

**INTERACTIVE SCALABLE INTERFACES
FOR MACHINE LEARNING INTERPRETABILITY**

A Dissertation
Presented to
The Academic Faculty

By

Frederick Hohman

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computational Science and Engineering

Georgia Institute of Technology

December 2020

Copyright © Frederick Hohman 2020

INTERACTIVE SCALABLE INTERFACES FOR MACHINE LEARNING INTERPRETABILITY

Approved by:

Duen Horng Chau, Advisor
School of Computational Science
and Engineering
Georgia Institute of Technology

Alex Endert
School of Interactive Computing
Georgia Institute of Technology

Chao Zhang
School of Computational Science
and Engineering
Georgia Institute of Technology

Nathan Hodas
Computing and Analytics Division
Pacific Northwest National Lab

Scott Davidoff
Human Centered Design
NASA Jet Propulsion Lab

Steven M. Drucker
Visualization and Interactive Data
Analysis
Microsoft Research

Date Approved: October 5, 2020

If you can't explain it simply, you don't understand it well enough.

Misattributed to Albert Einstein, adapted from Ernest Rutherford¹

¹Beautifully ironic, this quote has a disputed origin. While sometimes attributed to Richard Feynman due to his affinity for teaching, an older attribution to Einstein originates from claims made by Louis de Broglie in 1915. According to de Broglie, Einstein said something of this nature to him in Paris while they were both there celebrating the 100th year anniversary of the discovery and publication of the wave nature of light. It is of note, however, that there is no documented evidence surrounding what Einstein precisely said. Supposedly they were in a bar, presumably drinking. Moreover, the earliest known credit formally given to Einstein did not come until the early 1970s. In fact, the quote is likely actually adapted from an earlier, more colorful quote by Ernest Rutherford. But, while Rutherford died in 1937, the earliest known formal credit for his remark does not appear until the mid 1950s. In the end, these long delays reduce the trustworthiness of any of these attributions.

*To my parents and Jessica,
for everything.*

ACKNOWLEDGEMENTS

There is no possible way to list every person who has helped me throughout my time in graduate school. Doing a Ph.D. can be a selfish endeavor, but I am grateful to have benefitted from the support and guidance from incredible mentors, colleagues, friends, and family. So to everyone in my Ph.D. journey, and in life—this is for you.

First, to Polo, my advisor who has placed enormous time, effort, and energy into helping me become the researcher I am today: thank you. Polo has been nothing short of an exemplary advisor and truly a fantastic mentor. Through taking a chance on me after one year of graduate school, late nights editing my writing, supernatural responsiveness, and an incredible eye for detail, his relentless support has not only helped me solve problems but identify important problems to solve. Through Polo Club, Polo has created an amazing research group and community that I will remember and cherish forever.

My co-advisor, Alex, introduced me to visualization research during my first year at Georgia Tech. Thank you for listening to my half-baked ideas and helping me turn them into capable research. Chao Zhang, whom I met late during graduate school, graciously accepted to be on my thesis committee during his first semester as a professor.

Throughout my Ph.D., I have had the wonderful opportunity to be mentored by some of the smartest and kindest people during four formative internships.

Nathan Hodas hosted me for my first internship at Pacific Northwest National Lab and introduced me to deep learning and the problem of interpretability. Little did I know that this short time would sow the seeds of my thesis topic. Thank you for being patient with me as I transitioned from writing proofs to writing code.

Scott Davidoff, Hillary Mushkin, Maggie Hendrie, and Santiago Lombeyda taught me the importance of design as a process. They have put together a truly unique summer experience at NASA Jet Propulsion Lab to solve some of the most exciting scientific problems with data visualization. Thank you all for helping me prototype early and often, and ensuring users are considered at every stage. I have you all to thank (blame) for the dozens of poster-sized sketches hanging on the lab walls at Georgia Tech.

Steve Drucker and Rob DeLine at Microsoft Research gave me the freedom and flexibility to choose problems that were both widely important and foundational to my thesis topic. Thank you both for giving me confidence in my research, a critical piece of advice I needed to hear at this stage in graduate school.

Lastly, at Apple, Kanit Wongsuphasawat was brave to choose me as his first intern yet was extremely generous and eager to share his technical expertise learning alongside me. Kayur Patel reminded me to never hesitate to shoot for the moon and aim big. Thank you

both for widening my view of how to make impact.

Besides mentors, I have met some lifelong friends and collaborators over the past years that have been an incredible support system and backboard for brainstorming.

To my fellow labmates in Polo Club, thank you for your incredible daily support and enthusiasm. Robert Pienta, whom I met on my Ph.D. visitation weekend, helped me join Polo Club a year later and was my person at an arm's reach to answer the hundreds of questions I would eventually ask him. Thank you for sitting down with me and practically teaching me web programming, how to celebrate wins and brush off rejections, and most importantly to always see the bigger picture and fun in the work I do.

When Polo Club was smaller, Minsuk Kahng and Shang-Tse Chen helped me understand how research is done. To the lab now, including Nilaksh Das, Haekyu Park, Scott Freitas, Zijie Wang, Austin Wright, and Rahul Duggal, I am confident you all will push the lab forward and continue to be an amazing support group for each other. To the undergraduate students that worked with me closely, in particular, Dezhi Fang, Ángel Cabrera, and Will Epperson, thank for you giving me an opportunity to mentor students, and I am excited to see what you all do in the future. I also want to thank the Georgia Tech Visualization Lab for their warm and welcoming community.

Georgia Tech's School of Computational Science and Engineering is a special place to conduct interdisciplinary computational research, including the always helpful administrative staff, student body, and my year's CSE cohort that has been an amazing support system. Special shout-out to Jordi Wolfson-Pou and Caleb Robinson for their encouragement and companionship throughout our time at Georgia Tech.

I also want to thank my fantastic friends and fellow researchers that I have met along the way that continue to inspire me, including Matthew Conlen, Emily Wall, Arjun Srinivasan, Ian Stewart, Andrew Head, Mary Beth Kery, and Sara Stalla.

Besides academic mentors, colleagues, and friends, I need to give a final special thank you to my family.

To Jessica, the love of my life, who was crazy enough to start our relationship the summer before I started my Ph.D., our lives have been intertwined with my studies and research, yet from the beginning, your support has never faltered. This Ph.D. feels as much yours as it is mine. Thank you for sticking with me through every stressful night, code bug, paper deadline, and four summers apart. You remind me what is important in life.

Finally, to my parents, Steve and Rita, who always prioritized my education, thank you for giving me the opportunity, freedom, and patience to find and follow my passions. Thank you both for everything you have done to help me get to where I am today.

TABLE OF CONTENTS

Acknowledgments	v
List of Tables	xii
List of Figures	xiii
Summary	xix
Chapter 1: Introduction	1
1.1 Designing Machine Learning Interpretability for People	2
1.2 Thesis Overview	3
1.2.1 Part I: Enabling Machine Learning Interpretability	3
1.2.2 Part II: Scaling Deep Learning Interpretability	5
1.2.3 Part III: Communicating Interpretability with Interactive Articles	7
1.3 Thesis Statement	9
1.4 Research Contributions	9
1.5 Impact	10
1.6 Prior Publications and Authorship	11
Chapter 2: Background and Related Work	13
2.1 Definitions of Interpretability	13
2.1.1 Audience for Interpretability	13
2.1.2 Interpretability and AI Guidelines	14
2.2 Visual Analytics for Machine Learning Interpretability	15
2.2.1 Complementing Visualizations with Verbalizations.	16
2.3 Human Evaluation for Machine Learning Interpretability	17
2.4 Neural Network Interpretability	17
2.4.1 Understanding Neuron Activations	17
2.4.2 Towards Higher-level Deep Learning Interpretation	19
2.4.3 Visual Analytics for Neural Network Interpretability	20
I Enabling Machine Learning Interpretability	21
Chapter 3: GAMUT: Understanding How Data Scientists Understand Machine Learning	23
3.1 Introduction	23

3.2	Design Rationale	25
3.2.1	A Technology Probe for Model Interpretability	25
3.2.2	Assessing the Probe’s Features	26
3.2.3	Selecting the Probe’s Model Class	27
3.3	GAMUT	29
3.3.1	Shape Curve View	30
3.3.2	Instance Explanation View	30
3.3.3	Interactive Table	32
3.3.4	Implementation	32
3.4	User Study	33
3.4.1	Participants	33
3.4.2	Study Design	33
3.5	Results	34
3.5.1	RQ1: Reasons for Model Interpretability	34
3.5.2	RQ2: Global versus Local Explanation Paradigms	36
3.5.3	RQ3: Interactive Explanations	37
3.5.4	Usability	38
3.6	Limitations	38
3.7	Conclusion	39
Chapter 4: TELEGAM: Combining Visualization and Verbalization for Interpretability	41
4.1	Introduction	41
4.2	TELEGAM: Visualization & Verbalization	43
4.2.1	Design Goals	43
4.2.2	Model Class and Background	43
4.2.3	Realizing Design Goals in TELEGAM’s Interface	44
4.2.4	Generating Verbalizations	45
4.2.5	User-specified Verbalization Resolution	48
4.2.6	Illustrative Usage Scenario	49
4.3	Conclusion	49
II Scaling Deep Learning Interpretability	50
Chapter 5: INTERROGATIVE SURVEY for Visual Analytics in Deep Learning	52
5.1	Introduction	52
5.2	Our Contributions & Method of Survey	55
5.2.1	Our Contributions	55
5.2.2	Survey Methodology & Summarization Process	55
5.2.3	Related Surveys	56
5.2.4	Survey Overview & Organization	58
5.3	Common Terminology	58
5.4	Why Visualize Deep Learning	61
5.4.1	Interpretability & Explainability	61

5.4.2	Debugging & Improving Models	63
5.4.3	Comparing & Selecting Models	63
5.4.4	Teaching Deep Learning Concepts	64
5.5	Who Uses Deep Learning Visualization	65
5.5.1	Model Developers & Builders	65
5.5.2	Model Users	66
5.5.3	Non-experts	68
5.6	What to Visualize in Deep Learning	68
5.6.1	Computational Graph & Network Architecture	69
5.6.2	Learned Model Parameters	69
5.6.3	Individual Computational Units	70
5.6.4	Neurons in High-dimensional Space	70
5.6.5	Aggregated Information	71
5.7	How to Visualize Deep Learning	72
5.7.1	Node-link Diagrams for Network Architectures	72
5.7.2	Dimensionality Reduction & Scatter Plots	73
5.7.3	Line Charts for Temporal Metrics	74
5.7.4	Instance-based Analysis & Exploration	75
5.7.5	Interactive Experimentation	77
5.7.6	Algorithms for Attribution & Feature Visualization	78
5.8	When to Visualize in the Deep Learning Process	80
5.8.1	During Training	80
5.8.2	After Training	81
5.9	Where is Deep Learning Visualization	82
5.9.1	Application Domains & Models	82
5.9.2	A Vibrant Research Community: Hybrid, Apace, & Open-sourced .	84
5.10	Conclusion	85
Chapter 6: SUMMIT: Visualizing Activation and Attribution Summarizations	.	86
6.1	Introduction	86
6.2	Design Challenges	89
6.3	Design Goals	91
6.4	Model Choice and Background	92
6.5	Creating Attribution Graphs by Aggregation	92
6.5.1	Aggregating Neural Network Activations	93
6.5.2	Aggregating Inter-layer Influences	95
6.5.3	Combining Aggregated Activations and Influences to Generate Attribution Graphs	97
6.6	The SUMMIT User Interface	98
6.6.1	Embedding View: Learned Class Overview	98
6.6.2	Class Sidebar: Searching and Sorting Classes	99
6.6.3	Attribution Graph View: Visual Class Summarization	100
6.6.4	System Design	102
6.7	Neural Network Exploration Scenarios	103
6.7.1	Unexpected Semantics Within a Class	103

6.7.2	Mixed Class Association Throughout Layers	104
6.7.3	Discriminable Features in Similar Classes	105
6.7.4	Finding Non-semantic Channels	106
6.7.5	Informing Future Algorithm Design	107
6.8	Conclusion	107
III	Communicating Interpretability with Interactive Articles	108
Chapter 7: Machine Learning Literacy: Interactive Articles in Practice	110
7.1	PARAMETRIC PRESS	110
7.2	The Myth of The Impartial Machine	112
7.3	The Beginner’s Guide to Dimensionality Reduction	113
7.4	Summary	115
Chapter 8: Communicating with Interactive Articles	116
8.1	Introduction	116
8.2	Interactive Articles: Theory and Practice	117
8.2.1	Connecting People and Data	120
8.2.2	Making Systems Playful	123
8.2.3	Prompting Self-Reflection	126
8.2.4	Personalizing Reading	128
8.2.5	Reducing Cognitive Load	130
8.3	Challenges for Authoring Interactives	134
8.4	Critical Reflections	136
8.5	Looking Forward	139
IV	Conclusions	141
Chapter 9: Conclusions and Future Directions	142
9.1	Research Contributions	142
9.2	Impact	143
9.3	Future Directions	144
9.3.1	Multi-model Interpretability Interfaces	144
9.3.2	Understanding Adversarial Attacks	145
9.3.3	Making Interpretability Common Practice	145
9.3.4	Responsible Data-driven Decision Making	147
9.4	Conclusion	148
References	176

LIST OF TABLES

1.1	The publications () and interactive articles () mapped to the thesis outline. Selecting a work's title will navigate to a project page on the web.	12
5.1	Relevant visualization and AI venues ordered by: journals, conferences, workshops, open access journals, and preprint repositories. Within each, visualization venues precedes AI venues.	57
5.2	Foundational deep learning terminology used in this survey, sorted by importance. In a term's “meaning” (last column), defined terms are italicized.	60

LIST OF FIGURES

1.1	An overview of my interdisciplinary research where I design and develop interactive interfaces to enable machine learning interpretability at scale and for everyone	2
1.2	This thesis is composed of three parts, each addressing one research question. Each part is represented by one block with its research question, research answer, and example works that map to the chapters of the thesis. Selecting a block will link to its place in the document.	3
1.3	GAMUT and TELEGAM are interactive visualization systems that allow practitioners to interactively and scalably explain generalized additive models. We use GAMUT as a design probe to investigate the practice of machine learning interpretability with practitioners at a Microsoft. From our findings, we extend our work to TELEGAM that generates natural language verbalizations to complement explanatory visualizations for generalized additive models, and interactively links them through visual annotations.	4
1.4	A visual overview of our INTERROGATIVE SURVEY, and how each of the six questions, “Why, Who, What, How, When, and Where,” relate to one another.	5
1.5	With SUMMIT, users can scalably summarize and interactively interpret deep neural networks by visualizing <i>what</i> features a network detects and <i>how</i> they are related.	6
1.6	An interactive article published on PARAMETRIC PRESS that discusses machine learning’s impact on society that includes descriptive text, interactive graphics, data visualizations, bespoke animations, and live user-controlled simulations.	8

2.1 A common, widely shared example illustrating how neural networks learn hierarchical feature representations. Our work crystallizes these illustrations by systematically building a graph representation that describe <i>what</i> features a model has learned and <i>how</i> they are related. We visualize features learned at individual neurons and connect them to understand how high-level feature representations are formed from lower-level features. Ex. taken from Yann LeCun, 2015.	18
3.1 The GAMUT user interface tightly integrates multiple coordinated views. (A) The Shape Curve View displays GAM shape functions as line charts, and includes histograms of the data density for each feature. The charts can be normalized to better compare the impact each shape function has on the model. (B) The Instance Explanation View displays a waterfall chart for two data instances. Each chart encodes the cumulative impact each feature has on the final prediction for one data instance. (C) The Interactive Table displays the raw data in an interactive data grid where users can sort, filter, and compute nearest neighbors for data instances.	29
3.2 Interacting with GAMUT’s multiple coordinated views together. (A) Selecting the <i>OverallQual</i> feature from the sorted Feature Sidebar displays its shape curve in the Shape Curve View. (B) Brushing over either explanation for <i>Instance 550</i> or <i>Instance 798</i> shows the contribution of the <i>OverallQual</i> feature value for both instances. (C) Notice these two houses are similarly predicted (\$190,606 and \$188,620), but for different reasons!	31
3.3 GAMUT subjective ratings. In a preliminary usability evaluation, participants thought GAMUT was easy to use and enjoyable. Of GAMUT’s multiple coordinated views, all were rated favorably. This also supports our finding that both global and local explanations are valuable for understanding a model’s behavior.	39
4.1 The TELEGAM user interface contains (A) a model selector and parameters for the visualizations and verbalizations. (B) The Global Model View displays model feature-level verbalizations of GAM shape function charts that describe a feature’s overall impact on model predictions. (C) The Local Instance View displays two data instance’s waterfall charts, an explanatory visualization that shows the cumulative sum of the contribution each feature has on the final prediction. Alongside are instance-level verbalizations that, when brushed, highlight in orange) the corresponding marks of the visualization that the verbalization refers to. (D) Settings to interactively tune verbalization generation thresholds.	44

4.2 In TELEGAM, brushing a model feature verbalization displays a tooltip with features' corresponding shape function charts, a common GAM visualization. For example, here, the contribution of the linear-positive feature LotArea on overall model predictions approximately constantly increases as the feature value increases.	46
4.3 TELEGAM supports an initial interactive affordance to realize the simplicity-completeness explanation spectrum in an interface. As a user drags the slider, the resolution of the natural language explanation updates from “brief” to “detailed.” In the example above, the comparison summary for two instances is shown at three different levels of explanation resolution, including (A) brief, (B) default, and (C) detailed.	48
5.1 A visual overview of our interrogative survey, and how each of the six questions, “Why, Who, What, How, When, and Where,” relate to one another. Each question corresponds to one section of this survey, indicated by the numbered tag, near each question title. Each section lists its major subsections discussed in the survey.	53
5.2 Overview of representative works in visual analytics for deep learning. Each row is one work; works are sorted alphabetically by first author’s last name. Each column corresponds to a subsection from the six interrogative questions.	59
5.3 ActiVis [39]: a visual analytics system for interpreting neural network results using a novel visualization that unifies instance- and subset-level inspections of neuron activations deployed at Facebook.	67
5.4 Each point is a data instance’s high-dimensional activations at a particular layer inside of a neural network, dimensionally reduced, and plotted in 2D. Notice as the data flows through the network the activation patterns become more discernible (left to right) [39].	73
5.5 TensorFlow Playground [75]: a web-based visual analytics tool for exploring simple neural networks that uses direct manipulation rather than programming to teach deep learning concepts and develop an intuition about how neural networks behave.	78
5.6 Distill: The Building Blocks of Interpretability [70]: an interactive user interface that combines feature visualization and attribution techniques to interpret neural networks.	83

6.1	With Summit, users can scalably summarize and interactively interpret deep neural networks by visualizing <i>what</i> features a network detects and <i>how</i> they are related. In this example, INCEPTIONV1 accurately classifies images of <i>tench</i> (yellow-brown fish). However, SUMMIT reveals surprising associations in the network (e.g., using parts of people) that contribute to its final outcome: the “tench” prediction is dependent on an intermediate “hands holding fish” feature (right callout), which is influenced by lower-level features like “ <i>scales</i> ,” “ <i>person</i> ,” and “ <i>fish</i> ”. (A) Embedding View summarizes all classes’ aggregated activations using dimensionality reduction. (B) Class Sidebar enables users to search, sort, and compare all classes within a model. (C) Attribution Graph View visualizes highly activated neurons as vertices (“ <i>scales</i> ,” “ <i>fish</i> ”) and their most influential connections as edges (dashed purple edges).	87
6.2	A high-level illustration of how we take thousands of images for a given class, e.g., images from <i>white wolf</i> class, compute their top activations and attributions, and combine them to form an attribution graph that shows how lower-level features (“ <i>legs</i> ”) contribute to higher-level ones (“ <i>white fur</i> ”), and ultimately the final outcome.	89
6.3	A visual depiction of our approach for aggregating activations and influences for a layer l . Aggregating Activations: (A1) given activations at layer l , (A2) compute the max of each 2D channel, and (A3) record the top activated channels into an (A4) aggregated activation matrix, which tells us which channels in a layer most activate and represent every class in the model. Aggregating Influences: (I1) given activations at layer $l - 1$, (I2) convolve them with a convolutional kernel from layer l , (I3) compute the max of each resulting 2D activation map, and (I4) record the top most influential channels from layer $l - 1$ that impact channels in layer l into an (I5) aggregated influence matrix, which tells us which channels in the previous layer most influence a particular channel in the next layer.	93
6.4	Selectable network minimap animates the Embedding View.	99
6.5	Class Sidebar visual encoding.	99
6.6	An example substructure from the <i>lionfish</i> attribution graph that shows unexpected texture features, like “ <i>quills</i> ” and “ <i>stripes</i> ,” influencing top activated channels for a final layer’s “ <i>orange fish</i> ” feature (some <i>lion fish</i> are reddish-orange, and have white fin rays).	103

6.7 Using SUMMIT we can find classes with mixed semantics that shift their primary associations throughout the network layers. For example, early in the network, <i>horsecart</i> is most similar to mechanical classes (e.g., harvester, thresher, snowplow), towards the middle it shifts to be nearer to animal classes (e.g., bison, wild boar, ox), but ultimately returns to have a stronger mechanical association at the network output.	104
6.8 With attribution graphs, we can compare classes throughout layers of a network. Here we compare two similar classes: <i>black bear</i> and <i>brown bear</i> . From the intersection of their attribution graphs, we see both classes share features related to <i>bear-ness</i> , but diverge towards the end of the network using fur color and face color as discriminable features. This feature discrimination aligns with how humans might classify bears.	105
6.9 Using SUMMIT on INCEPTIONV1 we found non-semantic channels that detect irrelevant features, regardless of the input image, e.g., in layer mixed3a, channel 67 is activated by the frame of an image.	106
7.1 PARAMETRIC PRESS is our interactive publication, a born-digital magazine dedicated to showcasing the expository power that's possible when the audio, visual, and interactive capabilities of dynamic media are effectively combined.	111
7.2 A subset of the interactive graphics from The Myth of The Impartial Machine, demonstrating the trajectory and effect of sampling bias in data collection and model building.	113
7.3 The dataset and embeddings shown in The Beginner's Guide to Dimensionality Reduction as a reader progresses through the interactive article.	114
8.1 Exemplary interactive articles from around the web. In the interactive version of this figure, readers can hover over an article to enlarge its thumbnail and see more information.	117
8.2 Interactive articles are applicable to variety of domains, such as research dissemination, journalism, education, and policy and decision making. In the interactive version of this figure, readers can select the tabs to view different domains.	118
8.3 In the interactive version of this table, readers can sort a list of the interactive articles we discuss in this work.	119

8.4	The five affordances of interactive articles we discuss.	119
8.5	In the interactive version of this figure, readers can click the play button or scrub over the video frames to watch and control the animation.	120
8.6	In the example, “Extensive Data Shows Punishing Reach of Racism for Black Boys,” [253] the use of unit animation carries the main visualization of the story to highlight real people’s lives changing over time.	121
8.7	In the example, “Cutthroat Capitalism: The Game,” [265] readers play the role of a pirate commander, giving them a unique look at the economics that led to rise in piracy off the coast of Somalia.	122
8.8	In the interactive version of this figure, readers can drag a slider to change the number of boids in the simulation. Underneath the visualization, readers can also adjust the different parameters to find interesting configurations, for example comparing the left and right views above.	124
8.9	In the example, “Teachable Machines,” [274] a reader uses their own live video camera to train a machine learning image classifier in-browser without any extra computational resources.	125
8.10	In the example, “Visualizing Quaternions,” [288] a viewer can take control of an interactive video while narration continues in the background.	126
8.11	In the example, “The Gyllenhaal Experiment,” [296] readers are tasked to type the names of celebrities with challenging spellings. After submitting a guess, a visualization shows the reader’s entry against everyone else’s, scaled by the frequency of different spellings.	127
8.12	In the interactive version of this figure, readers can click and drag to make your guess of the data’s trend over time. Afterward, the real data is revealed.	127
8.13	In the example, “How To Remember Anything Forever-ish,” [303] readers use spaced repetition to learn about spaced repetition.	128
8.14	In the example, “How Much Hotter Is Your Hometown Than When You Were Born?,” [310] a reader enters their birthplace and birth year and is shown multiple visualizations describing the impact of climate on their hometown.	129
8.15	In the example, “Quantum Country,” [305] the interactive textbook uses spaced repetition and allows a reader to opt-in and save their progress while reading through dense material and mathematical notion over time..	130

8.16 In the interactive version of this figure, readers can click any point to listen to a different bird's chirp.	131
8.17 In the interactive version of this figure, readers can choose between 1 of 4 machine-generated images and brush over the circle callouts to display a short message about each region.	132
8.18 In the interactive version of this figure, readers can click to reveal, or remind oneself, what each mark of notation or variable represents in the equation.	133
8.19 In the interactive version of this figure, readers can drag the slider to display the theorem's statement in increasing levels of detail.	133
8.20 The Myth of the Impartial Machine was one of five articles published in PARAMETRIC PRESS. The article used techniques like animation, data visualizations, explanatory diagrams, margin notes, and interactive simulations to explain how biases occur in machine learning systems.	137
8.21 Interactive communication opportunities from both research and practice.	138
8.22 Example explorable explanations made in three weeks during the Explorables Jam covering topics from math, astronomy, computer graphics, and music.	139

SUMMARY

Data-driven paradigms now solve the world’s hardest problems by automatically learning from data. Unfortunately, what is learned is often unknown to both the people who train the models and the people they impact. This has led to a rallying cry for *machine learning interpretability*. But how we enable interpretability? How do we scale up explanations for modern, complex models? And how can we best communicate them to people?

Since machine learning now impacts people’s daily lives, we answer these questions taking a *human-centered perspective* by designing and developing interactive interfaces that enable interpretability at scale and for everyone. This thesis focuses on:

(1) **Enabling machine learning interpretability:** User research with practitioners guides the creation of our novel operationalization for interpretability, which helps tool builders design interactive systems for model and prediction explanations. We develop two such visualization systems, GAMUT and TELEGAM, which we deploy at Microsoft Research as a design probe to investigate the emerging practice of interpreting models.

(2) **Scaling deep learning interpretability:** Our first-of-its-kind INTERROGATIVE SURVEY reveals critical yet understudied areas of deep learning interpretability research, such as the lack of higher-level explanations for neural networks. Through SUMMIT, an interactive visualization system, we present the first scalable graph representation that summarizes and visualizes what features deep learning models learn and how those features interact to make predictions (e.g., InceptionNet trained on ImageNet with 1.2M+ images).

(3) **Communicating interpretability with interactive articles:** We use interactive articles, a new medium on the web, to teach people about machine learning’s capabilities and limitations, while developing a new interactive publishing initiative called the PARAMETRIC PRESS. From our success publishing interactive content at scale, we generalize and detail the affordances of INTERACTIVE ARTICLES by connecting techniques used in practice and the theories and empirical evaluations put forth by diverse disciplines of research.

This thesis contributes to *information visualization*, *machine learning*, and more importantly *their intersection*, including open-source interactive interfaces, scalable algorithms, and new, accessible communication paradigms. Our work is making significant impact in industry and society: our visualizations have been deployed and demoed at Microsoft and built into widely-used interpretability toolkits, our interactive articles have been read by 250,000+ people, and our interpretability research is supported by NASA.

CHAPTER 1

INTRODUCTION

Some of the world’s hardest problems are now being solved by data-driven, machine learning (ML) approaches. This has revolutionized conventional computing paradigms where people would author a function with exact and explicit rules that take in data as input to produce output. In a data-driven paradigm, an algorithm instead *learns* a function and its rules from pairs of input data and output examples. This change in problem solving has enabled us to master ancient board games, transform healthcare, co-create generative content, and understand the universe.

Unfortunately, often what is learned is unknown to both the people who develop the models and the people they impact. Research that audits machine learning technologies has shown numerous examples of models placing people in danger while also encoding and perpetuating societal biases. Examples include representing gender bias in facial analysis systems [1], propagating historical cultural stereotypes in text corpora into widely used models [2], biasing recidivism predictions by race [3], and completely breaking down given human-imperceptible adversarial attacks on computer vision applications such as autonomous driving [4]. These problems are exacerbated twofold: (1) *technologically*, when the most performant models suffer from hidden problems yet are transferred and applied in dozens of other domains, and (2) *societally*, when unaware people lose agency to automation due to a lack of understanding of data-driven technologies that have rapidly spread to nearly all aspects of our daily lives.

These problems have led to a rallying cry for **machine learning interpretability**: the desire to understand what a machine learning model has learned and how it makes decisions. Beyond understanding what a model may have learned to ensure the safety of people interacting with artificially intelligent (AI) technologies, there are numerous benefits to wielding interpretability [5]. These include ensuring models are fair to all people, building robust models that are protected against adversarial attacks and data drift, preserving user agency when interacting with personalized agents, providing a good user experience with data-driven technologies, helping understand our world through data-driven scientific discovery, and simply knowing whether models are right for the right reasons.

But what is interpretability, what do people expect from it, and how to we enable it in interactive systems and user interfaces? What do modern, complex models learn inside their internal representations? And how can we best represent and communicate these

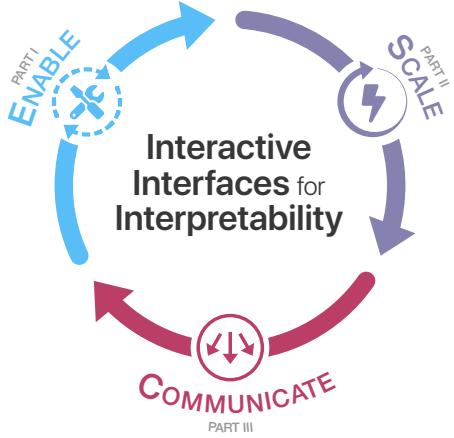


Figure 1.1: An overview of my interdisciplinary research where I design and develop interactive interfaces to **enable** machine learning interpretability at **scale** and for **everyone**.

models and their predictions to people? This dissertation presents new paradigms, methods, and interactive interfaces that address these challenges.

1.1 Designing Machine Learning Interpretability for People

Through my diverse research experience at national laboratories, NASA, Microsoft Research, and Apple over the past 5 years, it is clear that **applying machine learning is a people problem**. People gather data and people annotate data. People build models and people deploy models. People make decisions based on models and ultimately models impact people. *People* are at the center of every stage of the machine learning development process [6]. In other words, machine learning is inherently a human-centered problem, where people constantly make design decisions regarding the task, data, and model. I argue interpretability should be no different. Therefore, this thesis studies machine learning interpretability from a *human-centered perspective*.

