the initial projection layout, Frame 2 shows interactions performed within the projection, and Frame 3 shows the resulting projection updated by DeepSI$_{\text{vanilla}}$.

Further study for DeepSI$_{\text{vanilla}}$: However, DeepSI$_{\text{vanilla}}$ did work well at separating articles into two clusters in two opposite positions in the projection. For example, in Fig 5.4-3, smoking status articles (●) are separated from kidney disease articles (●). Exploring further, after resetting the model as shown in Fig. 5.5-2, three neurological disorders articles (●) are dragged to the bottom-left and three chronic kidney disease articles (●) are dragged to the top-right on the scatterplot view. After the layout updates, articles from these two dragged clusters are well placed in two opposite sides of the visualization in Fig. 5.5-3, ignoring articles in the other two clusters (about cancer ● and smoking status ●).

Qualitative Results

In terms of accuracy, DeepSI$_{\text{finetune}}$ grouped articles correctly based on the user-defined risk factors. In contrast, DeepSI$_{\text{vanilla}}$ did not provide a useful projection. The further study also confirmed that DeepSI$_{\text{vanilla}}$ can handle more straightforward tasks with only two clusters. The semantics of the ground truth knowledge provided by the contest organizers are more recognizable in the DeepSI$_{\text{finetune}}$ projection. Articles in each group are clearly clustered. However, DeepSI$_{\text{vanilla}}$ only partially separated in two separate directions, instead of into distinct clusters, which requires more cognitive effort to identify the boundary between the groups. In terms of efficiency, DeepSI$_{\text{finetune}}$ is more efficient than DeepSI$_{\text{vanilla}}$. DeepSI$_{\text{finetune}}$ needed a small number of interactions (moving three articles in each cluster, 12 dots movement in total) in one interactive SI loop to fine-tune the BERT model properly for this task. In contrast, in DeepSI$_{\text{vanilla}}$, the same amount of interactions only supported the simpler task with two clusters, and additional rounds of interaction still did not uncover all four clusters.

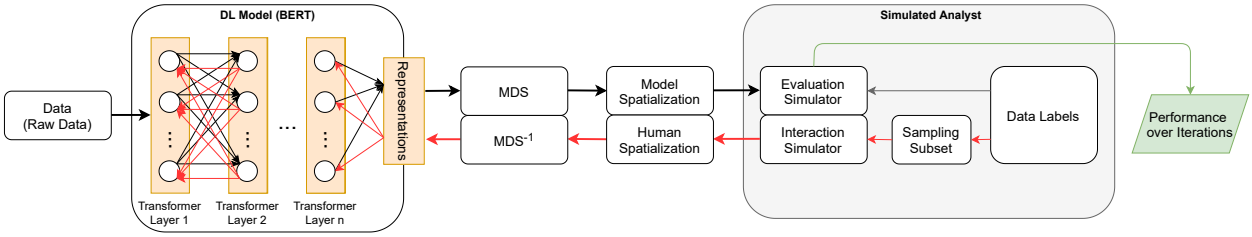


Figure 5.6: Simulation-based evaluation pipeline. The analyst is replaced by the 'simulated analyst' component where: analyst perception is simulated by the kNN classifier and analyst interaction by sampling a subset of ground truth. In each SI loop, the kNN classification is employed to calculate the accuracy of the model spatialization, which is updated by the underlying model $\text{DeepSI}_{\text{finetune}}$ and reflects the model performance.

5.4.2 Simulation-based Evaluation

From the machine learning algorithm perspective, DeepSI systems are transductive models [199] that interactively learn projections provided by the analyst. Therefore, the performance of DeepSI systems can be measured by the predicted projections. To conduct the quantitative comparisons between the predicted projections from DeepSI systems, we replaced the analyst with a simulation component (simulated analyst). As shown in Fig. 5.6, the simulated analyst uses the interaction simulator to generate a training projection (human spatialization) based on data labels, and the evaluation simulator to evaluate the accuracy of the predicted projection. After training iteration, the simulated analyst outputs a current projection accuracy. The projection accuracy over iterations reflects the learning curve [130] of the DeepSI model. Therefore, performances of both DeepSI models could be compared through their learning curves in both accuracy and efficiency perspectives.

Simulated Analyst

As shown in the simulation pipeline (Fig. 5.6), data labels are the ground truth to support both interaction simulator and evaluation simulator. First, the interaction simulator uses

these labels to calculate the pairwise distances between a subset of data samples, simulating the human-defined similarities between these samples. Further, the evaluation simulator uses these class labels to measure how well the predicted projection grouped data samples into correct classes based on their labels.

Interaction simulator: In each interaction, three samples from each class are selected using random sampling [161]. Then the interaction simulator calculates the pairwise distance $dist_L(y_i, y_j)$ of these selected samples based on:

$$dist_L(d_i, d_j) = \begin{cases} 0 & \text{if } d_i \text{ and } d_j \text{ have the same label} \\ \sqrt{2} & \text{otherwise} \end{cases}$$

As shown in the above Equation, if two selected samples have different labels, the distance between them is $dist_L(d_i, d_j) = \sqrt{2}$, because the analyst should move them away from each other to obtain the farthest distance on the 2D spatialization. If the two samples have the same label, the analyst should move them as close as possible ($dist_L(d_i, d_j) = 0$) in the projection, because they belong to the same cluster. Therefore, the interaction simulator provides the calculated pairwise distances between the selected samples as the training projection for DeepSI models.

Evaluation simulator: After the interaction simulator trains the DeepSI model, the trained model predicts a new projection. The predicted projection reflects the similarity relationships between samples in the low-dimensional spatialization. We used a kNN (K-nearest-neighbour) classifier [39] as the evaluation simulator to measure the predicted projection [13, 20]. The kNN classifier uses the neighbor information on the projection to train and predict the data classes. The performance of the learned kNN classifier can directly reflect the quality of the projection [173]. Concretely, we used the leave-one-out cross-

validation [122] and set k = 5 closest training examples to predict the unlabelled sample. We also explored other values for k, such as 3, 7, 9, 11, but these did not produce significant changes in the results. We could thus obtain the trained kNN classifier accuracy by comparing the predicted output with the ground truth.

Performance over iteration: A new accuracy from the kNN classifier was returned from the simulation pipeline in each iteration loop. These are accumulated into a plot of kNN classifier performance over the iterations of the simulated interaction loop. This learning curve shows how rapidly the DeepSI model learned during the interactive process.

Dataset and Task

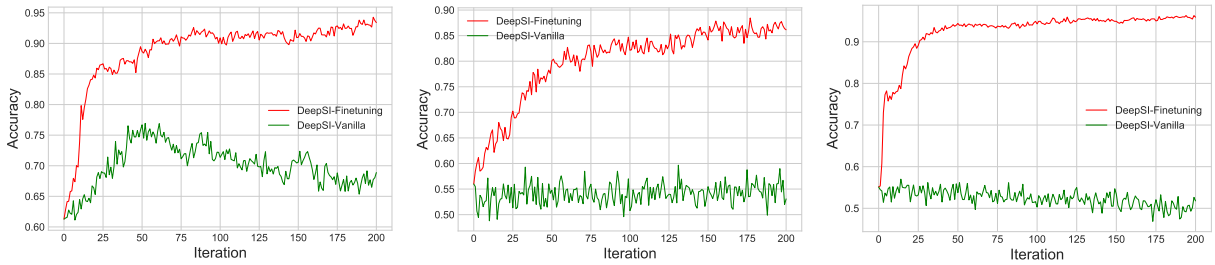
We explored three commonly used text corpora in natural language processing and visual text analysis tasks. These corpora contain different numbers of labels and are from different domains, providing a comprehensive evaluation of performance comparisons.

SST with two clusters: The SST dataset [157] is a collection of movie reviews with both fine-grained labels (out of five stars) and binary labels (positive and negative reviews). We used the binary version of the dataset, which contains 1821 reviews in total: 909 positive and 912 negative. The task, denoted as T_{sst}, used the SST dataset to train the DeepSI methods to obtain two clusters (positive and negative).

Vispubdata with three clusters: The Vispubdata dataset [85] contains academic papers published in the IEEE VIS conference series. These papers belong to one of the three conferences: InfoVis (Information Visualization), SciVis (Scientific Visualization), and VAST (Visual Analytics Science and Technology). We used the papers published between 2008 and 2018 (including 397 papers from InfoVis, 534 papers from SciVis, and 521 papers from VAST) in this task, denoted as T_{vis}. In T_{vis}, the simulated analyst need to iteratively drag papers

into these three conferences clusters to evaluate the DeepSI.

20 Newsgroups with four clusters: The 20 Newsgroup dataset [3] is a collection of newsgroup posts on 20 topics. Based on this dataset, we create the task (T_{news}) to classify four topics into different clusters in the spatialization. We picked four topics from the same subcategory 'rec' including: 594 reports from 'rec.autos', 598 reports from 'rec.motorcycles', 597 reports from 'rec.sport.baseball', and 600 reports from 'res.sport.hockey'.



(a) T_{sst} (b) T_{vis} (c) T_{news}

Figure 5.7: The accuracies of both $\text{DeepSI}_{\text{finetune}}$ and $\text{DeepSI}_{\text{vanilla}}$ updated projections over 200 iterations across the three tasks (T_{sst}, T_{vis}, and T_{news}) during the simulation-based experiment.

Quantitative Results

Fig. 5.7 shows the learning curves of both DeepSI methods in all three tasks. There is no crossing of the curves between these two models, and the performance curve of $\text{DeepSI}_{\text{finetune}}$ is above the curve of $\text{DeepSI}_{\text{vanilla}}$ through all iterations. This means $\text{DeepSI}_{\text{finetune}}$ showed better performance on all three tasks than $\text{DeepSI}_{\text{vanilla}}$. For model accuracy, $\text{DeepSI}_{\text{finetune}}$ converged to more than or nearly 90% accuracy in all three tasks. On the contrary, the best performance of $\text{DeepSI}_{\text{vanilla}}$ only showed slightly higher accuracy (less than 80%) than the initial performance in the first task (T_{sst}) with two clusters. For tasks with more than two clusters (T_{vis} in Fig. 7.8b and T_{news} in Fig. 7.8c), $\text{DeepSI}_{\text{vanilla}}$ did not show noticeable

[3] http://qwone.com/ jason/20Newsgroups/

accuracy increases. This is consistent with our findings in the case study (Sec. 5.4.1). For model efficiency, the performance over iterations of $\text{DeepSI}_{\text{finetune}}$ in all three tasks showed steeper increases and quickly approximated peak accuracy. Furthermore, the performance of $\text{DeepSI}_{\text{finetune}}$ increased fairly consistently compared to $\text{DeepSI}_{\text{vanilla}}$. This provides analysts with more consistent feedback over iterations.

5.5 Discussion

5.5.1 Generality and Applicability

In this paper, we use pretrained BERT as the specific DL model in $\text{DeepSI}_{\text{finetune}}$ to advance SI-enabled applications. Considering $\text{DeepSI}_{\text{finetune}}$ as a framework, it is general enough to apply other pretrained DL models into the semantic interaction pipeline for other VA tasks. First, other transformer-based models, such as RoBERTa [111], XLNet [190] and GPT-3 [21], can be applied directly in the DeepSI pipeline without any special configuration. In addition, fine-tunable DL models with other structures, such as CNN and RNN models [3], can also be integrated into a DeepSI pipeline by appending a special pooling layer to transform hidden states into proper representations. Further, other feature-based DL models could also be applied to the inteactive fine-tune process with specific designs. For example, ELMo [131] could be fine-tuned by using max-pool over the model's internal states and adding a softmax layer [132].

5.5.2 Scalability

Beyond measuring accuracy and efficiency, our two experiments also illuminated scalability. All our experiments were conducted on a desktop computer with an Intel i9-9900k processor,

32G Ram, and one NVIDIA GeForce RTX 2080Ti GPU, running Windows 10. In the case study, $\text{DeepSI}_{\text{finetune}}$ captured the analyst's intent and provided an accurate projection with 62 data points in real time. In the simulation-based experiment, $\text{DeepSI}_{\text{finetune}}$ also provided accurate projections that contains thousands of data points. During the simulation, MDS projection calculation ($O(n^2)$ algorithm) consumed the majority of the time. The amount of time required for the DL model update and prediction was negligible in comparison. This highlights the potential for improving the DR method in DeepSI.

5.5.3 Interactive Deep Metric Learning

Dis-Function [20] describes the WMDS-based SI model as an interactive distance function learning model from the interactive machine learning perspective. Likewise, $\text{DeepSI}_{\text{finetune}}$ can be regarded as an interactive deep metric learning model [92]. As shown in Fig. 5.3, in the model-updating direction, the underlying pretrained BERT model is trained interactively to output better presentations that can capture the analyst-desired distance relationships. Deep metric learning methods usually use metric loss functions [92] for labelled data, such as contrastive loss [67], triplet loss [77] and angular loss [168]. In contrast, $\text{DeepSI}_{\text{finetune}}$ uses a metric loss function specially designed for semantic interactions based on MDS^{-1}.

5.5.4 Limitations and Future Work

Our $\text{DeepSI}_{\text{finetune}}$ proved effective in capturing analysts' precise intents and displaying intuitive projections. However, current $\text{DeepSI}_{\text{finetune}}$ prototypes could be extended in two directions. First, it is important to make the internal status of the underlying model interpretable to analysts in order to facilitate them making hypotheses and decisions during the sensemaking process, which is known as interpretable machine learning [123]. Other than

the projection scatterplot, the current $\text{DeepSI}_{\text{finetune}}$ prototype does not provide any other visual hints about the status of the DL model. Therefore, we plan to add more specific visual designs in future work to better expose the effects of tuning the DL model.

In addition, $\text{DeepSI}_{\text{finetune}}$ uses the traditional metric learning method [9] as the interactive DR component to communicate between the DL model and the analyst. As discussed above (Sec. 5.5.3), a MDS-based loss function Loss_{SI} is used to interactively tune the DL model. Inversely, these deep metric learning loss functions, such as contrastive loss and triplet loss, could also be used as interactive DR components. We plan in future work to use deep metric learning loss functions as the interactive DR component in $\text{DeepSI}_{\text{finetune}}$.

5.6 Conclusion

In this work, we focused on DeepSI and the research question of how to integrate the DL model into the SI pipeline to leverage its capability to better capture the semantics behind user interactions. We identified two design requirements of effective DeepSI systems: the DL model is trained interactively in the SI pipeline and the DL model can be tuned properly with a small number of interactions. We presented $\text{DeepSI}_{\text{finetune}}$, which incorporates DL fine-tuning and MDS-based interactive DR methods into the DeepSI pipeline to meet these requirements. We performed two complementary experiments to measure the effectiveness of $\text{DeepSI}_{\text{finetune}}$, including a case study of a real-world task relating to COVID-19 and a simulation-based quantitative evaluation method on three commonly used text corpora. The results of these two experiments demonstrated that $\text{DeepSI}_{\text{finetune}}$ improves performance over the state-of-the-art alternative that uses DL only as a pre-processed feature extractor, indicating the importance of integrating the DL into the interactive loop.With a small number of semantic interactions as input, $\text{DeepSI}_{\text{finetune}}$ better captures the semantic intent of

the analyst behind these interactions.

Chapter 6

DeepSE: Semantic Explanation of Interactive Deep Learning

Visual analytics seeks to enable usable human-AI collaborative sensemaking with interactive visualizations. When powered by interactive deep learning, visual analytics applications can better capture analysts' precise intents and support complex decision-making tasks. The opaque nature of deep learning prevents incremental formalism, and thereby inhibits the sensemaking process. To address this problem **(RQ 3)**, we present a semantic explanation framework (SE) that generates and contextualizes counterfactual explanations into semantic interaction designs. SE consists of two key components: counterfactual engine and counterfactual projection. Counterfactual engine interprets black-box models by identifying representative counterfactual explanations for each data instance. Counterfactual projection contextualizes counterfactual examples into the (cognitive and model) data projection. SE is applied to a semantic interaction system for visual text analytics that is powered by deep learning via the BERT model. COVID-19 research documents are used in a case study to demonstrate how the system can help analysts sort articles related to specific risk factors.

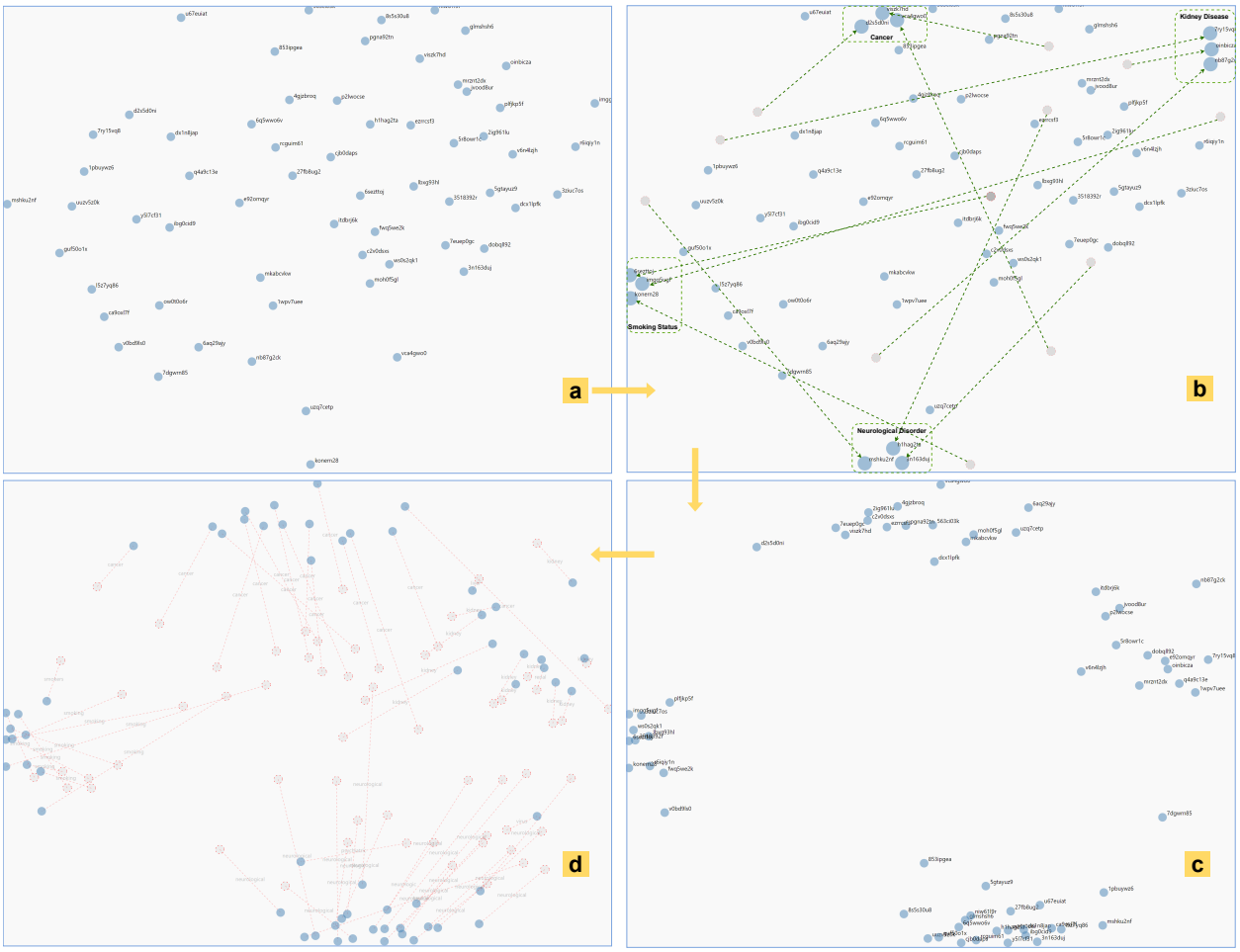


Figure 6.1: Screenshots during the analysis of COVID-19 research articles about four different risk factors using a semantic interaction system with the DL model BERT for visual text analytics: (a) The initial layout of all articles projected from pretrained BERT representations of the raw text data; (b) The analyst performs semantic interactions to provide visual feedback regarding articles about different risk factors; these interactions are then exploited to fine-tune BERT; (c) The resulting projection updated by the underlying fine-tuned BERT. However, the DL model lacks explanation of the meaning of the clusters created. Consequently, it is difficult to answer questions such as: what is the meaning of the cluster at the top? how does the model form these clusters? (d) Our proposed SE solution interprets the visual pattern of the model projection with counterfactual examples.