It is of paramount importance to understand what machine learning systems learn to help developers debug their models, practitioners apply machine learning appropriately, and people understand how to interact with machine learning in their daily lives. Since many different and diverse populations can benefit from interpretability, a deep understanding of the users of such technology is needed to know how to represent and communicate machine learning interpretability.

Interactive Scalable Interfaces for Machine Learning Interpretability



How to empower people to interactively explain ML?

CHAPTER 3
GAMUT Operationalize interpretability *CHI'19*

CHAPTER 4
TELEGAM Vis + text for better explanations *VIS'19*



How to make sense of large data & understand complex models?

CHAPTER 5
INTERROGATIVE SURVEY Summarize interpretability vis *TVCG'18*

CHAPTER 6
SUMMIT Higher-level explanations for neural networks *TVCG'20*



How to inform everyone about ML's impact on their lives?

CHAPTER 7
ML LITERACY Interactive mediums & platforms *VISCOMM'19, VISxAI'18*

CHAPTER 8
INTERACTIVE ARTICLES Formalizing interactive communication *Distill'20*

Figure 1.2: This thesis is composed of three parts, each addressing one research question. Each part is represented by one block with its research question, research answer, and example works that map to the chapters of the thesis. Selecting a block will link to its place in the document.

1.2 Thesis Overview

To *enable machine learning interpretability at scale for everyone*, this thesis investigates how to enable interpretability in practice (Part I), how to scale explanations to complex models (Part II), and how to communicate explanations to people through interactive articles and data visualizations (Part III). This thesis focuses on three complementary research questions in Figure 1.2, which are mapped to their corresponding parts, answers, and example works of this dissertation.

1.2.1 Part I: Enabling Machine Learning Interpretability

Building machine learning models is now common practice, but interpreting them is not. Without good models and the right tools to interpret them, data scientists risk making decisions based on hidden biases, spurious correlations, and false generalizations. This has led to a rallying cry for model interpretability. Yet the concept of interpretability remains nebulous, such that researchers and tool designers lack actionable guidelines for how to incorporate interpretability into models and accompanying tools. What do model developers and data scientists expect and want from machine learning interpretability? How can

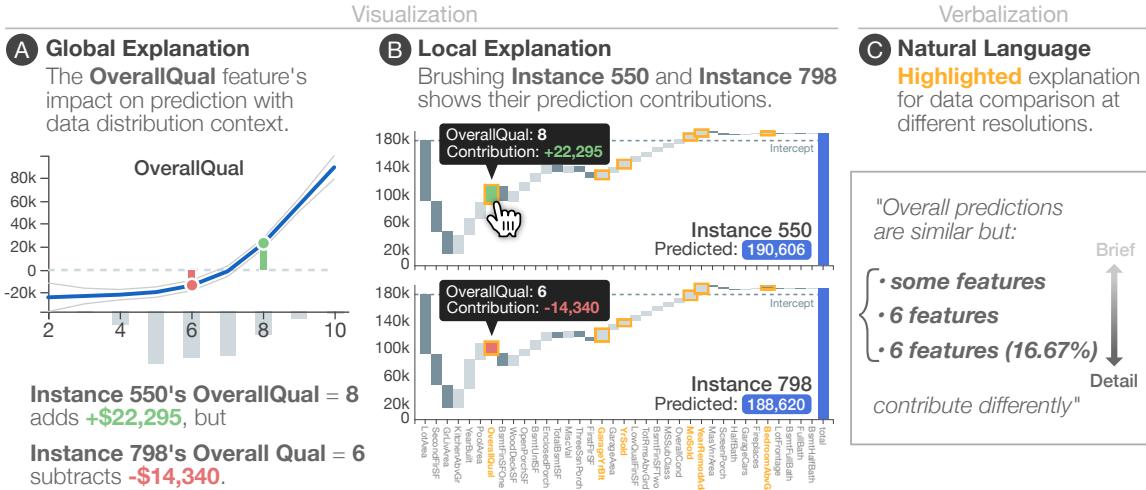


Figure 1.3: GAMUT and TELEGAM are interactive visualization systems that allow practitioners to interactively and scalably explain generalized additive models. We use GAMUT as a design probe to investigate the practice of machine learning interpretability with practitioners at a Microsoft. From our findings, we extend our work to TELEGAM that generates natural language verbalizations to complement explanatory visualizations for generalized additive models, and interactively links them through visual annotations.

we help practitioners understand their models and build interpretability into their machine learning systems and interactive user interfaces?

GAMUT: *Understanding How Data Scientists Understand Machine Learning (Chapter 3)*

Through an iterative design process with expert machine learning researchers and practitioners at Microsoft, we designed a visual analytics system, GAMUT (Figure 1.3A-B), to explore how interactive interfaces could better support model interpretation [7]. Using GAMUT as a probe, we investigated why and how practitioners interpret models, and how interface affordances can support them in answering questions about model interpretability. Our investigation showed that interpretability is not a monolithic concept: practitioners have different reasons to interpret models and tailor explanations for specific audiences, often balancing competing concerns of simplicity and completeness. Participants also asked to use GAMUT in their work, highlighting its potential to help practitioners understand their own data. GAMUT has been deployed at Microsoft, demoed for executive leadership at their internal TechFest, and incorporated into their open-source library InterpretML.

TELEGAM: *Combing Visualization and Verbalization for Interpretability (Chapter 4)*

Although visualizations are a powerful tool to interpret models, depending on the complexity of the model (e.g., number of features), interpreting these visualizations can be difficult

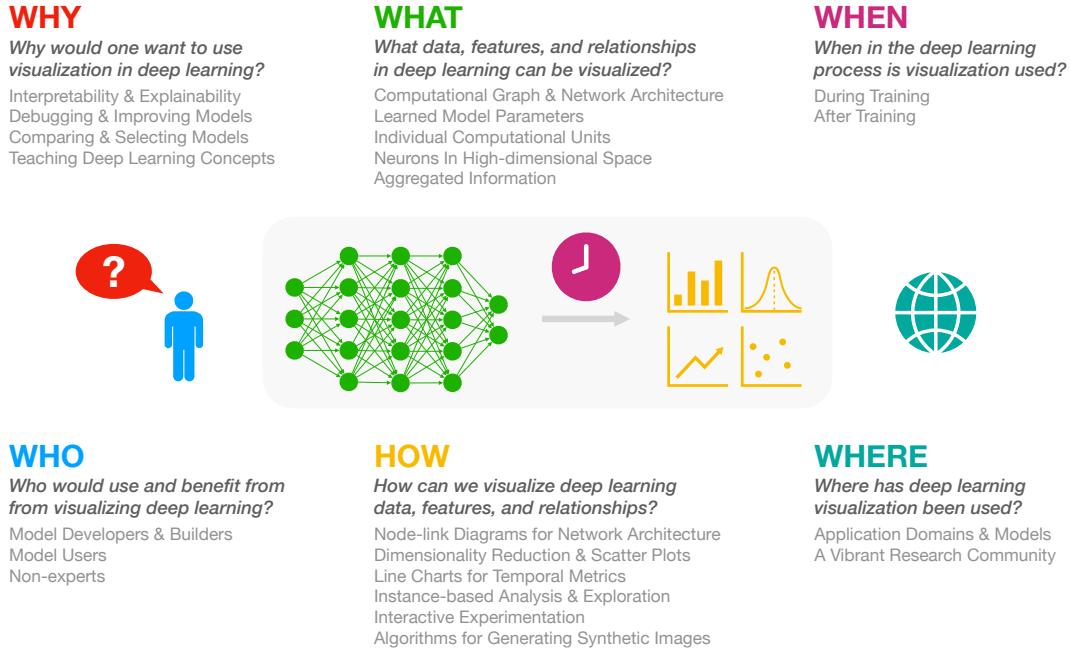


Figure 1.4: A visual overview of our INTERROGATIVE SURVEY, and how each of the six questions, “Why, Who, What, How, When, and Where,” relate to one another.

and may require additional expertise. Alternatively, textual descriptions, or verbalizations, can be a simple, yet effective way to communicate or summarize key aspects about a model, such as the overall trend in a model’s predictions or comparisons between pairs of data instances. With the potential benefits of visualizations and verbalizations in mind, we explore how the two can be combined to aid machine learning interpretability. Specifically, we extend our work in GAMUT and present an interactive interface, TELEGAM (Figure 1.3C), that demonstrates how visualizations and verbalizations can collectively support interactive exploration of machine learning models, for example, generalized additive models [8]. We discuss how TELEGAM can serve as a platform to conduct future studies for understanding user expectations and designing novel interfaces for interpretable machine learning.

1.2.2 Part II: Scaling Deep Learning Interpretability

Deep learning has recently seen rapid development and significant attention due to its state-of-the-art performance on previously-thought hard problems. However, because of the internal complexity and nonlinear structure of deep neural networks, the underlying decision making processes for these models are challenging and sometimes mystifying to

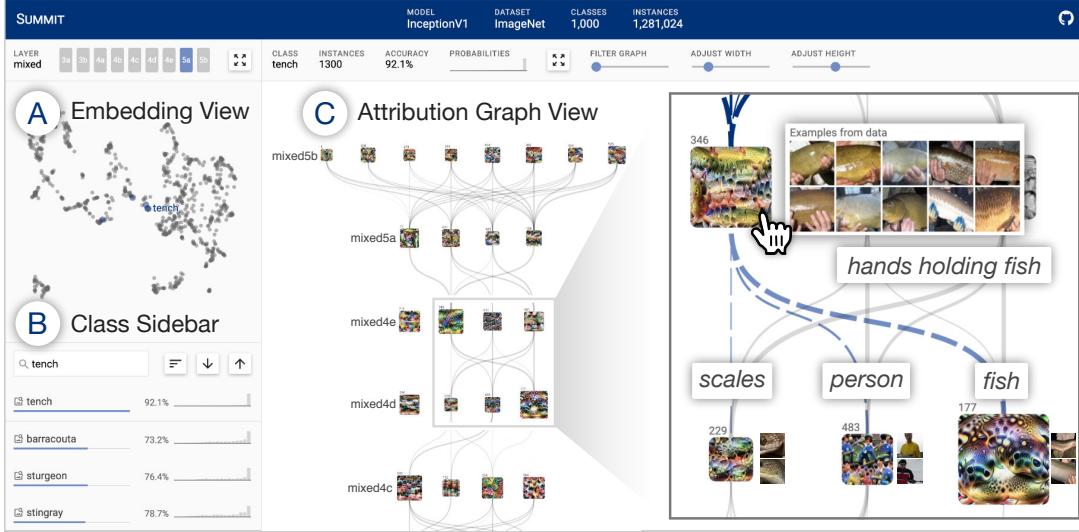


Figure 1.5: With SUMMIT, users can scalably summarize and interactively interpret deep neural networks by visualizing *what* features a network detects and *how* they are related.

understand. How can we equip people with tools for understanding when a model works correctly, when it fails, and ultimately how to improve its performance? And since deep neural networks often make predictions computed from millions of parameters that are optimized over millions of data instances, how do we ensure explanations for learned feature representations capture higher-level model and dataset structure?

Visual Analytics in Deep Learning: An Interrogative Survey (Chapter 5)

We present a survey of the role of visual analytics in deep learning research (Figure 1.4), which highlights its short yet impactful history and thoroughly summarizes the state-of-the-art using a human-centered interrogative framework, focusing on the *Five W's and How* (Why, Who, What, How, When, and Where) [9]. We highlight research directions and open research problems. This survey helps researchers and practitioners in both visual analytics and deep learning to quickly learn key aspects of this young and rapidly growing body of research, whose impact spans a diverse range of domains.

SUMMIT: Scaling Deep Learning Interpretability by Visualizing Activation and Attribution Summarizations (Chapter 6)

From our survey, we identified that existing work on interpreting neural network predictions for images often focuses on explaining predictions for single images or neurons. As predictions are often computed from millions of weights that are optimized over millions of images, such explanations can easily miss a bigger picture.

We present SUMMIT (Figure 1.5), an interactive system that scalably and systematically summarizes and visualizes what features a deep learning model has learned and how those features interact to make predictions [10]. SUMMIT introduces two new scalable summarization techniques: (1) *activation aggregation* discovers important neurons, and (2) *neuron-influence aggregation* identifies relationships among such neurons. SUMMIT combines these techniques to create the novel *attribution graph* that reveals and summarizes crucial neuron associations and substructures that contribute to a model’s outcomes. SUMMIT scales to large data, such as the ImageNet dataset with 1.2M images, and leverages neural network feature visualization and dataset examples to help users distill large, complex neural network models into compact, interactive visualizations. We present neural network exploration scenarios where SUMMIT helps us discover multiple surprising insights into a prevalent, large-scale image classifier’s learned representations and informs future neural network architecture design. The SUMMIT visualization runs in modern web browsers, is open-sourced, and supported by a NASA PhD fellowship.

1.2.3 Part III: Communicating Interpretability with Interactive Articles

So far, these interfaces for interpretability focus on data literate people with machine learning expertise. But machine learning now impacts everyone, therefore it is important that everyone knows how to use it, identify when it is wrong, and correct it. One of the most important challenge is how to represent explanations without technical overhead or requiring years of machine learning experience. How can we bring interpretability to everyone, including those without technical expertise? How do we ensure the broader public understands the capabilities and limitations of machine learning?

In contrast to traditional static media such as books and pictures, and moving media such as movies and animations, interactive articles are a new medium for communication that leverage the dynamic capabilities of the web to explain complex topics. These articles are characterized by interleaving text and interactive widgets – often utilizing animations, data visualizations, or simulations – that guide a reader through a primarily linear narrative. Interactive articles are becoming popular on the web: newspapers publish interactive articles that include dynamic graphics and visualizations; educators and technical communicators enrich text with interactions and multimedia in an effort to further engage their students and readers. In practice, these articles, while still relatively rare, bring broad readership, gain wide acclaim, and help educate people, but are difficult and time-consuming to author and are distributed among different communities.



Figure 1.6: An interactive article published on PARAMETRIC PRESS that discusses machine learning’s impact on society that includes descriptive text, interactive graphics, data visualizations, bespoke animations, and live user-controlled simulations.

Machine Learning Literacy: Interactive Articles in Practice (Chapter 7)

To help teach people about machine learning, we have written multiple interactive articles (Figure 1.6) on a diverse set of topics such as interpretability, fairness, and bias [11], common data science techniques such as dimensionality reduction [12], and launched a new open-source publishing initiative called PARAMETRIC PRESS to test these techniques in the wild—while empowering authors to tell data-driven stories and create explorable explanations [13]. PARAMETRIC PRESS provides an outlet to experiment with new interfaces that use interactivity, visualizations, and simulations to teach people about aspects of machine learning. Our articles went viral, which allowed us to analyze thousands of reader patterns to evaluate how this new medium is read and used in practice, a critical yet underexamined aspect of publishing interactive content.

Communicating with Interactive Articles (Chapter 8)

We have shown that there is growing excitement for using interactive articles for machine learning communication; however, since interactive articles are a new, highly flexible, and

expressive medium, there is little previous work for why they are useful and how they can benefit readers. With our knowledge and experience from successfully publishing interactive content at scale, we connect the dots between interactive articles such as ours and others featured in popular media publications and the techniques, theories, and empirical evaluations put forth by academic researchers across the fields of education, human-computer interaction, information visualization, and digital journalism [14]. After describing the affordances of interactive articles, we provide critical reflections from our own experience with open-source, interactive publishing. We conclude with discussing practical challenges and open research directions for authoring, designing, and publishing interactive articles.

1.3 Thesis Statement

A human-centered approach to designing and developing interactive interfaces for machine learning interpretability helps researchers and practitioners:

1. **enable** **interpretability** within interactive systems for model explanation,
2. **scale** **interpretability** to deep neural network feature representations, and
3. **communicate** **interpretability** to inform people about machine learning's capabilities, limitations, and impact on our lives.

1.4 Research Contributions

This thesis makes research contributions to multiple fields, including interactive data visualization, machine learning, and more importantly their intersection to **enable** (Part I), **scale** (Part II), and **communicate** (Part III) machine learning interpretability.

First formulation of machine learning interpretability system design.

- Through user research with practitioners, our work represents the *first operationalization of interpretability that defines a set of unique capabilities* interactive interpretability systems should support, and establishes a *model for future investigations* on understanding how people use interpretability tools in practice (Chapter 3).
- We *design, develop, and deploy a cohesive collection of interactive systems*, GAMUT (Chapter 3), TELEGAM (Chapter 4), and SUMMIT (Chapter 6), that showcases how our operationalization helps people understand models and their predictions across multiple modalities, data types, and explanation mediums.

New interactive and scalable techniques for global model understanding.

- GAMUT (Chapter 3) and TELEGAM (Chapter 4) *interactively combine global and local explanations*, commonly done separately, which not only give users the best of both worlds but we show is essential to effectively enabling interpretability.
- Our INTERROGATIVE SURVEY (Chapter 5), the *first comprehensive survey for visual analytics in deep learning*, helps practitioners quickly learn key aspects of this young and rapidly growing field.
- SUMMIT (Chapter 6) introduces *two new aggregation algorithms to create attribution graphs, the first scalable graph representation for understanding neural networks*, and combines feature visualization, graph visualization, and graph mining techniques to interactively explore neural network feature representations for *millions of images*.

New paradigm for amplifying research dissemination and interactive communication.

- SUMMIT’s (Chapter 6) *live demo and article amplify research dissemination* and engages people with state-of-the-art computing research while reducing the barrier to entry.
- The *viral success* of PARAMETRIC PRESS (Chapter 7) exemplifies the power of the web as a substrate for communicating complex ideas with dynamic media.
- Our work that connects the theory and practice of INTERACTIVE ARTICLES (Chapter 8) is itself *authored as an interactive article, the first work of its kind* that demonstrates interactive techniques alongside its discussion inline.

Open-source systems that broadens people’s access to interpretability.

- SUMMIT (Chapter 6) and TELEGAM (Chapter 4) are both *open-sourced* and accessible without any installation via *interactive web demos*.
- PARAMETRIC PRESS (Chapter 7) and our INTERACTIVE ARTICLES (Chapter 8) are also *open-sourced, including every article, visualization component, and the publishing engine itself* to allow authors to reuse templates for interactive articles.

1.5 Impact

Beyond the visualization and machine learning research communities, this thesis work has made significant broader impact to industry and society:

- GAMUT (Chapter 3) has been *deployed at Microsoft, was demoed for executive lead-*

ership at their internal TechFest, and has been *incorporated into their open-source interpretability toolkit InterpretML* (2,900+ stars on Github).

- PARAMETRIC PRESS (Chapter 7) and our other interactive articles *went viral*, have been *read by 250,000+ people*, *helped students learn about machine learning concepts*, and have *gathered acclaim for their mission and execution* (e.g., multiple Hacker News front page appearances, featured on Stack Overflow Blog, FastCompany review).
- The designed and developed interactive interfaces for interpretability, with a focus on SUMMIT (Chapter 6) have been invested in and recognized by a *NASA Space Technology Research PhD Fellowship at the Jet Propulsion Lab*, as well as a *Microsoft AI for Earth Award*.

This dissertation contributes novel interactive techniques, scalable algorithms, and open-source systems that *enable machine learning interpretability at scale for everyone*. Our research advances our technical understanding of responsible data-driven decision making, helps people uncover what machine learning models learn, and more importantly reinforces machine learning as a technique to *empower people, augmenting human intelligence and decision-making*. We hope our work will inspire and accelerate deeper engagement from both the human-computer interaction and machine learning communities to further innovate interactive interfaces for artificial intelligence.

1.6 Prior Publications and Authorship

While I am the principal author of the research included in this thesis, the research is the result of years of collaboration with my PhD advisor, Duen Horng (Polo) Chau, as well as many mentors and colleagues at Georgia Institute of Technology, Apple, Microsoft Research, NASA Jet Propulsion Lab, and Pacific Northwest National Lab. To reflect my collaborators' contributions, I will use the first-person plural throughout the thesis chapters. The research in this thesis that has been published previously is listed in Table 1.1.

Table 1.1: The publications () and interactive articles () mapped to the thesis outline. Selecting a work's title will navigate to a project page on the web.

Part I: Enabling Machine Learning Interpretability

GAMUT: A Design Probe to Understand How Data Scientists Understand Machine Learning Models. Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, Steven M. Drucker. *ACM Conference on Human Factors in Computing Systems (CHI)*, 2019 (Chapter 3).

TELEGAM: Combining Visualization and Verbalization for Interpretable Machine Learning. Fred Hohman, Arjun Srinivasan, Steven M. Drucker. *IEEE Visualization Conference (VIS)*, 2019 (Chapter 4).

Part II: Scaling Deep Learning Interpretability

Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. Fred Hohman, Minsuk Kahng, Robert Pienta, Duen Horng (Polo) Chau. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*. Berlin, Germany, 2018 (Chapter 5).

SUMMIT: Scaling Deep Learning Interpretability by Visualizing Activation and Attribution Summarizations. Fred Hohman, Haekyu Park, Caleb Robinson, Duen Horng (Polo) Chau. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*. Vancouver, Canada, 2020 (Chapter 6).

Part III: Communicating Interpretability with Interactive Articles

Launching the PARAMETRIC PRESS. Matthew Conlen, Fred Hohman. *Visualization for Communication at IEEE VIS (VisComm)*. Vancouver, Canada, 2019 (Chapter 7).

The Myth of the Impartial Machine. Alice Feng, Shuyan Wu, Fred Hohman, Matthew Conlen, Victoria Uren. *The Parametric Press, Issue 01*, 2019 (Chapter 7).

The Beginner's Guide to Dimensionality Reduction. Matthew Conlen, Fred Hohman. *Workshop on Visualization for AI Explainability at IEEE VIS (VISxAI)*. Berlin, Germany, 2018 (Chapter 7).

Communicating with Interactive Articles. Fred Hohman, Matthew Conlen, Jeffrey Heer, Duen Horng (Polo) Chau. *Distill*, 2020 (Chapter 8).

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Definitions of Interpretability

While existing definitions of interpretability center on human understanding, they vary in the aspect of the model to be understood: its internals [15], operations [16], mapping of data [17], or representation [18]. Hence, a formal, agreed upon definition remains open [19, 20]. These discussions make a distinction between *interpretability* (synonymous with explainability) and an *explanation*. An explanation is a collection of features from an interpretable domain that relate a data instance to a model’s outcome [17, 18]. An explanation can be truthful or deceptive, accurate or inaccurate, all with varying degrees of success. Therefore, multiple explanations are often used to gain an ultimate interpretation of a model. Miller argues that interpretability research should leverage the literature from philosophy, psychology, and cognitive science for the history of how people define, generate, select, evaluate, and present explanations [21]. This discussion is echoed and expanded upon in one of the few books solely dedicated to interpretability [22]. In this thesis, we build upon existing interpretability literature by using a human-centered approach to understand why data scientists need interpretability, how they use it, and how human-computer interaction methods can help design interfaces to explain models.

2.1.1 Audience for Interpretability

Rather than considering interpretability as a monolithic concept, it may be more useful to identify properties that AI systems should obey to ensure interpretability, such as simulability, decomposability, and algorithmic transparency [20]. Recent work argues that the sophistication and completeness of both interpretability and explanations depends on the audience [15, 18]. Model builders may prefer global, aggregate model explanations; whereas, model users may prefer local, specific decision examples. Both explanation paradigms will impact the interpretability of a system. In GAMUT, we support both global and local paradigms, and in its extension TELEGAM, we offer an interactive affordance that allows users to dynamically update the resolution of a verbalization to tailor the level of detail desired in explanation.

2.1.2 Interpretability and AI Guidelines

The GDPR’s recent declaration of the “right to explanation” [23] has sparked discussion for what this means in practice and what impact it will have on industry and research agendas [24]. While the updated version of the GDPR only requires explanation in limited contexts, AI and policy scholars expect explanations to be important in future regulations of AI systems [25]. Researchers have introduced a framework to turn the vague language of the GDPR into actionable guidelines, which include (1) identifying the factors that went into a decision, (2) knowing how varying a factor impacts a decision, and (3) comparing similar instances with different outcomes [25]. However, within this framework an AI-system need only satisfy one of the three above guidelines to be considered interpretable. Other useful post-hoc techniques for explaining decisions have also been proposed, such as using counterfactuals (that is, “What if” questions [26]), textual explanations, visualizations, local explanations, and representative examples of data [20]. We add to this existing work by contributing a list of capabilities that explainable interfaces should support to help people interpret models.

As AI-based systems become more common in our daily lives, including personal computing, large technology companies and industry research labs have recently released guidelines on designing systems involving human-AI interaction. Most notably, Microsoft, Google, and Apple have released guidelines that, while at times overlapping, each have a different approach for helping different audiences designing with data and machine learning in mind. Below we briefly describe each.

Microsoft’s Guidelines for Human-AI Interaction In a research paper published at ACM CHI 2019, Microsoft’s set of 18 ”Guidelines for Human-AI Interaction” [27] were derived using an extensive crowdsourced survey of over 168 potential guidelines originating from internal and external industry sources, public articles, and the academic literature. These guidelines were then organized and combined to fit within a single structure with a common style of “a rule of action, containing about 3-10 words and starting with a verb” and are structured according to their relevance during a user’s interaction with an AI feature or product. These primary stages are “initially, during interaction, when wrong, and over time.” The work also reports on a user study with HCI practitioners in order to evaluate the applicability and clearness of the guidelines.

Google’s People + AI Guidebook In an online book published in 2019, “Google’s People + AI Guidebook” describes how to follow a human-centered approach to working with AI based on “data and insights from Google product teams and academic research [28]”

The categorization of guidelines used here are organized around the process of product development, including considering user needs, data curation, accounting for mental models, explainability and trust, user feedback and control, and how to gracefully handle errors. Comparatively, these guidelines tend to be more prescriptive and detailed than others, offering atomic examples that illustrate subtle points around building with machine learning.

Apple’s Human Interface Guidelines for Machine Learning Included in Apple’s establish Human Interface Guidelines during WWDC 2019, their Machine Learning guidelines are based upon long standing design principles used within the company to design machine learning features and products [29]. Here the focus is almost entirely on user experience rather than AI development or functionality. These guidelines are divided into two main themes, the inputs of a model, further subdivided into explicit and implicit feedback, calibration, and corrections, and the outputs of a model, further subdivided into handling mistakes, multiple options, confidence, attribution, and limitations. These guidelines aim to help design the process by which machine learning products Each category of these sections aim to help design the processes by which learning products ask for, collect, use, and apply user data and interactions, and how preserve agency within users by displaying outputs that are understandable and actionable.

2.2 Visual Analytics for Machine Learning Interpretability

Previous work demonstrates that interaction between users and machine learning systems is a promising direction for collaboratively sharing intelligence [30]. Since then, interactive visual analytics has succeeded in supporting machine learning tasks [9, 5, 31, 32, 33]. Example tasks include interactive model debugging and performance analysis [34, 35, 36], feature ideation and selection [37, 38], instance subset inspection and comparison [39, 40], model comparison [41], and constructing interpretable and trustworthy visual analysis systems [42].

To address model interpretability, a burgeoning research field of explainable artificial intelligence (AI) has emerged, whose general goal is to create and evaluate effective explanations for model decisions to better understand what a model has learned [43]. Recently, information visualization [44] has been used as an medium for explanation [9, 45, 31, 32]. This is a natural fit, since visualization and interactive visual analytics [46] excel at graphical communication for complex ideas and meaningful summarization of information. While model explanations come in many forms (e.g., textual, graphical), two primary yet competing paradigms have emerged: global and local explanations. *Global explanations*

roughly capture the entire space learned by a model in aggregate, favoring simplicity over completeness. Conversely, *local explanations* accurately describe a single data instance’s prediction.

Most visual analytics systems specifically supporting machine learning interpretability do so on deep neural network models, a relatively newer trend. These systems are discussed in detail below. For non-neural network architectures, two visual analytics systems in particular are related to our work on GAMUT. Prospector [47] and the What-If Tool [48] use interactive partial dependence diagnostics and localized inspection techniques to allow data scientists to understand the outcomes for specific instances. These partial dependency charts are similar to the shape functions used in generalized additive models [22]. Both systems support using counterfactuals and modifying feature values on data instances to observe how changes could impact prediction outcome. In preliminary follow-up work, researchers investigated the effectiveness of providing instance explanations in aggregate, similarly identifying the distinction between global and local explanation paradigms [49]. We contribute to visual analytics literature by developing GAMUT, an interactive visualization system used as a design probe to investigate how data scientists use global and local explanation paradigms.

2.2.1 Complementing Visualizations with Verbalizations.

While visualizations are powerful tools to help people better understand ML models, they may not be sufficient, and depending on a user’s background, they can also be challenging to interpret. Recent work has begun to conjecture whether complementing visualizations with verbalizations can enhance model explanations. For instance, Sevastjanova et al. [50] present a design space discussing strategies for model explanation generation and presentation at the intersection of visualizations and verbalizations. In their design space, *post-hoc interpretability* describes when an explanation uses the relationship between the input and output of a model instead of the model’s inner mechanisms [51]. Specifically, following a strategy similar to recent visualization tools that systematically extract “data facts” to highlight potentially interesting observations in visualizations [52, 53, 54], in TELEGAM we heuristically analyze the data associated with model-level and instance-level visualizations, and present them as textual statements alongside visualizations. In other words, we adopt an overview and detail strategy [50] for generating explanations where visualizations are used to give an overview while the verbalizations highlight specific features or trends. Furthermore, TELEGAM also interactively links visualizations and verbalizations, supporting details-on-demand when presenting explanations [50].

2.3 Human Evaluation for Machine Learning Interpretability

Human-centered machine learning recognizes that machine learning work is inherently human work and explores the co-adaptation of humans and systems [55]. Therefore, artificial intelligence and machine learning systems should not only be developed with humans, but evaluated by humans. Unfortunately, the intrinsic probabilistic nature of machine learning models makes evaluation challenging. A taxonomy of evaluation approaches for interpretability includes application-grounded, human-grounded, and functionally grounded evaluations [19]. Our work in GAMUT falls into a human-grounded evaluation. Other studies have investigated the effectiveness of different explanations, taking initial steps toward identifying what factors are most important for providing human explanations [56]. Another study uses simulability as the main task that human subjects perform to compare the trust humans have in white-box and black-box linear regression models [57]. Using human trust as a metric of evaluation for the effectiveness of explanations has also been studied [18]. However, simulability and trust may not be ideal metrics to base evaluation on. An application-grounded evaluation for a pair of explainable machine learning interfaces deployed in the wild on a fraud detection team found that different explanation techniques yield widely varying results, yet are still considered reasonably valid and useful [58]. This is troublesome when in the case of incongruity domain experts were unaware of explanation disagreements and were eager to trust any explanation provided to them [58].

2.4 Neural Network Interpretability

Typically, a neural network is given an input data instance (e.g., an image) and computes transformations on this instance until ultimately producing a probability for prediction. Inside the network at each layer, each neuron (i.e., channel) detects a particular feature from the input. However, since deep learning models learn these features through training, research in interpretability investigates how to make sense of what specific features a network has detected. We provide an overview of existing activation-based methods for interpretability, a common approach to understand how neural networks operate internally that considers the magnitude of each detected feature inside hidden layers.

2.4.1 Understanding Neuron Activations

Neuron activations as features for interpretable explanations. There have been many approaches that use neuron activations as features for interpretable explanations of neural

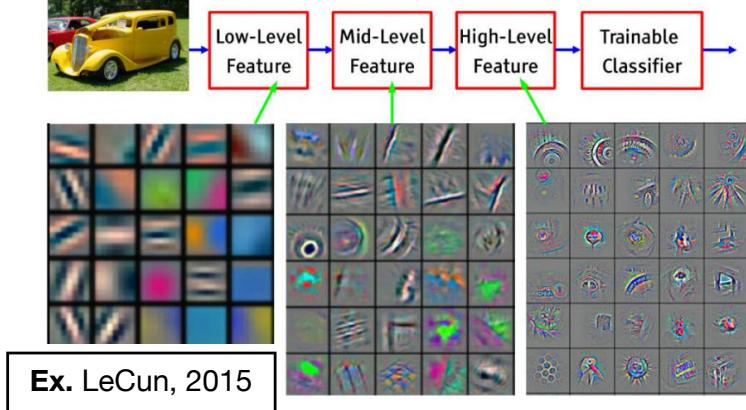


Figure 2.1: A common, widely shared example illustrating how neural networks learn hierarchical feature representations. Our work crystallizes these illustrations by systematically building a graph representation that describe *what* features a model has learned and *how* they are related. We visualize features learned at individual neurons and connect them to understand how high-level feature representations are formed from lower-level features. Ex. taken from Yann LeCun, 2015.

network decisions. TCAV vectorizes activations in each layer and uses the vectors in a binary classification task for determining an interpretable concept’s relevance (e.g., striped pattern) in model’s decision for a specific class (e.g., zebra) [59]. Network Dissection [60] and Net2Vec [61] propose methods to quantify interpretability by measuring alignment between filter activations and concepts. ActiVis visualizes activation patterns in an interactive table view, where the columns are neurons in a network and rows are data instances [39]. This table unifies instance-level and subset-level analysis, which enables users to explore inside neural networks and visually compare activation patterns across images, subsets, and classes.

Visualizing neurons with their activation. Instead of only considering the magnitude of activations, another technique called feature visualization algorithmically generates synthetic images that maximize a particular neuron [62, 63, 64, 65, 66, 67]. Since these feature visualizations optimize over a single neuron, users can begin to decipher what feature a single neuron may have learned. These techniques have provided strong evidence of how neural networks build their internal hierarchical representations [68]. Figure 2.1 presents widely shared examples of how neural networks learn hierarchical features by showing neuron feature visualizations. It is commonly thought that neurons in lower layers in a network learn low-level features, such as edges and textures, while neurons in later layers learn more complicated parts and objects. In our work, we crystallize this belief in SUMMIT by leveraging feature visualization to identify what features a model has detected, and how they are related.

2.4.2 Towards Higher-level Deep Learning Interpretation

It is not uncommon for modern, state-of-the-art neural networks to contain hundreds of thousands of neurons; visualizing all of them is ineffective. To address this problem, several works have proposed to extract only “important” neurons for a model’s predictions [67, 69, 70]. For example, Blocks, a visual analytics system, shows that class confusion patterns follow a hierarchical structure over the classes [71], and Activation Atlases, large-scale dimensionality reductions, show many averaged activations [67]. Both visualizations reveal interesting properties of neural networks. However, they either (1) consider activations independent of their learned connections, (2) depend on randomized sampling techniques, or (3) are computationally expensive. SUMMIT addresses these issues by: (1) combining both activations and relationships between network layers, as only knowing the most important neurons is not sufficient for determining how a model made a prediction—the relationships between highly contributing neurons are key to understanding how learned features are combined inside a network; (2) leveraging entire datasets; and (3) integrating scalable techniques.

Since feature visualization has shown that neurons detect more complicated features towards a network’s output, it is reasonable to hypothesize that feature construction is the collaborative combination of many different features from previous layers [72, 60, 61]. Our visualization community has started to investigate this hypothesis. For example, one of the earlier visual analytics approaches, CNNVis, derives neuron connections for model diagnosis and refinement, but did not scale to large datasets with many classes [73]. In the context of adversarial machine learning, AEVis uses backpropagation to identify where in a network the data paths of a benign and attacked instance diverge [69]. AEVis demonstrates its approach on single and small sets of images; it is unclear how the approach’s integral approximation optimization techniques scale to large, entire datasets, such as ImageNet. Another example, Building Blocks, proposes to use matrix factorization to group sets of neurons together within a layer and derive “compatible” neuron groups across layers [70]; however, the work suggests uncertainty in the proposed formulation. Our work draws inspirations from the above important prior research in neural network visualization. Our method introduced in SUMMIT makes advances to scale to large million-image datasets, providing new ways to interpret entire classes (vs. single-image explanations) by aggregating activations and influences across the model.

2.4.3 Visual Analytics for Neural Network Interpretability

To better facilitate interpretability, interactive visual analytics solutions have been proposed to help different user groups interpret models using a variety of interactive and visualization techniques. Predictive visual analytics supports experts conducting performance analysis of machine learning models by visualizing distributions of predicted instances, computing feature importance, and directly inspecting model and instance errors to support debugging [74, 31, 7, 35, 34]. Interactive visualization for explaining models to non-experts using direct manipulation has also seen attention due to the pervasiveness of machine learning in modern society and general interest from the public [75, 76, 77]. In Chapter 5, we present a first-of-its-kind survey of the role of visual analytics in deep learning that details dozens of systems using an interrogative structure [9]. This survey helps researchers and practitioners in both visual analytics and deep learning to quickly learn key aspects of this young and rapidly growing body of research, whose impact spans a diverse range of domains.

PART I

ENABLING MACHINE LEARNING

INTERPRETABILITY

Overview

Today, people excitedly apply machine learning to solve challenging and important problems that impact society at large. Knowing how these systems make decisions is fundamental to their fair, safe, and responsible use. While the broader community loosely describe this notion as machine learning interpretability, unfortunately there is no precise definition for what it means for a machine learning system to be interpretable. We argue this should not stop people from understanding how models make predictions and behave. Instead, we seek to **enable machine learning interpretability** through an operationalization to help practitioners better understand their models.

Part I begins by describing **GAMUT (Chapter 3)**, a novel interactive visualization interface that instantiates our machine learning interpretability operationalization and investigates the emerging practice of interpretability with professional machine learning developers. This chapter is adapted from work that was published and appeared at *CHI 2019* [7].

GAMUT: A Design Probe to Understand How Data Scientists Understand Machine Learning Models. ↗ Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, Steven M. Drucker. *ACM Conference on Human Factors in Computing Systems (CHI), 2019.*

Through our study, we find that depending on the complexity of the model (e.g., number of features), interpreting these visualizations can be difficult and may require additional expertise. Alternatively, textual descriptions, or verbalizations, can be a simple, yet effective way to communicate or summarize key aspects about a model, such as the overall trend in a model’s predictions or comparisons between pairs of data instances. With the potential benefits of visualizations and verbalizations in mind, we extend GAMUT to another novel interface, **TELEGAM (Chapter 4)**, to explore how the two can be combined to aid machine learning interpretability. This chapter is adapted from work that was published and appeared at *VIS 2019* [8].

TELEGAM: Combining Visualization and Verbalization for Interpretable Machine Learning. ↗ Fred Hohman, Arjun Srinivasan, Steven M. Drucker. *IEEE Visualization Conference (VIS), 2019.*

CHAPTER 3

GAMUT: UNDERSTANDING HOW DATA SCIENTISTS UNDERSTAND MACHINE LEARNING

Without good models and the right tools to interpret them, data scientists risk making decisions based on hidden biases, spurious correlations, and false generalizations. This has led to a rallying cry for model interpretability. Yet the concept of interpretability remains nebulous, such that researchers and tool designers lack actionable guidelines for how to incorporate interpretability into models and accompanying tools. Through an iterative design process with expert machine learning researchers and practitioners, we designed a visual analytics system, GAMUT, to explore how interactive interfaces could better support model interpretation. Using GAMUT as a probe, we investigated why and how professional data scientists interpret models, and how interface affordances can support data scientists in answering questions about model interpretability. Our investigation showed that interpretability is not a monolithic concept: data scientists have different reasons to interpret models and tailor explanations for specific audiences, often balancing competing concerns of simplicity and completeness. Participants also asked to use GAMUT in their work, highlighting its potential to help data scientists understand their own data.

3.1 Introduction

With recent advances in machine learning [78, 79, 80, 81], people are beginning to use ML to address important societal problems like identifying and predicting cancerous cells [82, 83], predicting poverty from satellite imagery to inform policy decisions [84], and locating buildings that are susceptible to catching on fire [85, 86]. Unfortunately, the metrics by which models are trained and evaluated often hide biases, spurious correlations, and false generalizations inside complex, internal structure. These pitfalls are nuanced, particularly to novices, and cannot be diagnosed with simple quality metrics, like a single accuracy number [87]. This is troublesome when ML is misused, with intent or ignorance, in situations where ethics and fairness are paramount. Lacking an explanation for how models perform can lead to biased and ill-informed decisions, like representing gender bias in facial analysis systems [1], propagating historical cultural stereotypes in text corpora into widely used AI components [2], and biasing recidivism predictions by race [3]. This is the problem of *model interpretability*.

Although there is no formal, agreed upon definition of model interpretability [20], existing research focuses on human understanding of the model representation [15, 17, 16, 21, 18]. Government policy makers are also joining the discussion through the recent General Data Protection Regulation (GDPR) requirements [23]. Articles 13 and 22 state a “right to explanation” for any algorithm whose decision impacts a person’s legal status [23]. Within the newly risen explainable artificial intelligence field, tools for interpretability have used information visualization [44] as a medium for explanation [9, 45, 31, 32] since they excel at graphical communication for complex ideas and meaningful summarization of information [46]. When considering what to visualize, two competing paradigms have emerged: global and local explanations. *Global explanations* roughly capture the entire space learned by a model in aggregate, favoring simplicity over completeness. Conversely, *local explanations* accurately describe a single data instance’s prediction.

In this work, we take a human-centered approach to studying model interpretability. Through an iterative design process with expert machine learning researchers and practitioners at a large technology company, we designed GAMUT, an interactive visual analytics system for model exploration that combines both global and local explanation paradigms. Using GAMUT as a probe into interpretability, we conducted a user study to investigate why and how data professional data scientists interpret models and how interface affordances support data scientists in answering question about model interpretability. In designing our probe, we sought a balance between low graphicacy skills needed to learn about the model and a high level of accuracy so that users of the probe would trust its predictions were accurate and realistic. Therefore, we ground our research on a class of models, called generalized additive models (GAMs) [88], that perform competitively to state-of-the-art models yet contain a relatively simple structure [5, 89, 90, 91]. The study included 12 professional data scientists with ranging levels of expertise in machine learning. Our investigation shows that interpretability is not a monolithic concept: data scientists have different reasons to interpret models and tailor explanations for specific audiences, often balancing the competing concerns of simplicity and completeness. We also observed that having a tangible, functional interface for data scientists helped ground discussions of machine learning interpretability. Participants also asked to use GAMUT in their work, highlighting its potential to help data scientists understand their own data. In this work, our contributions include:

- **A human-centered operationalization of model interpretability.** We contribute a list of capabilities that explainable machine learning interfaces should support to answer interpretability questions.

- **An interactive visualization system for generalized additive models (GAMs).**
GAMUT, an interactive visualization system built for exploring and explaining GAMs, iteratively designed with machine learning professionals.
- **A design probe evaluation with human subjects.** Results from a user study with professional data scientists using GAMUT as a design probe for understanding interpretability.

We hope the lessons learned from this work help inform the design of future interactive interfaces for explaining more kinds of models, including those with natural global and local explanations (e.g., linear regression, decision trees), as well as more complex models (e.g., neural networks).

3.2 Design Rationale

3.2.1 A Technology Probe for Model Interpretability

A technology probe is an “instrument that is deployed to find out about the unknown—returning with useful or interesting data,” and should balance three broad goals: *design*: inspire reflection on emerging technologies; *social science*: appreciate needs and desires of users; and *engineering*: field-testing prototypes [92]. Technology probes are a common approach for contextual research in human-computer interaction that invite user participation [93, 94].

While building and deploying ML models is now a standard software practice, interpreting models is not. We therefore use a technology probe to understand this emerging practice, balancing these three goals:

- *Engineering*: we iteratively developed an explainable interface that works on real data and models.
- *Social science*: we used qualitative methods for data collection to learn about data scientists’ behavior during an in-lab user study and quantitative measures for a preliminary usability assessment.
- *Design*: the visualization prototype inspired participants to reflect on interpretability and how they use it in their own work.

3.2.2 Assessing the Probe's Features

We took two approaches to design a visualization system to probe machine learning interpretability. First, we performed a literature survey to compare the many definitions of what makes a machine learning model interpretable. We focused on recent work that postulates interactive explanations will be key for understanding models better, as summarized in ?? . Second, we conducted a formative study through a series of interviews with both machine learning researchers and practitioners to gather questions a user should be able to ask a machine learning model or AI-powered system. The participants included 4 senior ML researchers and 5 ML practitioners (3 female and 6 male), who were recruited based on their expertise in ML and their interest in ML interpretability. Together, we synthesized our findings into the following list of capabilities that an explainable machine learning interface should support. While there is no guarantee of completeness, we, the authors and participants, find this list to be effective for operationalizing interpretability in explainable ML interfaces. Each capability provides an example interpretability question, which all reference a real-estate model that predicts the price of homes given the features of a house.

C1. Local instance explanations. PREDICTION

Given a single data instance, quantify each feature’s contribution to the prediction.

Example: Given a house and its predicted price of \$250,000, what features contributed to its price?

C2. Instance explanation comparisons. PREDICTION

Given a collection of data instances, compare what factors lead to their predictions.

Example: Given five houses in a neighborhood, what distinguishes them and their prices?

C3. Counterfactuals. PREDICTION

Given a single data instance, ask “what-if” questions to observe the effect that modified features have on its prediction.

Example: Given a house and its predicted price of \$250,000, how would the price change if it had an extra bedroom?

Example: Given a house and its predicted price of \$250,000, what would I have to change to increase its predicted price to \$300,000?

C4. Nearest neighbors. DATA

Given a single data instance, find data instances with similar features, predictions, or both.

Example: Given a house and its predicted price of \$250,000, what other houses have similar features, price, or both?

Example: Given a house and a binary model prediction that says to “buy”, what is the most similar real home that the model predicts “not to buy”?

C5. Regions of error. MODEL

Given a model, locate regions of the model where prediction uncertainty is high.

Example: Given a house price prediction model trained mostly on older homes ranging from \$100,000 - \$300,000, can I trust a model’s prediction that a newly built house costs \$400,000?

C6. Feature importance. MODEL

Given a model, rank the features of the data that are most influential to the overall predictions.

Example: Given a house price prediction model, does it make sense that the top three most influential features should be the square footage, year built, and location?

3.2.3 Selecting the Probe’s Model Class

Given the set of capabilities we uncovered during our formative study, our probe should work with a class of ML models having many ideal characteristics:

- The model should have a simple enough structure to allow the user to see the model globally.
- Understanding the model’s computation should require average math skills, to support non-expert users.
- Similarly, visualizing the model’s structure should require average graphicacy, i.e., data visualization literacy.
- The model should be compositional, so that the effect of features can be understood in isolation.
- The model should have high accuracy, so that deploying it is realistic.

Of course, no single class of model can be optimal for all these attributes [43]. For example, simpler models, like linear regression and decision trees, have simple global structure, but suffer from poor accuracy; more complex models, like deep neural networks, achieve superior performance at the cost of complex structure and lack of clear compositionality [89,

24, 18]. Our choice of model for the probe therefore represents a compromise among these criteria.

In essence, we sought a balance between low graphicacy skills needed to learn about the model and a high level of accuracy so that users of the probe would trust its predictions were accurate and realistic. One particular model class, the *generalized additive model* (GAM) [88], has recently attracted attention in the ML community. Thanks to modern ML techniques such as boosting [95], GAM performance on predictive tasks on tabular data competes favorably with more complex, state-of-the-art models, yet GAMs remain intelligible and more expressive than simple linear models [89, 90, 91]. Understanding a GAM requires only the ability to read a line chart. A GAM has a local explanation similar to linear regression, but also lends itself to a global explanation (shape function charts, described later), which other models lack; this allows us to test the relative value users place on having global understanding versus a purely local understanding of a model.

GAMs are a generalization of linear models. To illustrate the difference, consider a dataset $D = \{(\mathbf{x}_i, y_i)\}^N$ of N data points, where $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iM})$ is a feature vector with M features, and y_i is the target, i.e., the response, variable. Let x_j denote the j th variable in feature space. A typical linear regression model can then be expressed mathematically as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_N x_N$$

This model assumes that the relationships between the target variable y_i and features x_j are *linear* and can be captured in slope terms $\beta_1, \beta_2, \dots, \beta_N$. If we instead assume that the relationship between the target variable and features is *smooth*, we can write the equation for a GAM [88]:

$$y = \beta_0 + f_1(x_1) + f_2(x_2) + \cdots + f_N(x_N)$$

Notice here that the previous slope terms $\beta_1, \beta_2, \dots, \beta_N$ have been replaced by smooth, shape functions f_j . In both models β_0 is the model intercept, and the relationship between the target variable and the features is still additive; however, each feature now is described by one shape function f_j that can be nonlinear and complex (e.g., concave, convex, or “bendy”) [96].

Since each feature’s contribution to the final prediction can be understood by inspecting the shape functions f_j , GAMs are considered intelligible [89]. In this work, we omit the details of how to train GAMs, mean center shape functions, and distinguish their regression and classification versions, which are covered in the literature [97, 98, 90, 91]. We also note that GAM shape function charts differ from partial dependency (PD) [99] used in [47, 48]. PD assumes that features are uncorrelated, and PD averages over the other features not

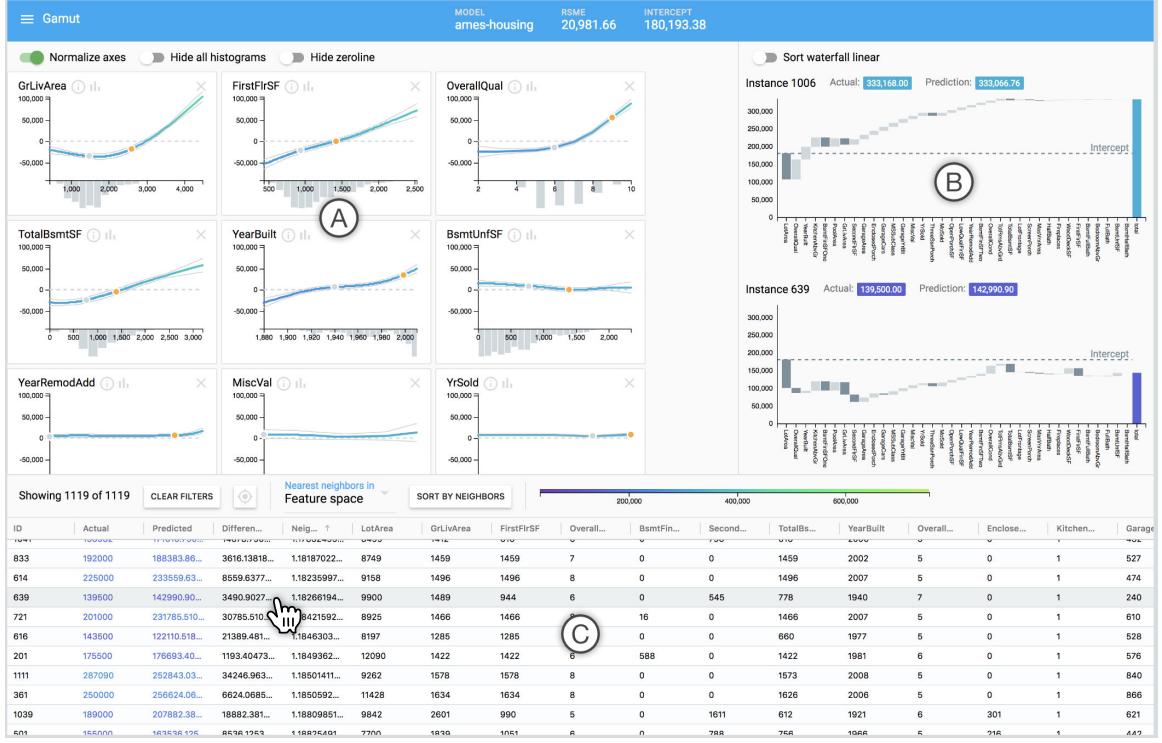


Figure 3.1: The GAMUT user interface tightly integrates multiple coordinated views. (A) **The Shape Curve View** displays GAM shape functions as line charts, and includes histograms of the data density for each feature. The charts can be normalized to better compare the impact each shape function has on the model. (B) **The Instance Explanation View** displays a waterfall chart for two data instances. Each chart encodes the cumulative impact each feature has on the final prediction for one data instance. (C) **The Interactive Table** displays the raw data in an interactive data grid where users can sort, filter, and compute nearest neighbors for data instances.

included in the chart. Therefore, PD only captures the effect of modifying one feature independent of the others, whereas GAM shape function charts, which are trained in parallel, are effectively the entire model—predictions are made by summing values from all charts together and take into account correlation among features to prevent multiple counting of evidence. All together, this makes GAMs uniquely suited as a model that maximizes our previous criteria and ties global and local explanations closely together.

3.3 GAMUT

Given the capabilities described in Section 3.2, we present GAMUT, an interactive visualization system that tightly integrates three coordinated views to support exploration of GAMs (Figure 3.1): the Shape Curve View (A); the Instance Explanation View (B); and the Interactive Table (C). To explain these views, we use an example real-estate model that uses a house’s features to predict its sale price in US dollars. The three views show different

aspects of a user-selected instance, in this case a chosen house. Throughout the description we link features to the capabilities **(C1)–(C6)** that the features support.

3.3.1 Shape Curve View

The Shape Curve View displays each feature’s shape function as a line chart (Figure 3.1A). The user can choose which features are displayed through the Feature Sidebar (Figure 3.2A): an ordered list the features of the data, sorted by importance to the model **(C6)**. We will first describe the encoding for one shape function chart. Consider the *OverallQual* feature and its shape function chart (Figure 3.2B). This chart shows the impact that the *OverallQual* feature has on the overall model predictions **(C6)**. The x-axis is the dimension of the feature, in this case, a rating of the house’s overall material and finish quality, between 2 and 10; the y-axis is the contribution of the feature to the output of a prediction, in this case, US dollars. The chart shows that having a rating of 9 adds \$50,000 to the predicted price, for example. Below the x-axis is a histogram of the data density for the dimension. This is useful for determining how many data points exists in a particular part of feature space **(C5)**, e.g., in Figure 3.2B, we see that most houses have a *OverallQual* of 5 to 8.

The selected instance’s specific feature values are shown as amber points on the shape function charts **(C1)**. A data instance has one value for every feature, i.e., one amber point on each shape function chart, which shows where the selected instance is located in the global model **(C5)**. The color of the line for each shape function encodes the final predicted value if we were to vary the selected amber point’s value to all other possible values. This is reinforced when a user brushes over a line chart: a new point, colored by its final prediction, is shown on the shape function curve, while projected crosshairs track with the mouse cursor, enabling users to ask interactive counterfactuals for any feature **(C3)**.

Since the Shape Curve View shows multiple shape function charts at once, we provide a Normalize toggle for accurate comparison. Turning Normalize on plots all the shape functions on a common scale, allowing visual comparison of the features’ different degrees of impact on the predictions. Charts with high slopes indicate more impact on predictions, whereas charts with relatively flat lines contribute only a little **(C6)**. Turning off Normalize plots each chart on its own scale, emphasizing the shape of low-impact (flat) features.

3.3.2 Instance Explanation View

The Instance Explanation View shows a visualization of individual instance predictions (Figure 3.1B) **(C1)**. A GAM converts each feature value of a data instance into its direct contribution on the final prediction. Since GAMs are additive models, to obtain a prediction

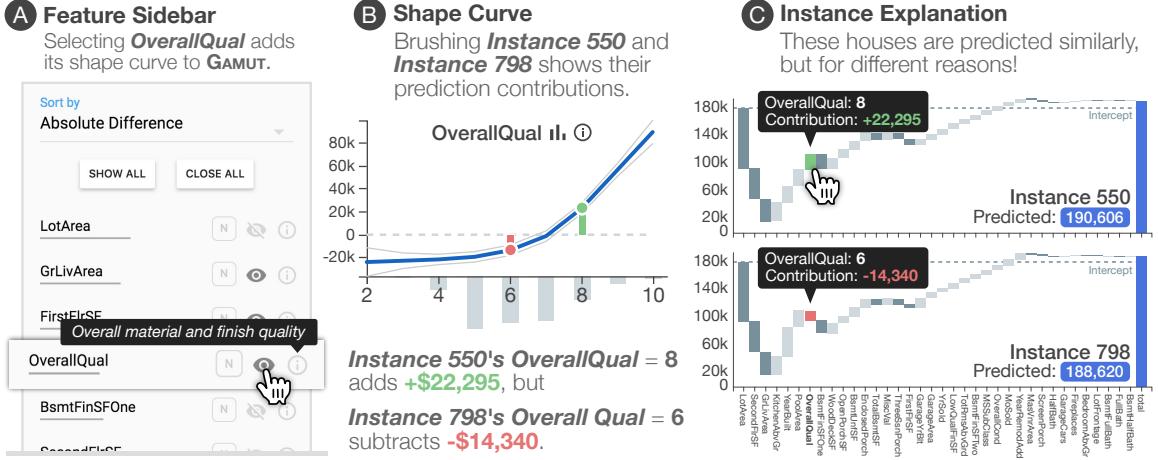


Figure 3.2: Interacting with GAMUT’s multiple coordinated views together. (A) Selecting the *OverallQual* feature from the sorted Feature Sidebar displays its shape curve in the Shape Curve View. (B) Brushing over either explanation for *Instance 550* or *Instance 798* shows the contribution of the *OverallQual* feature value for both instances. (C) Notice these two houses are similarly predicted (**\$190,606** and **\$188,620**), but for different reasons!

for a single data instance with M features, we compute the amount each feature contributes to the total prediction and add them all up. We also add the intercept (the average predicted value for the dataset), for a total of $M + 1$ values. The Instance Explanation View shows these $M + 1$ values as a waterfall chart (**C1**). The x-axis is a categorical axis of all the features, and the y-axis is the final prediction. These values can be positive or negative, as indicated by the dark and light gray shades of each of piece of the waterfall chart. The x-axis is sorted by the absolute value of each feature’s contribution; the leftmost values drive the majority of the overall prediction. For example, consider the waterfall chart in Figure 3.2C for Instance 550. From the colored tag, we see this house was predicted as costing **\$190,606**. We also see the first three features greatly reduce the price of the house (three dark gray rectangles), but the next four increase the price. Another interesting characteristic is the long tail of features towards the end of the waterfall chart; a single feature value hardly contributes to the over prediction alone, but together the small contributions account for a non-trivial amount of the final prediction.

The Instance Explanation View also allows easy comparison of multiple instances (Figure 3.2C). The first chart is the selected instance, which is pinned to the interface. This selected instance’s values are the same amber dots in the Shape Curve View. The second chart visualizes a different instance that updates as the user brushes over a different data instance from the Interactive Table, described in the next subsection. Since two instance predictions could have a different x-axis ordering, we impose the ordering of the selected instance on the second instance. Combined with automatically normalizing both y-axes for

the two waterfall charts, this enables direct comparison of both waterfall charts (**C2**).

Brushing over either waterfall chart provides several cues to aid comparison: a tooltip with the exact feature value and GAM contribution for both waterfall charts (Figure 3.2C) (**C1**); highlights in the corresponding shape function charts in the Shape Curve View; and plotting both points on the shape function charts (Figure 3.2B) (**C2**). The two instances, i.e., houses, shown in Figure 3.2C are close in predicted price,

(**\$190,606** and **\$188,620**), and have similar shapes in their waterfall charts. However, Instance 550 has an *OverallQual* of 8 which adds **+\$22,295** to the prediction cost; whereas, Instance 798 has a *OverallQual* of 6 which reduces the cost **-\$14,340**. While a few other values must differ to make up for this particular difference, we have found two houses that are predicted with similar prices, but achieve those prices by different means (**C4**).

3.3.3 Interactive Table

The Interactive Table is a scrollable data grid of the raw data used to train the model (Figure 3.1C). The rows of the data grid are individual data instances, and the columns are the features, plus five additional columns on the left: Instance ID; Actual value (or label) of the data instance; Predicted value (or label) of the data instance; Difference between actual and predicted value; and Nearest Neighbor Distance from the selected instance. The column headers provide familiar data grid features, like resizing, sorting, and filtering columns.

Brushing over a row in the Interactive Table updates the second waterfall chart in the Instance Explanation View and normalizes both waterfall charts to ensure direct comparison between the two visualized instances is accurate (**C2**). Brushing over a row also plots that instance's values on the Shape Curve View as gray points to compare against the selected instance's amber points described above (**C2**).

3.3.4 Implementation

GAMUT is a client-side web app, using D3 [100] for visualization and ag-Grid¹ for the data grid. We pre-train our GAMs in Python using the pyGAM [98] package. pyGAM uses splines to fit the GAM shape curves; however, more advanced techniques exist for training GAMS as cited in Section 3.2.

¹<https://www.ag-grid.com/>

3.4 User Study

We used GAMUT as a design probe during an in-lab study to understand how data scientists understand machine learning models and answer interpretability questions. We aimed to answer the following research questions:

- RQ1.** Why do data scientists need interpretability and how do they answer interpretability questions?
- RQ2.** How do data scientists use global explanations and local explanations?
- RQ3.** How does interactivity play a role in explainable machine learning interfaces?