6.1 Introduction

Semantic interaction (*SI*) [53] is an interaction methodology commonly utilized in visual analytic (*VA*) systems for human-AI collaborative sensemaking. With SI, analysts can express desired similarities in the data by intuitively performing actions within the visual projection. Analysts do not need to manipulate the parameters of the underlying model directly. The VA system infers analysts' intents from performed interactions and fine-tunes the underlying model. Analysts can concentrate on the work at hand, thereby enhancing their efficiency in the analytic tasks. As shown in Fig. 6.1b, the analyst can move several documents into four clusters to express their preferred data layout, intuitively defining high-level concepts related to covid risk factors. The underlying model learns new parameters to capture the new relationships between moved instances. Subsequently, the tuned model refreshes the whole projection to provide visual feedback about the remaining instances.

To determine the analyst's precise intent behind these interactions, increasingly powerful nonlinear models, such as deep learning (DL), have been utilized in SI systems. These black-box models offer new opportunities to power SI with more accurate steering [14]. However, they also make the new projection layout difficult to interpret, which inhibits users' mental formalization. As shown in Fig. 6.1c, the visual projection reveals the data similarities based only on projection distances through case-based reasoning. The DL model (BERT [42], a state-of-the-art model for NLP tasks) does not directly support the process of incremental formalization [152] in which analysts correlate the high-level projection layout to low-level features as evidence. The analyst has difficulty with (1) intuitively reasoning whether the projection has sufficiently captured their intent, and (2) formally defining their high-level concepts in terms of lower-level features. The analyst must pause the current sensemaking task, and comprehend the projection layout by examining each of the data points in detail. We argue that model explanation is the next bottleneck preventing analysts from focusing

on their sensemaking tasks with high efficiency.

In this paper, we aim to seamlessly integrate semantic explanation methods into existing SI systems. Central to this design goal are two research questions:

- How can semantic explanations be generated for black-box models inside SI systems that support incremental formalization for sensemaking tasks?
- How can the generated model explanations be naturally contextualized into the projection that represents the user's mental model?

To address these two questions, we propose the visual explanation framework called *Semantic Explanation (SE)*, analogous to SI. SE complements SI systems with two key components: the *counterfactual engine* and the *counterfactual projection*. The counterfactual engine interprets black-box models by creating representative counterfactual explanations for each of the data instances. The use of counterfactuals [167] is a model-agnostic approach that can generate unified explanations for different black-box models. Further, counterfactuals elicit causal thinking in humans and support incremental formalization. The counterfactual projection is carefully designed to visualize both data points and relevant counterfactuals on the same projection layout. This new type of projection augments the traditional visual metaphor with counterfactual explanations in a contextual fashion. The augmented projection also provides interactions to explore visual explanations in both local and global scope. With SE, analysts can focus on the analysis task at hand while incrementally formalizing their mental model, as shown in Fig. 6.1d.

We apply our SE framework to a semantic interaction system for visual text analytics that is powered by DL via the BERT model [42]. We call this novel system DeepSE. DeepSE helps analysts understand and explore projection results restructured by the interactively fine-tuned BERT model. With a case study of COVID-19 research documents, DeepSE

explains how the fine-tuned BERT model uses influential keywords in these documents to form particular projection patterns, such as clusters and outliers.

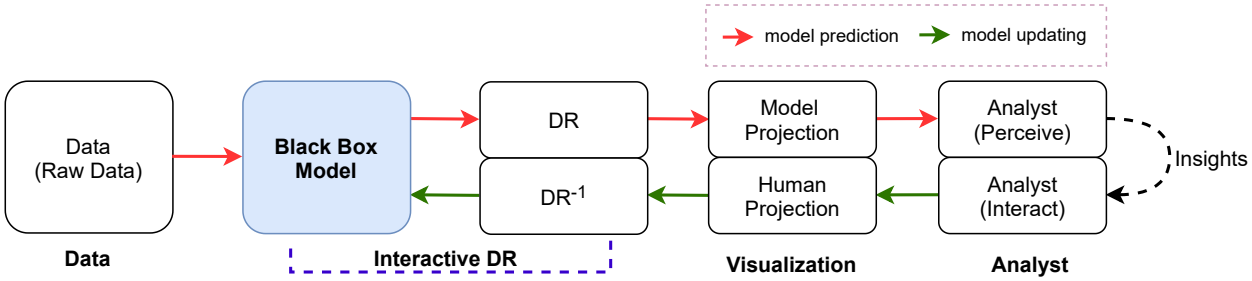


Figure 6.2: DeepSI pipeline [12] showing the communication between the analyst and VA system. The interactive DR component is responsible for capturing the analyst's intent from the human-modified projection (denoted as "human projection") and consequently updating the projection in response (denoted as "model projection"). The black-box model empowers the interactive DR in capturing the analyst's precise intent.

6.2 Semantic Explanation

This section describes the generalized SE framework for providing semantic explanations in semantic interaction systems.

6.2.1 Design Rationale

As shown in Fig. 6.3, SI is a useful interaction methodology for VA systems to shield analysts from directly manipulating underlying models. The cognitive connection between the user and the spatial layout, through the visual *proximity* $\approx$ *similarity* metaphor, makes SI an effective interaction methodology for analysts to express their intent without the need to prematurely formalize their analytic reasoning. With SI, the analyst performs spatial analytic interactions in the projection to externalize their internal reasoning about concepts based on exemplar cases (Step 1). For example, the analyst moves two articles apart because these

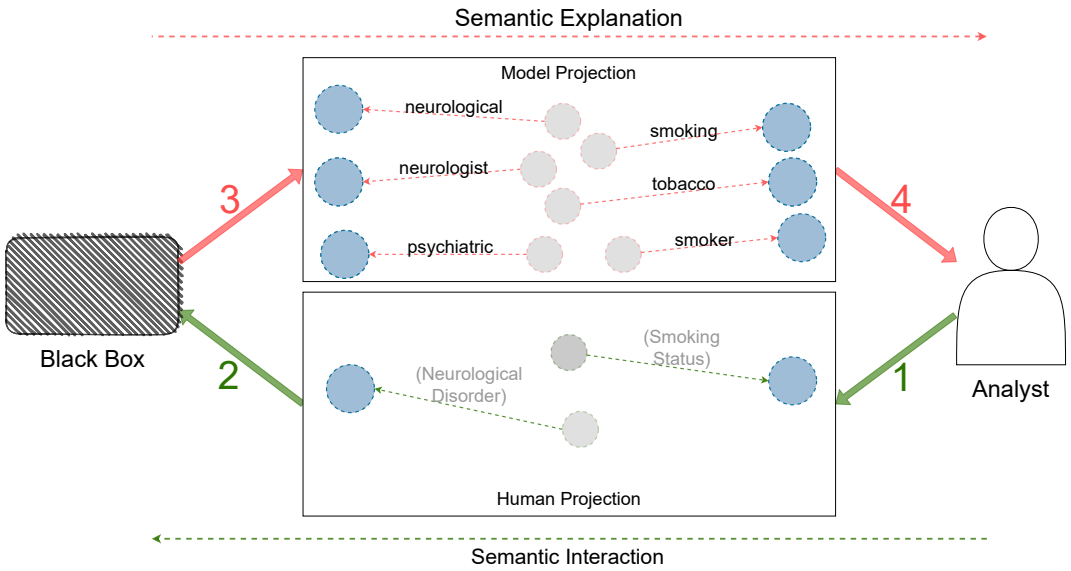


Figure 6.3: Design rationale of SE. With SI, the analyst performs informal but natural analytic interactions (Step 1) in the projection (human projection) without explicitly formalizing these relationships back to input features (words). The system interprets the associated analytical reasoning and updates the internal parameters (Step 2). Therefore, the analyst remains in the cognitive zone, focused on their sensemaking activity. Similarly, with SE, the system updates the data projection and also provides intuitive causal reasoning of the projection changes (Step 3). The analyst understands the projection updates via visual explanations, by receiving feedback about data features that support the projection. Thus, the analyst cognitively formalizes their understanding of the projected concepts (Step 4).

articles have different semantic meanings, *smoking status* vs. *neurological disorder*. Alternatively, they might create a spatial cluster of several similar articles related to neurological disorders as a covid risk factor. Their cluster defines a novel high-level semantic concept. They defined it using cases, rather than feature descriptions, as is often the situation in early stages of the sensemaking process before clear concept definitions have emerged. Based on this user-defined case-based similarity feedback from the semantic interactions, the model learns an updated high-dimensional representation of the data for re-projection (Step 2).

Similarly, the design rationale behind SE is also the connection between mental projection and computational projection. The system should explicitly formalize and contextualize the internal causal reasoning in updating the projection (Step 3). Specifically, the system

should explicitly show how essential features drive the underlying model to update all the observations, similar to users' spatial analytic interaction. For example, as shown in the model projection in Fig. 6.3, all the data points (six articles in this example) are moved into two separate clusters by the underlying model, similar to the analyst's interactions. Further, all the movements made by the model are interpreted with the essential features that contribute most to these movements. These visual explanations can be correlated with users' analytical reasoning behind their interactions. The explanations *neurological*, *neurologic* and *psychiatric* support the analyst's internal reasoning of *neurological disorder*. The explanations *smoking*, *tobacco* and *smoker* support the analyst's internal reasoning of *smoking status*. The analyst gains a deeper, more formalized understanding of the meaning of the clusters they created.

SI leverages the cognitive connection formed between the user and the spatial layout. These interactions carry semantic meanings in helping users define semantic concepts. Consequently, SE leverages the cognitive connection formed between the user and the spatial layout for black-box explanations. These explanations support the incremental formalization that correlates the abstract projection to input features (words) and helps analysts formally define the semantic concepts. Therefore, analysts can benefit from the SE by understanding the visual causal explanations of updated observations within a metaphor they are familiar with, namely spatial interactions in the projection (Step 4).

6.2.2 Model Pipeline

The SE framework is designed based on the rationale described in the previous subsection to integrate explanations into the SI pipeline. SE (Fig. 6.4) consists of two components in the model prediction direction: the counterfactual engine and the counterfactual projection.

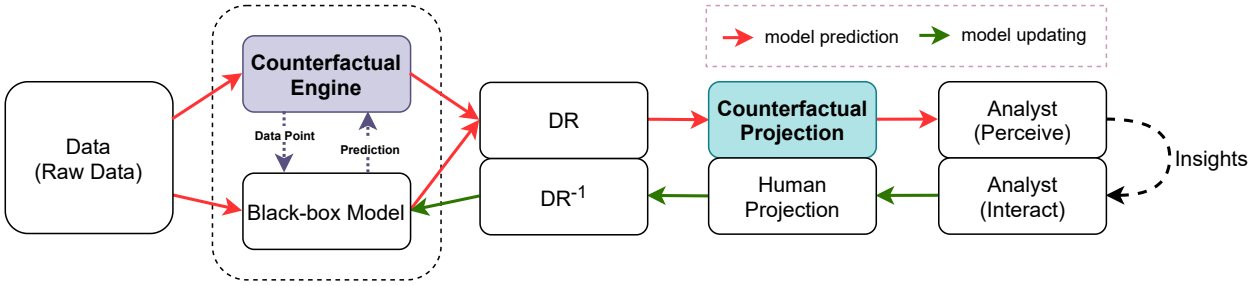


Figure 6.4: SE pipeline extends the DeepSI pipeline (Fig. 7.2) with two key components: counterfactual engine and counterfactual projection. Counterfactual engine interprets the black-box model by identifying representative counterfactual explanations for each of the data instances. Counterfactual projection visualizes both data points and relevant counterfactual explanations in the same projection. The counterfactuals, contextually projected with data instances, elicit cognitive causal reasoning between input features and the updated model output.

–**Counterfactual Engine:** Counterfactual explanations elicit causal thinking between the change of input features and prediction updates for a single data point. During the model prediction process, the updated model first calculates the new high-dimensional representations y for all data inputs x through the black-box model:

$$y = f_w(x) \tag{6.1}$$

where f_w is the black-box model with the current parameters w which get updated in the model updating direction. Meanwhile, the counterfactual engine utilizes the counterfactual method to generate representative explanations for each of the data points.

Existing counterfactual methods [167] are mainly applied to interpreting classification models where ground truth information is available in finding valid counterfactual examples. However, there is no ground truth in the context of interactive DR. We pose the new optimization objective in a unsupervised manner as follows:

$$\arg\max_{c} \left| (f_w(c) - y)^2 - d(c, x) \right| \tag{6.2}$$

where $d(\cdot, \cdot)$ is a function that measures the raw data distance between the counterfactual example c to the original data point x. The new optimization objective consists of *proximity* and *validity*. The proximity objective seeks to create a set of counterfactual examples c from the original instance x with small feature perturbations by minimizing their distances $d(c, x)$. The validity objective seeks to select important counterfactual examples c that make influential contributions to the changes in the high-dimensional representation space $(f_w(c) - f_w(x))^2$. Taken together, we seek counterfactuals where small data changes cause large representation changes.

–**Counterfactual Projection:** The original model projection is a traditional projection scatterplot to show similarities between data points based on their 2D distances. As shown in Eq. 6.3, the new counterfactual projection ($\boldsymbol{r}^{\boldsymbol{x}\cup\boldsymbol{c}}$) contains both data instances ($\boldsymbol{y}^{\boldsymbol{x}}$) and relevant counterfactuals ($\boldsymbol{y}^{\boldsymbol{c}}$) projected from the high-dimensional space. A new visual design is needed to display both data instances and their relevant counterfactuals on top of the basic scatterplot.

$$\boldsymbol{r}^{\boldsymbol{x}\cup\boldsymbol{c}} = \mathrm{DR}(\boldsymbol{y}^{\boldsymbol{x}} \cup \boldsymbol{y}^{\boldsymbol{c}}) \tag{6.3}$$

Inspired by flow-based scatterplot [25] and Praxis [23], we propose the counterfactual projection to visualize both data points and their relevant counterfactuals. Flow-based scatterplots augment the traditional metaphor using the sensitivity information as streamlines to highlight the local variation of one feature with respect to another feature dimension in the dataset for all data points. Praxis uses a proline to show how one input feature change could influence a single data point's model output. In our counterfactual projection (model projection in Fig. 6.3), we integrate counterfactuals for all data points into the traditional scatterplot with proline-like visual metaphors to highlight both local and global patterns of explanations. In addition, the counterfactual projection provides a series of interactions to explore explanations and avoid visual clutter.

The counterfactual projection *contextualizes* [174] explanation information of the underlying model into the existing visual metaphor for the data points. The visual update is non-intrusive and does not distort these existing artifacts on which the analyst is working. As a result, SE can augment the human mental model, providing new information relevant to the analyst's interests in context within their *space to think* [6]. Besides, extra detailed views can be employed to help analysts make sense of the feature influence within the data points of investigation, similar to StarSPIRE [19]. We describe the detailed visualization design and implementation of the counterfactual projection for the visual text analytics system in the following section.

These two components enable the model explanations with the nice properties described in the previous subsection. It is worth noting that the counterfactual examples do not represent where the data instances were before the interaction and re-projection. The counterfactual explanation shows the projection position of counterfactual instances without these essential words s from original data instances in current updated model.

6.3 DeepSE System

We apply our SE framework to a visual text analytics system, called *DeepSE*, that uses semantic interaction powered by deep learning with the BERT model. DeepSE builds on DeepSI [12] and adds the SE framework.

6.3.1 Overview

DeepSE is a web-based system that includes the backend interactive DR and the frontend visualization, implemented based on Fig. 7.2. The interactive DR component is implemented

in Python with Flask, consisting of the DL model BERT and a linear DR method MDS. BERT is a state-of-the-art DL model for natural language processing (NLP) tasks. The BERT model is trained interactively to generate better high-dimensional representations y for the data points x based on users' semantic interactions captured by DR. We used MDS as the linear DR method because MDS enables analysts to express their synthesis process by manipulating data point proximities to reflect their perceived similarity. The frontend visualization is implemented with HTML and D3. The projection is a traditional scatterplot. Analysts can reposition data points on the scatterplot to express their concepts.

The interactive process conducted using the original system without SE is shown in Fig 6.1. The interactive DR with the finetuned BERT model updated the projection layout without providing any explanations. We apply the SE framework to explain the projection updates made by the finetuned BERT model during the sensemaking process. In the following two subsections, we discuss the newly added counterfactual engine and the enhanced visualization in detail.

6.3.2 Counterfactual Engine

Here, we describe the algorithm details of the counterfactual engine that generates counterfactuals for the BERT model to explain the projection. Subsequently, we describe practical solutions to improve the counterfactual generation efficiency.

–**Evidence Counterfactual:** Inspired by the work on explaining document classifications [115], we define a small set of words s including k words $\{w_1, w_2,..., w_k\}$ as the features to change to create a counterfactual example c, such that removing these words from the instance x makes significant changes to the predicted high-dimensional representations y, thereby leading to significant changes in the projection. Therefore, the counterfactual en-

gine can create a diversity of m counterfactual examples $\{c_1, c_2, ..., c_m\}$ with different word sets $\{s_1, s_2, ..., s_m\}$ for a document x. In the current unsupervised context, we first set the size of the word set k to a fixed number. The goal is to find k words, such that removing these k words from the instance leads to the most significant high-dimensional representation changes. To provide a diversity of explanations, we select the top m most influential counterfactual examples for each document as representative explanations. Therefore, all the n documents and their relevant m representative explanations will be projected into the counterfactual projection.

Data: x = document;
k = size of word combination;
m = number of counterfactuals;
v = vocabulary of the document;
p = size of vocabulary to consider;
Result: m influential counterfactuals
begin
  calculate word attention in BERT model;
  v' = pick top-p words with highest attention from v;
  foreach *wordset s of (p choose k) word combinations in* v' **do**
    $c = x$ remove words s;
    validity = $(f_w(x) - f_w(c))^2$;
    store c, validity;
  end
  pick top-m c based on validity value;
end

Algorithm 1: Generating counterfactuals for a document

–**Practical Solution:** To create m representative counterfactual examples for each document, a straightforward approach would be to conduct a complete search through the space of k-word combinations. However, analysts need real-time feedback of the counterfactual explanations when performing interactive sensemaking tasks with VA systems. For this practical consideration, we employ two methods to reduce the search space of s. Our proposed algorithm is specified in Algorithm 1. First, we only keep a small proportion of words with

high attention values in the explanation word set s, where the attention value is calculated by the BERT model's internal attention maps [42]. Another practical solution is to set a small number of counterfactual examples m for a given instance. However, there is a tradeoff between speed and accuracy in this approach.