3.4.1 Participants

We invited 200 randomly selected professional data scientists at a large technology company and received 33 replies (17% response rate). We selected 12 participants (7 female, 5 male), all with bachelor’s degrees, 6 with graduate degrees. Half of the participants had only 1 year of experience with ML, while the other half had at least 3 years, with two participants having more than 5 years. One participant uses ML on a daily basis, five on a weekly basis, while the other six use ML less often. Ten of the participants reported they use visualization in their work, mostly dashboard-style analytics. Nine participants reported using tabular data in their own work. Six of participants reported that they have used explanations for models before; five said their explanations were static, with only one reporting their explanation being interactive. We compensated participants with a \$25 Amazon Gift card.

3.4.2 Study Design

The study duration was 1½-hours per participant. To start, each participant signed a consent form and filled out a background questionnaire. The session then consisted of a GAMUT tutorial, with a model that predicts the price of 1,000 diamonds, based on 9 features.

Participants thought aloud while using GAMUT to explore two models, one regression and one binary classification. Participants were free to choose one of three regression models that predict: the price of 506 houses in Boston, Massachusetts, based on 13 features (6 chose this); the price of 1,119 houses in Ames, Iowa, based on 36 features (5 chose this); or the quality of 1,599 wines, based on 11 features (1 chose this). Similarly, participants were free to choose one of three binary classification models that predict: the survival of 712 Titanic passengers, based on 7 features (4 chose this); heart disease in 261 patients,

based on 10 features (5 chose this); or diabetes in 392 patients, based on 8 features (2 chose this).

Once a participant chose a dataset, we provided them with the feature names and their textual descriptions. We then gave them 5 minutes to brainstorm their own hypotheses about the model, using their own intuition. We then allowed them to use GAMUT to explore the model, guided by a list of questions we provided (≈ 10 per dataset) that exercise GAMUT’s capabilities, ordered so that adjacent questions test different capabilities. All participants completed all the questions for one model in the allotted time, around 15 minutes. If they had not already addressed their initial questions, we returned to them to see if they were able to after. We then repeated this process for the second dataset. Each session ended with a usability questionnaire and an exit interview that asked participants to reflect on their process of explaining ML models in their own work, their process of using GAMUT, and if GAMUT could be useful for them.

3.5 Results

Every participant was successful at answering both their own and our prepared questions about the different models, despite being new to GAMs and GAMUT. We also observed that having a tangible, functional interface for data scientists helped ground the discussion of interpretability. In the following sections we summarize the results from our study, both during the participant usage of GAMUT and the conversations during the exit interviews.

3.5.1 RQ1: Reasons for Model Interpretability

Hypothesis generation. As participants used GAMUT, they constantly generated hypotheses about the data and model while observing different explanations. This was insightful, since after only a brief tutorial, the participants were comfortable answering a variety of questions about the models and started to reason about them in ways they could not before. We also noticed that participants were using the model to confirm prior beliefs about the data, slowly building trust that the model was producing accurate and believable predictions. However, participants were eager to rationalize explanations without first questioning the correctness of the explanation itself. While forming new hypotheses about one’s data and model can lead to deeper insight, this could be troublesome when participants trust explanations without healthy skepticism. While these results corroborate existing literature [49, 58], it suggests further studies to evaluate human trust in model explanations.

Data understanding. Participants also used interpretability as a lens into data, which prompted us to ask participants about this during the exit interviews. While a predictive model has its own uses, e.g., inference and task automation, many participants explained that they use models to gain insight into large datasets, as mentioned in [101]. One participant said, “*It’s more like a data digging process. So it’s finding the important features to help us understand the data better.*” While there are many academic and commercial tools for data exploration without statistical models, a model-based approach gave participants a new perspective on the data. About GAMUT, one participant said, “*This would help me and expedite my workflow to get to valuable nuggets of information, which is what [my stakeholders] are ultimately interested in.*” Related, another reason that emerged from the interviews was that data scientists use interpretability to understand the feature importance of a dataset. Most of our participants said that computing a metric (for which there are many) for feature importance across all features provides valuable information about what characteristics of a dataset are most important for making predictions. This allows data scientists to focus on accurately representing these features in a model. With regards to learning representations, a few participants said that interpretability also ensures customer privacy is upheld, by discovering what features are correlated with identifiable information so they can be removed.

Communication. Throughout the study, the prepared questions asked participants to communicate their process of discovering the answers. During the exit interviews, nearly every participant described a scenario in which they were using model explanations to communicate what features were predictive to stakeholders who wanted to deploy a model in the wild. One participant noted that “*different audiences require different explanations,*” describing a common trade-off between explanation simplicity and completeness. This was further supported by a participant who frequently presents reports to stakeholders: “*When you’re going to craft your story, ...you’re going to have to figure out what you want emphasize and what you want to minimize. But you have to always lay out everything. Know your audience and purpose.*” She also emphasized that she encourages fellow data scientists on her team to share knowledge about what they have learned to other non-scientists. Lastly, a participant said she uses explainable data analysis to change organizational behavior on her team, by using models to inspect and understand data quality. She described how some analysts claim they can predict a value, but neglect to explain why, which diminishes the impact: “*What are the features? How are you getting those features? What are the quality of those features? They’re just literally saying, ‘I’m forecasting the number—here’s the number you use.’ I’m going, ‘That just is not satisfying.’*” By using feature importance met-

rics, she ensures that the important features of data are accurately collected, recognizing that “clean” data creates better models.

Model building. Participants who have experience in developing models recognize that interpretability is also critical to model builders. Understanding characteristics about one’s data and model helps guide model improvement. Regarding the intelligibility versus accuracy trade-off, one participant said that he starts his work using simpler models to become familiar with the data, before moving onto more complex models. Having a solid understanding of one’s data is more important than incrementally improving model accuracy: “*I want to understand bit by bit how the dataset features work with each other, influence each other. That is my starting point.*” Another participant said his team uses two natural language processing models in production: a simpler, rule-based model that performs multiple checks before inference; if the checks pass, the data is passed to the more complex model for a final prediction.

3.5.2 RQ2: Global versus Local Explanation Paradigms

While using GAMUT, every participant used both the global and local explanations to answer interpretability questions, often moving between the two. This shows that global and local explanation paradigms are in fact complementary. Participants used the shape function charts of the model to explain a feature of the dataset, but grounded the explanation with local context using the data histogram. Conversely, participants described single-instance explanations using the global context of the shape function charts, i.e., overlaying the amber points of a waterfall chart on shape function charts. One participant said, “*If I want to see what the overall ecosystem is doing, [global explanations are] significantly better. If I wanted to find specific use cases that are interesting, then I’m going to use [local explanations] as case studies. So, I see it as having both.*”

Broadly speaking, we noticed the expertise of a participant correlated with which explanation paradigm they preferred: (1) the ML novices gravitated towards the local explanations, (2) more expert participants used global explanations more frequently, and (3) the most expert participants fluidly used both to reason about a prediction and a model. For example, a common practice in ML is to consider only the top features, since likely those are driving the prediction. However, one participant noticed that the visualizations in the Instance Explanation View argued otherwise—the long tail of a waterfall chart sometimes contributed a non-trivial percentage of a prediction—and observed that the top features were insufficient. This is an interesting example of how a local explanation can inform a global characteristic of a model.

The Interactive Table was a critical mechanism for linking global and local explanations. Participants frequently sorted columns (i.e., features) to see how data aggregates along a single feature, but also inspected many single data instances for exact feature values; to our surprise, sorting by nearest neighbors was only used a couple times per participant. Some participants were initially confused about whether a particular visualization was describing global or local model behavior (e.g., mistaking a waterfall chart to describe the global behavior of a model instead of a single data instance), suggesting that either the initial tutorial could be improved, or that the level of graphicacy required for GAMUT was higher than anticipated; regardless, by the end of every 1½-hour session, it was clear all participants understood how GAMUT’s representations connected together.

3.5.3 RQ3: Interactive Explanations

When choosing a model explanation, regardless of the type (e.g., textual, graphical), most explanations are static. Only recently has the notion of *interactive explanations* attracted attention. In GAMUT, interactivity refers to instance-based selection, brushing and linking between local and global views, quick comparison of instances and their explanations, sorting and filtering the Interactive Table, hovering over a shape function chart for asking counterfactuals, and computing nearest neighbors for a single instance.

Throughout the studies it became clear that interactivity was the primary mechanism for exploring, comparing, and explaining instance predictions and the chosen models by the participants. Interactivity was so fundamental for our participants’ understanding of the models, that when we prompted them to comment on interactivity, people could not conceive non-interactive means to answer both their hypotheses and prepared questions, even though the current best practice for understanding GAMs entails flipping through static print outs of all the shape function charts.

Participants liked the interactivity of GAMUT, but we think there is potential to alleviate redundant interactions by incorporating automated insight discovery techniques in explanation systems. Examples include algorithmically surfacing the most accurate explanations and finding the most relevant data (e.g., similar neighbors, counterfactuals) given interpretability-focused constraints.

Participants also suggested several additional features. First, while GAMUT supports comparing two instance explanations at once, participants wanted to compare multiple groups of instances (e.g., user-defined groups, or a group of nearest neighbors); they also wanted deeper comparison, such as changing the visual representation to a stacked bar chart to more easily compare the contributions of multiple instance by feature. Second, the more expert participants wanted more support for feature selection and importance, such as leav-

ing one feature out of the model and seeing its effect on performance. Lastly, we noticed most participants used counterfactuals often throughout their exploration, both as a direct task and as a sanity check for feature sensitivity; therefore, there could be opportunities to support automatic counterfactual identification in combination with computing nearest neighbors to enable data scientists to understand models faster and more confidently.

3.5.4 Usability

The exit questionnaire included a series of Likert-scale (7 point) questions about the utility and usefulness of the various views in GAMUT (Figure 3.3). From the high ratings, we are confident that GAMUT’s role as a design probe was not hampered by usability problems. Similarly, the uniformity of the feature ratings suggests that participants did not disfavor any particular feature because of a usability problem.

Even though GAMUT was designed as a probe, all 12 participants desired to use it to understand their own data. Some participants suggested using the system in its entirety, while others wanted to use specific parts of the interface, such as the Instance Explanation View, to include in reports to their stakeholders. One participant who frequently uses visual analytics tools said, “*I really like that it’s splitting out each of the individual features into its own chart. ...I can’t tell you how useful that is for me. Parameterizing dimensions is just not available with Tableau, Power BI, or anything else.*” Another participant wanted to use GAMUT to not only predict when customers would renew a product subscription, but to understand why and how they renew. A participant who frequently engages with legal discourse suggested a potential user for GAMUT that we had not considered: “*I definitely would use something like this, especially when it comes to privacy issues. I even would show this to lawyers.*” Several participants have followed up after the conclusion of the study and actively pursued using GAMUT in their teams with their own data.

3.6 Limitations

GAMUT only visualizes one class of ML model. While GAMUT’s design rationale, visualizations, and interactions were informed by multiple interviews and collaboration with ML researchers and practitioners, there could be an another complementary view that could have elicited better qualitative results during our user study. Regardless, to the best of our knowledge there is no existing interactive interface for GAMs. We think GAMUT is a useful interface for exploring GAMs, as supported by our usability ratings in Section 3.5 and participants desire to use GAMUT for their own work, perhaps by using GAMs to explain more complex models, as discussed in the following section.

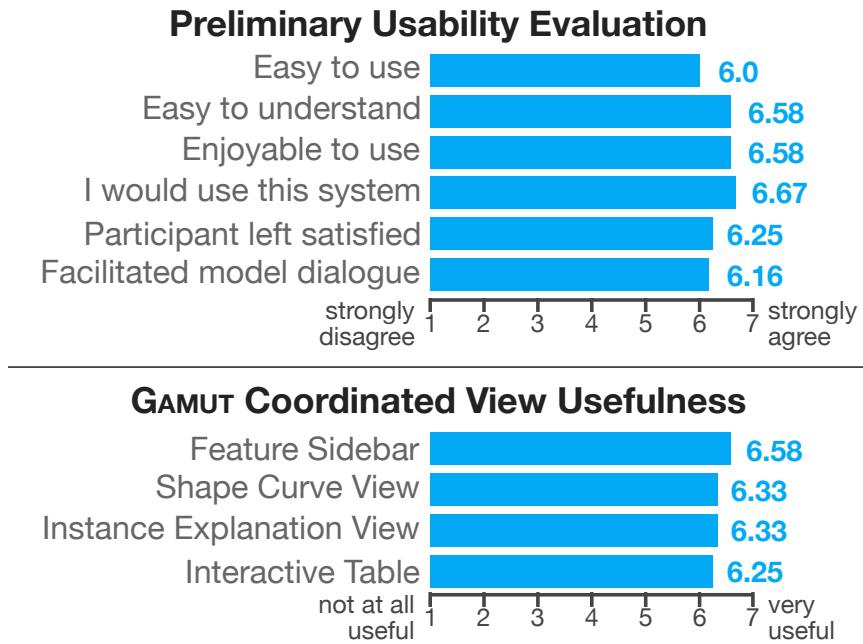


Figure 3.3: GAMUT subjective ratings. In a preliminary usability evaluation, participants thought GAMUT was easy to use and enjoyable. Of GAMUT’s multiple coordinated views, all were rated favorably. This also supports our finding that both global and local explanations are valuable for understanding a model’s behavior.

Understanding a model’s domain likely provides an advantage to understanding how a model works. Different participants entered the study with different domain knowledge. To mitigate this risk, we both provided a variety of models about approachable topics and allowed participants to choose the models that fit their own knowledge and expertise.

More technically, one participant with a PhD in statistics was concerned about correlated features and suggested that interaction terms should be considered.

3.7 Conclusion

In this work, through an iterative design process with expert machine learning researchers and practitioners at a large technology company, we identified a list of explainable machine learning interface capabilities, designed and developed an interactive visualization system, GAMUT, that embodied our capabilities, and used it as a design probe for machine learning interpretability through a human-subjects user study. Our results show that data scientists have many reasons for interpretability, answer interpretability questions using both global and local explanations, and like interactive explanations. GAMUT’s tightly interactive coordinated views enabled deeper understanding of both models and predictions. All participants wanted to use GAMUT on their own data in the course of their every day

work. From our study, it is clear there is a pressing need for better explanatory interfaces for machine learning, suggesting that HCI, design, and data visualization all have critical roles to play in a society where machine learning will increasingly impact humans.

CHAPTER 4

TELEGAM: COMBINING VISUALIZATION AND VERBALIZATION FOR INTERPRETABILITY

While machine learning continues to find success in solving previously-thought hard problems, interpreting and exploring ML models remains challenging. Recent work has shown that visualizations are a powerful tool to aid debugging, analyzing, and interpreting ML models. However, depending on the complexity of the model (e.g., number of features), interpreting these visualizations can be difficult and may require additional expertise. Alternatively, textual descriptions, or verbalizations, can be a simple, yet effective way to communicate or summarize key aspects about a model, such as the overall trend in a model’s predictions or comparisons between pairs of data instances. With the potential benefits of visualizations and verbalizations in mind, we explore how the two can be combined to aid ML interpretability. Specifically, we present a prototype system, TELEGAM, that demonstrates how visualizations and verbalizations can collectively support interactive exploration of ML models, for example, generalized additive models (GAMs). We describe TELEGAM’s interface and underlying heuristics to generate the verbalizations. We conclude by discussing how TELEGAM can serve as a platform to conduct future studies for understanding user expectations and designing novel interfaces for interpretable ML.

4.1 Introduction

While machine learning continues to find success in solving previously-thought hard problems with data, its pitfalls, such as encoding and perpetuating cultural and historical data bias inside complex models [3, 2, 1], have been the subjects of critical discussion surrounding its appropriate and ethical use [27, 28, 29]. In fact, governmental policy has been put in place, giving people a “right to explanation” for any model prediction that could impact their financial or legal status [23]. To understand how models learn and behave, interpretable, or explainable, artificial intelligence research has seen intense focus and progress [43]. Within this field, interactive data visualization has been used as a medium for communicating explanations for both models and predictions, allowing data scientists to better understand and debug their models [45, 9, 31, 32].

Previous work has shown that data scientists explain model results continuously to other groups of people: management, technical peers, and other stakeholders with invested inter-

est in an ML model or product [7]. However, often at the core of explainable ML sits an inherent trade-off between the completeness and simplicity of an explanation. To explain a single data instance’s prediction from a complex model with hundreds of features often requires significant effort that could consist of creating and interpreting many visualizations. These explanatory visualizations can also require high graphicacy, i.e., visualization literacy, from the people that create and use them for model iteration and decision making; this results in significant time and effort needed to understand what a visualization is showing and what is most important.

Alternatively, text or natural language has also been used as a medium to communicate model results and predictions [102, 50]. Text is useful for providing short, approximate explanations that provide most of the necessary information to understand a prediction without the cognitive burden of digesting a visualization. Natural language explanations could complement explanatory visualizations by helping people identify or verify inferences derived from a chart, identify prediction contributions they might have missed, or emphasize differences between predictions for multiple instances. However, systems combining both visual and natural language explanations for ML models remain largely underexplored. In this work, we investigate how system generated natural language explanations, or “verbalizations,” can complement explanatory visualizations. Such interfaces that combine visualizations and verbalizations could help data scientists better understand and debug their ML models, and aid them when communicating modeling results to other stakeholders. For example, systems could present verbalizations that are related to but not immediately observable in a visualization to help data scientists pivot between visualizations exploring different aspects of their models (e.g., global model-level explanations vs. local instance-level predictions) or drill-down into a model’s performance for specific instances (e.g., comparing predictions for two data instances).

To explore such possibilities, we extend recent work by Hohman et al. [7] on operationalizing model interpretability and contribute a prototype system demonstrating the potential of combining visualization and verbalization for explaining ML results. The system, TELEGAM, automatically generates natural language statements, or verbalizations, to complement explanatory visualizations for generalized additive models (GAMs). Incorporating the increasingly popular notion of interactively linking text and visualizations [103, 104, 105, 53], TELEGAM also lets users interact with verbalizations to visually manifest them in the explanatory visualization through simple annotations. By doing so, TELEGAM demonstrates how interfaces could better help data scientists fluidly understand and explain models along a completeness-simplicity spectrum and serve as a starting point for model analysis. TELEGAM can be accessed at: <https://poloclub.github.io/telegam/>.

4.2 TELEGAM: Visualization & Verbalization

4.2.1 Design Goals

Through a literature survey and formative studies with ML researchers and practitioners, Hohman et al. [7] synthesize six model-agnostic capabilities that an explainable ML interface should support. In this work, we focus on four of these and use them as design goals (**DG**) for our system. The design goals below each contain an example interpretability question, which all reference a real-estate model that predicts the price of homes given the features of a house.

- DG1. Local instance explanations.** Given a single data instance, quantify each feature’s contribution to the prediction.

Example: Given a house and its predicted price of \$250,000, what features contributed to its price?

- DG2. Instance explanation comparisons.** Given a collection of data instances, compare what factors lead to their predictions.

Example: Given five houses in a neighborhood, what distinguishes them and their prices?

- DG3. Feature importance.** Given a model, rank the features of the data that are most influential to the overall predictions.

Example: Given a house price prediction model, does it make sense that the top three most influential features should be the square footage, year built, and location?

- DG4. Counterfactuals.** Given a single data instance, ask “what-if” questions to observe the effect that modified features have on its prediction.

Example: Given a house and its predicted price of \$250,000, how would the price change if it had an extra bedroom?

4.2.2 Model Class and Background

In this work we consider a particular model class, the *generalized additive model* (GAM) [88], which has recently attracted attention in the ML community [106, 107], and satisfies our four **DGs**. Modern ML techniques have enabled GAMs to compete favorably with more complex, state-of-the-art models on tabular data prediction tasks; however, GAMs remain intelligible and more expressive than simple linear models [89, 90, 91]. A GAM provides

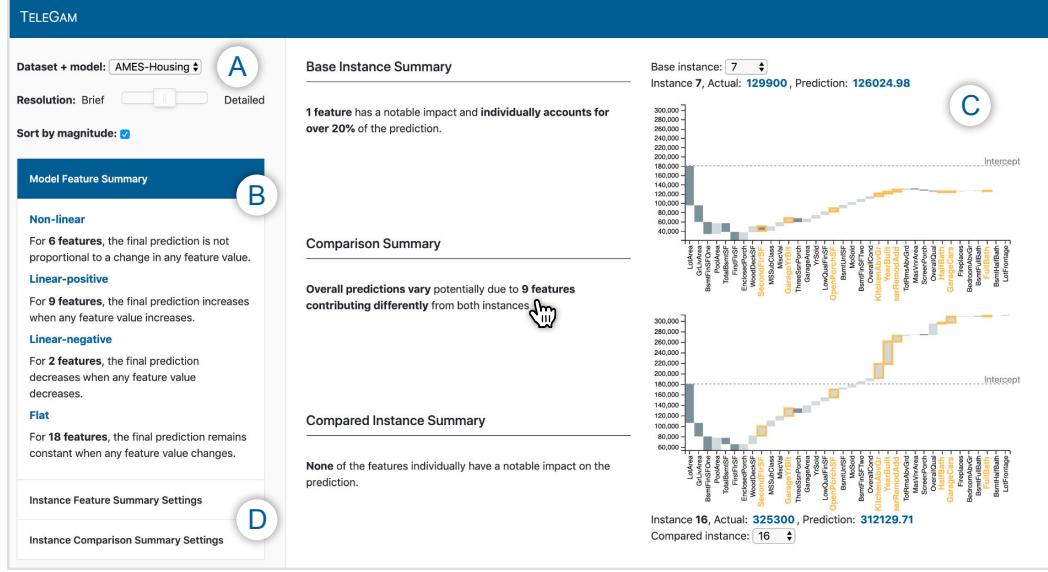


Figure 4.1: The TELEGAM user interface contains (A) a model selector and parameters for the visualizations and verbalizations. (B) **The Global Model View** displays model feature-level verbalizations of GAM shape function charts that describe a feature’s overall impact on model predictions. (C) **The Local Instance View** displays two data instance’s waterfall charts, an explanatory visualization that shows the cumulative sum of the contribution each feature has on the final prediction. Alongside are instance-level verbalizations that, when brushed, highlight in orange (the corresponding marks of the visualization that the verbalization refers to). (D) Settings to interactively tune verbalization generation thresholds.

both local instance explanations similar to linear regression, but also global feature explanations which other models lack.

GAMs are a generalization of linear models; GAMs replace linear model’s slope coefficients with smooth, shape functions. In both models, the relationship between the target variable and the features is still additive; however, each feature in a GAM is described by one shape function that can be nonlinear and complex (e.g., concave, convex, or “bendy”) [96]. Therefore, GAMs are considered intelligible [89] since each feature’s contribution to the final prediction can be understood by inspecting the shape functions. In this work, we omit the technical details and mathematical formulations of GAMs and their training, which are covered in the literature [97, 98, 90, 91].

4.2.3 Realizing Design Goals in TELEGAM’s Interface

We first give an overview of TELEGAM’s interface (Figure 4.1), deferring the details of the verbalizations to the next section. When a model is loaded (Figure 4.1A), the Global Model View (Figure 4.1B) displays sentences highlighting the features that may be interesting for the user to consider (**DG3**). Brushing over sentences displays a tooltip (??) showing the

GAM shape function line charts that present an overview of the feature values (on the x-axis) and model predictions (on the y-axis) corresponding to the features listed in the sentence. These visualizations also enable a user to ask counterfactuals, i.e., “what if” questions, by quantifying the increase or decrease of predictions based on a change in any feature value (**DG4**).

Selecting an instance from the dropdown in the Local Instance View (Figure 4.1C) displays the actual and predicted values for the instance. TELEGAM presents a waterfall chart similar to that in GAMUT [7] where the features are listed on the x-axis and the contribution to the prediction from each feature are represented by the height of the bars. The color of the bar indicates whether the contribution is positive (**light gray**) or negative (**dark gray**). By default, the features are sorted by the absolute magnitude of their contributions, i.e., the feature with the highest absolute contribution is shown on the left. A toggle is present (Figure 4.1A) to sort features by their actual contributions instead of their absolute contributions if desired. To summarize the contributions of features towards an instance’s prediction (**DG1**), along with displaying a waterfall chart, TELEGAM also presents a textual summary alongside the chart. Brushing over this sentence visually highlights the notable features and their corresponding bars in the waterfall chart in **orange**, as seen in Figure 4.1C. This also is useful for asking counterfactual questions by identifying which features could be changed to increase or decrease an instance’s final prediction (**DG4**).

With a base instance selected (Figure 4.1C, top), users can select a second instance for comparison (Figure 4.1C, bottom). To enable visual comparison, TELEGAM ensures the scale of the y-axis as well as the ordering of the features on the x-axis in both waterfall charts are normalized and consistent. In addition to showing a textual summary for each instance (Figure 4.1C: top-left, bottom-left), TELEGAM also generates a comparative summary highlighting the differences between the predictions, displaying possible features that may cause the difference (**DG2**). Similar to the individual instance summaries, brushing the comparative summary visually highlights the described features in the visualization **orange** (Figure 4.1C).

4.2.4 Generating Verbalizations

Following the design goals, TELEGAM presents three types of verbalizations to accompany feature and instance-level visualizations. We converged to these types of verbalizations based on interactions with participants during GAMUT’s user study [7] as well as other data scientists who frequently interact with GAMs. Specifically, we considered and collated comments with respect to communicating a model’s performance to different stakeholders. Then, using an iterative trial-and-error approach, we defined a set of heuristics to

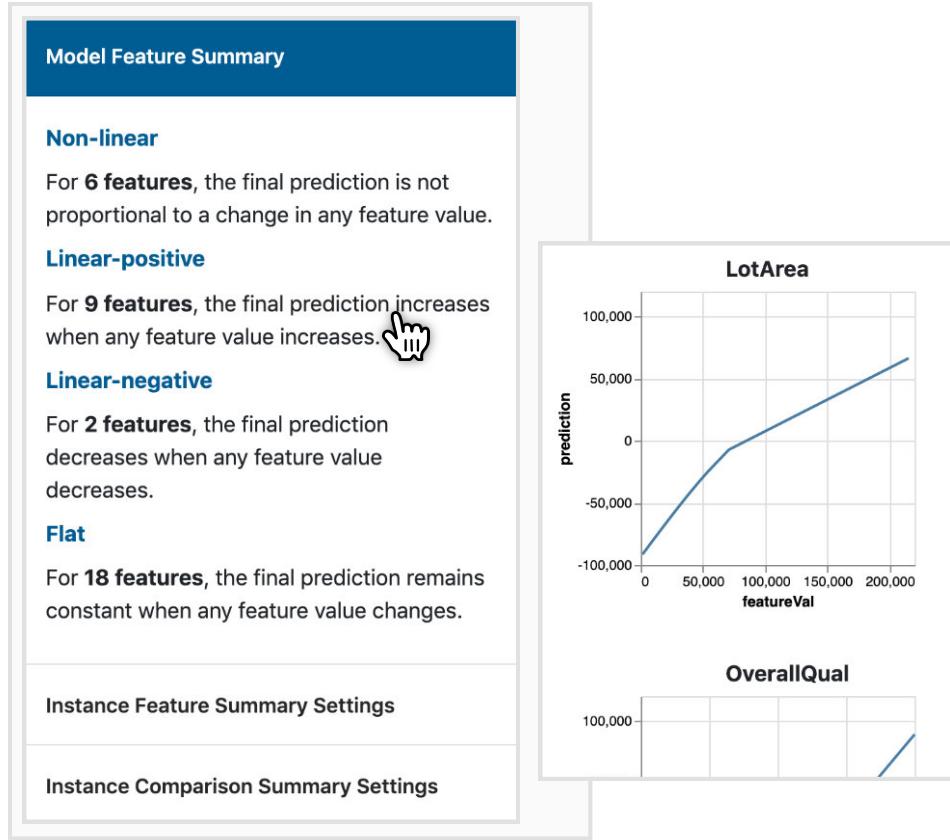


Figure 4.2: In TELEGAM, brushing a model feature verbalization displays a tooltip with features' corresponding shape function charts, a common GAM visualization. For example, here, the contribution of the linear-positive feature **LotArea** on overall model predictions approximately constantly increases as the feature value increases.

generate a set of verbalizations that were common across the observations. Note that the current set of verbalizations are only an initial step towards exploring how visualizations and verbalizations can be integrated in the context of GAMs, and are not exhaustive.

Instance Feature Summary. For an individual instance, TELEGAM verbalizes the features that have a notable impact on its final prediction. To generate this verbalization, we first compute the ratio of each feature's contribution with respect to the total prediction. If the ratio is greater than a predefined threshold τ_{contrib} , then that feature is included in the verbalization. In other words, a feature x_i is included in the verbalization if $f(x_i)/y > \tau_{\text{contrib}}$, where $f(x_i)$ is the feature's GAM prediction contribution and y is the final instance prediction. We empirically set τ_{contrib} to 0.15. For example, for the base instance in Figure 4.1C top, only one feature (**LotArea**) is included in the verbalization because it is the only feature that has a contribution of over 0.15 (or 15%) towards the instance's prediction.

Instance Comparison Summary. When verbalizing comparisons between two instances, TELEGAM identifies how similar, or different, the predictions for the instances are while highlighting which features may be contributing to the prediction difference. To do so, we first normalize both the total predictions and the individual feature contributions for all instances to $[0 - 1]$ so the comparison can be made in context of the entire dataset. Then, we check for the differences between the considered pair of normalized predictions and compare them to preset thresholds (τ_{minDiff} , τ_{maxDiff}) to generate the verbalization. Specifically, given two data instances and their normalized predictions y_1 and y_2 , the predictions are considered:

$$\begin{cases} \text{too similar} & \text{if } |y_1 - y_2| < \tau_{\text{minDiff}}, \\ \text{too different} & \text{else if } |y_1 - y_2| > \tau_{\text{maxDiff}}, \\ \text{moderately varying} & \text{else.} \end{cases}$$

where τ_{minDiff} and τ_{maxDiff} are empirically set to 0.25 and 0.75. For the second half of the verbalization, a feature is considered accountable for the difference between the final predictions of two instances if for any feature x_i their normalized feature contributions $f(x_{1,i})$ and $f(x_{2,i})$ satisfy

$$|f(x_{1,i}) - f(x_{2,i})| > \tau_{\text{featureContrib}}$$

where $\tau_{\text{featureContrib}}$ is empirically set to 0.25. For example, in Figure 4.1C, the verbalization states “*overall predictions vary*” because, in the context of the dataset, the two instances have moderately differing predictions which may be because of “*9 features contributing differently*,” since the normalized differences between the predictions for those features was over 0.25.

Model Feature Summary. TELEGAM highlights four groups of feature-level verbalizations based on the overall geometry of the shape function line charts—namely, features that have positively linear, negatively linear, non-linear, or flat geometry. To identify these groups, we used an agglomerative hierarchical clustering approach: a bottom-up technique for clustering data; in this case, we represent the shape function line charts in Figure 4.2 as time series and cluster them based on their overall geometry. We then inspected and labeled the clusters as the four groups listed above. Since some features may have expected predictions that are typically linear (e.g., the predicted price of a house increases with its square footage), these high-level groups and their corresponding verbalizations help users focus



Figure 4.3: TELEGAM supports an initial interactive affordance to realize the simplicity-completeness explanation spectrum in an interface. As a user drags the slider, the resolution of the natural language explanation updates from “brief” to “detailed.” In the example above, the comparison summary for two instances is shown at three different levels of explanation resolution, including (A) brief, (B) default, and (C) detailed.

on features that are potentially more interesting (e.g. those with non-linear geometry) while still summarizing every feature.

4.2.5 User-specified Verbalization Resolution

Professional data scientists have different reasons to interpret models and tailor explanations for specific audiences, often balancing competing concerns of simplicity and completeness [7, 15, 18]. Previous work has also suggested interfaces where users could specify the resolution of presented explanations; this can help adapt to users with differing preferences or expertise levels [50]. TELEGAM supports an initial interactive affordance to realize this simplicity-completeness tradeoff spectrum. Located in Figure 4.1A, a slider adjusts how detailed verbalizations should be. Currently, there are three positions ranging from “brief” to “detailed.” As a user drags the slider from one end to the other, the verbalizations update to provide more (or less) detail about a data instance’s prediction.

For example, Figure 4.3 shows three different slider positions for verbalization summarization for the comparison of two instance predictions. When set to “Brief” (Figure 4.3A), the verbalization is composed only of text to describe the difference between the two instance’s predictions. Dragging the slider right, in the second position (Figure 4.3B), the sentence updates and displays the exact number of features that the two predictions differ in. Finally, in the “Detailed” position (Figure 4.3C), the sentence updates and lists the actual prediction values, the number of differing features, and what percent of the total features the instances differ in. This is only one realized example of how a system could provide explanations based on user-specified resolutions to better communicate results to differing stakeholders invested in an ML model.

4.2.6 Illustrative Usage Scenario

We now demonstrate how the different views of TELEGAM could be used to interpret a GAM through a hypothetical usage scenario. June is a data scientist at a real-estate firm exploring the available properties to gain insight into the company’s portfolio. As June loads a pre-trained model into TELEGAM to understand the data and predictions, the system automatically displays textual statements summarizing the major feature trends (Figure 4.1B). By interacting with these statements, June explores how the different features are represented inside the model (Figure 4.2). Next, recollecting a property (data instance) they recently visited ($id=7$) but did not sell despite it being affordable, June inspects it as the base instance (Figure 4.1C, top). Based on their understanding of the individual feature trends from Global Model View and through the visualization in the Local Instance View, they infer that the *LotArea* feature is the primary factor determining the property’s value. The text alongside the chart simultaneously confirms this inference (Figure 4.1C, top-left).

To understand potential factors that make a house more saleable, June compares the selected property to another ($id=16$) that recently sold although it was more expensive. Through a combination of the juxtaposed waterfall charts and the verbalizations comparing the two charts, June notes that the differences in price arise from multiple non-salient features (e.g., *OpenPorchSF*, *SecondFlrSF*) that they would have otherwise missed without a visual linking between the text and the charts (Figure 4.1B). Adjusting the comparison verbalization resolution (Figure 4.3), TELEGAM further reveals the specifics of the contributing feature quantity and distribution differences. Finally, to prepare a report to share with their colleagues, June sets the detail level of the verbalizations to “Brief” and captures a screenshot, moving onto other instances and continuing their analysis.

4.3 Conclusion

Through the design of TELEGAM, we show how combining visualizations and verbalizations can support interactive exploration and interpretation of ML models, demonstrated using GAMs. TELEGAM also represents an initial step towards a broader research goal that aims to understand if verbalizations can enhance interpretability by augmenting ML visualization tools with explanations.

PART II

SCALING DEEP LEARNING

INTERPRETABILITY

Overview

We have seen the benefits that interactive interfaces for machine learning interpretability bring to practitioners; however, GAMUT and TELEGAM both demonstrate this opportunity using generalized additive models, which while expressive for tabular data, currently do not compete with more complex models for tasks within computer vision and natural language processing. For image and text data, deep learning and neural network approaches have seen a surge in success and popularity despite their infamous lack of interpretability. Next, we show how to **scale interpretability** to larger and more complex model architectures, such as deep neural networks.

Part II begins by presenting a **survey (Chapter 5)** of visual analytics in deep learning with a focus on interpretability, highlighting its short yet impactful history. The survey thoroughly summarizes the state-of-the-art using a human-centered interrogative framework, focusing on the *Five W's and How* (Why, Who, What, How, When, and Where). This chapter is adapted from work that was published and appeared in *TVCG 2018* [9].

Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. ↗ Fred Hohman, Minsuk Kahng, Robert Pienta, Duen Horng (Polo) Chau. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2018.

From this survey, we find there is a lack of high-level explanations for neural network predictions and learned feature representations. In response, we designed and developed **SUMMIT (Chapter 6)**, an interactive system that scalably and systematically summarizes and visualizes what features a deep learning model has learned and how those features interact to make predictions. This chapter is adapted from work that was published and appeared in *TVCG 2020* [10].

SUMMIT: Scaling Deep Learning Interpretability by Visualizing Activation and Attribution Summarizations. ↗ Fred Hohman, Haekyu Park, Caleb Robinson, Duen Horng (Polo) Chau. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2020.

CHAPTER 5

INTERROGATIVE SURVEY FOR VISUAL ANALYTICS IN DEEP LEARNING

Deep learning has recently seen rapid development and received significant attention due to its state-of-the-art performance on previously-thought hard problems. However, because of the internal complexity and nonlinear structure of deep neural networks, the underlying decision making processes for why these models are achieving such performance are challenging and sometimes mystifying to interpret. As deep learning spreads across domains, it is of paramount importance that we equip users of deep learning with tools for understanding when a model works correctly, when it fails, and ultimately how to improve its performance. Standardized toolkits for building neural networks have helped democratize deep learning; visual analytics systems have now been developed to support model explanation, interpretation, debugging, and improvement. We present a survey of the role of visual analytics in deep learning research, which highlights its short yet impactful history and thoroughly summarizes the state-of-the-art using a human-centered interrogative framework, focusing on the *Five W's and How* (Why, Who, What, How, When, and Where). We conclude by highlighting research directions and open research problems. This survey helps researchers and practitioners in both visual analytics and deep learning to quickly learn key aspects of this young and rapidly growing body of research, whose impact spans a diverse range of domains.

5.1 Introduction

Deep learning is a specific set of techniques from the broader field of machine learning that focus on the study and usage of *deep* artificial neural networks to learn structured representations of data. First mentioned as early as the 1940s [108], artificial neural networks have a rich history [109], and have recently seen a dominate and pervasive resurgence [78, 65, 110] in many research domains by producing state-of-the-art results [111, 112] on a number of diverse big data tasks [113, 114]. For example, the premiere machine learning, deep learning, and artificial intelligence conferences have seen enormous growth in attendance and paper submissions since early 2010s. Furthermore, open-source toolkits and programming libraries for building, training, and evaluating deep neural networks have become more robust and easy to use, democratizing deep learning. As a result, the barrier to

Visual Analytics in Deep Learning Interrogative Survey Overview

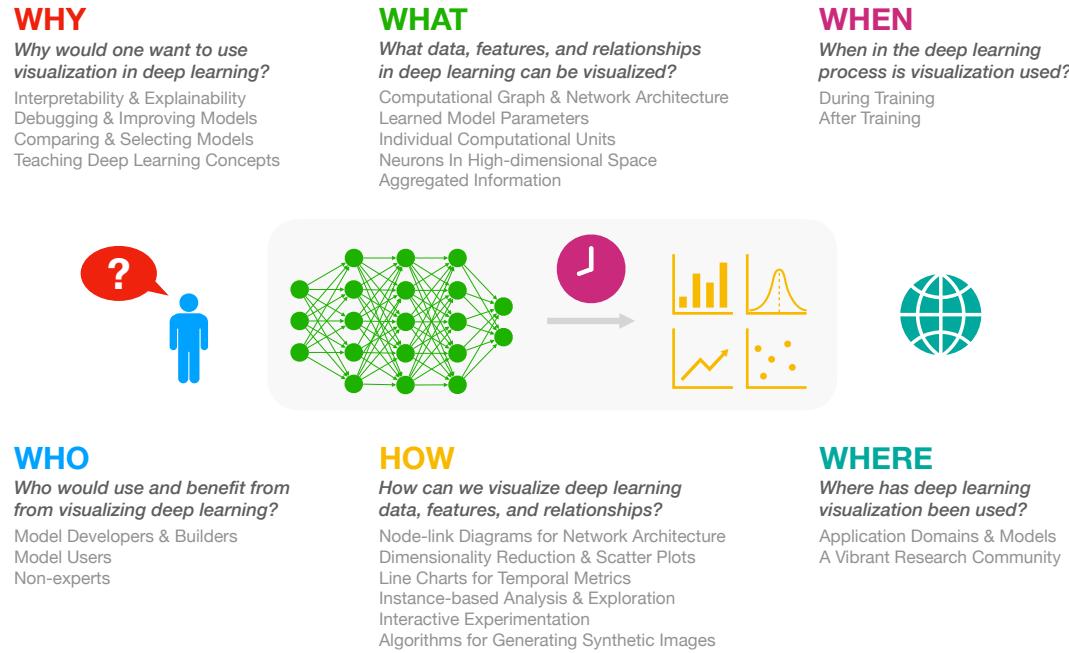


Figure 5.1: A visual overview of our interrogative survey, and how each of the six questions, “Why, Who, What, How, When, and Where,” relate to one another. Each question corresponds to one section of this survey, indicated by the numbered tag, near each question title. Each section lists its major subsections discussed in the survey.

developing deep learning models is lower than ever before and deep learning applications are becoming pervasive.

While this technological progress is impressive, it comes with unique and novel challenges. For example, the lack of interpretability and transparency of neural networks, from the learned representations to the underlying decision process, is an important problem to address. Making sense of why a particular model misclassifies test data instances or behaves poorly at times is a challenging task for model developers. Similarly, end-users interacting with an application that relies on deep learning to make critical decisions may question its reliability if no explanation is given by the model, or become baffled if the explanation is convoluted. While explaining neural network decisions is important, there are numerous other problems that arise from deep learning, such as AI safety and security (e.g., when using models in applications such as self-driving vehicles), and compromised trust due to bias in models and datasets, just to name a few. These challenges are often compounded, due to the large datasets required to train most deep learning models. As worrisome as these problems are, they will likely become even more widespread as more

AI-powered systems are deployed in the world. Therefore, a general sense of model understanding is not only beneficial, but often required to address the aforementioned issues.

Data visualization and visual analytics excel at knowledge communication and insight discovery by using encodings to transform abstract data into meaningful representations. In the seminal work by Zeiler and Fergus [68], a technique called *deconvolutional networks* enabled projection from a model’s learned feature space back to the pixel space. Their technique and results give insight into what types of features deep neural networks are learning at specific layers, and also serve as a debugging tool for improving a model. This work is often credited for popularizing visualization in the machine learning and computer vision communities in recent years, putting a spotlight on it as a powerful tool that helps people understand and improve deep learning models. However, visualization research for neural networks started well before [115, 116, 117]. Over just a handful of years, many different techniques have been introduced to help interpret what neural networks are learning. Many such techniques generate static images, such as attention maps and heatmaps for image classification, indicating which parts of an image are most important to the classification. However, interaction has also been incorporated into the model understanding process in visual analytics tools to help people gain insight [73, 74, 75]. This hybrid research area has grown in both academia and industry, forming the basis for many new research papers, academic workshops, and deployed industry tools.

In this survey, we summarize a large number of deep learning visualization works using the *Five W’s and How* (Why, Who, What, How, When, and Where). Figure 5.1 presents a visual overview of how these interrogative questions reveal and organize the various facets of deep learning visualization research and their related topics. By framing the survey in this way, many existing works fit a description as the following fictional example:

*To interpret representations learned by deep models (**why**), model developers (**who**) visualize neuron activations in convolutional neural networks (**what**) using t-SNE embeddings (**how**) after the training phase (**when**) to solve an urban planning problem (**where**).*

This framing captures the needs, audience, and techniques of deep learning visualization, and positions new work’s contributions in the context of existing literature.

We conclude by highlighting prominent research directions and open problems. We hope that this survey acts as a companion text for researchers and practitioners wishing to understand how visualization supports deep learning research and applications.

5.2 Our Contributions & Method of Survey

5.2.1 Our Contributions

- C1.** We present a comprehensive, timely survey on visualization and visual analytics in deep learning research, using a human-centered, interrogative framework. This method enables us to position each work with respect to its *Five Ws and How* (Why, Who, What, How, When, and Where), and flexibly discuss and highlight existing works' multifaceted contributions.
- Our human-centered approach using the *Five W's and How* — based on how we familiarize ourselves with new topics in everyday settings — enables readers to quickly grasp important facets of this young and rapidly growing body of research.
 - Our interrogative process provides a framework to describe existing works, as well as a model to base new work off of.
- C2.** To highlight and align the cross-cutting impact that visual analytics has had on deep learning across a broad range of domains, our survey goes beyond visualization-focused venues, extending a wide scope that encompasses most relevant works from many top venues in artificial intelligence, machine learning, deep learning, and computer vision. We highlight how visual analytics has been an integral component in solving some of AI's biggest modern problems, such as neural network interpretability, trust, and security.
- C3.** As deep learning, and more generally AI, touches more aspects of our daily lives, we highlight important research directions and open problems that we distilled from the survey. These include improving the capabilities of visual analytics systems for furthering interpretability, conducting more effective design studies for evaluating system usability and utility, advocating humans' important roles in AI-powered systems, and promoting proper and ethical use of AI applications to benefit society.

5.2.2 Survey Methodology & Summarization Process

We selected existing works from top computer science journals and conferences in visualization (e.g., IEEE Transactions on Visualization and Computer Graphics (TVCG)), visual analytics (e.g., IEEE Conference on Visual Analytics Science and Technology (VAST)) and deep learning (e.g., Conference on Neural Information Processing Systems (NIPS) and the International Conference on Machine Learning (ICML)). Since deep learning visualization

is relatively new, much of the relevant work has appeared in workshops at the previously mentioned venues; therefore, we also include those works in our survey. Table 5.1 lists some of the most prominent publication venues and their acronyms. We also inspected preprints posted on arXiv (<https://arxiv.org/>), an open-access, electronic repository of manuscript preprints, whose computer science subject has become a hub for new deep learning research. Finally, aside from the traditional aforementioned venues, we include non-academic venues with significant attention such as Distill, industry lab research blogs, and research blogs of influential figures. Because of the rapid growth of deep learning research and the lack of a perfect fit for publishing and disseminating work in this hybrid area, the inclusion of these non-traditional sources are important to review, as they are highly influential and impactful to the field.

Visualization takes many forms throughout the deep learning literature. This survey focuses on visual analytics for deep learning. We also include related works from the AI and computer vision communities that contribute novel static visualizations. So far, the majority of work surrounds convolutional neural networks (CNNs) and image data; more recent work has begun to visualize other models, e.g., recurrent neural networks (RNNs), long short-term memory units (LSTMs), and generative adversarial networks (GANs). For each work, we recorded the following information if present:

- Metadata (title, authors, venue, and year published)
- General approach and short summary
- Explicit contributions
- Future work
- Design component (e.g. user-centered design methodologies, interviews, evaluation)
- Industry involvement and open-source code

With this information, we used the *Five W’s and How* (Why, Who, What, How, When, and Where) to organize these existing works and the current state-of-the-art of visualization and visual analytics in deep learning.

5.2.3 Related Surveys

While there is a larger literature for visualization for machine learning, including predictive visual analytics [32, 31, 35] and human-in-the-loop interactive machine learning [118,

Table 5.1: Relevant visualization and AI venues ordered by: journals, conferences, workshops, open access journals, and preprint repositories. Within each, visualization venues precedes AI venues.

TVCG	IEEE Transactions on Visualization and Computer Graphics
VAST	IEEE Conference on Visual Analytics Science and Technology
InfoVis	IEEE Information Visualization
VIS	IEEE Visualization Conference (VAST+InfoVis+SciVis)
CHI	ACM Conference on Human Factors in Computing Systems
NIPS	Conference on Neural Information Processing Systems
ICML	International Conference on Machine Learning
CVPR	Conference on Computer Vision and Pattern Recognition
ICLR	International Conference on Learning Representations
VADL	IEEE VIS Workshop on Visual Analytics for Deep Learning
HCML	CHI Workshop on Human Centered Machine Learning
IDEA	KDD Workshop on Interactive Data Exploration & Analytics
	ICML Workshop on Visualization for Deep Learning
WHI	ICML Workshop on Human Interpretability in ML
	NIPS Workshop on Interpreting, Explaining and Visualizing Deep Learning
	NIPS Interpretable ML Symposium
FILM	NIPS Workshop on Future of Interactive Learning Machines
	ACCV Workshop on Interpretation and Visualization of Deep Neural Nets
	ICANN Workshop on Machine Learning and Interpretability
Distill	Distill: Journal for Supporting Clarity in Machine Learning
arXiv	arXiv.org e-Print Archive

119], to our knowledge there is no comprehensive survey of visualization and visual analytics for deep learning. Regarding deep neural networks, related surveys include a recent book chapter that discusses visualization of deep neural networks related to the field of computer vision [120], an unpublished essay that proposes a preliminary taxonomy for visualization techniques [121], and an article that focuses on describing interactive model analysis, which mentions deep learning in a few contexts while describing a high-level framework for general machine learning models [122]. A recent overview article by Choo and Liu [123] is the closest in spirit to our survey. Our survey provides wider coverage and more detailed analysis of the literature.

Different from all the related articles mentioned above, our survey provides a comprehensive, human-centered, and interrogative framework to describe deep learning visual analytics tools, discusses the new, rapidly growing community at large, and presents the major research trajectories synthesized from existing literature.

5.2.4 Survey Overview & Organization

Section 5.3 introduces common deep learning terminology. Figure 5.1 shows a visual overview of this survey’s structure and Figure 5.2 summarizes representative works. Each interrogative question (Why, Who, What, How, When, and Where) is given its own section for discussion, ordered to best motivate why visualization and visual analytics in deep learning is such a rich and exciting area of research.

5.5 Why do we want to visualize deep learning?

Why and for what purpose would one want to use visualization in deep learning?

5.5 Who wants to visualize deep learning?

Who are the types of people and users that would use and stand to benefit from visualizing deep learning?

5.6 What can we visualize in deep learning?

What data, features, and relationships are inherent to deep learning that can be visualized?

5.7 How can we visualize deep learning?

How can we visualize the aforementioned data, features, and relationships?

5.8 When can we visualize deep learning?

When in the deep learning process is visualization used and best suited?

5.9 Where is deep learning visualization being used?

Where has deep learning visualization been used?

5.3 Common Terminology

To enhance readability of this survey, and to provide quick references for readers new to deep learning, we have tabulated a sample of relevant and common deep learning terminology used in this work, shown in Table 5.2. The reader may want to refer to Table 5.2 throughout this survey for technical terms, meanings, and synonyms used in various contexts of discussion. The table serves as an introduction and summarization of the state-of-the-art. For definitive technical and mathematical descriptions, we encourage the reader to refer to excellent texts on deep learning and neural network design, such as the *Deep Learning* textbook [124].

Author	Year	WHY			WHO			WHAT			HOW			WHEN		WHERE				
		Interpretability & Explainability	Debugging & Improving Models	Comparing & Selecting Models	Model Developers & Builders	Model Users	Non-experts	Computational Graph & Network Architecture	Learned Model Parameters	Individual Computational Units	Neurons in High-dimensional Space	Aggregated Information	Node-link Diagrams for Network Architecture	Dimensionality Reduction & Scatter Plots	Line Charts for Temporal Metrics	Instance-based Analysis & Exploration	Interactive Experimentation	Algorithms for Attribution & Feature Visualization	During Training	After Training
Abadi, et al.	2016	■■■			■■													■■	■■	arXiv
Bau, et al.	2017	■■■	■■		■■				■■									■■	■■	CVPR
Bilal, et al.	2017	■■■	■■		■■				■■	■■	■■							■■	■■	TVCG
Bojarski, et al.	2016	■■■	■■		■■				■■									■■■	■■■	arXiv
Bruckner	2014	■■■	■■		■■															MS Thesis
Carter, et al.	2016	■■■		■■■■■	■■■■■	■■■■■	■■■■■											■■■	■■■	Distill
Cashman, et al.	2017	■■■	■■		■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■	■■■	VADL
Chae, et al.	2017	■■■	■■		■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■	■■■	VADL
Chung, et al.	2016	■■■	■■		■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■	■■■	FILM
Goyal, et al.	2016	■■■			■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■	■■■	arXiv
Harley	2015	■■■			■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■	■■■	ISVC
Hohman, et al.	2017	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	CHI
Kahng, et al.	2018	■■■	■■		■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	TVCG
Karpathy, et al.	2015	■■■			■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	arXiv
Li, et al.	2015	■■■			■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	arXiv
Liu, et al.	2017	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	TVCG
Liu, et al.	2018	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	TVCG
Ming, et al.	2017	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	VAST
Norton & Qi	2017	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	VizSec
Olah	2014	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	Web
Olah, et al.	2018	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	Distill
Pezzotti, et al.	2017	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	TVCG
Rauber, et al.	2017	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	TVCG
Robinson, et al.	2017	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	GeoHum.
Rong, et al.	2016	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	ICML VIS
Smilkov, et al.	2016	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	NIPS Workshop
Smilkov, et al.	2017	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	ICML VIS
Strobelt, et al.	2017	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	TVCG
Tzeng & Ma	2005	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■		■■■■■	■■■■■	■■■■■							■■■■■	■■■■■	VIS
Wang, et al.	2018	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■		■■■■■■■■■	■■■■■■■■■	■■■■■■■■■							■■■■■■■■■	■■■■■■■■■	TVCG
Webster, et al.	2017	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■		■■■■■■■■■	■■■■■■■■■	■■■■■■■■■							■■■■■■■■■	■■■■■■■■■	Web
Wongsuphasawat, et al.	2018	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■		■■■■■■■■■	■■■■■■■■■	■■■■■■■■■							■■■■■■■■■	■■■■■■■■■	TVCG
Yosinski, et al.	2015	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■		■■■■■■■■■	■■■■■■■■■	■■■■■■■■■							■■■■■■■■■	■■■■■■■■■	ICML DL
Zahavy, et al.	2016	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■		■■■■■■■■■	■■■■■■■■■	■■■■■■■■■							■■■■■■■■■	■■■■■■■■■	ICML
Zeiler, et al.	2014	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■		■■■■■■■■■	■■■■■■■■■	■■■■■■■■■							■■■■■■■■■	■■■■■■■■■	ECCV
Zeng, et al.	2017	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■		■■■■■■■■■	■■■■■■■■■	■■■■■■■■■							■■■■■■■■■	■■■■■■■■■	VADL
Zhong, et al.	2017	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■		■■■■■■■■■	■■■■■■■■■	■■■■■■■■■							■■■■■■■■■	■■■■■■■■■	ICML VIS
Zhu, et al.	2016	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■	■■■■■■■■■		■■■■■■■■■	■■■■■■■■■	■■■■■■■■■							■■■■■■■■■	■■■■■■■■■	ECCV

Figure 5.2: Overview of representative works in visual analytics for deep learning. Each row is one work; works are sorted alphabetically by first author's last name. Each column corresponds to a subsection from the six interrogative questions.

Table 5.2: Foundational deep learning terminology used in this survey, sorted by importance. In a term’s “meaning” (last column), defined terms are italicized.

Technical Term	Synonyms	Meaning
Neural Network	Artificial neural net, net	Biologically-inspired models that form the basis of deep learning; approximate functions dependent upon a large and unknown amount of inputs consisting of <i>layers of neurons</i>
Neuron	Computational unit, node	Building blocks of <i>neural networks</i> , entities that can apply <i>activation functions</i>
Weights	Edges	The trained and updated parameters in the <i>neural network</i> model that connect <i>neurons</i> to one another
Layer	Hidden layer	Stacked collection of <i>neurons</i> that attempt to extract features from data; a <i>layer's</i> input is connected to a previous <i>layer's</i> output
Computational Graph	Dataflow graph	Directed graph where nodes represent operations and edges represent data paths; when implementing <i>neural network</i> models, often times they are represented as these
Activation Functions	Transform function	Functions embedded into each <i>layer</i> of a <i>neural network</i> that enable the network represent complex non-linear decisions boundaries
Activations	Internal representation	Given a trained network one can pass in data and recover the <i>activations</i> at any <i>layer</i> of the network to obtain its current representation inside the network
Convolutional Neural Network	CNN, convnet	Type of <i>neural network</i> composed of convolutional <i>layers</i> that typically assume image data as input; these <i>layers</i> have depth unlike typical <i>layers</i> that only have width (number of <i>neurons</i> in a <i>layer</i>); they make use of filters (feature & pattern detectors) to extract spatially invariant representations
Long Short-Term Memory	LSTM	Type of <i>neural network</i> , often used in text analysis, that addresses the vanishing gradient problem by using memory gates to propagate gradients through the network to learn long-range dependencies
Loss Function	Objective function, cost function, error	Also seen in general ML contexts, defines what success looks like when learning a representation, i.e., a measure of difference between a <i>neural network's</i> prediction and ground truth
Embedding	Encoding	Representation of input data (e.g., images, text, audio, time series) as vectors of numbers in a high-dimensional space; oftentimes reduced so data points (i.e., their vectors) can be more easily analyzed (e.g., compute similarity)
Recurrent Neural Network	RNN	Type of <i>neural network</i> where recurrent connections allow the persistence (or “memory”) of previous inputs in the network's internal state which are used to influence the network output
Generative Adversarial Networks	GAN	Method to conduct unsupervised learning by pitting a generative network against a discriminative network; the first network mimics the probability distribution of a training dataset in order to fool the discriminative network into judging that the generated data instance belongs to the training set
Epoch	Data pass	A complete pass through a given dataset; by the end of one <i>epoch</i> , a <i>neural network</i> will have seen every datum within the dataset once

5.4 Why Visualize Deep Learning

5.4.1 Interpretability & Explainability

The most abundant, and to some, the most important reason why people want to visualize deep learning is to understand how deep learning models make decisions and what representations they have learned, so we can place trust in a model [20]. This notion of general model understanding has been called the *interpretability* or *explainability* when referring to machine learning models [20, 17, 21]. However, neural networks particularly suffer from this problem since oftentimes real world and high-performance models contain a large number of parameters (in the millions) and exhibit extreme internal complexity by using many non-linear transformations at different stages during training. Many works motivate this problem by using phrases such as “opening and peering through the black-box,” “transparency,” and “interpretable neural networks,” [117, 125, 126], referring the internal complexity of neural networks.

Discordant Definitions for Interpretability

Unfortunately, there is no universally formalized and agreed upon definition for explainability and interpretability in deep learning, which makes classifying and qualifying interpretations and explanations troublesome. In Lipton’s work “The Mythos of Model Interpretability [20],” he surveys interpretability-related literature, and discovers diverse motivations for why interpretability is important and is occasionally discordant. Despite this ambiguity, he attempts to refine the notion of interpretability by making a first step towards providing a comprehensive taxonomy of both the desiderata and methods in interpretability research. One important point that Lipton makes is the difference between interpretability and an explanation; an explanation can show predictions without elucidating the mechanisms by which models work [20].

In another work originally presented as a tutorial at the International Conference on Acoustics, Speech, and Signal Processing by Montavona et al. [17], the authors propose exact definitions of both an interpretation and an explanation. First, an interpretation is “the mapping of an abstract concept (e.g., a predicted class) into a domain that the human can make sense of.” They then provide some examples of interpretable domains, such as images (arrays of pixels) and text (sequences of words), and noninterpretable domains, such as abstract vector spaces (word embeddings). Second, an explanation is “the collection of features of the interpretable domain, that have contributed for a given example to produce a decision (e.g., classification or regression).” For example, an explanation can be a heatmap

highlighting which pixels of the input image most strongly support an image classification decision, or in natural language processing, explanations can highlight certain phrases of text.

However, both of the previous works are written by members of the AI community, whereas work by Miller titled “Explanation in Artificial Intelligence: Insights from the Social Sciences” [21] postulates that much of the current research uses only AI researchers’ intuition of what constitutes a “good” explanation. He suggests that if the focus on explaining decisions or actions to a human observer is the goal, then if these techniques are to succeed, the explanations they generate should have a structure that humans accept. Much of Miller’s work highlights vast and valuable bodies of research in philosophy, psychology, and cognitive science for how people define, generate, select, evaluate, and present explanations, and he argues that interpretability and explainability research should leverage and build upon this history [21]. In another essay, Offert [127] argues that to make interpretability more rigorous, we must first identify where it is impaired by intuitive considerations. That is, we have to “consider it precisely in terms of what it is not.” While multiple works bring different perspectives, Lipton makes the keen observation that for the field to progress, the community must critically engage with this problem formulation issue [20]. Further research will help solidify the notions of interpretation and explanation.

Interpretation as Qualitative Support for Model Evaluation in Application Domains

While research into interpretation itself is relatively new, its impact has already been seen in applied deep learning contexts. A number of applied data science and AI projects that use deep learning models include a section on interpretation to qualitatively evaluate and support the model’s predictions and the work’s claims overall. An example of this is an approach for end-to-end neural machine translation. In the work by Johnson et al. [128], the authors present a simple and efficient way to translate between multiple languages using a single model, taking advantage of multilingual data to improve neural machine translation for all languages involved. The authors visualize an embedding of text sequences, for example, sentences from multiple languages, to support and hint at a universal interlingua representation. Another work that visualizes large machine learning embeddings is by Zahavy et al. [126], where the authors analyze deep Q-networks (DQN), a popular reinforcement learning model, to understand and describe the policies learned by DQNs for three different Atari 2600 video games. An application for social good by Robinson et al. [129] demonstrates how to apply deep neural networks on satellite imagery to perform population prediction and disaggregation, jointly answering the questions “where do people live?” and “how many people live there?”. In general, they show how their methodology can be

an effective tool for extracting information from inherently unstructured, remotely-sensed data to provide effective solutions to social problems.