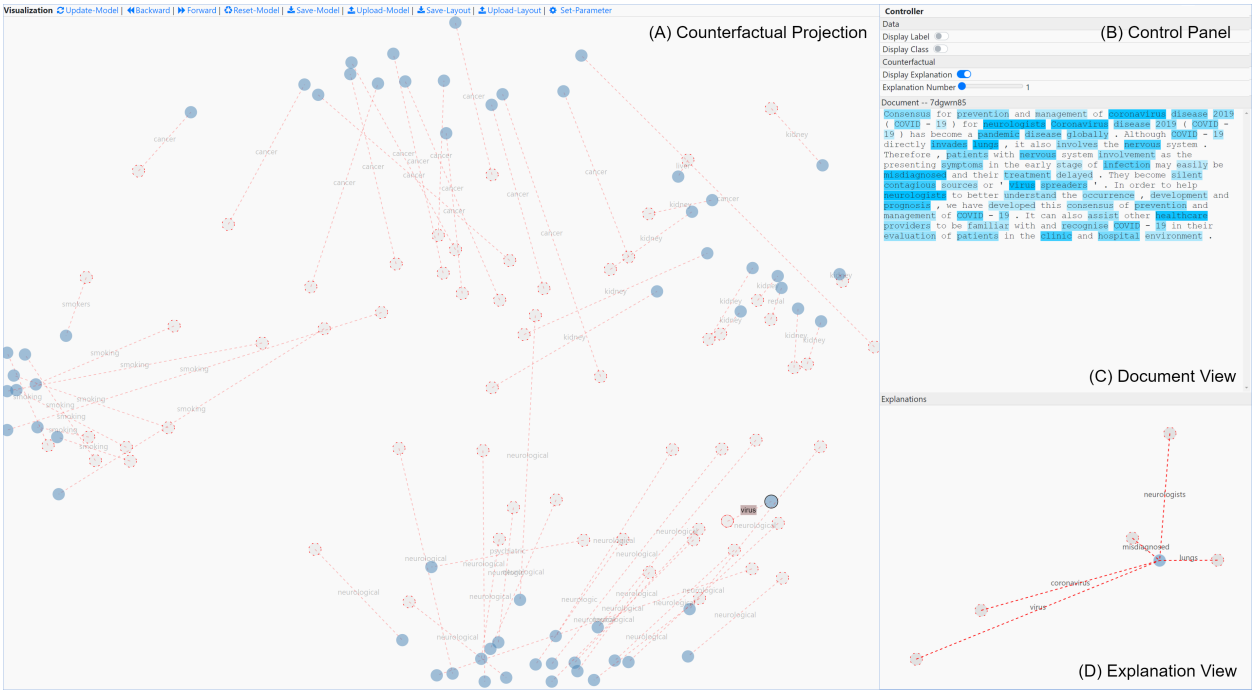


Figure 6.5: The visual interface of DeepSE with DL explanations. The counterfactual projection (A) visualizes both instances and their relevant counterfactuals on the scatterplot based on their projection similarity. The control panel (B) provides visual settings of instances and explanations in the counterfactual projection. The document view (C) and explanation view (D) show the corresponding document content and the relevant explanations for the selected data point in the counterfactual projection.

6.3.3 Visualization Design

As shown in Fig. 6.5, the visual interface consists of four views: the main *counterfactual projection* view, *control panel*, *document view*, and *explanation view*. The interactive visual design aims to contextualize the counterfactual explanations into data projection. Addition-

ally, relevant interactions are provided by the visualization for analysts to explore explanations as needed.

–**Counterfactual Projection:** The counterfactual projection is the main view and entry point for analysts. This view provides visualization for both predicted data points and their relevant representative counterfactuals (shown in Eq. 6.3). Similar to the original scatterplot, data points are rendered as solid blue dots and positioned based on their similarities. In addition to data points, counterfactual examples are also visualized as dots but in a smaller size and with transparent gray color so that analysts can easily distinguish between data instances and explanations. The dotted red line segment between the data instance and the counterfactual example shows the projection (DR) changes induced by the input feature changes, precisely the removed word set s to create the counterfactual example. The text label overlying the counterfactual line shows the word set s removed from the original instance to create the counterfactual example at the other end of the line. Analysts can intuitively gain insight that removing the word set from the data instances will lead to the movement of the instance to the counterfactual position. In reverse, these counterfactual lines let analysts obtain a sense of the data flow from the counterfactual examples to the instances because of owning the word set. Therefore, analysts gain an overall comprehension of how the projection is influenced by important words in the documents (Fig. 6.5).

This view provides a series of interactions for explanation explorations. First, the analyst can view the details of a data point by selecting it. The selected data point, together with relevant explanations, will be highlighted in the scatterplot. Selecting a data point also triggers the document view and explanation view to show more details about the selected instance. In addition, the analyst can also hover the mouse on an explanation dot to highlight this explanation and similar explanations for other data points, as shown in Fig. 6.6. When hovering on one explanation of *cancer*, all *cancer* explanations are highlighted. This provides

global patterns for the local-level explanations.

–**Control Panel:** The control panel allows the analyst to change the display of the counterfactual projection. One instance could have multiple relevant explanations shown in the scatterplot. An excess of counterfactuals shown in the scatterplot will lead to visual clutter and inhibit analysts from interpreting the layout. As shown in the panel, the analyst can choose to hide or display counterfactuals (dotted lines and counterfactual example dots) in the projection by toggling the *Display Explanation* button. The analyst can also select to display only the top counterfactual explanations for each of the instances by setting the *Explanation Number*. Besides setting explanations, the analyst can also toggle text labels for data points in the counterfactual projection to focus on the important parts of the layout.

–**Document View:** When a document in the scatterplot is selected, the document view updates correspondingly to display the full content of the selected document for analysts to review. Specifically, the document view shows the document content as a text heatmap visualization. The word importance is calculated based on the internal attention maps of the BERT model.

–**Explanation View:** The explanation view enables analysts to inspect the full explanations for the selected document. This is the detailed view of all the m representative explanations for the selected instance.

6.4 Case Study: COVID-19

We demonstrate the efficacy of DeepSE through a case study. COVID-19 has become a global pandemic. It is essential that medical researchers quickly find relevant documents about a specific research question, given the extensive coronavirus literature. In this case study, we

show how DeepSE can help a medical researcher to identify articles related to specific risk factors for COVID-19, and analyze the data projection to understand the underlying model.

–**Dataset and Task:** The dataset consists of 62 academic articles, including four main risk factors: cancer (15 articles), chronic kidney disease (13 articles), neurological disorders (23 articles), and smoking status (11 articles). These articles are carefully chosen from the COVID-19 Open Research Dataset (CORD-19) [169] by an expert. We used these four risk factors as the ground truth for the test task and loaded these 62 articles into the DeepSE prototype. Therefore, the test task was to organize these 62 articles into four clusters with our DeepSE prototype such that each cluster represented articles of a specific risk factor.

–**Update Projection:** Fig. 6.1 shows the process of the communication between the analyst and system. Frame a shows the initial layout updated by the default pretrained BERT model. In the initial projection layout, all the articles are spatially intermixed. This means that the pretrained BERT model cannot distinguish these articles by their related risk factors. The analyst begins the exploration and performs SI to provide visual feedback on 12 of the documents regarding articles about different risk factors to reflect the perceived connections between articles (in Frame b). Frame c shows the model projection updated by the underlying model. The overall data points show four clear clusters but do not show how the clusters are formed. Without SE, the analyst must check each of the documents to ensure all articles are correctly grouped and to learn about related topics.

–**Configure Visualization:** Therefore, the analyst first turns on the *Display Explanation* to show explanations and make sense of the model projection with the help of SE. To avoid visual clutter and focus on the visual layout, the analyst hides the text labels by turning off the *Display Label* button, and sets the *Explanation Number* to 1 to display only the most essential explanation for each of the documents.

–**Explain Layout:** As shown in Fig. 6.5, the counterfactual projection shows all the data points and their relevant counterfactual explanations. In the scatterplot, almost all the counterfactual dots are in the more internal area of the scatterplot, compared with data points. The global trends of counterfactual lines show the data flow from the center to the margins. There are essential words that contribute most to the forming of these four clusters. This indicates the updated model pushes all the instances distant from the center to form these four clusters in a clear manner because of some essential words related to covid risk factors.

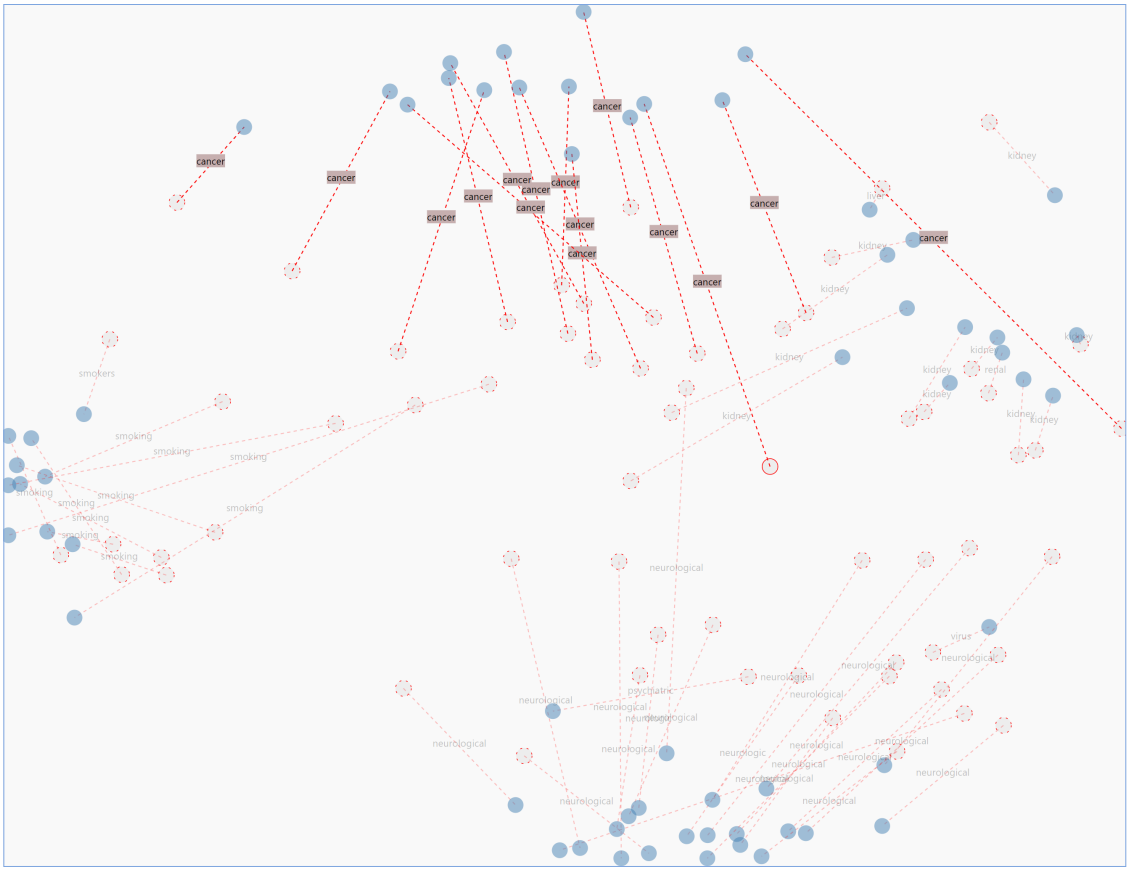


Figure 6.6: *Cancer* cluster explanation. Counterfactuals with the word *cancer* cause the upward movement of data points into the cluster from the center of the projection, indicating that *cancer* is an influential word in forming the cluster.

–**Explain Cluster:** To check the semantics of the projection in detail, the analyst begins

their investigation of each cluster. When the analyst hovers the mouse on one explanation about *cancer* in the top cluster, all other related explanations are also highlighted. As shown in Fig. 6.6, the word *cancer* contributes the most in forming the cluster. The analyst can make the direct connection between this model generated cluster on the top with the manually created *cancer* group in Fig. 6.1b. Another example in Fig. 6.7 shows the words *smoking* and *smokers* have a dominant effect on forming the leftmost cluster. The other two clusters of *kidney disease* and *neurological disorder* also present similar patterns.

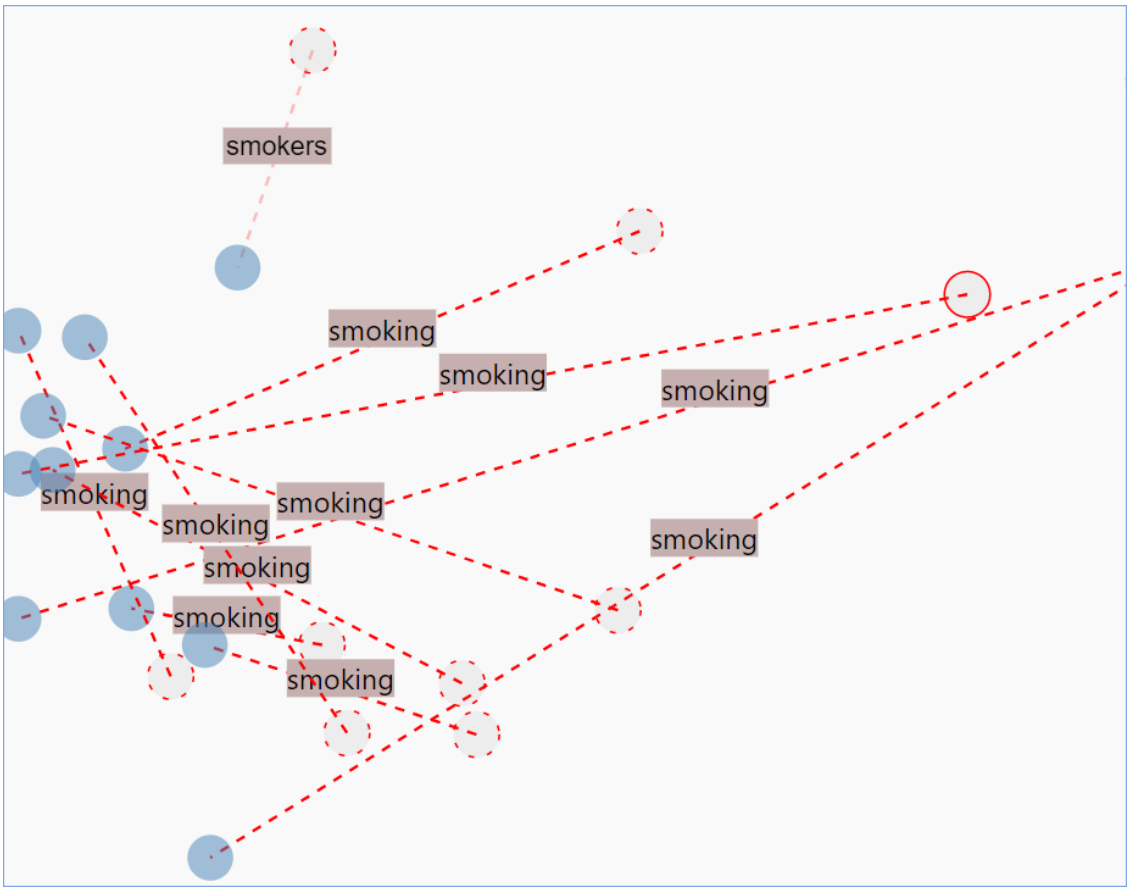


Figure 6.7: Counterfactual explanations show that the cluster *smoking status* is formed by documents with the keyword *smoking* or *smoker*. The underlying model moves documents from the center of the projection to the left because of these keywords.

–**Explain Outlier:** The analyst further examines the bottom cluster of *neurological disorder*. Then the analyst finds an outlier on the top right of the cluster, highlighted in Fig. 6.5.

The analyst clicks the outlier to investigate why the outlier is relatively far away from the cluster than other documents inside this cluster. The document view and explanation view of Fig. 6.5 display the detailed explanations about the outlier. The heatmap background in document view shows important words *coronavirus*, *virus*, *neurologists*, and *lungs* calculated based on attention maps inside the BERT model. The explanation view also shows consistent outcomes. The key word *neurologists* pushes the document down to the correct cluster in the bottom, while the word *lungs* pushes it toward the leftmost *smoking status* cluster. The other two words, *virus* and *coronavirus*, push the document away from the correct cluster to the center of the layout.

–**Draw Conclusions:** Based on previous projection reasoning, the analyst claims that the model constructed clusters shown in the projection are consistent with the clusters created by themselves with SI. The visual explanations show these model-generated clusters capture even more accurate semantic meanings than the analyst expressed. If the current explanations do not convince the analyst, they can further update the layout and tune the underlying BERT model. Therefore, the analyst can focus on the sensemaking tasks while gaining insights into the model and building trust.

6.5 Discussion

6.5.1 Generalizations

In terms of the DL model, DeepSE currently uses BERT, but could be generalized to other DL models for NLP, such as GPT-3, RNN, or to any other black box model. This is because the counterfactual explanation is a model-agnostic explanation method. The counterfactual engine currently assumes text data, but could be generalized to other data types by designing

new methods for generating candidate counterfactuals c and corresponding function d.

6.5.2 Performance

The most significant issue with the counterfactual explanation method is the time complexity. The time complexity of the counterfactual engine for each interaction depends on word combination size k, data size n, and vocabulary size v. For each data point, the counterfactual engine needs to find m counterfactual explanations from a large search space of v *choose* k word combinations. We use a heuristic method based on BERT's internal attention-maps to reduce v to a small constant amount by selecting the top v' attention words. This limits the search space to v' *choose* k, for all n documents. By default we further limit $k = 1$. In the case study (with $n = 61$, $m = 5$ and $v' = 16$), performance for the counterfactual engine is near real time (0.83s), and is faster than the more expensive semantic interaction updates to the BERT model (1.65s) and MDS projection (1.89s). To enable $k > 1$, our attention-based heuristics could be scaled up to find word tuples that contain words that pay high attention to each other in BERT. Beyond heuristics, optimization methods have been proposed to search counterfactuals based on the objective defined in Eq. 6.2. The system could also employ progressive visualization methods to calculate and load explanations on-demand.

6.5.3 Importance of Contextual Explanation

From user feedback, we discovered a potential misunderstanding of the meaning of the length and orientation of the lines in the detail explanation view. In the design, the detail explanation is shown in a separate view so that it can be zoomed in to accentuate the length of short explanation lines. Unfortunately, by divorcing the explanation from the projection context, some users misunderstood its meaning. However, users did understand the meaning

of the explanations within the main projection. Thus, this finding confirms the importance and necessity of contextualizing the explanations into the original counterfactual projection. Therefore, a revised design replaces the separate explanation view with a focus+context lens that displays the zoomed detail explanation within the projection. As shown in Fig. 6.8, when the analyst clicks a particular data point, the fisheye lens will appear over the counterfactual projection and show the detailed explanations about the data point in the current position. While we have explored contextual explanations for the projection, further research could explore contextual explanations for the highlighted words in the document view.

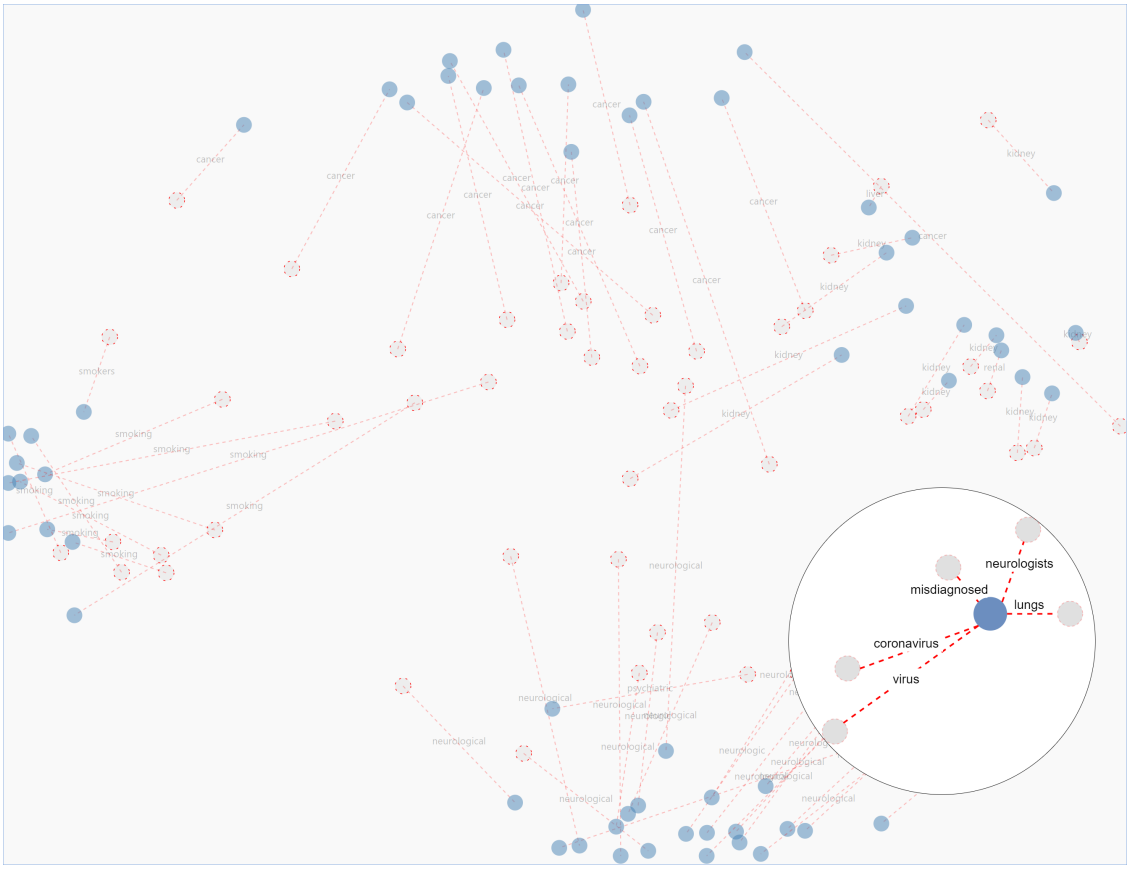


Figure 6.8: Revised visual interface displays the explanation details for the selected document through a fisheye lens, instead of a separate detailed view.