These are only a few domains where visualization and deep learning interpretation have been successfully used. Others include building trust in autonomous driving vehicles [130], explaining decisions made by medical imaging models, such as MRIs on brain scans, to provide medical experts more information for making diagnoses [131], and using visual analytics to explore automatically-learned features from street imagery to gain perspective into identity, function, demographics, and affluence in urban spaces, which is useful for urban design and planning [132].

In this survey we will mention interpretation and explanation often, as they are the most common motivations for deep learning visualization. Later, we will discuss the different visualization techniques and visual analytics systems that focus on neural network interpretability for embeddings [133], text [134, 135, 136], quantifying interpretability [137], and many different image-based techniques stemming from the AI communities [66, 68, 138, 65, 139].

5.4.2 Debugging & Improving Models

Building machine learning models is an iterative design process [140, 141, 142], and developing deep neural networks is no different. While mathematical foundations have been laid, deep learning still has many open research questions. For example, finding the exact combinations of model depth, layer width, and finely tuned hyperparameters is nontrivial. In response to this, many visual analytics systems have been proposed to help model developers build and debug their models, with the hope of expediting the iterative experimentation process to ultimately improve performance [74, 143, 144]. Oftentimes this requires monitoring models during the training phase [145, 69], identifying misclassified instances and testing a handful of well-known data instances to observe performance [39, 71, 146], and allowing a system to suggest potential directions for the model developer to explore [147]. This reason for why we wish to visualize deep learning ultimately provides better tools to speed up model development for engineers and researchers so that they can quickly identify and fix problems within a model to improve overall performance.

5.4.3 Comparing & Selecting Models

While certainly related to model debugging and improvement, model comparison and selection are slightly different tasks in which visualization can be useful [148, 149, 40]. Oftentimes model comparison describes the notion of choosing a single model among an

ensemble of well-performing models. That is, no debugging needs to be done; all models have “learned” or have been trained semi-successfully. Therefore, the act of selecting a single, best-performing model requires inspecting model metrics and visualizing parts of the model to pick the one that has the highest accuracy, the lowest loss, or is the most generalizable, while avoiding pitfalls such as memorizing training data or overfitting.

Some systems take a high-level approach and compare user-defined model metrics, like accuracy and loss, and aggregate them on interactive charts for performance comparison [150]. Other frameworks compare neural networks trained on different random initializations (an important step in model design) to discover how they would affect performance, while also quantifying performance and interpretation [137]. Some approaches compare models on image generation techniques, such as performing image reconstruction from the internal representations of each layer of different networks to compare different network architectures [151]. Similar to comparing model architectures, some systems solely rely on data visualization representations and encodings to compare models [152], while others compare different snapshots of a single model as it trains over time, i.e., comparing a model after n_1 epochs and the same model after n_2 epochs of training time [153].

5.4.4 Teaching Deep Learning Concepts

Apart from AI experts, another important reason why we may wish to visualize deep learning is to educate non-expert users about AI. The exact definition of non-experts varies by source and is discussed further in Subsection 5.5.3. An example that targets the general public is Teachable Machines [154], a web-based AI experiment that explores and teaches the foundations of an image classifier. Users train a three-way image classifier by using their computer’s webcam to generate the training data. After providing three different examples of physical objects around the user (e.g., holding up a pencil, a coffee mug, and a phone), the system then performs real-time inference on whichever object is in view of the webcam, and shows a bar chart with the corresponding classification scores. Since inference is computed in real-time, the bar charts wiggles and jumps back and forth as the user removes an object, say the pencil, from the view and instead holds up the coffee mug. The visualization used is a simple bar chart, which provides an approachable introduction into image classification, a modern-day computer vision and AI problem.

Another example for teaching deep learning concepts, the Deep Visualization Toolbox [155] discussed later in this survey, also uses a webcam for instant feedback when interacting with a neural network. Taking instantaneous feedback a step further, some works have used direct manipulation to engage non-experts in the learning process. TensorFlow Playground [75], a robust, web-based visual analytics tool for exploring simple

neural networks, uses direct manipulation to reinforce deep learning concepts, and importantly, evokes the user’s intuition about how neural networks work. Other non-traditional mediums have been used to teach deep learning concepts and build an intuition for how neural networks behave too. Longform, interactive scrollytelling works focusing on particular AI topics that use interactive visualizations as supporting evidence are gaining popularity. Examples include “How to Use t-SNE Effectively,” where users can play with hundreds of small datasets and vary single parameters to observe their effect on an embedding [156], and a similar interactive article titled “Visualizing MNIST” that visualizes different types of embeddings produced by different algorithms [157].

5.5 Who Uses Deep Learning Visualization

This section describes the groups of people who may stand to benefit from deep learning visualization and visual analytics. We loosely organize them into three *non-mutually exclusive* groups by their level of deep learning knowledge (most to least): *model developers*, *model users*, and *non-experts*. Note that many of the works discussed can benefit multiple groups, e.g., a model developer may use a tool aimed at non-experts to reinforce their own intuition for how neural networks learn.

5.5.1 Model Developers & Builders

The first group of people who use deep learning visualization are individuals whose job is primarily focused on developing, experimenting with, and deploying deep neural networks. These model developers and builders, whether they are researchers or engineers, have a strong understanding of deep learning techniques and a well-developed intuition surrounding model building. Their knowledge expedites key decisions in deep learning workflows, such as identifying which types of models perform best on which types of data. These individuals wield mastery over models, e.g., knowing how to vary hyperparameters in the right fashion to achieve better performance. These individuals are typically seasoned in building large-scale models and training them on high-performance machines to solve real-world problems [122]. Therefore, tooling and research for these users is much more technically focused, e.g., exposing many hyperparameters for detailed model control.

Of the existing deep learning visual analytics tools published, a handful tackle the problem of developing tools for model developers, but few have seen widespread adoption. Arguably the most well-known system is TensorBoard [150]: Google’s included open-source visualization platform for its dataflow graph library TensorFlow. TensorBoard includes a number of built-in components to help model developers understand, debug, and optimize

TensorFlow programs. It includes real-time plotting of quantitative model metrics during training, instance-level predictions, and a visualization of the computational graph. The computational graph component was published separately by Wongsuphasawat et al. [74] and works by applying a series of graph transformations that enable standard layout techniques to produce interactive diagrams of TensorFlow models.

Other tools, such as DeepEyes [144], assist in a number of model building tasks, e.g., identifying stable layers during the training process, identifying unnecessary layers and degenerated filters that do not contribute to a model’s decisions, pruning such entities, and identifying patterns undetected by the network, indicating that more filters or layers may be needed. Another tool, *Blocks* [71], allows a model builder to accelerate model convergence and alleviate overfitting, through visualizing class-level confusion patterns. Other research has developed new metrics beyond measures like loss and accuracy, to help developers inspect and evaluate networks while training them [145].

Some tools also address the inherent iterative nature of training neural networks. For example, ML-o-scope [158] utilizes a time-lapse engine to inspect a model’s training dynamics to better tune hyperparameters, while work by Chae et al. [147] visualizes classification results during training and suggests potential directions to improve performance in the model building pipeline. Lastly, visual analytics tools are beginning to be built for expert users who wish to use models that are more challenging to work with. For example, DGMTracker [69] is a visual analytics tool built to help users understand and diagnose the training process of deep generative models: powerful networks that perform unsupervised and semi-supervised learning where the primary focus is to discover the hidden structure of data without resorting to external labels.

5.5.2 Model Users

The second group of people who may benefit from deep learning visualization are model users. These are users who may have some technical background but are neural network novices. Common tasks include using well-known neural network architectures for developing domain specific applications, training smaller-scale models, and downloading pre-trained model weights online to use as a starting point. This group of users also include machine learning artists who use models to enable and showcase new forms of artistic expression.

An example visual analytics system for these model users is ActiVis [39]: a visual analytics system for interpreting the results of neural networks by using a novel visual representation that unifies instance- and subset-level inspections of neuron activations. Model users can flexibly specify subsets using input features, labels, or any intermediate outcomes

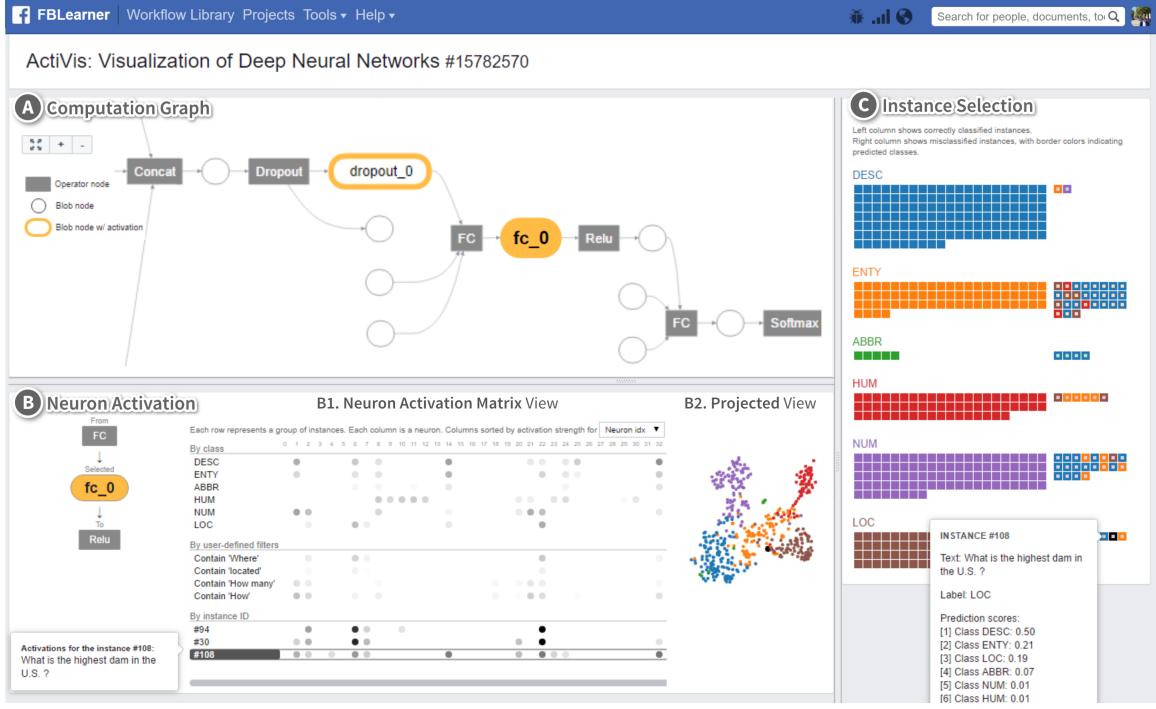


Figure 5.3: ActiVis [39]: a visual analytics system for interpreting neural network results using a novel visualization that unifies instance- and subset-level inspections of neuron activations deployed at Facebook.

in a machine learning pipeline. ActiVis was built for engineers and data scientists at Facebook to explore and interpret deep learning models results and is deployed on Facebook’s internal system. LSTMVis [143] is a visual analysis tool for recurrent neural networks with a focus on understanding hidden state dynamics in sequence modeling. The tool allows model users to perform hypothesis testing by selecting an input range to focus on local state changes, then to match these states changes to similar patterns in a large dataset, and finally align the results with structural annotations. The LSTMVis work describes three types of users: architects, those who wish to develop new deep learning methodologies; trainers, those who wish to apply LSTMs to a task in which they are domain experts in; and end users, those who use pretrained models for various tasks. Lastly, Embedding Projector [133], while not specifically deep learning exclusive, is a visual analytics tool to support interactive visualization and interpretation of large-scale embeddings, which are common outputs from neural network models. The work presents three important tasks that model users often perform while using embeddings; these include exploring local neighborhoods, viewing the global geometry to find clusters, and finding meaningful directions within an embedding.

5.5.3 Non-experts

The third group of people whom visualization could aid are non-experts in deep learning. These are individuals who typically have no prior knowledge about deep learning, and may or may not have a technical background. Much of the research targeted at this group is for educational purposes, trying to explain what a neural network is and how it works at a high-level, sometimes without revealing deep learning is present. These group also includes people who simply use AI-powered devices and consumer applications.

Apart from Teachable Machines [154] and the Deep Visualization Toolbox [155] mentioned in Subsection 5.4.4, TensorFlow Playground [75], a web-based interactive visualization of a simple dense network, has become a go-to tool for gaining intuition about how neural networks learn. TensorFlow Playground uses direct manipulation experimentation rather than coding, enabling users to quickly build an intuition about neural networks. The system has been used to teach students about foundational neural network properties by using “living lessons,” and also makes it straightforward to create a dynamic, interactive educational experience. Another web-browser based system, ShapeShop [159], allows users to explore and understand the relationship between input data and a network’s learned representations. ShapeShop uses a feature visualization technique called class activation maximization to visualize specific classes of an image classifier. The system allows users to interactively select classes from a collection of simple shapes, select a few hyperparameters, train a model, and view the generated visualizations all in real-time.

Tools built for non-experts, particularly with an educational focus, are becoming more popular on the web. A number of web-based JavaScript frameworks for training neural networks and inference have been developed; however, ConvNetJS (<http://cs.stanford.edu/people/karpathy/convnet.js/>) and TensorFlow.js (<https://js.tensorflow.org/>) are the most used and have enabled developers to create highly interactive explorable explanations for deep learning models.

5.6 What to Visualize in Deep Learning

This section discusses the technical components of neural networks that could be visualized. This section is strongly related to the next section, Section 5.7 “How,” which describes how the components of these networks are visualized in existing work. By first describing *what* may be visualized (this section), we can more easily ground our discussion on *how* to visualize them (next section).

5.6.1 Computational Graph & Network Architecture

The first thing that can be visualized in a deep learning model is the model architecture. This includes the *computational graph* that defines how a neural network model would train, test, save data to disk, and checkpoint after epoch iterations [150]. Also called the dataflow graph [150], this defines how data flows from operation to operation to successfully train and use a model. This is different than the neural network's edges and weights, discussed next, which are the parameters to be tweaked during training. The dataflow graph can be visualized to potentially inform model developers of the types of computations occurring within their model, as discussed in Subsection 5.7.1.

5.6.2 Learned Model Parameters

Other components that can be visualized are the learned parameters in the network during and after training.

Neural Network Edge Weights Neural network models are built of many, and sometimes diverse, constructions of layers of computational units [124]. These layers send information throughout the network by using edges that connect layers to one another, oftentimes in a linear manner, yet some more recent architectures have shown that skipping certain layers and combining information in unique ways can lead to better performance. Regardless, each node has an outgoing edge with an accompanying *weight* that sends signal from one neuron in a layer to potentially thousands of neurons in an adjacent layer [75]. These are the parameters that are tweaked during the backpropagation phase of training a deep model, and could be worthwhile to visualize for understanding what the model has learned, as seen in Subsection 5.7.1.

Convolutional Filters Convolutional neural networks are built using a particular type of layer, aptly called the *convolutional layer*. These convolutional layers apply filters over the input data, oftentimes images represented as a two-dimensional matrix of values, to generate smaller representations of the data to pass to later layers in the network. These filters, like the previously mentioned traditional weights, are then updated throughout the training process, i.e., learned by the network, to support a given task. Therefore, visualizing the learned filters could be useful as an alternate explanation for what a model has learned [68, 155], as seen in Subsection 5.7.6.

5.6.3 Individual Computational Units

Albeit reductionist, neural networks can be thought as a collection of layers of neurons connected by edge weights. Above, we discussed that the edges can be visualized, but the neurons too can be a source of data to investigate.

Activations When given a trained model, one can perform inference on the model using a new data instance to obtain the neural network’s output, e.g., a classification or a specific predicted value. Throughout the network, the neurons compute *activations* using activation functions (e.g., weighted sum) to combine the signal from the previous layer into a new node [124, 155]. This mapping is one of the characteristics that allows a neural network to learn. During inference, we can recover the activations produced at each layer. We can use activations in multiple ways, e.g., as a collection of individual neurons, spatial positions, or channels [70]. Although these feature representations are typically high-dimensional vectors of the input data at a certain stage within the network [70], it could be valuable in helping people visualize how input data is transformed into higher-level features, as seen in Subsection 5.7.2. Feature representations may also shed light upon how the network and its components respond to particular data instances [155], commonly called instance-level observation; we will discuss this in detail in Subsection 5.7.4 and Subsection 5.7.5.

Gradients for Error Measurement To train a neural network, we commonly use a process known as backpropagation [124]. *Backpropagation*, or sometimes called the backpropagation of errors, is a method to calculate the gradient of a specified loss function. When used in combination with an optimization algorithm, e.g., gradient descent, we can compute the error at the output layer of a neural network and redistribute the error by updating the model weights using the computed gradient. These gradients flow over the same edges defined in the network that contain the weights, but flow in the opposite direction., e.g., from the output layer to the input layer. Therefore, it could be useful to visualize the gradients of a network to see how much error is produced at certain outputs and where it is distributed [160, 161], as mentioned in Subsection 5.7.6.

5.6.4 Neurons in High-dimensional Space

Continuing the discussion of visualizing activations of a data instance, we can think of the feature vectors recovered as vectors in a high-dimensional space. Each neuron in a layer then becomes a “dimension.” This shift in perspective is powerful, since we can now take advantage of high-dimensional visualization techniques to visualize extracted activa-

tions [162, 163]. Sometimes, people use neural networks simply as feature vector generators, and defer the actual task to other computational techniques, e.g., traditional machine learning models [129, 65]. In this perspective, we now can think of deep neural networks as feature generators, whose output embeddings could be worth exploring. A common technique is to use dimensionality reduction to take the space spanned by the activations and embed it into 2D or 3D for visualization purposes [163, 133, 162], as discussed in Subsection 5.7.2.

5.6.5 Aggregated Information

Groups of Instances As mentioned earlier, instance-level activations allow one to recover the mapping from data input to a feature vector output. While this can be done for a single data instance, it can also be done on collections of instances. While at first this does not seem like a major differentiation from before, instance groups provide some unique advantages [39, 152]. For example, since instance groups by definition are composed of many instances, one can compute all the activations simultaneously. Using visualization, we can now compare these individual activations to see how similar or different they are from one another. Taking this further, with instance groups, we can now take multiple groups, potentially from differing classes, and compare how the distribution of activations from one group compares or differs from another. This aggregation of known instances into higher-level groups could be useful for uncovering the learned decision boundary in classification tasks, as seen in Subsection 5.7.2 and Subsection 5.7.4.

Model Metrics While instance- and group-level activations could be useful for investigating how neural networks respond to particular results a-priori, they suffer from scalability issues, since deep learning models typically wrangle large datasets. An alternative object to visualize are model metrics, including loss, accuracy, and other measures of error [150]. These summary statistics are typically computed every epoch and represented as a time series over the course of a model’s training phase. Representing the state of a model through a single number, or handful of numbers, abstracts away much of the subtle and interesting features of deep neural networks; however, these metrics are key indicators for communicating how the network is progressing during the training phase [144]. For example, is the network “learning” anything at all or is it learning “too much” and is simply memorizing data causing it to overfit? Not only do these metrics describe notions of a single model’s performance over time, but in the case of model comparison, these metrics become more important, as they can provide a quick and easy way to compare multiple models at once. For this reason, visualizing model metrics can be an important

and powerful tool to consider for visual analytics, as discussed in Subsection 5.7.3.

5.7 How to Visualize Deep Learning

In the previous section, we described what technical components of neural networks could be visualized. In this section, we summarize how the components are visualized and interacted with in existing literature. For most neural network components, they are often visualized using a few common approaches. For example, network architectures are often represented as node-link diagrams; embeddings of many activations are typically represented as scatter plots; and model metrics over epoch time are almost always represented as line charts. In this section, we will also discuss other representations, going beyond the typical approaches.

5.7.1 Node-link Diagrams for Network Architectures

Given a neural network’s dataflow graph or model architecture, the most common way to visualize where data flows and the magnitude of edge weights is a node-link diagram. Neurons are shown as nodes, and edge weights as links. For computational and dataflow graphs, Kahng et al. [39] describe two methods for creating node-link diagrams. The first represents only operations as nodes, while the second represents both operations and data as nodes. The first way is becoming the standard due to the popularity of TensorBoard [150] and the inclusion of its interactive dataflow graph visualization [74]. However, displaying large numbers of links from complex models can generate “hairball” visualizations where many edge crossings impede pattern discovery. To address this problem, Wongsuphasawat et al. [74] extracts high-degree nodes (responsible for many of the edge crossings), visualizes them separately from the main graph, and allow users to define super-groups within the code. Another approach to reduce clutter is to place more information on each node; DGM-Tracker [69] provides a quick snapshot of the dataflow in and out of a node by visualizing its activations within each node.

Regarding neural network architecture, many visual analytics systems use node-link diagrams (neurons as nodes, weights as links) [117, 75, 77, 73, 160]. The weight magnitude and sign can then be encoded using color or link thickness. This technique was one of the the first to be proposed [117], and the trend has continued on in literature. Building on this technique, Harley [77] visualizes the convolution windows on each layer and how the activations propagate through the network to make the final classification. Similar to the dataflow graph examples above, some works include richer information inside each node besides an activation value, such as showing a list of images that highly activate

Activation patterns are more discernible as data flows through the network

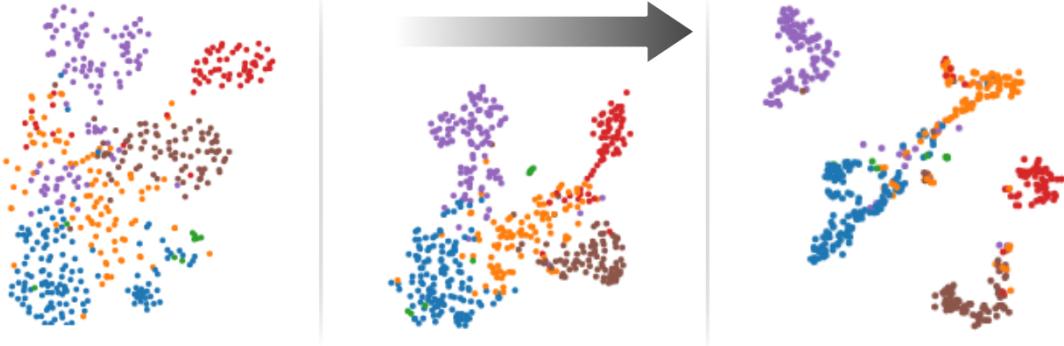


Figure 5.4: Each point is a data instance’s high-dimensional activations at a particular layer inside of a neural network, dimensionally reduced, and plotted in 2D. Notice as the data flows through the network the activation patterns become more discernible (left to right) [39].

that neuron or the activations at a neuron as a matrix [73]. As mentioned in the dataflow graph visualizations, node-link diagrams for network architecture work well for smaller networks [75], but they also suffer from scalability issues. CNNVis [73], a visual analytics system that visualizes convolutional neural networks, proposes to use a bi-clustering-based edge bundling technique to reduce visual clutter caused by too many links.

5.7.2 Dimensionality Reduction & Scatter Plots

In Section 5.6, “What,” we discussed different types of high-dimensional embeddings: text can be represented as vectors in word embeddings for natural language processing and images can be represented as feature vectors inside of a neural network. Both of these types of embeddings are mathematically represented as large tensors, or sometimes as 2D matrices, where each row may correspond to an instance and each column a feature.

The most common technique to visualize these embeddings is performing dimensionality reduction to reduce the number of columns (e.g., features) to two or three. Projecting onto two dimensions would mean computing (x, y) coordinates for every data instance; for three dimensions, we compute an additional z component, resulting in (x, y, z) . In the 2D case, we can plot all data instances as points in a scatter plot where the axes may or may not have interpretable meaning, depending on the reduction technique used, e.g., principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE) [163]. In the 3D case, we can still plot each data instance as a point in 3D space and use interactions to pan, rotate, and navigate this space [133]. These types of embeddings are often included in visual analytics systems as one of the main views [144, 160], and are also used in application papers as static figures [128, 126]. However, viewing a 3D space on a 2D

medium (e.g., computer screen) may not be ideal for tasks like comparing exact distances.

Since each reduced point corresponds to an original data instance, another common approach is to retrieve the original image and place it at the reduced coordinate location. Although the image size must be greatly reduced to prevent excessive overlap, viewing all the images at once can provide insight into what a deep learning model has learned, as seen in the example in [151] where the authors visualize ImageNet test data, or in [164] where the authors create many synthetic images from a single class and compare the variance across many random initial starting seeds for the generation algorithm. We have discussed the typical case where each dot in the scatter plot is a data instance, but some work has also visualized neurons in a layer as separate data instances [145]. Another work studies closely how data instances are transformed as their information is passed through the deep network, which in effect visualizes how the neural network separates various classes along approximated decision boundaries [162]. It is also possible to use time-dependent data and visualize how an embedding changes over time, or in the case of deep learning, over epochs [165]. This can be useful for evaluating the quality of the embedding during the training phase.

However, these scatter plots raise problems too. The quality of the embeddings greatly depends on the algorithm used to perform the reduction. Some works have studied how PCA and t-SNE differ, mathematical and visually, and suggest new reduction techniques to capture the semantic and syntactic qualities within word embeddings [166]. It has also been shown that popular reduction techniques like t-SNE are sensitive to changes in the hyperparameter space. Wattenberg meticulously explores the hyperparameter space for t-SNE, and offers lessons learned and practical advice for those who wish to use dimensionality reduction methods [156]. While these techniques are commonplace, there are still iterative improvements that can be done using clever interaction design, such as finding instances similar to a target instance, i.e., those “near” the target in the projected space, helping people build intuition for how data is spatially arranged [133].

5.7.3 Line Charts for Temporal Metrics

Model developers track the progression of their deep learning models by monitoring and observing a number of different metrics computed after each epoch, including the loss, accuracy, and different measure of errors. This can be useful for diagnosing the long training process of deep learning models. The most common visualization technique for visualizing this data is by considering the metrics as time series and plotting them in line charts [150]. This approach is widely used in deep learning visual analytics tools [144, 160]. After each epoch, a new entry in the time series is computed, therefore some tools, like TensorBoard,

run alongside models as they train and update with the latest status [150]. TensorBoard focuses much of its screen real-estate to these types of charts and supports interactions for plotting multiple metrics in small multiples, plotting multiple models on top of one another, filtering different models, providing tooltips for the exact metric values, and resizing charts for closer inspection. This technique appears in many visual analytics systems and has become a staple for model training, comparison, and selection.

5.7.4 Instance-based Analysis & Exploration

Another technique to help interpret and debug deep learning models is testing specific data instances to understand how they progress throughout a model. Many experts have built up their own collection of data instances over time, having developed deep knowledge about their expected behaviors in models while also knowing their ground truth labels [39, 35]. For example, an instance consisting of a single image or a single text phrase is much easier to understand than an entire image dataset or word embedding consisting of thousands of numerical features extracted from an end user’s data. This is called instance-level observation, where intensive analysis and scrutiny is placed on a single data instance’s transformation process throughout the network, and ultimately its final output.

Identifying & Analyzing Misclassified Instances

One application of instance-level analysis is using instances as unit tests for deep learning models. In the best case scenario, all the familiar instances are classified or predicted correctly; however, it is important to understand *when* a specific instance can fail and *how* it fails. For example, in the task of predicting population from satellite imagery, the authors showcase three maps of areas with high errors by using a translucent heatmap overlaid on the satellite imagery [129]. Inspecting these instances reveals three geographic areas that contain high amounts of man-made features and signs of activity but have no registered people living in them: an army base, a national lab, and Walt Disney World. The visualization helps demonstrate that the proposed model is indeed learning high-level features about the input data. Another technique, HOGgles [167], uses an algorithm to visualize feature spaces by using object detectors while inverting visual features back to natural images. The authors find that when visualizing the features of misclassified images, although the classification is wrong in the image space, they look deceptively similar to the true positives in the feature space. Therefore, by visualizing feature spaces of misclassified instances, we can gain a more intuitive understanding of recognition systems.

For textual data, a popular technique for analyzing particular data instances is to use

color as the primary encoding. For example, the background of particular characters in a phrase of words in a sentence would be colored using a divergent color scheme according to some criteria, often their activation magnitudes [135, 136, 168]. This helps identify particular data instances that may warrant deeper inspection (e.g., those misclassified) [35].

When pre-defined data instances are not unavailable (e.g., when analyzing a new dataset), how can we guide users towards important and interesting instances? To address this problem, a visual analytics system called *Blocks* [71] uses confusion matrices, a technique for summarizing the performance of a classification algorithm, and matrix-level sorting interactions to reveal that class error often occurs in hierarchies. *Blocks* incorporates these techniques with a sample viewer in the user interface to show selected samples potentially worth exploring.

Analyzing Groups of Instances

Instead of using individual data instances for testing and debugging a model, it is also common for experts to perform similar tasks using groups of instances [35]. While some detail may be lost when performing group-level analysis it allows experts to further test the model by evaluating its average and aggregate performance across different groups.

Much of the work using this technique is done on text data using LSTM models [143]. Some approaches compute the saliency for groups of words across the model and visualize the values as a matrix [134], while others use matrix visualizations to show the activations of word groups when represented as feature vectors in word embeddings [146, 169]. One system, ActiVis [39], places instance group analysis at the focus of its interactive interface, allowing users to compare preset and user-defined groups of activations. Similar to the matrix visualization that summarizes activations for each class in CNNVis [73], ActiVis also uses a scrolling matrix visualization to unify both instance-level and group-level analysis into a single view where users can compare the activations of the user-defined instances.

However, sometimes it can be challenging to define groups for images or text. For textual data, people often use words to group documents and provide aggregated data. ConceptVector [170] addresses the instance group generation problem by providing an interactive interface to create interesting groups of concepts for model testing. Furthermore, this system also suggests additional words to include in the user-defined groups, helping guide the user to create semantically sound concepts.

5.7.5 Interactive Experimentation

Interactive experimentation, another interesting area that integrates deep learning visualization, makes heavy use of interactions for users to experiment with models [5]. By using direct manipulation for testing models, a user can pose “what if?” questions and observe how the input data affects the results. Called *explorable explanations* [171], this type of visual experimentation is popular for making sense of complex concepts and systems.

Models Responding to User-provided Input Data

To engage the user with the desired concepts to be taught, many systems require the user to provide some kind of input data into the system to obtain results. Some visual analytics systems use a webcam to capture live videos, and visualize how the internals of neural network models respond to these dynamic inputs [155]. Another example is a 3D visualization of a CNN trained on the classic MNIST dataset that shows the convolution windows and activations on images that the user draws by hand [77]. MNIST is a small, popular dataset consisting of thousands of 28×28 px images of handwritten digits (0 to 9). MNIST is commonly used as a benchmark for image classification models¹. For example, drawing a “5” in the designated area passes that example throughout the network and populates the visualization with the corresponding activations using a node-link diagram. A final example using image data is ShapeShop [159], a system that allows a user to select data from a bank of simple shapes to be classified. The system then trains a neural network and using the class activation maximization technique to generate visualizations of the learned features of the model. This can be done in real-time, therefore a user can quickly train multiple models with different shapes to observe the effect of adding more diverse data to improve the internal model representation.

An example using textual data is the online, interactive Distill article for handwriting prediction [136], which allows a user to write words on screen, and in real-time, the system draws multiple to-be-drawn curves predicting what the user’s next stroke would be, while also visualizing the model’s activations. Another system uses GANs to interactively generate images based off of user’s sketches [172]. By sketching a few colored lines, the system presents the user with multiple synthetic images using the sketch as a guideline for what to generate. A final example is the Adversarial Playground [173], a visual analytics system that enables users to compare adversarially-perturbed images, to help users understand why an adversarial example can fool a CNN image classifier. The user can select from one of the MNIST digits and adjust the strength of adversarial attack. The system then compares

¹<http://yann.lecun.com/exdb/mnist/>

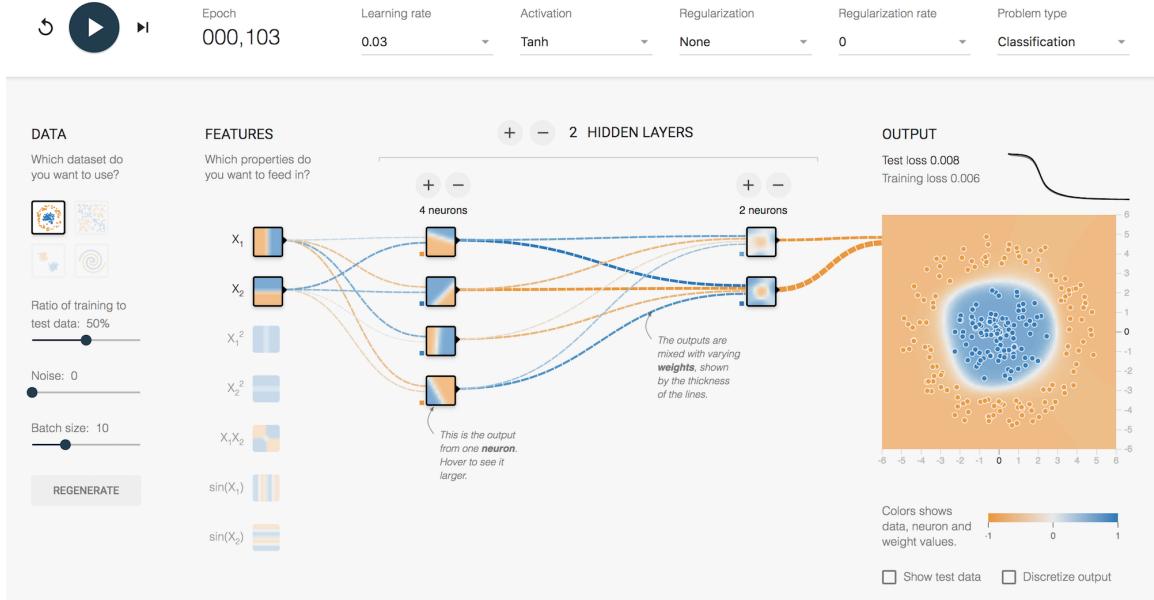


Figure 5.5: TensorFlow Playground [75]: a web-based visual analytics tool for exploring simple neural networks that uses direct manipulation rather than programming to teach deep learning concepts and develop an intuition about how neural networks behave.

the classifications scores in a bar chart to observe how simple perturbations can greatly impact classification accuracy.

How Hyperparameters Affect Results

While deep learning models automatically adjust their internal parameters, their hyperparameters still require fine-tuning. These hyperparameters can have major impact on model performance and robustness. Some visual analytics systems expose model hyperparameters to the user for interactive experimentation. One example previously mentioned is TensorFlow Playground [75], where users can use direct manipulation to adjust the architecture of a simple, fully-connected neural network, as well as the hyperparameters associated with its training, such as the learning rate, activation function, and regularization. Another example is a Distill article that meticulously explores the hyperparameters of the t-SNE dimensionality reduction method [156]. This article tests dozens of synthetic datasets in different arrangements, while varying hyperparameters such as the t-SNE perplexity and the number of iterations to run the algorithm for.

5.7.6 Algorithms for Attribution & Feature Visualization

The final method for how to visualize deep learning hails from the AI and computer vision communities. These are algorithmic techniques that entail image generation. Given

a trained a model, one can select a single image instance and use one of the algorithmic techniques to generate a new image of the same size that either highlights important regions of the image (often called *attribution*) or is an entirely new image that supposedly is representative of the same class (often called *feature visualization*) [62, 70]. In these works, it is common to see large, full-page figures consisting of hundreds of such images corresponding to multiple images classes [174]. However, it is uncommon to see interactivity in these works, as the primary contribution is often about algorithms, not interactive techniques or systems. Since the focus of this interrogative survey is on visual analytics in deep learning, we do not discuss in detail the various types of algorithmic techniques. Rather, we mention the most prominent techniques developed, since they are impactful to the growing field of deep learning visualization and could be incorporated into visual analytics systems in the future. For more details about these techniques, such as input modification, deconvolutional methods [68], and input reconstruction methods, we refer our readers to the taxonomies [175] and literature surveys for visualizing learned features in CNNs [120, 176], and a tutorial that presents the theory behind many of these interpretation techniques and discusses tricks and recommendations to efficiently use them on real data [17].

Heatmaps for Attribution, Attention, & Saliency One research area generates translucent heatmaps that overlay images to highlight important regions that contribute towards classification and their sensitivity [65, 177, 178, 138, 179]. One technique called visual backpropagation attempts to visualize which parts of an image have contributed to the classification, and can do so in real-time in a model debugging tool for self-driving vehicles [130]. Another technique is to invert representations, i.e., attempt to reconstruct an image using a feature vector to understand the what a CNN has learned [180, 181, 151]. Prediction difference analysis is a method that highlights features in an image to provide evidence for or against a certain class[131]. Other work hearkens back to more traditional computer vision techniques by exploring how object detectors emerge in CNNs and attempts to give humans object detector vision capabilities to better align humans and deep learning vision for images [182, 167]. Visualizing CNN filters is also popular, and has famously shown to generate dream-like images, becoming popular in artistic tasks [64, 183]. Some work for interpreting *visual question answering (VQA)* models and tasks use these heatmaps to explain which parts of an image a VQA model is looking at in unison with text activation maps when answering the given textual questions [168]. However, recent work has shown that some of these methods fail to provide correct results and argue that we should develop explanation methods that work on simpler models before extending them to the more complex ones [176].

Feature Visualization For feature visualization, while some techniques have proven interesting [184], one of the most studied techniques, class activation maximization, maximizes the activation of a chosen, specific neuron using an optimization scheme, such as gradient ascent, and generates synthetic images that are representative of what the model has learned about the chosen class [66]. This led to a number of works improving the quality of the generated images. Some studies generated hundreds of these non-deterministic synthetic images and clustered them to see how variations in the class activation maximization algorithm affects the output image [164]. In some of their most recent work on this topic, Ngyuen et al. [139] present hundreds of high-quality images using a deep generator network to improve upon the state-of-the-art, and include figures comparing their technique to many of the existing and previous attempts to improve the quality of generated images. The techniques developed in this research area have improved dramatically over the past few years, where now it is possibly to synthetically generate photorealistic images [185]. A recent comparison of feature visualization techniques highlights their usefulness [62]; however, the authors note that they remain skeptical of their trustworthiness, e.g., do neurons have a consistent meaning across different inputs, and if so, is that meaning accurately reified by feature visualization [70]?

5.8 When to Visualize in the Deep Learning Process

This section describes when visualizing deep learning may be most relevant and useful. Our discussion primarily centers around the training process: an iterative, foundational procedure for using deep learning models. We identify two distinct, non-mutually exclusive times for when to visualize: *during training* and *after training*. Some works propose that visualization be used both during and after training.

5.8.1 During Training

Artificial neural networks learn higher-level features that are useful for class discrimination as training progress [186]. By using visualization during the training process, there is potential to monitor one’s model as it learns to closely observe and track the model’s performance [162].

Many of the systems in this category run in a separate web-browser alongside the training process, and interface with the underlying model to query the latest model status. This way, users can visually explore and rigorously monitor their models in real time, while they are trained elsewhere. The visualization systems dynamically update the charts with metrics recomputed after every epoch, e.g., the loss, accuracy, and training time. Such metrics

are important to model developers because they rely on them to determine if a model (1) has begun to learn anything at all; (2) is converging and reaching the peak of its performance; or (3) has potentially overfitted and memorized the training data. Therefore, many of the visual analytics systems used during training support and show these updating visualizations as a primary view in the interface [150, 75, 160, 144, 69, 147]. One system, Deep View [145], visualizes model metrics during the training process and uses its own defined metrics for monitoring (rather than the loss): a discriminability metric, which evaluates neuron evolution, and a density metric which evaluates the output feature maps. This way, for detecting overfitting, the user does not need to wait long to view to infer overfitting; they simply observe the neuron density early in training phase.

Similarly, some systems help reduce development time and save computational resources by visualizing metrics that indicate whether a model is successfully learning or not, allowing a user to stop the training process early [75]. By using visualization during model training, users can save development time through model steering [160] and utilizing suggestions for model improvement [147]. Lastly, another model development time minimization focuses on diagnosing neurons and layers that are not training correctly or are misclassifying data instances. Examples include DeepEyes [144], a system that identifies stable and unstable layers and neurons so users may prune their models to speed up training; *Blocks* [71], a system that visualizes class confusion and reveals that confusion patterns follow a hierarchical structure over the classes which can then be exploited to design hierarchy-aware architectures; and DGMTracker [69], a system that proposes a credit assignment algorithm that indicates how other neurons contribute to the output of particular failing neurons.

5.8.2 After Training

While some works support neural network design during the iterative model building process, there are other works that focus their visualization efforts after a model has been trained. In other words, these works assume a trained model as input to the system or visualization technique. Note that many, if not most, of the previously mentioned algorithmic techniques developed in the AI fields, such as attribution and feature visualization, are performed after training. These techniques are discussed more in Subsection 5.7.6.

The Embedding Projector [133] specializes in visualizing 2D and 3D embeddings produced by trained neural networks. While users can visualize typical high-dimensional datasets in this tool, the Embedding Projector tailors the experience towards embeddings commonly used deep learning. Once a neural network model has been trained, one can compute the activations for a given test dataset and visualize the activations in the Embed-

ding Projector to visualize and explore the space that the network has learned. Instead of generating an overview embedding, another previously discussed system, the Deep Visualization Toolbox [155], uses a trained model to visualize live activations in a large small-multiples view to understand of what types of filters a convolutional network has learned.

More traditional visual analytics systems have also been developed to inspect a model after it has finished training. ActiVis [39], a visual analytics system for neural network interpretation deployed at Facebook reports that Facebook engineers and data scientists use visual analytics systems often in their normal workflow. Another system, RNNVis [152], visualizes and compares different RNN models for various natural language processing tasks. This system positions itself as a natural extension of TensorFlow; using multiple TensorFlow models as input, the system then analyzes the trained models to extract learned representations in hidden states, and further processes the evaluation results for visualization. Lastly, the LSTMVis [143] system, a visual analysis tool for RNN interpretability, separates model training from the visualization. This system takes a model as input that must be trained separately, and from the model, gathers the required information to produce the interactive visualizations to be rendered in a web-based front-end.

5.9 Where is Deep Learning Visualization

For the last question of the interrogative survey, we divide up “Where” into two subsections: where deep learning visualization research has been applied, and where deep learning visualization research has been conducted, describing the new and hybrid community. This division provides a concise summary for practitioners who wish to investigate the usage of the described techniques for their own work, and provides new researchers with the main venues for this research area to investigate existing literature.

5.9.1 Application Domains & Models

While many non-neural approaches are used for real-world applications, deep learning has successfully achieved state-of-the-art performance in several domains. Previously in Subsection 5.4.1, we presented works that apply neural networks to particular domains and use visualizations to lend qualitative support to their usual quantitative results to strengthen users’ trust in their models. These domains included neural machine translation [128], reinforcement learning [126], social good [129], autonomous vehicles [130], medical imaging diagnostics [131], and urban planning [132].

Next we summarize the types of models that have been used in deep learning visualization. Much of the existing work has used image-based data and models, namely CNNs,

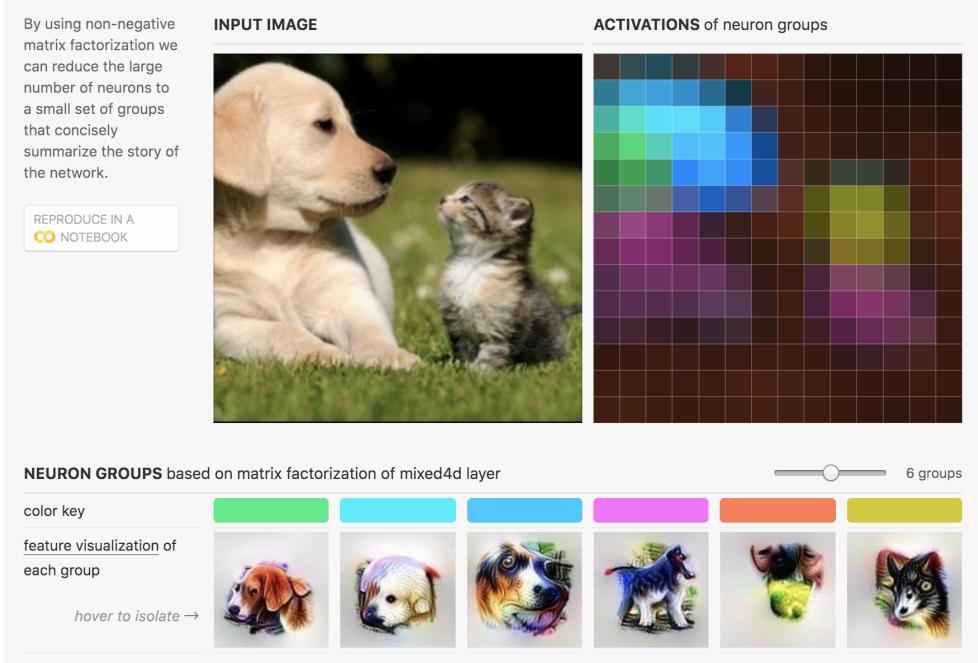


Figure 5.6: Distill: The Building Blocks of Interpretability [70]: an interactive user interface that combines feature visualization and attribution techniques to interpret neural networks.

to generate attribution and feature visualization explanations for what a model has learned from an image dataset. CNNs, while not exclusively used for images, have become popular in the computer vision community and are often used for image classification and interactive, image-based creative tasks [172, 187]. Besides images, sequential data (e.g., text, time series data, and music) has also been studied. This research stems from the natural language processing community, where researchers typically favor RNNs for learning representations of large text corpora. These researchers make sense of large word embeddings by using interactive tools that support dimensionality reduction techniques to solve problems such as sequence-to-sequence conversion, translation, and audio recognition. Research combining both image and text data has also been done, such as image captioning and visual question answering [188, 189]. Harder still are new types of networks called *generative adversarial networks*, or GANs for short, that have produced remarkable results for data generation [190], e.g., producing real-looking yet fake images [191]. While GANs have only existed for a couple of years, they are now receiving significant research attention. To make sense of the learned features and distributions from GANs, two visual analytics systems, DGMTracker [69] and GANViz [192], focus on understanding the training dynamics of GANs to help model developers better train these complex models, often consisting of multiple dueling neural networks.

5.9.2 A Vibrant Research Community: Hybrid, Apace, & Open-sourced

As seen from this survey, bringing together the visualization communities with the AI communities has led to the design and development of numerous tools and techniques for improving deep learning interpretability and democratization. This hybrid research area has seen accelerated attention and interest due to its widespread impact, as evidenced by the large number of works published in just a few years, as seen in Figure 5.2. A consequence of this rapid progress is that deep learning visualization research are being disseminated across multiple related venues. In academia, the premiere venues for deep learning visualization research consists of two main groups: the information visualization and visual analytics communities; and the artificial intelligence and deep learning communities. Furthermore, since this area is relatively new, it has seen more attention at multiple workshops at the previously mentioned academic conferences, as tabulated in Table 5.1.

Another consequence of this rapidly developing area is that new work is immediately publicized and open-sourced, without waiting for it to be “officially” published at conferences, journals, etc. Many of these releases take the form of a preprint publication posted on arXiv, where a deep learning presence has thrived. Not only is it common for academic research labs and individuals to publish work on arXiv, but companies from industry are also publishing results, code, and tools. For example, the most popular libraries² for implementing neural networks are open-source and have consistent contributions for improving all areas of the codebase, e.g., installation, computation, and deployment into specific programming languages’ open-source environments.

Some works have a corresponding blog post on an industry research blog,³ which, while non-traditional, has large impact due to their prominent visibility and large readership. While posting preprints may have its downsides (e.g., little quality control) the communities have been promoting the good practices of open-sourcing developed code and including direct links within the preprints; both practices are now the norm. Although it may be overwhelming to digest the amount of new research published daily, having access to the work with its code could encourage reproducibility and allow the communities to progress faster. In summary, given the increasing interest in deep learning visualization research and its importance, we believe our communities will continue to thrive, and will positively impact many domains for years to come.

²Popular libraries include TensorFlow [150], Keras, PyTorch, Caffe, PyTorch, and Theano.

³High impact industry blogs include: Google Research Blog, OpenAI, Facebook Research Blog, the Apple Machine Learning Journal, NVIDIA Deep Learning AI, and Uber AI

5.10 Conclusion

We presented a comprehensive, timely survey on visualization and visual analytics in deep learning research, using a human-centered, interrogative framework. Our method helps researchers and practitioners in visual analytics and deep learning to quickly learn key aspects of this young and rapidly growing body of research, whose impact spans a broad range of domains. Our survey goes beyond visualization-focused venues to extend a wide scope that also encompasses relevant works from top venues in AI, ML, and computer vision. We highlighted visual analytics as an integral component in addressing pressing issues in modern AI, helping to discover and communicate insight, from discerning model bias, understanding models, to promoting AI safety.

CHAPTER 6

SUMMIT: VISUALIZING ACTIVATION AND ATTRIBUTION SUMMARIZATIONS

Deep learning is increasingly used in decision-making tasks. However, understanding how neural networks produce final predictions remains a fundamental challenge. Existing work on interpreting neural network predictions for images often focuses on explaining predictions for single images or neurons. As predictions are often computed from millions of weights that are optimized over millions of images, such explanations can easily miss a bigger picture. We present SUMMIT, an interactive system that scalably and systematically summarizes and visualizes what features a deep learning model has learned and how those features interact to make predictions. SUMMIT introduces two new scalable summarization techniques: (1) *activation aggregation* discovers important neurons, and (2) *neuron-influence aggregation* identifies relationships among such neurons. SUMMIT combines these techniques to create the novel *attribution graph* that reveals and summarizes crucial neuron associations and substructures that contribute to a model’s outcomes. SUMMIT scales to large data, such as the ImageNet dataset with 1.2M images, and leverages neural network feature visualization and dataset examples to help users distill large, complex neural network models into compact, interactive visualizations. We present neural network exploration scenarios where SUMMIT helps us discover multiple surprising insights into a prevalent, large-scale image classifier’s learned representations and informs future neural network architecture design. The SUMMIT visualization runs in modern web browsers and is open-sourced.

6.1 Introduction

Deep learning is increasingly used in decision-making tasks, due to its high performance on previously-thought hard problems and a low barrier to entry for building, training, and deploying neural networks. Inducing a model to discover important features from a dataset is a powerful paradigm, yet this introduces a challenging *interpretability* problem — it is hard for people to understand what a model has learned. This is exacerbated in situations where a model could have impact on a person’s safety, financial, or legal status [23]. Definitions of interpretability center around *human understanding*, but they vary in the aspect of the model to be understood: its internals [15], operations [16], mapping of data [17], or

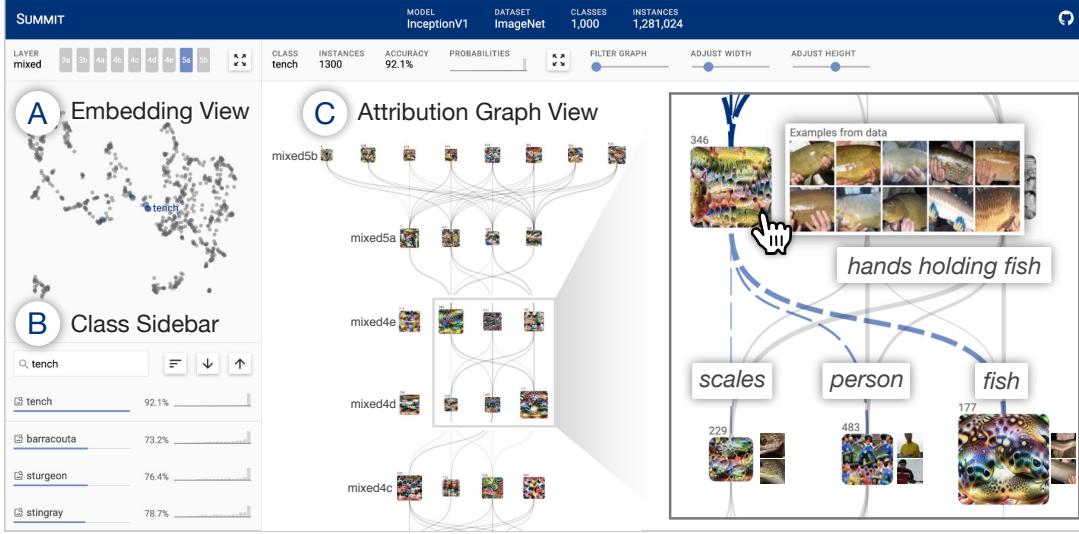


Figure 6.1: With SUMMIT, users can scalably summarize and interactively interpret deep neural networks by visualizing *what* features a network detects and *how* they are related. In this example, INCEPTIONV1 accurately classifies images of **tench** (yellow-brown fish). However, SUMMIT reveals surprising associations in the network (e.g., using parts of people) that contribute to its final outcome: the “tench” prediction is dependent on an intermediate “hands holding fish” feature (right callout), which is influenced by lower-level features like “scales,” “person,” and “fish”. **(A) Embedding View** summarizes all classes’ aggregated activations using dimensionality reduction. **(B) Class Sidebar** enables users to search, sort, and compare all classes within a model. **(C) Attribution Graph View** visualizes highly activated neurons as vertices (“scales,” “fish”) and their most influential connections as edges (dashed purple edges).

representation [18]. Although recent work has begun to operationalize interpretability [7], a formal, agreed-upon definition remains open [19, 20].

Existing work on interpreting neural network predictions for images often focuses on explaining predictions for single images or neurons [193, 178, 62, 70]. As large-scale model predictions are often computed from millions of weights optimized over millions of images, such explanations can easily miss a bigger picture. Knowing how entire classes are represented inside of a model is important for trusting a model’s predictions and deciphering what a model has learned [18], since these representations are used in diverse tasks like detecting breast cancer [194, 195], predicting poverty from satellite imagery [84], defending against adversarial attacks [196], transfer learning [197, 198], and image style transfer [199]. For example, high-performance models can learn unexpected features and associations that may puzzle developers. Conversely, when models perform poorly, developers need to understand their causes to fix them [39, 18]. As demonstrated in Figure 6.1, INCEPTIONV1, a prevalent, large-scale image classifier, accurately classifies images of **tench** (yellow-brown fish). However, our system, SUMMIT, reveals surprising associations in the network that contribute to its final outcome: **tench** is dependent on an intermediate person-

related “hands holding fish” feature (right callout) influenced by lower-level features like “*scales*,” “*person*,” and “*fish*”. There is a lack of research in developing scalable summarization and interactive interpretation tools that simultaneously reveal important neurons and their relationships. SUMMIT aims to fill this critical research gap.

Contributions. In this work, we contribute:

- **SUMMIT, an interactive system for scalable summarization and interpretation** for exploring entire learned classes in prevalent, large-scale image classifier models, such as INCEPTIONV1 [110]. SUMMIT leverages neural network feature visualization [62, 63, 64, 65, 66] and dataset examples to distill large, complex neural network models into compact, interactive graph visualizations (Section 6.6).
- **Two new scalable summarization techniques** for deep learning interpretability: (1) *activation aggregation* discovers important neurons (Subsection 6.5.1), and (2) *neuron-influence aggregation* identifies relationships among such neurons (Subsection 6.5.2). These techniques scale to large data, e.g., ImageNet ILSVRC 2012 with 1.2M images [114].
- **Attribution graph, a novel way to summarize and visualize entire classes**, by combining our two scalable summarization techniques to reveal crucial neuron associations and substructures that contribute to a model’s outcomes, simultaneously highlighting *what* features a model detects, and *how* they are related (Figure 6.2). By using a graph representation, we can leverage the abundant research in graph algorithms to extract attribution graphs from a network that show neuron relationships and substructures within the entire neural network that contribute to a model’s outcomes (Subsection 6.5.3).
- **An open-source, web-based implementation** that broadens people’s access to interpretability research without the need for advanced computational resources. Our work joins a growing body of open-access research that aims to use interactive visualization to explain complex inner workings of modern machine learning techniques [171, 76, 75]. Our computational techniques for aggregating activations, aggregating influences, generating attribution graphs and their data, as well as the SUMMIT visualization, are open-sourced.¹ The system is available at the following public demo link: <https://fredhohman.com/summit/>.

¹Visualization: <https://github.com/fredhohman/summit>.
Code: <https://github.com/fredhohman/summit-notebooks>.
Data: <https://github.com/fredhohman/summit-data>.

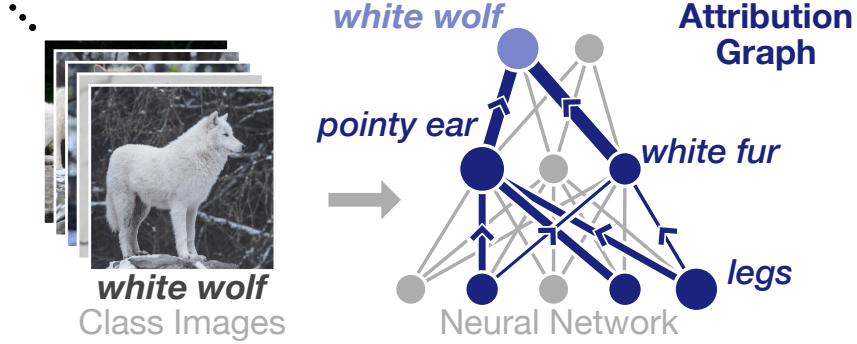


Figure 6.2: A high-level illustration of how we take thousands of images for a given class, e.g., images from *white wolf* class, compute their top activations and attributions, and combine them to form an **attribution graph** that shows how lower-level features (“legs”) contribute to higher-level ones (“white fur”), and ultimately the final outcome.

Neural network exploration scenarios. Using SUMMIT, we investigate how a widely-used computer vision model hierarchically builds its internal representation that has merely been illustrated in previous literature. We present neural network exploration scenarios where SUMMIT helps us discover multiple surprising insights into a prevalent, large-scale image classifier’s learned representations and informs future neural network architecture design (Section 6.7).

Broader impact for visualization in AI. We believe our summarization approach that builds entire class representations is an important step for developing higher-level explanations for neural networks. We hope our work will inspire deeper engagement from both the information visualization and machine learning communities to further develop human-centered tools for artificial intelligence [171, 45].

6.2 Design Challenges

Our goal is to build an interactive visualization tool for users to better understand how neural networks build their hierarchical representation. To develop our summarization techniques and design SUMMIT, we identified five key challenges.

- C1. **SCALABILITY** **Scaling up explanations and representations to entire classes, and ultimately, datasets of images.** Much of the existing work on interpreting neural networks focuses on visualizing the top independent activations or attributions for a single image [193, 178, 62, 70]. While this can be useful, it quickly becomes tiresome to inspect these explanations for more than a handful of images. Furthermore, since every image may contain different objects, to identify which concepts are rep-

representative of the learned model for a specific class, users must compare many image explanations together to manually find commonalities.

- C2. INFLUENCE** **Discovering influential connections in a network that most represents a learned class.** In dense neural network models, scalar edge weights directly connect neurons in a previous layer to neurons in a following layer; in other words, the activation of single neuron is expressed as a weighted sum of the activations from neurons in the previous layer [124]. However, this relationship is more complicated in convolutional neural networks. Images are convolved to form many 2D activation maps, that are eventually summed together to form the next layers activations. Therefore, it becomes non-trivial to determine the effect of a single convolutional filter's effect on later layers.
- C3. VISUALIZATION** **Synthesizing meaningful, interpretable visualizations with important channels and influential connections.** Given a set of top activated neurons for a collection of images, and the impact convolutional filters have on later layers, how do we combine these approaches to form a holistic explanation that describes an entire class of images? Knowing how entire classes are represented inside of a model is important for trusting a model's predictions [18], aiding decision making in disease diagnosis [194, 195], devising security protocols [196], and fixing under-performing models [39, 18].
- C4. INTERACTION** **Interactive exploration of hundreds of learned class representations in a model.** How do we support interactive exploration and interpretation of hundreds or even thousands of classes learned by a prevalent, large-scale deep learning model? Can an interface support both high-level overviews of learned concepts in a network, while remaining flexible to support filtering and drilling down into specific features? Whereas **C1** focuses on the summarization approaches to scale up representations, this challenge focuses on interaction approaches for users to work with the summarized representations.
- C5. RESEARCH ACCESS** **High barrier of entry for understanding large-scale neural networks.** Currently, deep learning models require extensive computational resources and time to train and deploy. Can we make understanding neural networks more accessible without such resources, so that everyone has the opportunity to learn and interact with deep learning interpretability?

6.3 Design Goals

Based on the identified design challenges (Section 6.2), we distill the following main design goals for SUMMIT, an interactive visualization system for summarizing what features a neural network has learned.