6.5.4 Incremental Formalization via DL Model

Currently, DeepSE helps analysts formalize their mental model by correlating their high-level concepts expressed through the projection to specific low-level feature words. However, this is only the first step in supporting the incremental formalization. SE only provides a 2-layer mapping between abstract concepts and input features. Yet, the multi-layer structure of DL networks offer an opportunity to provide a richer abstraction hierarchy to assist analysts in formalizing complex concepts. Through a stack of neural layers, DL models learn representations of data with multiple levels of abstractions. Visualizing information from these layers could reveal additional relationships between documents and words. For example, a hierarchical concept map could be overlayed on the projection.

Furthermore, the system should assist formalization in an incremental fashion. Users continue to interact and refine their concepts in repeated loops of the pipeline. At each iteration of the loop, the model updates. Currently, DeepSE shows the current model explanation for the current projection. However, it might be useful to specifically provide an incremental explanation that emphasizes the difference from the previous iteration of the model.

6.5.5 Counterfactual Projection Distortion

There is a tradeoff between distortion and explanation in current counterfactual projection. To contextualize explanations into the spatial layout of data, we project both the data and their relevant counterfactual examples at the same time via the DR approach MDS (shown in Eq. 6.3). However, the new added counterfactual examples affect the original data mapping from high-dimensional to lower dimensional spaces. It is highly like that the MDS projection is distorted by the counterfactual explanation. In other words, the layout patterns observed in the original data projection might not be exactly the same ones existing in the

counterfactual projection. Possible solutions of counterfactual projection distortions include projecting each set independently and then overlaying them, or projecting the counterfactuals into a pinned projection of data.

6.5.6 Hierarchical Explanations

Since the space of explanations is large, interactive visualization and navigation of the explanations is needed. Currently, DeepSE divides the explanations between overview projection and detail explanation view. The visualization mantra of overview first, zoom and filter, then details on demand, could be applied to enable visual explanations at different levels of hierarchy. The counterfactual projection could provide cluster-level visual explanations as an entry point, such as a cluster tag cloud, or edge-bundling of counterfactual lines for clusters. Users can first search through each of the clusters (either automatically or manually selected) based on the summarized cluster-level explanations. Then they can zoom in on an individual cluster to investigate more details, such as more diverse counterfactuals for that cluster and pairwise document relationships.

6.6 Conclusion

In this work, we proposed SE to contextualize explanations into VA systems with black-box models for sensemaking tasks. SE is designed as an output analogy to SI input. With SE, analysts can better formalize concepts from the underlying black-box model while maintaining focus on the analytic task at hand. The SE framework contains two main components: the counterfactual engine and the counterfactual projection. These key elements forge the natural connections between cognitive projections and computational projections, thereby

yielding contextual explanations. We implemented DeepSE, a visual text analysis system with interactive BERT model, based on the SE framework. We demonstrated the utility of DeepSE via a usage study with three specific visual pattern explanations.

Chapter 7

NeuralSI: Neural Design of Semantic Interaction Systems

The constructed SI pipeline bridge the DL training loop and the analyst sensemaking loop via the interactive DR. However, the traditional DR method does not ideally introduce semantic interactions as ground truth into the training process of the DL model. The whole batch processing of the DR model does not fit the gradient descent optimization method for DL training. Furthermore, the interactive DR does not provide consistent projections through different communication iterations. The usage of traditional DR as the bridge between the human and DL inhibits collaborative sensemaking. An appropriate interactive neural network-based DR design is needed to advance the SI pipeline to train the DL model and provide consistent projections as feedback. This chapter focuses on the **RQ 4** to replace the traditional interactive DR component of the SI pipeline with a projection network for effective DL model training and consistent model projection.

An increasing number of studies have utilized interactive deep learning as the analytic model of visual analytics systems for complex sensemaking tasks. These systems follow the standard visual analytics framework where the analytic model and the visualization method are two separate components. While these systems better capture analysts' intents, due to incompatibilities between these two components they cannot support several desired properties for visual analytics, including out-of-sample capability, stability, and inference speed. To

avoid this issue, we propose the neural design framework of visual analytics for interactive deep learning. In our framework, we replace the visualization method with a neural network and append it to the deep learning model as the task-specific output layer. Therefore, both the analytic model and visualization method form one integrated end-to-end trainable deep neural network. In order to understand what promotes the performance of the neural design, we systematically explored and studied the major design aspects of our framework. When choosing the best design options and combinations, we can implement a neural system with comparable performance to state-of-the-art standard systems with interactive deep learning.

7.1 Introduction

Visual analytics (VA) [38] enables human-AI collaborative sensemaking through interactive visualizations. The standard VA framework consists of three separate components: data, visualization, and analytic model [142]. The analytic model captures the user's intents based on their visual interactions and abstractions from the raw data. The visualization component is an automatic model that generates visualizations from the data and patterns extracted by the analytic model, and it is the primary interface between the analyst and the analytic model.

Semantic interaction (SI) [53] is a visual interaction methodology whih has been comomly applied to VA systems. SI-enabled systems let the analyst directly manipulate interactive projections of data. The semantic meaning of these interactions indicates the relationships the analyst wishes to find within the data during the sensemaking process [133]. Through these intuitive and natural interactions, the analyst can remain within the cognitive zone [65], and the system thereby enhances the analyst's efficiency in performing analytic tasks. As shown in Figure 7.1, SI-enabled systems follow the standard VA framework: the analytic

model ultilizes a distance metric learning method to infer the intent behind the analyst's semantic interactions; The visualization component is an interactive dimensionality reduction (DR) [143] that acts as a bridge between the high-dimensional space updated by the metric learning method and the low-dimensional visual space manipulated by the analyst.

Figure 7.1: SI pipeline that follows the standard VA framework (adapted from [53, 143]). The interactive DR component serves as the visualization method responsible for capturing the analyst's intent and updating the projection in response. The metric learning method [20] is the analytic model responsible for inferring the intents from interactive DR and providing the updated data relevance as feedback.

Deep learning (DL) [104] is the state-of-the-art representation learning method [10], that can automatically extract abstract and useful hierarchical representations from raw data. The capability of learning nonlinear patterns has boosted the integration of interactive DL into VA systems as the analytic model for complex sensemaking tasks [12, 60, 100]. These systems follow the standard VA framework, where the analytic model and the visualization method are two separate components. During the human-DL co-learning process, these two components are executed individually in sequence: first the analytic model (interactive DL) generates user- and task-specific representations to capture analysts' precise intents; then the visualization method transforms the generated DL representations to a new visual layout totally from scratch. While these systems better capture analysts' intents [12, 13], due to incompatibilities between the two components they cannot support several desired properties for visual analytics, including out-of-sample capability, stability, and inference speed [57, 67, 74].

This work aims to address the incompatibility of components in the standard VA framework and propose the neural design framework of visual analytics for interactive DL. Specifically, we focus the neural design framework on visual analytics systems with semantic interaction.

In our framework, called NeuralSI, both the analytic model and visualization method are combined into one integrated end-to-end trainable deep neural network. The integrated DL model has two parts: (1) a pretrained DL as the backbone, which serves as the analytic model to infer users' concepts and update representation vectors; (2) a small neural network projection head appended to the backbone, which functions as the visualization method to communicate with analysts.

In order to understand what enables effective NeuralSI systems, we investigated three major design aspects in our framework: (1) projection head architecture (linear vs. nonlinear projection head); (2) projection parameter initialization (random vs. MDS-based initialization); (3) loss function for visual interactions (contrastive loss vs. MSE loss). We then systematically studied these design choices of our framework by using BERT as the default backbone in the framework and then applying it to three visual text analytics tasks: Stanford sentiment treebank (SST), Vispubdata, and 20 newsgroups. By combining our findings from the experiments, we considerably improved the neural design of VA systems with SI and showed comparable performance with the state-of-the-art model DeepSI.

Specifically, we claim the following contributions:

1. Identification of a neural design framework of visual analysis for interactive deep learning, NeuralSI, that integrates the analytic model and visualization method into one end-to-end trainable neural network;

2. Investigation of three design aspects to boost the NeuralSI framework performance: projection head architecture, projection parameter initialization, and loss function for visual interactions;

3. A systematic study of the NeuralSI framework combining three design choices and a quantitative comparison with the state-of-the-art model DeepSI.

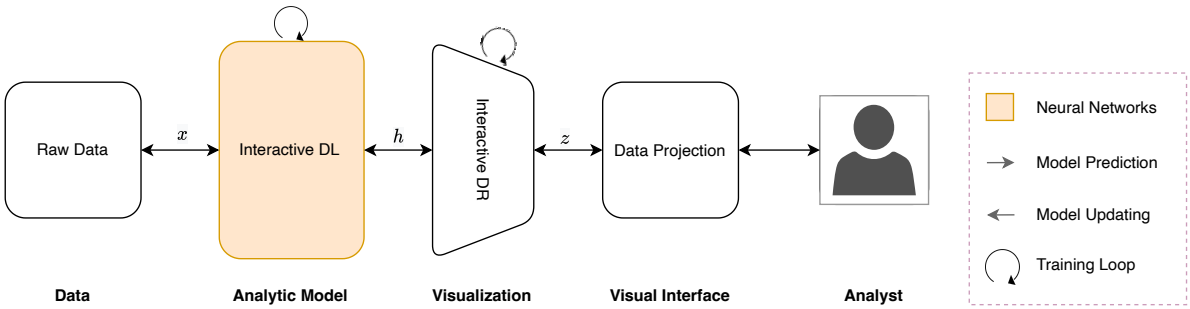


Figure 7.2: DeepSI framework: embedding DL within the standard SI pipeline as the analytic model (adapted from [12, 142]). The interactive DR serves as the visualization component. The usage of traditional DR models makes the visualization component separate from the analytic model.

7.2 Background

In order to frame the discussion of our NeuralSI framework, this section briefly describes DeepSI, the state-of-the-art SI model that uses DL as the underlying analytic model. As shown in Figure 7.2, the DeepSI pipeline follows the standard VA framework, which contains three components: data, DL as the analytic model, and interactive DR (specifically MDS) as the visualization method [142]. DL is the analytic model responsible for generating user- and task-specific representations to capture analysts' precise intents. The interactive DR (MDS) is responsible for updating the visual interface (data projection) based on the model patterns extracted by DL. As the traditional DR, MDS is a nonparametric model and can directly pass semantics of users' interactions to DL training without information loss. This enables DeepSI a state-of-the-art performance.

We illustrate the detailed human-DL co-learning process of using DeepSI as follows. In the forward model-prediction direction, the analytic model (DL) generates new representation vectors for the raw data, x, through forward propagation with current DL parameters,

($\boldsymbol{w}_{\text{backbone}}$), given by the equation

$$\boldsymbol{h} = \text{f}(x, w_{\text{backbone}}) \tag{7.1}$$

where h is the output of the DL $f(\cdot)$. Next, the interactive DR (MDS) projects the high-dimensional DL representations h to the 2D visual interface through the following equation:

$$\boldsymbol{z} = \arg\min_{z} \sum_{i<j\leq n} \Big(||z_i - z_j|| - ||h_i - h_j|| \Big)^2 \tag{7.2}$$

Importantly, for the updated $\boldsymbol{h}$ in each iteration, MDS needs to train the new low-dimensional embeddings $\boldsymbol{z}$ from scratch. Based on the updates of $\boldsymbol{h}$, which represent the new out-of-sample data, a new MDS should be trained from scratch to learn z in each loop.

In the backward model-updating direction, the analyst provides visual feedback by repositioning n data points within the projection. Then, the interactive DR captures the human-defined similarities between n moved data points, $dist(z_i, z_j)$, and uses them to steer the DL model parameters, $\boldsymbol{w}_{\text{backbone}}$, to generate better high-dimensional representations, h, such that the similarity of the representations reflects the proximity of the points in the modified projection, as follows:

$$w_{\text{backbone}} = \arg\min_{w_{\text{backbone}}} \sum_{i<j\leq n} \Big(dist(z_i, z_j) - ||\text{f}(x_i) - \text{f}(x_j))|| \Big)^2 \tag{7.3}$$

The optimization objective is to tune the DL weights, $\boldsymbol{w}_{\text{backbone}}$, to minimize the difference between low-dimensional and high-dimensional distances of n moved data points through backpropagation. After the backpropagation, the updated $\boldsymbol{w}_{\text{backbone}}$ is used in the forward propagation to calculate the new representations, $\boldsymbol{h}$, shown in (7.1).

In DeepSI, the analytic model and visualization method are two separate components. Dur-

ing the human-DL co-learning process, these two components are executed individually in sequence: first the interactive DL generates user- and task-specific representations to capture analysts' precise intents; then the traditional DR method, MDS, has to transform the generated DL representations to a new visual layout totally from scratch. Incompatibilities between these two components cause the system's loss of the desired properties for visual analytics, including out-of-sample capability, stability, and inference speed [57, 67, 117]. To avoid these drawbacks, we propose NeuralSI. Because NeuralSI uses neural networks for both visualization method and the analytic model, it forms one integrated end-to-end trainable deep neural network.

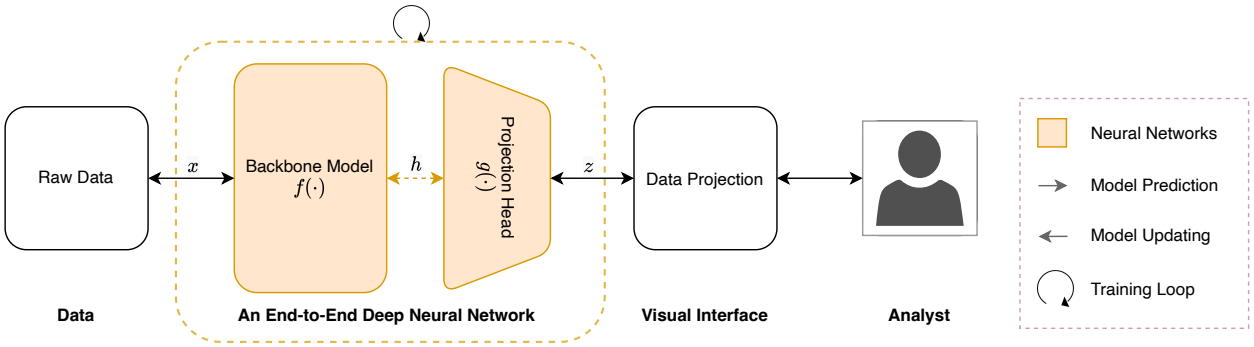


Figure 7.3: The NeuralSI framework uses a deep neural network as the analytic model and a relatively shallow projection neural network as the visualization method. These two components (the analytic model and visualization) form one integrated end-to-end trainable neural network responsible for concept learning and visualization generation.

7.3 Neural Design of Semantic Interaction

In this section, we first describe the NeuralSI framework and then discuss three main design aspects that affect the framework: projection head architecture, projection parameter initialization, and loss function.

7.3.1 The NeuralSI Framework

We propose NeuralSI, a neural deisgn VA framework of semantic interaction which uses an end-to-end trainable deep neural network. The integrated deep neural network is responsible for both the analytic model and visualization method. During the human-DL co-learning process, these two parts are executed in the same training loop as an integrated unit to capture users' intents and generate new visualizations. Therefore, NeuralSI avoids the limitations of existing systems by supporting the desired properties, including out-of-sample capability, stability, and inference speed discussed in section 7.2.

Inspired by SimCLR [30], we describe the integrated neural network in two parts based on their functionalities:

- *A backbone model* $f(\cdot)$ functioning as a deep neural network for extracting representation vectors, h, from input data examples, x, for downstream tasks. In this model, w_{backbone} represents the internal parameters and contains a large number of variables. The framework allows for various options in the network architecture without any constraints. The backbone model serves as the analytic model that infers users' intents, discovers data patterns, and presents them as visual feedback.

- *A projection head* $g(\cdot)$ which is a small neural network appended to the backbone model. The projection head maps the high-dimensional representations, h, to the visual projection, z, so that the user can gain insights from the visual layout. The projection head is the visualization method for visual projection generation. $w_{\text{projection}}$ represents the internal parameters of the projection head. From the DL perspective, the projection head is the task-specific output layer for supervised DR [82].

Figure 7.3 illustrates the detailed usage of NeuralSI through the human-DL co-learning

process. Before the co-learning process, the two parts (the backbone model and the projection head) of the deep neural network are initialized separately. The parameters of the backbone model, w_{backbone}, are trained with a massive number of data to learn general representations that can then be easily adapted to downstream tasks. The initialization of the projection head paramters, $w_{\text{projection}}$, is described in section 7.3.3.

In the forward model-prediction direction, the end-to-end neural network directly generates the visual layout z (2D data projection) from x, with the DL parameters ($\boldsymbol{w}_{\text{backbone}}$ and $\boldsymbol{w}_{\text{projection}}$):

$$\boldsymbol{z} = \text{g}(\text{f}(x, w_{\text{backbone}}), w_{\text{projection}}) \tag{7.4}$$

As shown in (7.4), the high-dimensional DL representations h, calculated by $\text{f}(x)$, are directly passed to the projection head, and are then transformed to the 2D spatialization by $\text{g}(h)$.

In the backward model-updating direction, the analyst modifies the visual layout by repositioning samples to express the preferred similarities. Then, the loss function L (described in section 7.3.4) uses the moved data points, z_i, to steer the parameters of the end-to-end neural network, $g(f(x))$. The process is illustrated in the following equation:

$$w = \arg\min_{w} \mathcal{L}(g(f(x)), z) \tag{7.5}$$

The optimization objective is to fine-tune DL parameters (both $\boldsymbol{w}_{\text{backbone}}$ and $\boldsymbol{w}_{\text{projection}}$) to minimize the difference between predicted $\hat{z}$ and the user-updated z variables. All internal parameters of the DL model (i.e., $\boldsymbol{w}_{\text{backbone}}$ for f, and $\boldsymbol{w}_{\text{projection}}$ for g) are updated in order, from the projection head to the backbone, by a gradient descent optimization algorithm. After the backpropagation, the updated parameters are used in the forward propagation to calculate new projections ($\boldsymbol{z}$ in Equation 7.4).

In the human-DL co-learning process described above, three major design aspects affect the

framework performance: the projection head architecture, projection parameter initialization, and loss function. In the following subsections, we detail each of the different design aspects.

7.3.2 Projection Head Architecture

The projection head, $g(\cdot)$, is the output layer of NeuralSI particularly designed for the downstream task—interactive DR. A variety of neural network architectures have been proposed for DR [57, 67, 74]. For simplicity, we constrained the architecture of the projection head to two basic one-layer neural networks, linear projection and nonlinear projection.

Linear Projection:

that is a basic one-layer linear transformation. As shown in (7.6), linear projection maps the high-dimensional representation, h, to the 2D projection, z:

$$z = g(h) = w_{\text{projection}} * h \tag{7.6}$$

Through linear transformation, a consistent data structure for the high-dimensional space is preserved in the 2D visualization. However, normalization is needed after the transformation to fit the projected outputs to the limited visual space.

Nonlinear Projection:

that is a linear projection followed by a Sigmoid function [70]:

$$z = g(h) = \sigma(w_{\text{projection}} * h) \tag{7.7}$$

Though it makes the projection inconsistent with the high-dimensional space, the sigmoid activation function, σ, constrains the scale of z so that it falls within the limited visual space [57, 117]. Thereby, the projection result can be directly used in the visual layout.

7.3.3 Projection Parameter Initialization

Unlike the nonparametric DR methods used in DeepSI, the nueral projection head of NeuralSI contains trainable paramters $w_{\text{projection}}$. Projection parameter initialization is critical to the overall performance of NeuralSI systems. First, before the human-DL co-learning process, an effective initialization scheme can initially generate a moderate projection layout for data explorations. In addition, the properly initialized weights of the projection head also enhance the performance of the subsequent co-learning process. We explore two basic but commonly used parameter initialization schemes for the projection head, namely: random initialization and MDS-based initialization.

Random Initialization

We used the robust initialization method proposed by Kaiming et al. [72], known as He Initialization, for the projection head.

MDS-based Initialization

Aside from random initialization, neural network projections use two common initialization schemes: (1) initialization from the projections generated by traditional MDPs (discussed in section 2.5), such as T-SNE [113], and (2) self-supervised learning, such as the use of Autoencoders [74]. These two initialization schemes have equal performance, as discussed in the Generalized Autoencoder (GAE) framework [171]. To simplify the comparison with

DeepSI, we initialized the projection head by training its parameters with the MDS projection results.

7.3.4 Loss Functions

NeuralSI systems ultilize users' visual interactions on the data projections as the instructions to tune the intergrated deep neural network $g(f(x))$. Through repositioning projected data points (z) on the visual space, analysts intuitively express their preferences of the dataset. A variety of loss functions with different tuning behaviors can be used to train $g(f(x))$ to infer the concepts bebind users' interactions. For purposes of simplicity and generalization, we focused on the two most commonly used loss functions: contrastive loss [30], and MSE loss [57, 117].

Contrastive Loss

When regarding the SI training as the deep metric learning task, the existing loss functions that could be employed in NeuralSI include contrastive loss, triplet loss [77] and N-pair loss [158]. In the deep metric learning task, the similarity information comes from the repositioned data points on the visual layout. The data points to generate model projections with the desired "similarity $\approx$ proximity" property. Contrastive loss is the most common deep metric learning loss [92]. It is also the primary loss function used in existing SI systems [12, 20, 149], and can be expressed as:

$$\mathcal{L}_{contrastive} = \sum_{i<j\leq n} \Big(||z_i - z_j|| - ||\hat{z}_i - \hat{z}_j|| \Big)^2 \tag{7.8}$$

where z_i are the positions of all moved data points, and $\hat{z}_i$ are the positions predicted by the projection $g(f(x))$. Contrastive loss enables NeuralSI only to need neighborhood relationships between the moved samples.

MSE Loss

When regarding the SI training as the multi-output regression task [17], the NeuralSI system should train the projection z to learn the data projection embedding directly as the supervised multi-output regression task. We used Mean Squared Error (MSE) [87] in this paper for the its simplicity and better performance in comparison to other commonly used regression loss functions, such as mean absolute error (MAE) [180] and log hyperbolic cosine (log-cosh) [29]. MSE is commonly used in MDP neural network training [57, 117] and is expressed as:

$$\mathcal{L}_{MSE} = \sum_{i=1}^{n} (z_i - \hat{z}_i)^2 \tag{7.9}$$

where z_i are positions of all moved data points, and $\hat{z}_i$ are the positions predicted by the projection $g(f(x))$.

7.4 Experiment Design

This section illustrates the experimental design that was used to answer the following two research questions:

- RQ1: How do the three design aspects affect the NeuralSI framework?
- RQ2: How does the performance of NeuralSI with the best configuration compare with the state-of-the-art SI system?

For these two questions, we performed a systematic study of our NeuralSI framework using a simulation-based evaluation method with three text analytic tasks. The experiment design is detailed in the following subsections.

7.4.1 Simulation-Based Evaluation

SI systems interactively learn projections provided by the analyst in an online learning process [79]. To perform quantitative comparisons of these design aspects, we employ the simulation-based evaluation method in the experiments. This algorithm centered method replaces the analyst with a simulation component. In each training loop, the simulated analyst first performs semantic interactions on data points randomly picked from the dataset to train the SI system and then conducts quantitative measurement of the updated visual projection produced by the tuned SI system. In each training iteration, the simulated analyst outputs a current projection accuracy. The projection accuracy over iterations reflects the learning curve [130] of the SI system and is used to systematically compare the NeuralSI implementations with different choices in three design aspects. Admittedly, compared with human-centered analysis, the simulation-based method can only measure the performance of SI systems in clustering tasks. However, the quantitive comparisons enables this method provide an objective analysis of these designs options. For a more detailed illustration of the simulation-based evaluation, please refer to [12, 13].

7.4.2 Dataset and Task

To measure the performance of NeuralSI with the simulation-based evaluation, we applied visual text analytic tasks to the system. There are two reasons for this. First, VA systems with SI are primarily used for visual text analytics tasks, such as intelligence analysis [50] and

research publication exploration [117]. In addition, the state-of-the-art SI system, DeepSI, has been evaluated with visual text analytic tasks, making it a suitable benchmark for comparison purposes.

For this text analysis task, we utilized the three text corpora settings used in DeepSI: Stanford sentiment treebank (SST) dataset [157] with two clusters, denoted as T_{sst}; the Vispubdata dataset [85] with three clusters, denoted as T_{vis}; and 20 newsgroup dataset [1] with four clusters, denoted as T_{news}. These three tasks contained different numbers of clusters. Among these three target tasks, T_{sst} and T_{news} are closer to the pre-training tasks of BERT model. while T_{vis} is more domain-specific and especially difficult .

7.4.3 Prototype Setting

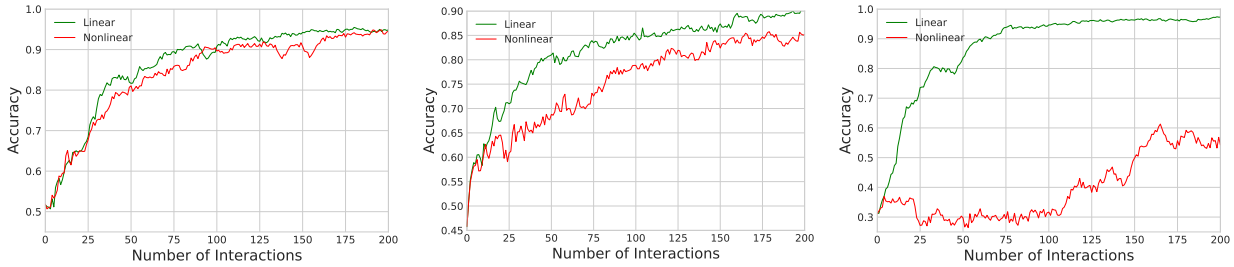
To implement the NeuralSI system in visual text analytic tasks, we used the pretrained BERT [42], a state-of-the-art DL model for NLP tasks, as the default backbone of the framework. We used the base BERT model ($\text{BERT}_{\text{BASE}}$) (bert-base-uncased, 12-layers, 768-hidden, 12-heads, 110M parameters) and appended a MEAN pooling layer to generate representations with a fixed size. Thus, for the given document, the backbone output a 768-dimensional hidden representation, h. In addition, we used the default Adam optimizer [95] to optimize the model parameters inside the NeuralSI (w_{backbone} and $w_{\text{projection}}$). Further, we used the suggested learning rate ($3e^{-5}$) for the fine-tuning of the BERT models [42] in our experiments. For the simulation process, we trained 200 interaction iterations. In each iteration, we randomly selected ten samples and applied semantic interactions to them.

[1] http://qwone.com/ jason/20Newsgroups/

7.4.4 Experiment Process

To address RQ1, we systematically study the three design aspects in our framework using the simulation-based evaluation method on three text analytic tasks. To avoid integration effects caused by multiple design options, we evaluated one design aspect each time and kept other design factors constant. Unless otherwise specified, the three design choices defaulted to linear projection, random initialization, and contrastive loss. After examining the effects of design choices, we combined the findings, built a NeuralSI with the best design options, and compared it with the state-of-the-art method to answer RQ2.

7.5 Results and Findings



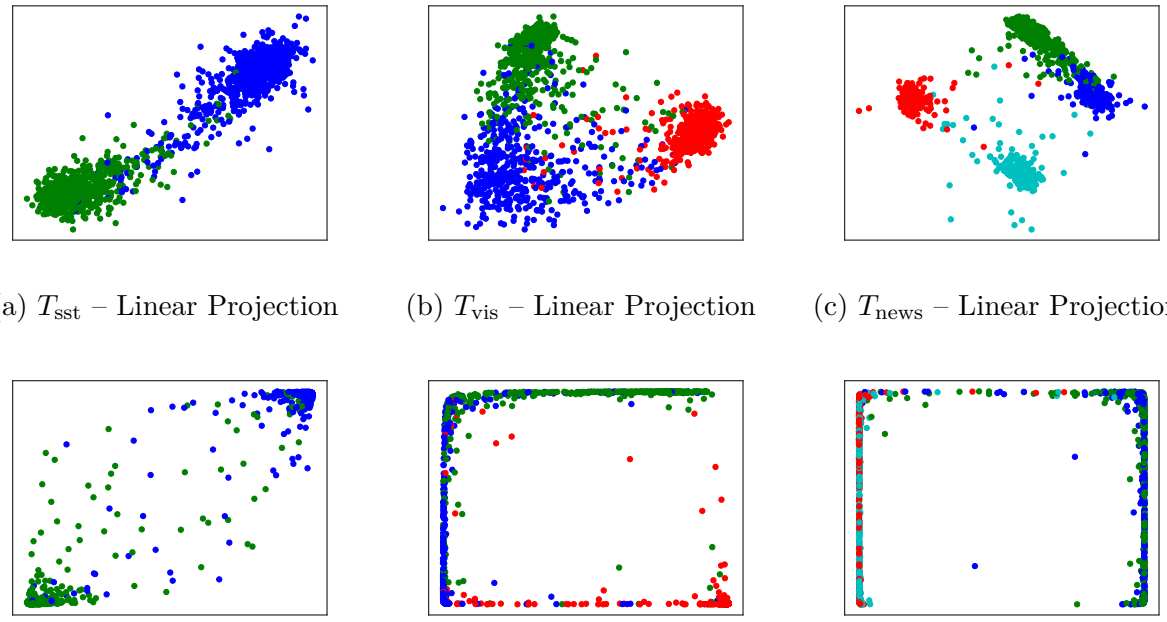
(a) T_{sst} with two clusters (b) T_{vis} with three clusters (c) T_{news} with four clusters

Figure 7.4: The accuracies of both $\text{NeuralSI}_{\text{linear}}$ and $\text{NeuralSI}_{\text{nonlinear}}$ updated projections over 200 iterations across the three tasks (T_{sst}, T_{vis}, and T_{news}) during the simulation-based experiment.

7.5.1 Projection Head Architecture

Results: Figure 7.4 shows the learning curves of the NeuralSI using different projection head architectures in each of the three tasks. We used random initialization to prepare the projection head and contrastive loss as the default loss function for this comparison. The

NeuralSI with the linear projection head was denoted as $\text{NeuralSI}_{\text{linear}}$, and the NeuralSI with the nonlinear projection head was denoted as $\text{NeuralSI}_{\text{nonlinear}}$. The projection parameters were randomly initialized in this experiment, and contrastive loss was used to train the model. We observed that the linear projection head performed better than the nonlinear projection in all three tasks. Regarding model accuracy, $\text{NeuralSI}_{\text{linear}}$ converged to more than 90% accuracy in all three tasks. In contrast, the best performance of the $\text{NeuralSI}_{\text{nonlinear}}$ only showed comparable performance in (T_{sst}) with two clusters. For tasks with more than two clusters (T_{vis} and T_{news}), it only showed small accuracy increases after being trained with a large number of interactions.



(a) T_{sst} – Linear Projection (b) T_{vis} – Linear Projection (c) T_{news} – Linear Projection

(d) T_{sst} – Nonlinear Projection (e) T_{vis} – Nonlinear Projection (f) T_{news} – Nonlinear Projection

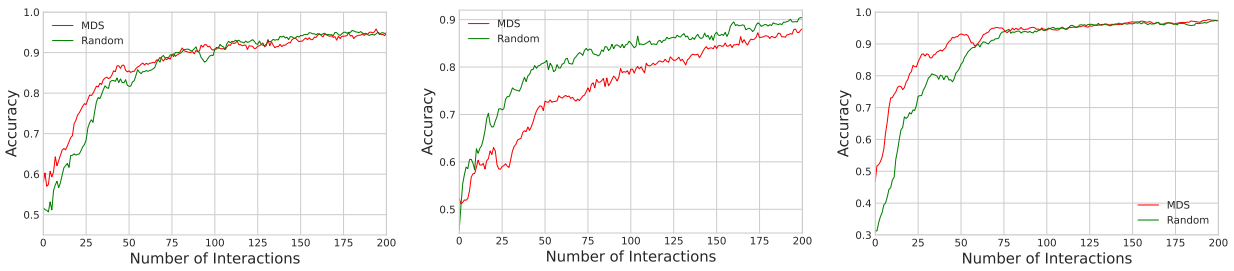
Figure 7.5: The visual projections generated by the $\text{NeuralSI}_{\text{linear}}$ and $\text{NeuralSI}_{\text{nonlinear}}$ models over 200 iterations for three tasks (T_{sst}, T_{vis}, and T_{news}) during the simulation-based experiment.

We conjectured that the better performance of the linear projection head was caused by the linear transformation allowing the high-dimensional representation space to preserve

consistent data patterns from the visual space. However, the nonlinear projection model was shown to make capturing the consistent metric of the 2D visual space by the backbone challenging, especially in tasks with more than two clusters. Accordingly, when projecting the high-dimensional representations, h, to the visual space, z, the visual space was distorted by the sigmoid function. This made the accuracy of the predicted projection even worse.

To verify our hypothesis, we compared the visual projections generated by the NeuralSI$_{\text{linear}}$ and NeuralSI$_{\text{nonlinear}}$ models. As shown in Figure 7.5, the linear projection head generated elliptical-shaped clusters, and data points in the same class were gathered around the centroid in all three tasks. In contrast, clusters created by $NeuralSI_{nonlinear}$ were distorted because the nonlinear sigmoid function squeezed them into the limited visual space.

Findings: *The design of the projection head, which serves as the visualization component, should provide consistent connections between the high-dimensional space and the visual space. In this perspective, the linear projection head is a better architecture for NeuralSI systems to maintain consistent connections.* It is recommended to use nonlinear projections as the task-specific output layers of DL models for classification tasks [30]. However, our results proved that linear projections perform better for interactive DR tasks.



(a) T_{sst} (b) T_{vis} (c) T_{news}

Figure 7.6: The accuracies of the NeuralSI$_{\text{random}}$ and NeuralSI$_{\text{mds}}$ models when updating projections over 200 iterations for each of the three tasks (T_{sst}, T_{vis}, and T_{news}) during the simulation-based experiment.

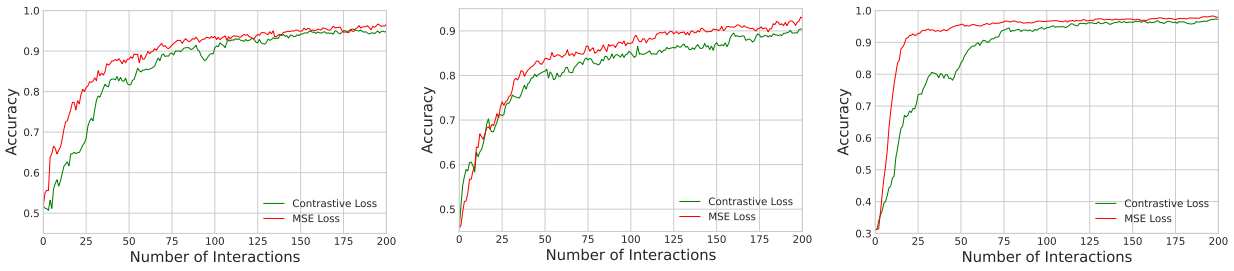
7.5.2 Parameter Initialization

Results: Figure 7.6 shows the performance comparison between NeuralSI with random initialization ($\text{NeuralSI}_{\text{random}}$) and with MDS-based initialization ($\text{NeuralSI}_{\text{mds}}$). We used the contrastive loss as the default loss function and linear projection as the default projection head in these two NeuralSI systems to make the comparisons fair. When the number of interactions is 0, $\text{NeuralSI}_{\text{mds}}$ had better initial accuracies for all three tasks. This indicates that MDS-based initialization allowed the projection head to generate better projections for data exploration before the co-learning process. With enough interactions, the accuracy of each model became more similar as training interactions increased. With the small number of interactions, $\text{NeuralSI}_{\text{mds}}$ kept a significant advantage over $\text{NeuralSI}_{\text{random}}$ in two general purpose tasks (T_{sst} and T_{news}). In the domain-specific task, T_{vis}, $\text{NeuralSI}_{\text{mds}}$ had lower accuracy gains than $\text{NeuralSI}_{\text{random}}$ as learning from training interactions. This result indicates that MDS-initialized parameters hinder the further fine-tuning of the end-to-end deep neural network in domain-specific tasks.

Findings: *For general-purpose datasets and tasks, MDS-based initialization could increase the speed of the learning process without hindering the accuracy gains from training with more visual interactions. However, if the analysis task is domain-specific, it hinders the training performance of NeuralSI systems.* This finding is consistent with the transfer learning and domain adaptation methodologies of reusing pretrained deep neural networks on new tasks [191].

7.5.3 Loss Functions

Results: Figure 7.7 shows the impact of different loss functions on NeuralSI performance. We denote NeuralSI trained with contrastive loss as $\text{NeuralSI}_{\text{contrastive}}$ and NeuralSI trained



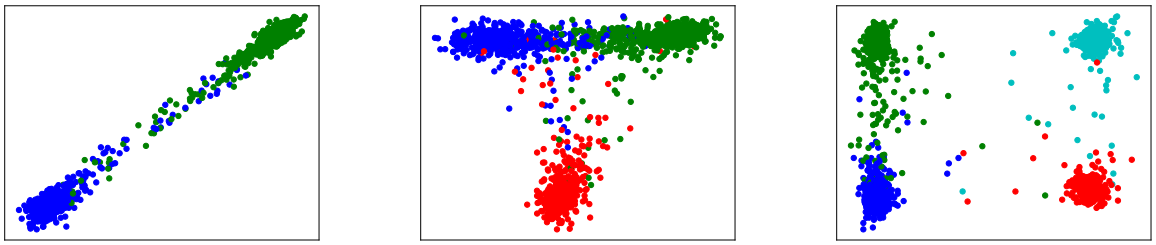
(a) T_{sst} (b) T_{vis} (c) T_{news}

Figure 7.7: The accuracies of the NeuralSI$_{\text{contrastive}}$ and NeuralSI$_{\text{mse}}$ models' projections over 200 iterations for the three tasks (T_{sst}, T_{vis}, and T_{news}) during the simulation-based experiment.

with MSE loss as NeuralSI$_{\text{mse}}$. We found that NeuralSI$_{\text{contrastive}}$ and NeuralSI$_{\text{mse}}$ have comparable performances, though NeuralSI$_{\text{mse}}$ has a slightly better performance in tasks with a smaller number of clusters (T_{sst} and T_{vis}). In the T_{news}, which has more clusters, there is a significant gap between the two loss functions. With more training interactions, the gap between the loss functions increases. NeuralSI$_{\text{mse}}$ approximates to peak accuracy with half number of interactions (25 interactions). In contrast, the performance of NeuralSI$_{\text{contrastive}}$ increased relatively slowly.

To further investigate the reason, we compared the visual projections generated by these two NeuralSI systems. As shown in Figure 7.8, NeuralSI$_{\text{mse}}$ is better at separating different clusters and making use of the whole visual projection. While in projections generated by NeuralSI$_{\text{contrastive}}$ (Figure 7.5), different clusters are not well separated, especially in T_{news} with four clusters. The MSE loss trains the neural network to learn the direct data positions on the visual interface, which makes a good balance between the intra-cluster cohesion and inter-cluster separation [22]. In contrast, the contrastive loss focuses on the training of relative distances between data points [20]. It's difficult for contrastive loss to preserve consistent clustering patterns.

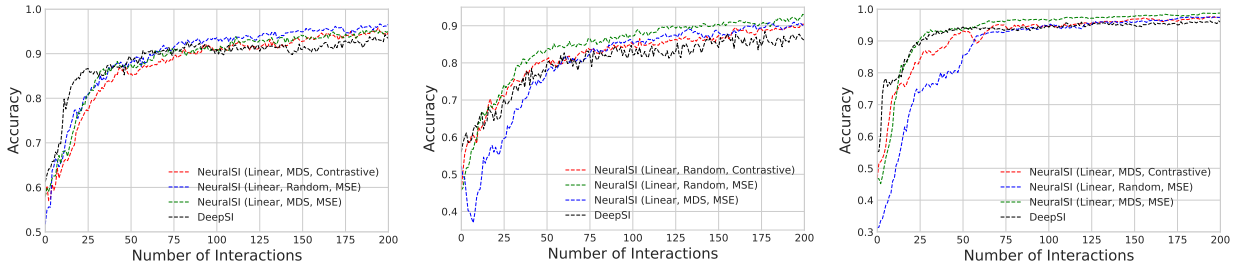
Findings: *NeuralSI benefits more from the supervised learning methodology that uses MSE*



(a) T_{sst} – MSE Projection (b) T_{vis} – MSE Projection (c) T_{news} – MSE Projection

Figure 7.8: The visual projections generated by $\text{NeuralSI}_{\text{mse}}$ over 200 iterations for the three tasks (T_{sst}, T_{vis}, and T_{news}) during the simulation-based experiment.

loss, especially for complex concepts defined by more clusters. This is consistent with our findings in section 7.5.1, which also highlighted the importance of the metric and conceptual consistency between the high-dimensional representation space and the low-dimensional visual space.



(a) T_{sst} (b) T_{vis} (c) T_{news}

Figure 7.9: The accuracies of the DeepSI and NeuralSI models' updated projections over 200 iterations for the three tasks (T_{sst}, T_{vis}, and T_{news}) during the simulation-based experiment.

7.5.4 Design Option Combinations

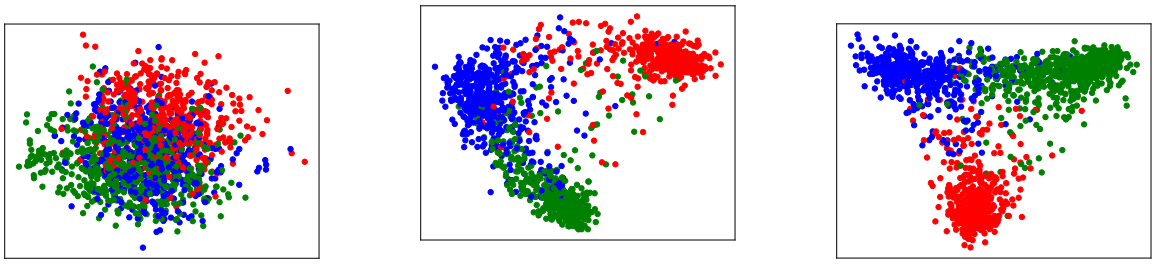
To save space and make the comparison more manageable, we only explored the design option combinations that promoted performance the most. For tasks T_{sst} and T_{news}, we evaluated the design combination of linear projection head, MDS-based initialization, and

MSE Loss. For the task T_{vis}, we explored the combination of linear projection head, random initialization, and MSE loss.

Results: As shown in Figure 7.11b, the learning curve of NeuralSI$_{\text{(linear, mds, mse)}}$ shown in blue experienced a big drop at the first several loops of interactions. The drop also occurred in the learning curve of NeuralSI$_{\text{(linear, mds, mse)}}$ shown in green in Figure 7.11c. We hypothesis that this drop was caused by the inconsistency between the MDS-based initialized projection layouts and the training layouts that the MSE loss learned. In contrast, the contrastive loss was shown to be less likely to have this inconsistency because it only utilizes the pairwise distance information from the training layout. This let NeuralSI$_{\text{(mds, contrastive)}}$ show a better performance than NeuralSI$_{\text{(mds, mse)}}$ at the beginning of the interactive learning process, while NueralSI benefits more from MSE loss than contrastive loss.

To verify our hypothesis, we compared the initial projection made by the DR model MDS, and two projections generated by NeuralSI$_{\text{(mds, contrastive)}}$ and NeuralSI$_{\text{(mds, mse)}}$ over 200 iterations for the task T_{vis}. As shown in Figure 7.10, NeuralSI$_{\text{(mds, contrastive)}}$ (Figure 7.10b) is more consistent with the MDS projections. In both projections, the red cluster is in the top right area of both projections, and the blue and green clusters are in the bottom left area. Instead of learning a consistent projection, NeuralSI$_{\text{(mds, contrastive)}}$ rotated the red cluster to the bottom part of the projection.

Findings: *Different design options may conflict with each other. The inconsistency between the initialization method and the loss function slightly hindered the speed at the beginning of the learning process.* To further boost the learning performance, a more consistent design between the three parts is needed. For example, the MDS initialization and MSE loss use the information on the data points' location, and there might be rotation inconsistencies between the fixed positions.



(a) T_{sst} – MDS Projection (b) T_{vis} – NeuralSI$_{\text{(mds, contrastive)}}$ (c) T_{news} – NeuralSI$_{\text{(mds, mse)}}$

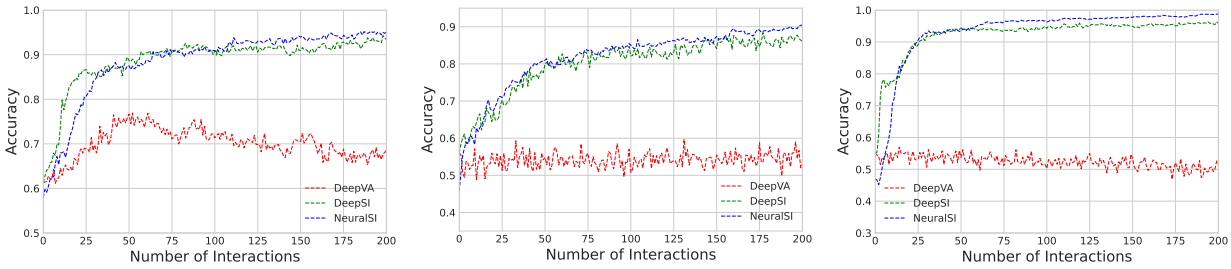
Figure 7.10: Subfigure (a) shows the VIS dataset projection made by the DR model MDS. Subfigures (b) and (c) show the visual projections generated by NeuralSI$_{\text{(mds, contrastive)}}$ and NeuralSI$_{\text{(mds, mse)}}$ over 200 iterations for the task T_{vis} during the simulation-based experiment, respectively.

7.5.5 Comparison with DeepSI

Results: As shown in Figure 7.9 and Figure 7.11, we also compared the best designed NeuralSI systems with the state-of-the-art SI model, DeepSI. At the several initial loops with fewer interactions, DeepSI showed a better performance than all NeuralSI systems. The result is obvious because DeepSI uses the traditional nonparametric DR method as the visualization. There was no need to train extra parameters except parameters inside the backbone model. In addition, in the domain-specific tasks T_{vis}, DeepSI only showed slightly better performance over NeuralSI systems. This result is consistent with our finding in section 7.5.2, indicating that the projection head initialization is not the key factor leading to the performance improvement in domain-specific tasks.

With more interactions, the gap between DeepSI and NeuralSI decreases. After a small number of interaction loops (4 loops in T_{vis}, 12 loops in T_{news} and 30 loops in T_{sst}), NeuralSI even had slightly better performance until approximating to the peak accuracy in all three tasks. This result, together with findings in section 7.5.1, indicates that the projection head does not limit the backbone model's capability of learning nonlinear patterns for sensemaking

tasks.



(a) T_{sst} (b) T_{vis} (c) T_{news}

Figure 7.11: The accuracies of the DeepVA, DeepSI and NeuralSI models' updated projections over 200 iterations for the three tasks (T_{sst}, T_{vis}, and T_{news}) during the simulation-based experiment.

Findings: *With a careful combination of the three design options, NeuralSI systems can show comparable performance to DeepSI. However, there remains room for improvement by developing more appropriate parameter initialization methods, especially for general-purpose tasks.* On the whole, NeuralSI can benefit from the design option combinations, which show comparable performances with DeepSI. However, at the initial steps of the co-learning process, DeepSI with a nonparametric DR gains a better accuracy than NearualSI systems with parametric projection heads. This is consistent with the few-shot learning finding that the nonparametric task-specific output layer is better at training DL models with fewer examples [66, 172].

7.6 Limitations and Future Work

The NeuralSI framework proved effective in the neural design of SI systems for human-DL collaborations. For simplicity of comparison, this paper was limited to three datasets for visual text analytic tasks. More experiments are needed to apply NeuralSI to a variety of analytics tasks, for example, adapting CNN models as the backbone for visual concept

analytics [14]. Further, a user study is needed to measure NeuralSI through the human perspective. Specifically, we wish to investigate how NeuralSI systems boost human performance in complex sensemaking tasks [13, 149, 177].

NeuralSI is the straightforward neural design of VA systems for interactive DL. In the future, we plan to explore more advanced design choices in three aspects of NeuralSI: the projection head architecture, projection parameter initialization, and loss function. First, more complex neural structures can be utilized in designing the projection head. Research on ML4VIS [170], using machine learning models to design effective visualizations automatically, received more attention recently. Increasingly complex neural networks have been used to generate visualizations. For example, Data4Vis [44] uses Seq2seq LSTM [75] to generate visualizations in the declarative language Vega-Lite[145]. As for the projection head [132], task-specific multi-layer neural networks, such as graph embedding [188], could be utilized to generate data projections.

In addition, advanced initialization methods could be used to improve the learning performance of NeuralSI further. For example, weight imprinting [135] combines the best properties of the neural classifier with embedding approaches for solving the low-shot learning problem [31]. It can be redesigned for solving the interactive DR tasks in our NeuralSI framework.

Finally, a particular loss function of NeuralSI could be designed for semantic interactions. One solution could be using multiple loss functions [116, 186], such as combining both the contrastive loss and the MSE loss discussed in section 7.3.4. A more advanced approach is designing one loss function that can combine the advantages of both the contrastive loss and MSE loss so that the loss function is consistent with both the initialization method and users' visual interactions.

7.7 Conclusion

This work focused on the neural design of VA systems for interactive deep learning, NeuralSI. NeuralSI replaces both the analytic model and the visualization components with one integrated end-to-end trainable deep neural network. The whole deep neural network can be divided into two parts based on the functionality: the pretrained backbone for concept learning and the projection head for visualization generation. This design gives NearualSI systems the substantial advantages of neural networks, such as out-of-sample capability, stability, and inference speed. We explored three design aspects to boost the NeuralSI framework performance: projection head architecture, projection parameter initialization, and loss function. We systematically studied the effects of different design options and design combinations. By combining our findings, we can improve the NeuralSI performance considerably and design NeuralSI systems with comparable inference ability to the state-of-the-art SI models.

Chapter 8

Conclusion and Future Work

In this chapter, I conclude this dissertation and discuss future work. I begin by reviewing the major contributions from a broad perspective. Then I discuss opportunities for future work.

8.1 Conclusion

Existing SI systems explore the integration of traditional machine learning models into the human-in-the-loop VA pipeline. This thesis has demonstrated how to instead power SI systems with state-of-the-art DL for human-AI sensemaking. The complexity of the model structure and the training process makes DL models challenging to integrate into the standard SI pipeline.

To tackle these challenges, I have proposed several frameworks, including DeepVA, DeepSI, DeepSE, and NeuralSI. These frameworks integrate interactive and interpretable DL into the human-in-the-loop sensemaking pipeline. Following these frameworks, SI systems can capture the precise intent behind analysts' semantic interactions even when the number of interactions is small (DeepVA, DeepSI), then interpret the captured intent with semantic explanations (DeepSE). Semantic explanations build analysts' trust and help them gain knowledge and make better decisions. Moreover, NeuralSI redesigns the pipeline to enable these SI systems with desired properties, such as out-of-sample capability, stability, and

inference speed.

In Chapters 3 and 4, I developed and validated SI systems with pretrained DL representations, called DeepVA. Compared with traditional SI systems, DeepVA is more effective and efficient in supporting interactive sensemaking tasks via SI. DeepVA offers new insight into effectively bridging the gap between human cognition and computational models. In Chapter 5, I designed the DeepSI framework that integrates deep learning into the human-in-the-loop interactive sensemaking pipeline. Compared with DeepVA, DeepSI more accurately captures users' complex mental models with fewer interactions. In Chapter 6, I designed the DeepSE framework. DeepSE generates counterfactual explanations from DL models and contextualizes these explanations into the visual interfaces. DeepSE can provide intuitive explanations of the interactive human-steered DL model during human-in-the-loop analysis. Finally, in Chapter 7, I proposed neural design of the SI system powered with DL (NeuralSI). NeuralSI replaces the visualization component in the SI system with a projection neural network. The DL and projection neural networks combine into one integrated end-to-end trainable neural network. The neural design enables SI systems with substantial advantages of neural networks, including out-of-sample capability, stability, and inference speed.

The subsections below revisit these research designs and discuss their contributions and broader implications.

8.1.1 Coupling Cognition and Computation

In Chapters 3 and 4, I developed and validated SI systems with pretrained DL (DeepVA) instead of traditional ML. Results show that even fixed pretrained DL representations can better capture users' concepts. This work offers new insight into effectively bridging the gap

between human cognition and computational models. From the ML perspective, using the pretrained DL model as the feature extractor is a straightforward data preprocessing method for everyday tasks. However, from the perspective of human-AI collaboration, it is an initial but essential step in building connections between cognition and computation spaces via abstract DL representation. It is a convenient and efficient communication between cognition and computation. In this thesis, I explored the use of DR to visualize DL representations for sensemaking tasks.

DeepVA supports the usage of CNN and Word2Vec for visual concept analysis and text analysis. We envision the DeepVA framework being adaptable to broader applications with different analytic tasks. A similarly straightforward strategy can be applied to the original VA pipeline to support other analytic tasks and data types. Overall, the hope is that the DeepVA pipeline will accelerate research on coupling the concepts between cognition and computations using deep learning representations for human-AI collaborative analytic tasks.

8.1.2 Interactive DL for Visual Analytics

In Chapter 5, I designed the DeepSI framework that integrates deep learning into the human-in-the-loop interactive sensemaking pipeline. The interactively tuned DL model can create user- and task-specific representations for complex tasks.

From the HCI perspective, DeepSI uses interactive DL as the underlying analytic model of VA systems. The critical research is designing the visual interfaces and the relevant interactions compatible with the underlying deep neural networks. For ML, the interactive DL is the backbone model to tune high-dimensional representations for downstream tasks. The critical research is designing the downstream task-specific neural output layers and loss functions.

These two perspectives address separate but related facets of the same problem. Both the interactive DL and the visualization components need specific designs for compatible and effective combination. In DeepSI, one mean pooling layer is appended to the DL model to generate fixed-length encoding vectors from raw data. To avoid information loss, the nonparametric DR model, MDS, is used as the default visualization method rather than the commonly used parametric or probabilistic DR.

In DeepSI, I used pretrained BERT as the specific DL model to advance SI-enabled applications. The DeepSI framework is general enough to apply other pretrained DL models into the semantic interaction pipeline for other VA tasks. More generally, the interactive DL design could also be applied to the standard VA framework [142] with specific designs.

8.1.3 Explainable AI Supports Sensemaking

In Chapter 6, I designed the DeepSE framework. DeepSE generates counterfactual explanations of DL and contextualizes explanations into the visual interface. DeepSE can provide intuitive explanations of the interactive human-steered DL model during human-in-the-loop sensemaking. Our prototype DeepSE currently interprets nonlinear DR models powered with deep learning for text data. As a model-agnostic explanation method, SE can be generalized to different DR models and applied to other data types. For example, SE can be integrated into SI systems with weighted MDS, such as Andromeda for numerical data analysis [149] or DeepVA for visual concept analysis [14].

Existing eXplainable AI (XAI) techniques are applied to interpret the model trained on the dataset to build trust or improve model performance. However, this thesis explores the usage of XAI in interactive DL models in support of sensemaking. Specifically, this framework extracts explanations to assist users in figuring out even more complicated concepts or plots.

The purpose is to assist analysts with incremental formalization, specifically, transferring their high-level blur concepts into basic features.

By applying XAI techniques for the incremental formalization process of humans, I hope to highlight the importance of utilizing XAI for visual analytics and expand the scope of application of XAI to boost users' complex sensemaking in visual analytics tasks. We denote this task as XAI4VA and plan to explore further in this direction.

8.1.4 End-to-end Neural Network as VA System

End-to-end deep neural networks have been applied to a diverse set of target tasks [132, 166] by appending task-specific neural layers to the backbone models. For example, a softmax layer is utilized for classification tasks [110] and a CRF layer for NER tasks [83]. Likewise, in Chapter 7, I proposed the NeuralSI framework, an end-to-end trainable neural network for VA systems with SI.

Existing SI systems use the standard framework consisting of two components: the analytic model and the visualization method. NeuralSI replaces both components with the integrated neural network. This design enables VA systems with the substantial advantages of end-to-end neural networks, including simplicity, out-of-sample capability, stability, applicability, and inference speed [57, 67, 74].

Three major design choices were explored in building an effective deep neural network for SI: projection head architecture, projection parameter initialization, and loss function. Experimental findings highlight the importance of designing consistent connections between high-dimensional representation and low-dimensional visualization spaces. On the strength of these findings, the neural design of the SI system can achieve performance levels comparable with those of the state-of-the-art SI model.

NeuralSI demonstrates the possibility of using an end-to-end network for other visual analytic tasks by replacing the output layers with other task-specific neural networks for visualization generation and synthesis[44, 137, 170].

8.2 Future Work

This dissertation presents several frameworks to power VA systems with DL for human-AI sensemaking. While these frameworks can be generalized to other analytic tasks, four directions can be explored to advance their performance and broaden their applicability.

8.2.1 The Usage of More Advanced DL Techniques

Deep learning (DL) [104] is a state-of-the-art representation learning method [10] that can automatically extract abstract and useful hierarchical representations from raw data [10]. For simplicity and generality, this dissertation presents the straightforward usage of pretrained DL as the underlying analytic model. During the interactive learning process, fine-tuning—the commonly used model adaptation method—is used for model steering [132]. Recently, advanced techniques have been proposed to enable DL models with more transferability and applicability. These developments offer the opportunity to use more advanced DL techniques to boost the model performance in various ways.

Self-Supervised Learning

First, more advanced pretrained deep neural networks with self-supervised learning can be used as the analytic model to boost system performance. Self-Supervised Learning (SSL) allows DL models to learn general domain knowledge from the vast amount of unlabeled data

instead of relying on expensively collected manual labels. BERT [42], used in Chapter 5, 6, and 7, is the most used self-supervised model for language understanding. BERT is pretrained using two semi-supervised tasks (Masked LM and Next Sentence Prediction) from two large corpora: BooksCorpus (800M words) [200] and English Wikipedia (2,500M words) [27].

Recently, more complex deep neural networks with a large quantity of unlabeled data, such as GPT-3 [21] and SimCLR [30], have been trained. This enables DL models with the capability for few-shot learning. Using these more advanced DL models can effectively capture users' intents based on an even smaller number of interactions, thereby, dramatically boosting the efficiency and performance human-AI sensemaking.

Few-Shot Learning

From the ML perspective, SI systems with interactive DL are designed to solve few-shot learning problems [31]. SI systems should capture the precise intents behind users' semantic interactions even when the interactions are few. In SI systems, the analyst repositions a subset of data points shown in the projection space (2D) to demonstrate how they should be arranged to express the analyst's mental model. The underlying machine learning model should find a new set of parameters to capture the intent, then update the data projection as the prediction for the analyst to perceive and gain insights. In this way, few-shot learning can also boost the performance of human-AI sensemaking.

As a few-shot learning problem, the sensemaking process with semantic interactions can be described as follows. The subset of data points moved by the analyst belongs to the support set ($\mathcal{D}_s = \{(x_i, y_i)\}_{i=1}^{N_s}$). All the observations on the visual interface belong to the query set ($\mathcal{D}_q = \{(x_i, y_i)\}_{i=1}^{N_q}$). Each interaction loop can be treated as a few-shot episode. The

target $y_i \in R^2$ is a two-dimensional vector, the 2D coordinates in the projection, instead of the set of classes used in the few-shot learning literature. The goal is to teach a function $\mathcal{F}$ to exploit the support set $\mathcal{D}_s$ to predict the projection of all the observations x, where $(x, y) \in \mathcal{D}_q$, by the following equation:

$$\hat{y} = \mathcal{F}(x; \mathcal{D}_s). \tag{8.1}$$

Based on the above definition of "few-shot semantic interaction," we can apply existing few-shot learning techniques [31, 43, 155] to SI systems with interactive deep learning.

Semi-Supervised Learning

Semantic interaction for sensemaking is also described as a semi-supervised DR task [149, 197]. However, existing SI systems, including the frameworks proposed in this thesis, focus on using only a few labeled data points for model training and concept learning. A large number of unlabeled observations will not be used.

Semi-supervised learning uses unlabeled observations together with a limited set of labeled data for efficient model training. For example, a straightforward technique in the semi-supervised learning literature is penalizing the Shannon Entropy of the predictions on the unlabeled data [64]. More advanced semi-supervised techniques include LDS [121], GAN-based methods [41], and graph-based approaches [96]. These self-supervised methods can also be used with the frameworks proposed in previous chapters to further augment their performance.

Data Augmentation

Data augmentation [187] aims to introduce more data for model training by transforming the original dataset. It is a simple but effective way to improve the performance of DL. In recent years, significant advances have been made in designing data augmentations for different tasks, including NLP [193], computer vision [40, 97], and speech [71, 125]. Data augmentation has also been applied in the unlabeled data for semi-supervised tasks in a process known as unsupervised data augmentation (UDA) [187].

Both augmentation methods can be applied to our proposed SI frameworks for interactive DL. First, we employ supervised data augmentation to generate more semantic interactions (instructions) based on users' interactions. Second, we employ unsupervised data augmentation to synthesize more unlabeled data used in the semi-supervised method. This method can advance the existing SI system without any configuration. Furthermore, it can easily be combined with other deep learning techniques discussed in this section.

8.2.2 Advanced Explanation Designs for Sensemaking

In Chapter 6, I proposed the semantic explanation of SI systems with interactive DL. Results shown that it is an intuitive and effective explanation method to support sensemaking. However, the straightforward method has two limitations in supporting complex sensemaking.

First, DeepSE provides a two-layer mapping between abstract concepts and input features learned across all the processes. Concepts generated during the sensemaking process are usually too complex to be explained by a group of basic features. A stack of neural layers, DL models learn representations of data with multiple levels of abstractions [10]. Semantic explanation does not fully use the hierarchical structures of deep neural networks. Interpreting a richer abstraction hierarchy (multi-layer explanations) from interactive DL offers an

opportunity to assist analysts in formalizing complex concepts. The structural relationships between high-level concepts and low-level input features provide formal representations and naturally support the incremental formalization process [152].

Second, DeepSE indicates how well deep learning models capture the concepts behind users interactions through sensemaking processing. It explains the overall effects introduced by user interactions accumulated through all the processes. However, users might change their overall hypothesis during the complex sensemaking process or introduce new biases in different iterations. The overall explanation can not display the effect changes introduced by interactions on the current iteration. A separate explanation of the interaction effects in each iteration is necessary to support the incremental updates of their mental models during the sensemaking process.

Two advanced explanations could be explored and designed to address these limitations.

Explanation in Hierarchical Levels

The goal is to enhance the projection explanation and visualization proposed in Chapter 6 to better support incremental formalization and interpret the projection with formal semantic structure. Two steps are needed to support the hierarchical explanations of SI systems with interactive DL: hierarchical explanation extraction and hierarhical explanation visualization.

Hierarchical Explanation Extraction: both local and global explanations can be utilized to generate hierarchical explanations [163]. In local hierarchical explanation [28], the system should reveal prediction behaviors of the deep learning model by detecting feature interactions with respect to model predictions. For global hierarchical explanation [109], the system is required to have the summarization ability to generate higher-level hierarchical explanations from the model's local behavior. Combining these two methods can generate

hierarchical explanations at different levels of granularity and abstraction. In SI with interactive DL, three key levels of hierarchical explanations are needed: instance, cluster, and global.

Formal Representation Visualization: The formal representations in three levels generated through hierarchical explanation extraction can boost incremental formalization. Proper visualization design is required to contextualize and visualize the formal representations into the original workspace. One reasonable solution is to design the visual analysis systems with multiple visualization views to support these three levels: instance view, cluster view, and the global workspace view.

The global workspace view is the entry point for analysts and displays the projected data points as a scatterplot. In the workspace view, analysts can rearrange the projection to refine concepts via semantic interactions or by selecting one data point or cluster of data points to analyze. In addition, the analyst can circle a group of data points to identify them as a cluster for interpretation. Cluster view displays the explanations for the clusters defined and selected in the workspace. Selecting a data point in the workspace also triggers the instance view. The instance view displays the raw data and instance-level hierarchical explanations. SI systems provide a natural visual interface for analysts to explore explanations at different levels.

Explanation in Incremental Fashion

Explanations in SI systems should assist users' concept formalization in an incremental fashion. Users continue to interact and refine their concepts in repeated loops of the sensemaking pipeline. At each iteration of the loop, the model updates based on users' interactions. It might be useful to specifically provide an incremental explanation that emphasizes the dif-

ference from the previous iteration of the model. Explanations in the incremental fashion can quantify the effect of interactions performed in different loops and thereby assist users in forming concepts and hypotheses.

One straightforward solution for gaining incremental explanations is utilizing the difference between explanations calculated in different iterations. For example, feature weights show the importance of each feature in model predictions. In the incremental explanation, we use the difference of the feature weights between the current model and the model from the last iteration. Another possible solution is using neurons activated only in the current iteration. These representative neurons indicate the concept update triggered by user interactions performed in the current loop.

8.2.3 System Design from the HCI Perspective

Existing SI systems follow the standard VA framework [142], in which the analyst and DL models are in their own learning loops [174]. The sensemaking loop [133] of analysts is incremental, while the training loop [93] of deep learning is batch-oriented. It is challenging to adapt the training loop of deep learning to better capture the incremental intent behind users' interactions and subsequently support incremental formalization [152] for the human sensemaking loop.

This thesis focuses mainly on SI system design and implementation from the interactive DL perspective. Specifically, I concentrated on improving the performance of human-DL sensemaking by adapting ML techniques into the SI system. These techniques include the usage of DL as the analytic model (Chapter 3 and 4), interactive DL steering for better inference capability (chapter 5), semantic explanation of interactive DL for concept understanding (Chapter 6), and neural design of the SI system for inference speed (Chapter 7).

Further improvements that might be possible by integrating more advanced DL techniques are discussed in section 8.2.1). Augmenting system performance from the HCI perspective is another indispensable new direction. Four possible improvements could be explored in this context: semantic interaction, user study, crowdsourcing, and sensemaking loop.

Semantic Interactions

Semantic interaction includes a variety of interactions commonly used during the sensemaking process [48]. For example, in visual text analytics, the commonly used semantic interactions include document movement, text highlighting, annotation, document coloring, level of visual detail, and query terms [53]. Furthermore, semantic interactions have been combined with other visual interactions to utilize multiple analytic models. For example, Wenskovitch et al. have used cluster creation and projection interaction to merge the interactive DR and clustering algorithm into one SI pipeline [175].

These interactions give users more expressiveness when using SI systems for sensemaking tasks. For simplicity and clarity of demonstration, this thesis uses only the most common semantic interaction, projection interaction, which is also known as observation-level interaction [51, 149]. More types of semantic interactions should be applied to the SI systems in the context of real-world tasks.

User Study

I performed several algorithm-level quantitative evaluations to objectively measure the effectiveness of the proposed interactive DL tuning and explanation methods. However, extensive user studies are needed to evaluate the proposed SI systems in the wild: to discover how analysts perform real-world sensemaking tasks using the proposed SI systems. Specifically,

I wish to investigate how these designs and systems boost human performance in complex sensemaking tasks [149, 177]. User study provides a complementary evaluation of SI systems for human-AI sensemaking.

User studies [6, 176] have been performed that examine grouping and organizational sensemaking tasks. These studies provide numerous design recommendations to improve the usability of SI systems. It is likely that, in the context of human-in-the-loop analysis, the interaction sequence also influences SI systems with interactive ML. Therefore, user studies to examine how users perform interactions sequentially could provide meaningful suggestions for SI system design.

Sensemaking Loop

The purpose of this thesis is to design SI systems with interactive DL for complex sensemaking tasks [133]. The overall sensemaking process is divided into two major loops of activities: a foraging loop and a sensemaking loop. The foraging loop involves processes of information retrieval from a large dataset, while the sensemaking loop involves actions of mental model building from the retrieved information. During the design and implementation, I focused on designing systems only for the synthesis loop, where analysts work on the data points shown in the workspace. Therefore, more work needs to be undertaken in the future to support the whole sensemaking process (both the foraging loop and the sensemaking loop).

To support both loops, Bradel et al. have proposed a multi-model semantic interaction system, StarSPIRE [19]. StarSPIRE uses multiple models at multiple levels of data scale: the foraging loop at the large scale and the sensemaking loop at the small scale. A similar strategy is required to support the foraging loop in the DeepSI framework (Chapter 5), because the traditional DR component of DeepSI is time-consuming. The NeuralSI framework

can naturally support the foraging loop because the neural projection of NeuralSI can handle large-scale data retrieval. Tasks remaining for future research include implementing the system for both loops and validating its performance.

8.2.4 Application to General Problems

While claiming that the frameworks proposed in this thesis (DeepVA, DeepSI, DeepSE, and NeuralSI) can be generalized to other problems, I have demonstrated their application to only two sensemaking tasks: visual text analytics and visual concept learning. I plan to expand these systems to broader areas in the future. Here I explore their application in the following three problems.

Numerical Analysis

The majority of previous SI systems were designed for numerical data [45, 51, 149, 175]. The numerical analysis tasks used in these works are relatively simpered enough to be solved by linear DR models. A variety of numerical tasks of different difficulty levels are needed to achieve a complementary evaluation of SI systems with interactive DL. In addition, for numerical tasks of different difficulty levels, the usage of DL models of different complexities should be considered.

Education

SI-enabled VA systems provide analysts with a user-friendly interface where users can intuitively interact with the data to express their intents. Therefore, SI systems can help non-expert users understand the underlying data and analytic models. Chen et al. [32] have proposed an immersive SI approach called “be the data” to educate students about data

analysis techniques, such as dimension reduction. Likewise, SI systems with interactive DL can be used to teach students abstract deep learning techniques. For example, by exploring the 2D projection of high-dimensional DL representations, students can better understand how DL models generate abstract concepts from raw data.

Deep Learning

I explored the usage of deep learning as the analytic model in SI systems to encourage analyst sensemaking. Inversely, the SI system can also facilitate the development of DL by researchers and practitioners. For a more detailed illustration of VA for DL, we refer readers to the exhaustive surveys [35, 78]. Here I discuss the usage of the proposed frameworks in deep learning to achieve four different purposes: interpretability and explainability, debugging and improving models, comparing and selecting models, teaching deep learning concepts.

DeepSE, proposed in Chapter 6, provides a semantic explanation of underlying interactive DL to understand the informal concepts expressed by analysts. DeepSE can also promote understanding how the trained deep learning models make decisions and what representations these models have learned. With a clear understanding, DL practitioners and users can place trust in DL models [35, 107].

In addition, the DeepSI and NeuralSI frameworks proposed in Chapters 5 and 7 are designed to use interactive DL training to capture intents based on users' interactions during the sensemaking process. To support debugging and improving models, semantic interactions let developers and builders effectively explore a handful of data instances and identify whether they are misclassified in the workspace.

In Chapter 5, I conducted a case study to compare the performance of DeepSI and DeepVA

models. Specifically, the same semantic interactions are applied to both DeepSI and DeepVA systems. Based on the updated visual layout, the analyst can determine the better system for the current task. Likewise, using semantic interaction with interactive DL, model users can compare different models and choose the best-performing one for future usage.

Finally, when highlighting the DL part of the human-AI co-learning process, the proposed frameworks can naturally support non-experts by teaching deep learning concepts through intuitive interactions, thus building an intuition for how neural networks behave [153].

Bibliography

[1] Agnar Aamodt and Enric Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1):39–59, 1994. ISSN 0921-7126. doi: 10.3233/aic-1994-7104.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[3] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018. ISSN 2405-8440. doi: https://doi.org/10.1016/j.heliyon.2018.e00938. URL http://www.sciencedirect.com/science/article/pii/S2405844018332067.

[4] Saleema Amershi, James Fogarty, and Daniel Weld. Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceeding CHI '12 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, May 2012. URL https://www.microsoft.com/en-us/research/publication/regroup-interactive-machine-learning-demand-group-creation-social-networks/.

[5] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35 (4):105–120, 2014.

[6] Christopher Andrews, Alex Endert, and Chris North. Space to think: large high-

resolution displays for sensemaking. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 55–64, 2010.

[7] Ben Athiwaratkun and Keegan Kang. Feature representation in convolutional neural networks. *arXiv preprint arXiv:1507.02313*, 2015.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[9] Aurélien Bellet, Amaury Habrard, and Marc Sebban. Metric learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 9(1):1–151, 2015. doi: 10.2200/S00626ED1V01Y201501AIM030. URL https://doi.org/10.2200/S00626ED1V01Y201501AIM030.

[10] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. ISSN 0162-8828. doi: 10.1109/tpami.2013.50.

[11] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, volume 1, 2010.

[12] Yali Bian and Chris North. Deepsi: Interactive deep learning for semantic interaction. In *Proceedings of the 26th International Conference on Intelligent User Interfaces*, IUI '21, page 177–188, College Station, TX, USA, 2021. Association for Computing Machinery. ISBN 978-1-4503-8017-1/21/04. doi: 10.1145/3397481.3450670. URL https://doi.org/10.1145/3397481.3450670.

[13] Yali Bian, Michelle Dowling, and Chris North. Evaluating semantic interaction on

word embeddings via simulation. *EValuation of Interactive VisuAl Machine Learning systems, an IEEE VIS 2019 Workshop.*, 2019.

[14] Yali Bian, John Wenskovitch, and Chris North. Deepva: Bridging cognition and computation through semantic interaction and deep learning. In *Proceedings of the IEEE VIS Workshop MLUI 2019: Machine Learning from User Interactions for Visualization and Analytics. VIS'19.*, 10/2019 2019.

[15] Yali Bian, Chris North, Eric Krokos, and Sarah Joseph. Semantic explanation of interactive dimensionality reduction. In *2021 IEEE Visualization Conference (VIS)*, pages 26–30, 2021. doi: 10.1109/VIS49827.2021.9623322.

[16] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[17] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larranaga. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233, 2015.

[18] Nadia Boukhelifa, Anastasia Bezerianos, and Evelyne Lutton. Evaluation of interactive machine learning systems. In *Human and Machine Learning*, pages 341–360. Springer, 2018.

[19] L. Bradel, C. North, L. House, and S. Leman. Multi-model semantic interaction for text analytics. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 163–172, Oct 2014. doi: 10.1109/VAST.2014.7042492.

[20] Eli T Brown, Jingjing Liu, Carla E Brodley, and Remco Chang. Dis-function: Learning distance functions interactively. *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 83–92, 2012. doi: 10.1109/VAST.2012.6400486.

[21] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[22] EJ Bynen. *Cluster analysis: Survey and evaluation of techniques*, volume 1. Springer Science & Business Media, 2012.

[23] Marco Cavallo and Çagatay Demiralp. A visual interaction framework for dimensionality reduction based data exploration. In Regan L. Mandryk, Mark Hancock, Mark Perry, and Anna L. Cox, editors, *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, Montreal, QC, Canada, April 21-26, 2018*. ACM, 2018. doi: 10.1145/3170427.3186508. URL https://doi.org/10.1145/3170427.3186508.

[24] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.

[25] Y. Chan, C. D. Correa, and K. Ma. Flow-based scatterplots for sensitivity analysis. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pages 43–50, Oct 2010. doi: 10.1109/VAST.2010.5652460.

[26] Angelos Chatzimparmpas, Rafael M. Martins, and Andreas Kerren. t-visne: Interactive assessment and interpretation of t-sne projections. *IEEE Transactions on Visualization and Computer Graphics*, 26(8):2696–2714, 2020. doi: 10.1109/TVCG.2020.2986996.

[27] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling, 2014.

[28] Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. Generating hierarchical explanations on text classification via feature interaction detection, 2020.

[29] Pengfei Chen, Guangyong Chen, and Shengyu Zhang. Log hyperbolic cosine loss improves variational auto-encoder. 2018.

[30] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[31] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[32] Xin Chen, Jessica Zeitz Self, Leanna House, and Chris North. Be the data: A new approach for lmmersive analytics. In *2016 Workshop on Immersive Analytics (IA)*, pages 32–37, 2016. doi: 10.1109/IMMERSIVE.2016.7932380.

[33] F. Cheng, Y. Ming, and H. Qu. Dece: Decision explorer with counterfactual explanations for machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1438–1447, 2021. doi: 10.1109/TVCG.2020.3030342.

[34] H. Cheng, A. Cardone, S. Jain, E. Krokos, K. Narayan, S. Subramaniam, and A. Varshney. Deep-learning-assisted volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 25(2):1378–1391, Feb 2019. ISSN 1077-2626. doi: 10.1109/TVCG.2018.2796085.

[35] J. Choo and S. Liu. Visual analytics for explainable deep learning. *IEEE Computer Graphics and Applications*, 38(04):84–92, jul 2018. ISSN 0272-1716. doi: 10.1109/MCG.2018.042731661.

[36] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[37] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[38] Kristin A Cook and James J Thomas. Illuminating the path: The research and development agenda for visual analytics. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 01 2005.

[39] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1):21–27, September 2006. ISSN 0018-9448. doi: 10.1109/TIT.1967.1053964. URL https://doi.org/10.1109/TIT.1967.1053964.

[40] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data, 2019.

[41] Zihang Dai, Zhilin Yang, Fan Yang, William W. Cohen, and Ruslan Salakhutdinov. Good semi-supervised learning that requires a bad gan, 2017.

[42] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[43] Guneet S. Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification, 2019.

[44] Victor Dibia and Çağatay Demiralp. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE Comput. Graph. Appl.*, 39(5):33–46, sep 2019. ISSN 0272-1716. doi: 10.1109/MCG.2019.2924636. URL https://doi.org/10.1109/MCG.2019.2924636.

[45] M. Dowling, J. Wenskovitch, J. T. Fry, S. Leman, L. House, and C. North. Sirius: Dual, symmetric, interactive dimension reductions. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):172–182, 2019.

[46] Michelle Dowling, John Wenskovitch, Peter Hauck, Adam Binford, Nicholas Polys, and Chris North. A bidirectional pipeline for semantic interaction. In *Proc. Workshop on Machine Learning from User Interaction for Visualization and Analytics (at IEEE VIS 2018)*, volume 11, 2018.

[47] Michelle Dowling, Nathan Wycoff, Brian Mayer, John Wenskovitch, Scotland Leman, Leanna House, Nicholas Polys, Chris North, and Peter Hauck. Interactive Visual Analytics for Sensemaking with Big Text. *Big Data Research*, 16:49–58, 2019. ISSN 2214-5796. doi: 10.1016/j.bdr.2019.04.003.

[48] A. Endert, P. Fiaux, and C. North. Semantic interaction for sensemaking: Inferring analytical reasoning for model steering. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2879–2888, Dec 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.260.

[49] A. Endert, R. Chang, C. North, and M. Zhou. Semantic interaction: Coupling cognition and computation through usable interactive analytics. *IEEE Computer Graphics and Applications*, 35(4):94–99, July 2015. ISSN 0272-1716. doi: 10.1109/MCG.2015.91.

[50] Alex Endert, Chao Han, Dipayan Maiti, Leanna House, Scotland Leman, and Chris North. Observation-level interaction with statistical models for visual analytics. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 121–130. IEEE.

[51] Alex Endert, Chao Han, Dipayan Maiti, Leanna House, Scotland Leman, and Chris North. Observation-level interaction with statistical models for visual analytics. *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 121–130, 2011. doi: 10.1109/vast.2011.6102449.

[52] 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*, CHI '12, pages 473–482, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207741. URL http://doi.acm.org/10.1145/2207676.2207741.

[53] 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.

[54] Alex Endert, Remco Chang, Chris North, and Michelle Zhou. Semantic Interaction: Coupling Cognition and Computation through Usable Interactive Analytics. *IEEE Computer Graphics and Applications*, 35(4):94–99, jul 2015. ISSN 02721716. doi: 10.1109/MCG.2015.91. URL https://ieeexplore.ieee.org/document/7160906/.

[55] Martin J Eppler and Jeanne Mengis. The concept of information overload-a review of literature from organization science, accounting, marketing, mis, and related disciplines (2004). *Kommunikationsmanagement im Wandel*, pages 271–305, 2008.

[56] Mateus Espadoto. *Learning Multidimensional Projections with Neural Networks*. PhD thesis, University of Groningen, 2021.

[57] Mateus Espadoto, Nina Sumiko Tomita Hirata, and Alexandru C Telea. Deep learning multidimensional projections. *Information Visualization*, page 1473871620909485, 2020.

[58] Jerry Alan Fails and Dan R Olsen Jr. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45, 2003.

[59] Rebecca Faust, David Glickenstein, and Carlos Scheidegger. DimReader: Axis lines that explain non-linear projections. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):481–490, 2018. ISSN 1077-2626. doi: 10.1109/tvcg.2018.2865194.

[60] Sebastian Gehrmann, Hendrik Strobelt, Robert Krüger, Hanspeter Pfister, and Alexander M. Rush. Visual Interaction with Deep Learning Models through Collaborative Semantic Inference. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):884–894, 2020. ISSN 1077-2626. doi: 10.1109/tvcg.2019.2934595.

[61] Aindrila Ghosh, Mona Nashaat, James Miller, and Shaikh Quader. Interpretation of structural preservation in low-dimensional embeddings. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020. doi: 10.1109/TKDE.2020.3005878.

[62] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[63] Oscar Gomez, Steffen Holter, Jun Yuan, and Enrico Bertini. Vice: Visual counterfactual explanations for machine learning models. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, page 531–535, New York,

NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371186. doi: 10.1145/3377325.3377536. URL https://doi.org/10.1145/3377325.3377536.

[64] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. In *CAP*, pages 281–296, 2005.

[65] Tera Marie Green, William Ribarsky, and Brian Fisher. Building and applying a human cognition model for visual analytics. *Information visualization*, 8(1):1–13, 2009.

[66] Aakriti Gupta, Kapil Thadani, and Neil O'Hare. Effective Few-Shot Classification with Transfer Learning. pages 1061–1066, 2020. doi: 10.18653/v1/2020.coling-main.92.

[67] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006. doi: 10.1109/CVPR.2006.100.

[68] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, July 1995. ISSN 0162-8828. doi: 10.1109/34.391417.

[69] Chao Han, Leanna House, and Scotland C. Leman. Expert-guided generative topographical modeling with visual to parametric interaction. *PLOS ONE*, 11(2):1–14, 02 2016. doi: 10.1371/journal.pone.0129122. URL https://doi.org/10.1371/journal.pone.0129122.

[70] Jun Han and Claudio Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *Proceedings of the International Workshop on*

Artificial Neural Networks: From Natural to Artificial Neural Computation, IWANN '96, page 195–201, Berlin, Heidelberg, 1995. Springer-Verlag. ISBN 3540594973.

[71] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep speech: Scaling up end-to-end speech recognition, 2014.

[72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[73] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[74] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. ISSN 0036-8075. doi: 10.1126/science.1127647. URL https://science.sciencemag.org/content/313/5786/504.

[75] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[76] Nathan Oken Hodas and Alex Endert. Adding semantic information into data models by learning domain expertise from user interaction. *CoRR*, abs/1604.02935, 2016. URL http://arxiv.org/abs/1604.02935.

[77] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In Aasa Feragen, Marcello Pelillo, and Marco Loog, editors, *Similarity-Based Pattern Recognition*, pages 84–92, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24261-3.

[78] Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[79] Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289, 2021.

[80] Leanna House, Scotland Leman, and Chao Han. Bayesian visual analytics: Bava. *Stat. Anal. Data Min.*, 8(1):1–13, February 2015. ISSN 1932-1864. doi: 10.1002/sam.11253. URL http://dx.doi.org/10.1002/sam.11253.

[81] X. Hu, L. Bradel, D. Maiti, L. House, C. North, and S. Leman. Semantics of directly manipulating spatializations. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2052–2059, 2013.

[82] Jian Huang, Yuling Jiao, Xu Liao, Jin Liu, and Zhou Yu. Deep Dimension Reduction for Supervised Representation Learning. *arXiv*, 2020.

[83] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[84] F. Hughes and D. Schum. Discovery-proof-choice, the art and science of the process of intelligence analysis-preparing for the future of intelligence analysis. *Washington, DC: Joint Military Intelligence College*, 2003.

[85] Petra Isenberg, Florian Heimerl, Steffen Koch, Tobias Isenberg, Panpan Xu, Chad Stolper, Michael Sedlmair, Jian Chen, Torsten Möller, and John Stasko. vispubdata.org: A metadata collection about IEEE visualization (VIS) publications. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2199–2206, Septem-

ber 2017. doi: 10.1109/TVCG.2016.2615308. URL https://tobias.isenberg.cc/VideosAndDemos/Isenberg2017VMC.

[86] Itseez. Open source computer vision library. https://github.com/itseez/opencv, 2015.

[87] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[88] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[89] Matthew Johnson and Alonso Vera. No ai is an island: The case for teaming intelligence. *AI Magazine*, 40(1):16–28, Mar. 2019. doi: 10.1609/aimag.v40i1.2842. URL https://ojs.aaai.org/index.php/aimagazine/article/view/2842.

[90] E. Kandogan. Just-in-time annotation of clusters, outliers, and trends in point-based data visualizations. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 73–82, Oct 2012. doi: 10.1109/VAST.2012.6400487.

[91] Youn-ah Kang and John Stasko. Characterizing the intelligence analysis process: Informing visual analytics design through a longitudinal field study. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 21–30, 2011. doi: 10.1109/VAST.2011.6102438.

[92] Mahmut Kaya and Hasan Şakir Bilge. Deep Metric Learning: A Survey. *Symmetry*, 11(9):1066, 2019. doi: 10.3390/sym11091066.

[93] Henry J Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10):947–954, 1960.

[94] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279*, 2017.

[95] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[96] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[97] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=2999134.2999257.

[98] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 957–966. JMLR.org, 2015. URL http://dl.acm.org/citation.cfm?id=3045118.3045221.

[99] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):221–230, Jan 2017. ISSN 1077-2626. doi: 10.1109/TVCG.2016.2598446.

[100] Bum Chul Kwon, Min-Je Choi, Joanne Taery Kim, Edward Choi, Young Bin Kim, Soonwook Kwon, Jimeng Sun, and Jaegul Choo. RetainVis: Visual Analytics with Interpretable and Interactive Recurrent Neural Networks on Electronic Medical Records. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):299–309, 2018. ISSN 1077-2626. doi: 10.1109/tvcg.2018.2865027.

[101] Jack Lanchantin, Ritambhara Singh, Beilun Wang, and Yanjun Qi. Deep motif dashboard: Visualizing and understanding genomic sequences using deep neural networks. In *Pacific Symposium on Biocomputing 2017*, pages 254–265. World Scientific, 2017.

[102] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[103] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org, 2014. URL http://dl.acm.org/citation.cfm?id=3044805.3045025.

[104] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436, 2015.

[105] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. Ivisclustering: An interactive visual document clustering via topic modeling. *Comput. Graph. Forum*, 31(3pt3):1155–1164, June 2012. ISSN 0167-7055. doi: 10.1111/j.1467-8659.2012.03108.x. URL https://doi.org/10.1111/j.1467-8659.2012.03108.x.

[106] Scotland C Leman, Leanna House, Dipayan Maiti, Alex Endert, and Chris North. Visual to Parametric Interaction (V2PI). *PLOS ONE*, 8(3):e50474, March 2013.

[107] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[108] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):91–100, Jan 2017. ISSN 1077-2626. doi: 10.1109/TVCG.2016.2598831.

[109] Ninghao Liu, Xiao Huang, Jundong Li, and Xia Hu. On interpretation of network embedding via taxonomy induction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1812–1820, 2018.

[110] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7, 2016.

[111] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[112] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004. ISSN 1573-1405. doi: 10.1023/B:VISI.0000029664.99615.94. URL https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[113] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[114] Catherine C Marshall, Frank M Shipman III, and James H Coombs. Viki: Spatial hypertext supporting emergent structure. In *Proceedings of the 1994 ACM European conference on Hypermedia technology*, pages 13–23, 1994.

[115] David Martens and Foster Provost. Explaining data-driven document classifications. *MIS Q.*, 38(1):73–100, March 2014. ISSN 0276-7783. doi: 10.25300/MISQ/2014/38.1.04. URL https://doi.org/10.25300/MISQ/2014/38.1.04.

[116] Alberto Gonzalez Martinez. *Exploratory Analysis of Research Publications Collections with Human Steerable Black-Box Models. Towards Generalizing Inverse Computations for Semantic Interaction.* PhD thesis, University of Hawai'i at Manoa, 2021.

[117] Alberto González Martínez, Billy Troy Wooton, Nurit Kirshenbaum, Dylan Kobayashi, and Jason Leigh. Exploring Collections of research publications with Human Steerable AI. pages 339–348, 2020. doi: 10.1145/3311790.3396646.

[118] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[119] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[120] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[121] Takeru Miyato, Shin ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training, 2016.

[122] Matthew Mullin and Rahul Sukthankar. Complete cross-validation for nearest neighbor classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 639–646, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

[123] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.

[124] Luis Gustavo Nonato and Michal Aupetit. Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2650–2673, 2017. ISSN 1077-2626. doi: 10.1109/tvcg.2018.2846735.

[125] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*, Sep 2019. doi: 10.21437/interspeech.2019-2680. URL http://dx.doi.org/10.21437/Interspeech.2019-2680.

[126] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[127] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[128] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[129] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

[130] Claudia Perlich, Foster Provost, and Jeffrey S. Simonoff. Tree induction vs. logistic regression: A learning-curve analysis. *J. Mach. Learn. Res.*, 4(null):211–255, December 2003. ISSN 1532-4435. doi: 10.1162/153244304322972694. URL https://doi.org/10.1162/153244304322972694.

[131] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[132] Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. To tune or not to tune? adapting pretrained representations to diverse tasks. *CoRR*, abs/1903.05987, 2019. URL http://arxiv.org/abs/1903.05987.

[133] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. pages 2–4, 2005. URL https://analysis.mitre.org/proceedings/Final_Papers_Files/206_Camera_Ready_Paper.pdf.

[134] Meg Pirrung, Nathan Hilliard, Artëm Yankov, Nancy O'Brien, Paul Weidert, Courtney D. Corley, and Nathan O. Hodas. Sharkzor: Interactive deep learning for image

triage, sort and summary. *CoRR*, abs/1802.05316, 2018. URL http://arxiv.org/abs/1802.05316.

[135] Hang Qi, Matthew Brown, and David G. Lowe. Low-shot learning with imprinted weights, 2018.

[136] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014.

[137] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR, 2016.

[138] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL http://arxiv.org/abs/1908.10084.

[139] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL http://doi.acm.org/10.1145/2939672.2939778.

[140] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2016.

[141] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[142] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey Ellis, and Daniel A Keim. Knowledge generation model for visual analytics. *IEEE transactions on visualization and computer graphics*, 20(12):1604–1613, 2014.

[143] 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, 2016.

[144] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[145] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization & Computer Graphics (Proc. InfoVis)*, 2017. doi: 10.1109/tvcg.2016.2599030. URL http://idl.cs.washington.edu/papers/vega-lite.

[146] Susan S Schiffman, M Lance Reynolds, and Forrest W Young. *Introduction to multidimensional scaling: Theory, methods, and applications*. Emerald Group Publishing, 1981.

[147] Jessica Zeitz Self, Xinran Hu, Leanna House, and Chris North. Designing for interactive dimension reduction visual analytics tools to explore high-dimensional data. Technical report, Technical report, Department of Computer Science, Virginia Tech, Blacksburg, Virginia, 2015.

[148] Jessica Zeitz Self, Radha Krishnan Vinayagam, J. T. Fry, and Chris North. Bridging the gap between user intention and model parameters for human-in-the-loop data

analytics. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, HILDA '16, pages 3:1–3:6, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4207-0. doi: 10.1145/2939502.2939505. URL http://doi.acm.org/10.1145/2939502.2939505.

[149] Jessica Zeitz Self, Michelle Dowling, John Wenskovitch, Ian Crandell, Ming Wang, Leanna House, Scotland Leman, and Chris North. Observation-Level and Parametric Interaction for High-Dimensional Data Analysis. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2), 2018. ISSN 2160-6455. doi: 10.1145/3158230.

[150] F. Shaheen, B. Verma, and M. Asafuddoula. Impact of automatic feature extraction in deep learning architecture. In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, Nov 2016. doi: 10.1109/DICTA.2016.7797053.

[151] Frank M. Shipman and Catherine C. Marshall. Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representations in interactive systems. *Computer Supported Cooperative Work (CSCW)*, 8(4):333–352, Dec 1999. ISSN 1573-7551. doi: 10.1023/A:1008716330212. URL https://doi.org/10.1023/A:1008716330212.

[152] Frank M Shipman III and Raymond McCall. Supporting knowledge-base evolution with incremental formalization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 285–291. ACM, 1994.

[153] Patrice Y. Simard, Saleema Amershi, David M. Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. Machine teaching: A new paradigm for building machine learning systems, 2017.

[154] Daniel Smilkov, Shan Carter, D Sculley, Fernanda B Viégas, and Martin Wattenberg. Direct-Manipulation Visualization of Deep Networks. *arxiv.org*, August 2017.

[155] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, 2017. URL http://arxiv.org/abs/1703.05175.

[156] Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. Parsing with compositional vector grammars. In *In Proceedings of the ACL conference*, 2013.

[157] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[158] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1857–1865, 2016.

[159] Hyesook Son, Seokyeon Kim, Hanbyul Yeon, Yejin Kim, Yun Jang, and Seung-Eock Kim. Visual analysis of spatiotemporal data predictions with deep learning models. *Applied Sciences*, 11(13), 2021. ISSN 2076-3417. doi: 10.3390/app11135853. URL https://www.mdpi.com/2076-3417/11/13/5853.

[160] J. Stahnke, M. Dörk, B. Müller, and A. Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):629–638, 2016. doi: 10.1109/TVCG.2015.2467717.

[161] Yves Tillé. *Sampling algorithms*. Springer, 2006.

[162] Julio Torales, Marcelo O'Higgins, João Mauricio Castaldelli-Maia, and Antonio Ventriglio. The outbreak of covid-19 coronavirus and its impact on global mental health. *International Journal of Social Psychiatry*, 66(4):317–320, 2020. doi: 10.1177/0020764020915212. URL https://doi.org/10.1177/0020764020915212. PMID: 32233719.

[163] Michael Tsang, Youbang Sun, Dongxu Ren, and Yan Liu. Can i trust you more? model-agnostic hierarchical explanations, 2018.

[164] Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13, 2009.

[165] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[166] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[167] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *CoRR*, abs/1711.00399, 2017. URL http://arxiv.org/abs/1711.00399.

[168] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2593–2601, 2017.

[169] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, et al. Cord-19: The covid-19 open research dataset. *ArXiv*, 2020.

[170] Qianwen Wang, Zhutian Chen, Yong Wang, and Huamin Qu. A survey on ml4vis: Applying machinelearning advances to data visualization. *IEEE Transactions on Visualization and Computer Graphics*, page 1–1, 2021. ISSN 2160-9306. doi: 10.1109/tvcg.2021.3106142. URL http://dx.doi.org/10.1109/TVCG.2021.3106142.

[171] Wei Wang, Yan Huang, Yizhou Wang, and Liang Wang. Generalized autoencoder: A neural network framework for dimensionality reduction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 496–503, 2014. doi: 10.1109/CVPRW.2014.79.

[172] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Computing Surveys*, 53(3): 1–34, 2020. ISSN 0360-0300. doi: 10.1145/3386252.

[173] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

[174] J. Wenskovitch and C. North. Interactive artificial intelligence: Designing for the "two black boxes" problem. *Computer*, 53(8):29–39, Aug 2020. ISSN 1558-0814. doi: 10.1109/MC.2020.2996416.

[175] John Wenskovitch and Chris North. Observation-level interaction with clustering and dimension reduction algorithms. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, pages 1–6, 2017.

[176] John Wenskovitch and Chris North. An examination of grouping and spatial organization tasks for high-dimensional data exploration, 2020.

[177] John Wenskovitch, Lauren Bradel, Michelle Dowling, Leanna House, and Chris North.

The effect of semantic interaction on foraging in text analysis. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 13–24. IEEE, 2018.

[178] John Wenskovitch, Michelle Dowling, and Chris North. With respect to what? simultaneous interaction with dimension reduction and clustering projections. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, page 177–188, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371186. doi: 10.1145/3377325.3377516. URL https://doi.org/10.1145/3377325.3377516.

[179] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65, Jan 2020. ISSN 1941-0506. doi: 10.1109/TVCG.2019.2934619.

[180] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005.

[181] Daniela M. Witten and Robert Tibshirani. Supervised multidimensional scaling for visualization, classification, and bipartite ranking. *Computational Statistics & Data Analysis*, 55(1):789–801, 2011. ISSN 0167-9473. doi: 10.1016/j.csda.2010.07.001.

[182] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52, 1987. ISSN 0169-7439. doi: https://doi.org/10.1016/0169-7439(87)80084-9. URL http://www.sciencedirect.com/science/article/pii/0169743987800849. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[183] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[184] Pak Chung Wong, Beth Hetzler, Christian Posse, Mark Whiting, Susan Havre, Nick Cramer, Anuj Shah, Mudita Singhal, Alan Turner, and Jim Thomas. In-spire infovis 2004 contest entry. In *Proceedings of the IEEE Symposium on Information Visualization*, INFOVIS '04, pages 216.2–, Washington, DC, USA, Oct 2004. IEEE Computer Society. ISBN 0-7803-8779-3. doi: 10.1109/INFOVIS.2004.37. URL http://dx.doi.org/10.1109/INFOVIS.2004.37.

[185] Kanit Wongsuphasawat, Daniel Smilkov, James Wexler, Jimbo Wilson, Dandelion Mane, Doug Fritz, Dilip Krishnan, Fernanda B Viégas, and Martin Wattenberg. Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):1–12, Jan 2018. ISSN 1077-2626. doi: 10.1109/TVCG.2017.2744878.

[186] Billy Troy Wooton. *Facilitating Visual to Parametric Interaction with Deep Contrastive Learning*. PhD thesis, University of Hawai'i at Manoa, 2020.

[187] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised Data Augmentation for Consistency Training. 2019.

[188] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: A general framework for dimensionality reduc-

tion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007. doi: 10.1109/TPAMI.2007.250598.

[189] Jun Yang, Yu-Gang Jiang, Alexander G. Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval*, MIR '07, pages 197–206, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-778-0. doi: 10.1145/1290082.1290111. URL http://doi.acm.org/10.1145/1290082.1290111.

[190] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020.

[191] J Yosinski, J Clune, Y Bengio Advances in neural, and 2014. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3320–3328, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969197.

[192] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13(3):55–75, Aug 2018. ISSN 1556-603X. doi: 10.1109/MCI.2018.2840738.

[193] Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B14TlG-RW.

[194] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Ma-

lik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[195] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *Computer Vision – ECCV 2014*, pages 818–833. Springer, Cham, Cham, September 2014.

[196] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. 2020. https://d2l.ai.

[197] Daoqiang Zhang, Zhi-Hua Zhou, and Songcan Chen. Semi-Supervised Dimensionality Reduction. pages 629–634, 2007. doi: 10.1137/1.9781611972771.73.

[198] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4): 43–52, 2010.

[199] X. Zhu and A. Goldberg. *Introduction to Semi-Supervised Learning*. Morgan & Claypool, 2009. ISBN 9781598295481. URL https://ieeexplore.ieee.org/document/6813505.

[200] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, 2015.