- G1. Aggregating activations by counting top activated channels.** Given the activations for an image, we can view them channel-wise, that is, a collection of 2D matrices where each encodes the magnitude of a detected feature by that channel’s learned filter. We aim to identify which channels have the strongest activation for a given image, so that we can record only the topmost activated channels for every image, and visualize which channels, in aggregate, are most commonly firing a strong activation (**C1**). This data could then be viewed as a feature of vector for each class, where the features are the counts of images that had a specific channel as a top channel (Subsection 6.5.1).
- G2. Aggregating influences by counting previous top influential channels.** We aim to identify the most influential paths data takes throughout a network. If aggregated for every image, we could use intermediate outputs of the fundamental convolutional operation used inside of CNNs (**C2**) to help us determine which channels in a previous layer have the most impact on future channels for a given class of images (Subsection 6.5.2).
- G3. Finding what neural networks look for, and how they interact.** To visualize how low-level concepts near early layers of a network combine to form high-level concepts towards later layers, we seek to form a graph from the entire neural network, using the aggregated influences as an edge list and aggregated activations as vertex values. With a graph representation, we could leverage the abundant research in graph algorithms, such as Personalized PageRank, to extract a subgraph that best captures the important vertices (neural network channels) and edges (influential paths) in the network (Subsection 6.5.3). Attribution graphs would then describe the most activated channels and attributed paths between channels that ultimately lead the network to a final prediction (**C3**).
- G4. Interactive interface to visualize classes attribution graphs of a model.** We aim to design and develop an interactive interface that can visualize entire attribution graphs (Section 6.6). Our goal is to support users to freely inspect any class within a large neural network classifier to understand what features are learned and how they relate

to one another to make predictions for any class (**C4**). Here, we also want to use state-of-the-art deep learning visualization techniques, such as pairing feature visualization with dataset examples, to make channels more interpretable (Subsection 6.6.3).

G5. Deployment using cross-platform, lightweight web technologies. To develop a visualization that is accessible for users without specialized computational resources, in SUMMIT we use modern web browsers to visualize attribution graphs (Section 6.6). We also open-source our code to support reproducible research (**C5**).

6.4 Model Choice and Background

In this work, we demonstrate our approach on INCEPTIONV1 [110], a prevalent, large-scale convolutional neural network (CNN) that achieves top-5 accuracy of 89.5% on the ImageNet dataset that contains over 1.2 millions images across 1000 classes. INCEPTIONV1 is composed of multiple inception modules: self-contained groups of parallel convolutional layers. The last layer of each inception module is given a name of the form “mixed{number}{letter},” where the {number} and {letter} denote the location of a layer in the network; for example, mixed3b (an earlier layer) or mixed4e (a later layer). In INCEPTIONV1, there are 9 such layers: mixed3{a,b}, mixed4{a,b,c,d,e}, and mixed5{a,b}. While there are more technical complexities regarding neural network design within each inception module, we follow existing interpretability literature and consider the 9 mixed layers as the primary layers of the network [62, 70]. Although our work makes this model choice, our proposed summarization and visualization techniques can be applied to other neural network architectures in other domains.

6.5 Creating Attribution Graphs by Aggregation

SUMMIT introduces two new scalable summarization techniques: (1) *activation aggregation* discovers important neurons, and (2) *neuron-influence aggregation* identifies relationships among such neurons. SUMMIT combines these techniques to create the novel *attribution graph* that reveals and summarizes crucial neuron associations and substructures that contribute to a model’s outcomes. Attribution graphs tell us *what* features a neural network detects, and *how* those features are related. Below, we formulate each technique, and describe how we combine them to generate attribution graphs (Subsection 6.5.3) for CNNs.

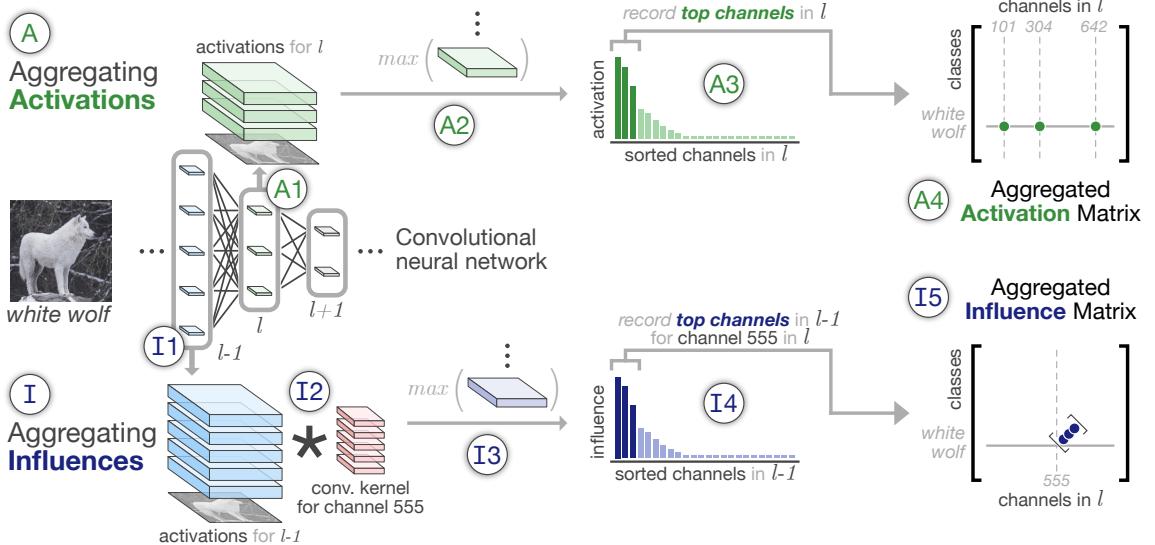


Figure 6.3: A visual depiction of our approach for aggregating activations and influences for a layer l . **Aggregating Activations:** (A1) given activations at layer l , (A2) compute the max of each 2D channel, and (A3) record the top activated channels into an (A4) aggregated activation matrix, which tells us which channels in a layer most activate and represent every class in the model. **Aggregating Influences:** (I1) given activations at layer $l-1$, (I2) convolve them with a convolutional kernel from layer l , (I3) compute the max of each resulting 2D activation map, and (I4) record the top most influential channels from layer $l-1$ that impact channels in layer l into an (I5) aggregated influence matrix, which tells us which channels in the previous layer most influence a particular channel in the next layer.

6.5.1 Aggregating Neural Network Activations

We want to understand *what* a neural network is detecting in a dataset. We propose summarizing how an image dataset is represented throughout a CNN by aggregating individual image **activations** at each channel in the network, over all of the images in a given class. This aggregation results in a matrix, A^l for each layer l in a network, where an entry A_{cj}^l roughly represents how *important* channel j (from the l^{th} layer) is for representing images from class c . This measure of importance can be defined in multiple ways, which we discuss formally below.

A convolutional layer contains C_l image kernels (parameters) that are convolved with an input image, X , to produce an output image, Y , that contains C_l corresponding channels. For simplicity, we assume that the hyperparameters of the convolutional layer are such that X and Y will have the same height H and width W , i.e., $X \in \mathbb{R}^{H \times W \times C_{l-1}}$ and $Y \in \mathbb{R}^{H \times W \times C_l}$. Each channel in Y is a matrix of values that represent how strongly the corresponding kernel *activated* in each spatial position. For example, an edge detector kernel will produce a channel, also called an activation map, that has larger values at locations where an edge

is present in the input image. As kernels in convolutional layers are learned during model training, they identify different features that discriminate between different image classes. It is commonly thought that CNNs build hierarchical feature representations of input images, learning simple edge and shape detectors in early layers of the network, which are combined to form texture detectors, and finally relevant object detectors in later layers of the network [68] (see Figure 2.1).

A decision must be made on how to aggregate activations over spatial locations in a channel and aggregate activations over all images in a given class. Ultimately, we want to determine channel importance in a CNN’s representation of a class. As channels roughly represent concepts, we choose the maximum value of a channel as an indicator of how strongly a concept is present, instead of other functions, such as mean, that may dampen the magnitude of relevant channels.

Alongside Figure 6.3, our method for aggregation is as follows:

- **Compute activation channel maximums for all images.** For each image, (A1) obtain its activations for a given layer l and (A2) compute the maximum value per channel. This is equivalent to performing Global Max-pooling at each layer in the network. Now for each layer, we will have a matrix Z^l , where an entry Z_{ij}^l represents the maximum activation of image i over the j^{th} channel in layer l .
- **Filter by a particular class.** We consider all rows of Z^l whose images belong to the same class, and want to aggregate the maximum activations from these rows to determine which channels are important for detecting the class.
- **Aggregation Method 1: taking top k_{M1} channels.** For each row, we set the top k_{M1} largest elements to 1 and others to 0, then sum over rows. Performing this operation for each class in our dataset will result in a matrix A^l from above where an entry A_{cj}^l is the count of the number of times that the j^{th} channel is one of the top k_{M1} channels by maximum activation for all images in class c . This method ignores the actual maximum activation values, so it will not properly handle cases where a single channel activates strongly for images of a given class (as it will consider $k_{M1} - 1$ other channels), or cases where many channels are similarly activated over images of a given class (as it will *only* consider k_{M1} channels as “important”). This observation motivates our second method.
- **Aggregation Method 2: taking top $k_{M2}\%$ of channels by weight.** We first scale rows of Z^l to sum to 1 by dividing by the row sums, $Z_{ij}^{l\prime} = \frac{Z_{ij}^l}{\sum_{n=1}^N Z_{nj}^l}$, where N is the number of images. Instead of setting the top k_{M2} elements to 1, as in **Method 1**, we

set the m largest elements of each row to 1 and the remaining to 0. Here, m is the largest index such that $\sum_{j \in \text{sorted}}^m Z_i^l Z_{ij}^l \leq k_{M2}$, where k_{M2} is some small percentage. In words, this method first sorts all channels by their maximum activations, then records channels, starting from the largest activated, until the cumulative sum of probability weight from the recorded channels exceeds the threshold. Contrary to **Method 1**, this method adaptively chooses channels that are important for representing a given image, producing a better final class representation.

Empirically, we noticed the histograms of max channel activations was often power law distributed, therefore we use **Method 2** to (A3) record the top $k_{M2} = 3\%$ of channels to include in the (A4) **Aggregated Activations** matrix A^l . In terms of runtime, this process requires only a forward pass through the network.

6.5.2 Aggregating Inter-layer Influences

Aggregating activations at each convolutional layer in a network will only give a local description of which channels are important for each class, i.e., from examining A^l we will not know *how* certain channels come to be the most representative for a given class. Thus, we need a way to calculate how the activations from the channels of a previous layer, $l - 1$, **influence** the activations at the current layer, l . In dense layers, this influence is trivial to compute: the activation at a neuron in l is computed as the weighted sum of activations from neurons in $l - 1$. The influence of a single neuron from $l - 1$ is then proportional to the activation of that neuron multiplied by the associated weight to the neuron being examined from l . In convolutional layers, calculating this influence is more complicated: the activations at a channel in l are computed as the 3D convolution of all of the channels from $l - 1$ with a learned kernel tensor. This operation can be broken down (shown formally later in this section) as a summation of the 2D convolutions of each channel in $l - 1$ with a corresponding slice of the appropriate kernel. The summations of 2D convolutions are similar in structure to the weighted-summations performed by dense layers, however the corresponding “influence” of a single channel from $l - 1$ on the output of a particular channel in l is a 2D feature map. We can summarize this feature map into a scalar influence value by using any type of reduce operation, which we discuss further below.

We propose a method for (1) quantifying the *influence* a channel from a previous layer has on the activations of a channel in a following layer, and (2) aggregating influences into a tensor, I^l , that can be interpreted similarly to the A^l matrix from the previous section. Formally, we want to create a tensor I^l for every layer l in a network, where an entry I_{cij}^l

represents how important channel i from layer $l - 1$ is in determining the output of channel j in layer l , for all images in class c .

First, using the notation from the previous section, we consider how a single channel of Y is created from the channels of X . Let $K^{(j)} \in \mathbb{R}^{H \times W \times C_{l-1}}$ be the j^{th} kernel of our convolutional layer. Now the operation of a convolutional layer can be written as:

$$Y_{\cdot,\cdot,j} = \underbrace{X * K^{(j)}}_{\text{3D convolution}} = \sum_{i=1}^{C_{l-1}} \underbrace{X_{\cdot,\cdot,i} * K_{\cdot,\cdot,i}^{(j)}}_{\text{2D convolution}} \quad (6.1)$$

In words, (I1) each channel from X is (I2) convolved with a slice of the j^{th} kernel, and the resulting maps are summed to produce a single channel in Y . We care about the 2D quantity $X_{\cdot,\cdot,i} * K_{\cdot,\cdot,i}^{(j)}$ as it contains exactly the contributions of a *single* channel from the previous layer to a channel in the current layer.

Second, we must summarize the quantity $X_{\cdot,\cdot,i} * K_{\cdot,\cdot,i}^{(j)}$ into a scalar influence value. Similarly discussed in Subsection 6.5.1, this can be done in many ways, e.g., by summing all values, applying the Frobenius norm, or taking the maximum value. Each of these summarization methods (i.e., 2D to 1D reduce operations) may lend itself well to exposing interesting connections between channels later in our pipeline. We chose to (I3) take the maximum value of $X_{\cdot,\cdot,i} * K_{\cdot,\cdot,i}^{(j)}$ as our measure of influence for the image classification task, since this task intuitively considers the largest magnitude of a feature, e.g., how strongly a “dog ear” or “car wheel” feature is expressed, instead of summing values for example, which might indicate how many places in the image a “dog ear” or “car wheel” is being expressed. Also, this mirrors our approach for aggregating activations above.

Lastly, we must aggregate these influence values between channel pairs in consecutive layers, for all images in a given class, i.e., create the proposed I^l matrix from the pairwise channel influence values. This process mirrors the aggregation described previously (Subsection 6.5.1), and we follow the same framework. Let L_{ij}^l be the scalar influence value computed by the previous step *for a single image in class c* , between channel i in layer $l - 1$ and channel j in layer l . We increment an entry (c, i, j) in the tensor I_{cij}^l if L_{ij}^l is one of the top k_{M1} largest values in the column $L_{\cdot,j}^l$ (mirroring **Method 1** from Subsection 6.5.2), or if L_{ij}^l is in the top $k_{M2}\%$ of largest values in $L_{\cdot,j}^l$ (mirroring **Method 2** (Subsection 6.5.1)).

Empirically, we noticed the histograms of max influence values were not as often power law distributed as in the previous aggregation of activations, therefore we use **Method 1** to (I4) record the top $k_{M1} = 5$ channels to include in the (I5) **Aggregated Influence** matrix I^l . Note that INCEPTIONV1 contains inception modules, groups of branching parallel convolution layers. Our influence aggregation approach handles these layer depth imbalances by merging paths using the minimum of any two hop edges through an inner layer; this

guarantees all edge weights between two hop channels are maximal. In terms of runtime, this process is more computationally expensive than aggregating activations, since we have to compute all intermediate 2D activation maps; however, with a standard GPU equipped machine is sufficient. We discuss our experimental setup later in Subsection 6.6.4.

6.5.3 Combining Aggregated Activations and Influences to Generate Attribution Graphs

Given the aggregated activations A^l and aggregated influences I^l we aim to combine them into a single entity that describes both *what* features a neural network is detecting and *how* those features are related. We call these **attribution graphs**, and we describe their generation below.

In essence, neural networks are directed acyclic graphs: they take input data, compute transformations of that data at sequential layers in the network, and ultimately produce an output. We can leverage this graph structure for our desired representation. Whereas a common network graph has vertices and connecting edges, our vertices will be the channels of a network (for all layers of the network), and edges connect channels if the channel in the previous layer has a strong influence to a channel in an later, adjacent layer.

Using graph algorithms for neural network interpretability. Consider the aggregated influences I^l as an edge list; therefore, we can build an “entire graph” of a neural network, where edges encode if an image had a path from one channel to another as a top influential path, and the weight of an edge is a count of the number of images for a given class with that path as a top influential path. Now, for a given class, we want to extract the subgraph that best captures the important vertices (channels) and edges (influential paths) in the network. Since we have instantiated a typical network graph, we can now leverage the abundant research in graph algorithms. A natural fit for our task is the Personalized PageRank algorithm [200, 201], which scores each vertex’s importance in a graph, based on both the graph structure and the weights associated with the graph’s vertices and edges. Specifically, SUMMIT operates on the graph produced from all the images of a given class; the algorithm is initialized by and incorporates both vertex information (aggregated activations A^l) and edge information (aggregated influences I^l) to find a subgraph most relevant for all the provided images. We normalize each layer’s personalization from A^l by dividing by max A^l value for each layer l so that each layer has a PageRank personalization within 0 to 1. This is required since each layer has a different total number of possible connections (e.g., the first and last layers, mixed3a and mixed5b, only have one adjacent layer, therefore their PageRank values would be biased small). In summary, we make the full graph of a neural network where vertices are channels from all layers in the network with

a personalization from A^l , and edges are influences with weights from I^l .

Extracting attribution graphs. After running Personalized PageRank for 100 iterations, the last task is to select vertices based on their computed PageRank values to extract an attribution graph. There are many different ways to do this; below we detail our approach. We first compute histograms of the PageRank vertex values for each layer. Next, we use the methodology described in Subsection 6.5.1 for **Method 2**, where we continue picking vertices with the largest PageRank value until we have reached $k_{M2}\%$ weight for each layer independently. Empirically, here we set $k_{M2} = 7.5\%$ after observing that the PageRank value histograms are roughly power law, indicating that there are only a handful of channels determined important. Regarding the runtime, the only relevant computation is running PageRank on the full neural network graph, which typically has a few thousands vertices and a few hundred thousand edges. Using the Python NetworkX² implementation [201, 200], Personalized PageRank runs in ~ 30 seconds for each class.

6.6 The SUMMIT User Interface

From our design goals in Section 6.3 and our aggregation methodology in Section 6.5, we present SUMMIT, an interactive system for scalable summarization and interpretation for exploring entire learned classes in large-scale image classifier models (Figure 6.1).

The header of SUMMIT displays metadata about the visualized image classifier, such as the model and dataset name, the number of classes, and the total number data instances within the dataset. As described in Section 6.4, here we are using INCEPTIONV1 trained on the 1.2 million image dataset ImageNet that contains 1000 classes. Beyond the header, the SUMMIT user interface is composed of three main interactive views: the Embedding View, the Class Sidebar, and the Attribution Graph View. The following section details the representation and features of each view and how they tightly interact with one another.

6.6.1 Embedding View: Learned Class Overview

The first view of SUMMIT is the Embedding View, a dimensionality reduction overview of all the classes in a model (Figure 6.1A). Given some layer l 's A^l matrix, recall an entry in this matrix corresponds to the number of images from one class (row) that had one channel (column) as a top channel. We can consider A as a feature matrix for each class where the number of channels in a layer corresponds to the number of features. For reduction and visualization, the Embedding View uses UMAP: a non-linear dimensionality reduction that better preserves global data structure, compared to other techniques like t-SNE, and

²NetworkX: <https://networkx.github.io/>



Figure 6.4: Selectable network minimap animates the Embedding View.

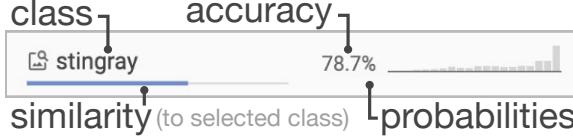


Figure 6.5: Class Sidebar visual encoding.

often provides a better “big picture” view of high-dimensional data while preserving local neighbor relations [202]. Each dot corresponds to one class of the model, with spatial position encoding their similarity. To explore this embedding, users can freely zoom and pan in the view, and when a user zooms in close enough, labels appear to describe each class (point) so users can easily see how classes within the model compare. Clicking on a point in the Embedding View will update the selection for the remaining views of SUMMIT, as described below.

Selectable neural network minimap. At the top of the Embedding View sits a small visual representation of the considered neural network; in this case, INCEPTIONV1’s primary mixed layers are shown (Figure 6.4). Since we obtain one A^l matrix for every layer l in the model, to see how the classes related to one another at different layer depths within the network, users can click on one of the other layers to animate the Embedding View. This is useful for obtaining model debugging hints and observing at a high-level how classes are represented throughout a network’s layers.

6.6.2 Class Sidebar: Searching and Sorting Classes

Underneath the Embedding View sits the Class Sidebar (Figure 6.1B): a scrollable list of all the class of the model, containing high-level class performance statistics. The first class at the top of the list is the selected class, whose attribution graph is shown in the Attribution Graph View, to be discussed in the next section. The Class Sidebar is sorted by the similarity of the selected class to all other classes in the model. For the similarity metric, we compute the cosine similarity using the values from A^l . Each class is represented as a horizontal bar that contains the class’s name, a purple colored bar that indicates its similarity to the selected class (longer purple bars indicate similar classes, and vice versa), the class’s top-1 accuracy for classification, and a small histogram of all the images’ predicted probabilities within that class (i.e., the output probabilities from the final layer) (Figure 6.5).

From this small histogram, users can quickly see how well a class performs. For example, classes with power law histograms indicate high accuracy, whereas classes with normal distribution histograms indicate underperformance. Users can then hypothesize whether a model may be biasing particular classes over others, or if underperforming classes have problems with their raw data.

Scrolling for context. To see where a particular class in the sidebar is located in the Embedding View, users can hover over a class to highlight its point and label the Embedding View above (Figure 6.1A-B). Since the Class Sidebar is sorted by class similarity, to see where similar classes lie compared to the selected class, all classes in the Class Sidebar visible to the user (more technically, in the viewbox of the interface) are also highlighted in the Embedding View (Figure 6.1A-B). Scrolling then enables users to quickly see where classes in the Class Sidebar lie in the Embedding View as classes become less similar to the originally selected class to visualize.

Sorting and selecting classes. To select a new class to visualize, users can click on any class in the Class Sidebar to update the interface, including resorting the Class Sidebar by similarity based on the newly selected class and visualize the new class’s attribution graph in the Attribution Graph View. Users can also use the search bar to directly search for a known class instead of freely browsing the Class Sidebar and Embedding View. Lastly, the Class Sidebar has two additional sorting criteria. Users can sort the Class Sidebar by the accuracy, either ascending or descending, to see which classes in the model have the highest and lowest predicted accuracy, providing a direct mechanism to begin to inspect and debug underperforming classes.

6.6.3 Attribution Graph View: Visual Class Summarization

The Attribution Graph View is the main view of SUMMIT (Figure 6.1C). A small header on top displays some information about the class, similar to that in the Class Sidebar, and contains a few controls for interacting with the attribution graph, to be described later.

Visualizing attribution graphs. Recall from Subsection 6.5.3 that an attribution graph is a subgraph of the entire neural network, where the vertices correspond to a class’s important channels within a layer, and the edges connect channels based on their influence from the convolution operation. Our graph visualization design draws inspiration from recent visualization works, such as CNNVis [73], AEVis [69], and Building Blocks [70], that have successfully leveraged graph based representations for deep learning interpretability. In the main view of SUMMIT, an attribution graph is shown in a zoomable and panable canvas that visualizes the graph vertically, where the top corresponds to the last mixed network layer in the network, mixed5b, and the bottom layer corresponds to the first mixed layer,

mixed3a (Figure 6.1C). In essence, the attribution graph is a directed network with vertices and edges; in SUMMIT, we replace vertices with the corresponding channel’s feature visualization. Each layer, denoted by a label, is a horizontal row of feature visualizations of the attribution graph. Each feature visualization is scaled by its magnitude of the number of images within that class that had that channel as a top channel in their prediction, i.e., the value from A^l . Edges are drawn connecting each channel to visualize the important paths data takes during prediction. Edge thickness is encoded by the influence from one channel to another, i.e., the value from I^l .

Understanding attribution graph structure. This novel visualization reveals a number of interesting characteristics about how classes behave inside a model. First, it shows how neural networks build up high-level concepts from low-level features, for example, in the *white wolf* class, early layers learn fur textures, ear detectors, and eye detectors, which all contribute to form face and body detectors in later layers. Second, the number of visualized channels per layer roughly indicates how many features are needed to represent that class within the network. For example, in layer mixed5a, the *strawberry* class only has a few large channels, indicating this layer has learned specific object detectors for strawberries already, whereas in the same layer, the *drum* class has many smaller channels, indicating that this layer requires the combination of multiple object detectors working together to represent the class. Third, users can also see the overall structure of the attribution graph, and how a model has very few important channels in earlier layers, but as the the network progress, certain channels grow in size and begin to learn high-level features about what an image contains.

Inspecting channels and connections in attribution graphs. Besides displaying the feature visualization at each vertex, there are a number of different complementary data that is visualized to help interpret what a model has learned for a given class attribution graph. It has been shown that for interpreting channels in a neural network, feature visualization is not always enough [62]; however, displaying example image patches from the entire dataset next to a feature visualization helps people better understand what the channel is detecting. We apply a similar approach, where hovering over a channel reveals 10 image patches from the entire dataset that most maximize this specific channel (Figure 6.1C). Pairing feature visualization with dataset examples helps understand what the channel is detecting in the case where a feature visualization alone is hard to decipher. When a user hovers over a channel, SUMMIT also highlights the edges that flow in and out of that specific channel by coloring the edges and animating them within the attribution graph. This is helpful for understanding which and how much channels in a previous layer contribute to a new channel in a later layer. Users can also hover over the edges of an attribution graph to

color and animate that specific edge and its endpoint channels, similar to the interaction used when hovering over channels. Lastly, users can get more insight into what feature a specific channel has learned by hovering left to right on a channel to see the feature visualization change to display four other feature visualizations generated with *diversity*: a technique used to create multiple feature visualizations for a specific channel at once that reveals different areas of latent space that a channel has learned [62]. This interaction is inspired from commercial photo management applications where users can simply hover over an image album’s thumbnail to quickly preview what images are inside.

Dynamic drill down and filtering. When exploring an attribution graph, users can freely zoom and pan the entire canvas, and return to the zoomed-out overview of the visualization via a button included in the options bar above the attribution graph. In the case of a large attribution graph where there are too many channels and edges, in the options bar there is a slider that when dragged, filters the channels of the attribution graph by their importance from A^l. This interaction technique draws inspiration from existing degree-of-interest graph exploration research, where users can dynamically filter and highlight a subset of the most important channels (vertices) and connections (edges) based on computed scores [203, 204, 205, 206]. Dragging the slider triggers an animation where the filtered-out channels and their edges are removed from the attribution graph, and the remaining visualization centers itself for each layer. With the additional width and height sliders, these interactions add dynamism to the attribution graph, where it fluidly animates and updates to users deciding the scale of the visualization.

6.6.4 System Design

To broaden access to our work, SUMMIT is web-based and can be accessed from any modern web-browser. SUMMIT uses the standard HTML/CSS/JavaScript stack, and D3.js³ for rendering SVGs. We ran all our deep learning code on a NVIDIA DGX 1, a workstation with 8 GPUs, with 32GB of RAM each, 80 CPU cores, and 504GB of RAM. With this machine we could generate everything required for *all 1000 ImageNet* classes—aggregating activations, aggregating influences, and combining them with PageRank (implementation from NetworkX) to form attribution graphs—and perform post-processing under 24 hours. However, visualizing a single class on one GPU takes only a few minutes. The *Lucid* library is used for creating feature visualizations,⁴ and dataset examples are used from the appendix⁵ of [62].

³D3.js: <https://d3js.org/>

⁴Lucid: <https://github.com/tensorflow/lucid>

⁵<https://github.com/distillpub/post--feature-visualization>

Attribution graph substructure in *lionfish* class.

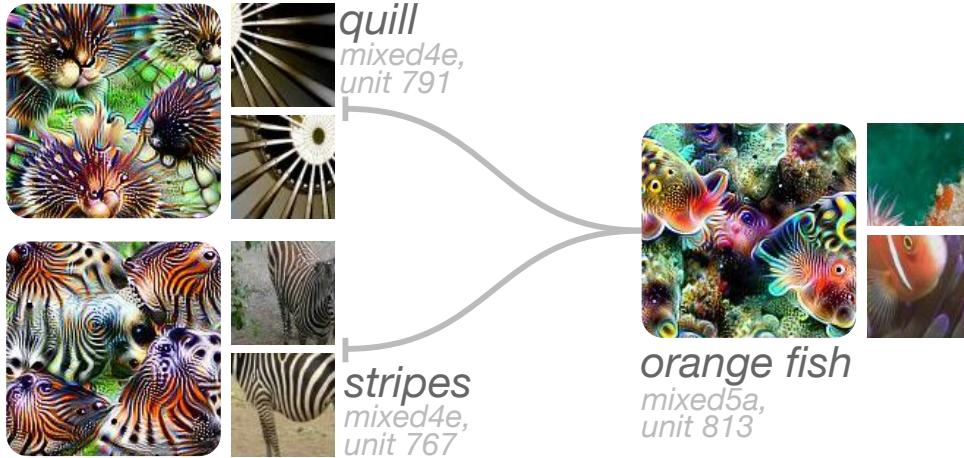


Figure 6.6: An example substructure from the *lionfish* attribution graph that shows unexpected texture features, like “quills” and “stripes,” influencing top activated channels for a final layer’s “orange fish” feature (some *lionfish* are reddish-orange, and have white fin rays).

6.7 Neural Network Exploration Scenarios

6.7.1 Unexpected Semantics Within a Class

A problem with deploying neural networks in critical domains is their lack of interpretability, specifically, can model developers be confident that their network has learned what they think it has learned? We can answer perplexing questions like these with SUMMIT. For example, in Figure 6.1, consider the *tench* class (a type of yellow-brown fish). Starting from the first layer, as we explore the attribution graph for *tench* we notice there are no fish or water feature, but there are many “finger”, “hand”, and “people” detectors. It is not until a middle layer, mixed4d, that the first fish and scale detectors are seen (Figure 6.1C, callout); however, even these detectors focus solely on the body of the fish (there is no fish eye, face, or fin detectors). Inspecting dataset examples reveals many image patches where we see people’s fingers holding fish, presumably after catching them. This prompted us to inspect the raw data for the *tench* class, where indeed, most of the images are of a person holding the fish. We conclude that, unexpectedly, the model uses people detectors and in combination with brown fish body and scale detectors to represent the *tench* class. Generally, we would not expect “people” as an essential feature for classifying fish.

This surprising finding motivated us to seek another class of fish that people do not normally hold to compare against, such as a *lionfish* (due to their venomous spiky fin rays). Visualizing the *lionfish* attribution graph confirms our suspicion (Figure 6.6): there are not any people object detectors in its attribution graph. However, we discover yet another

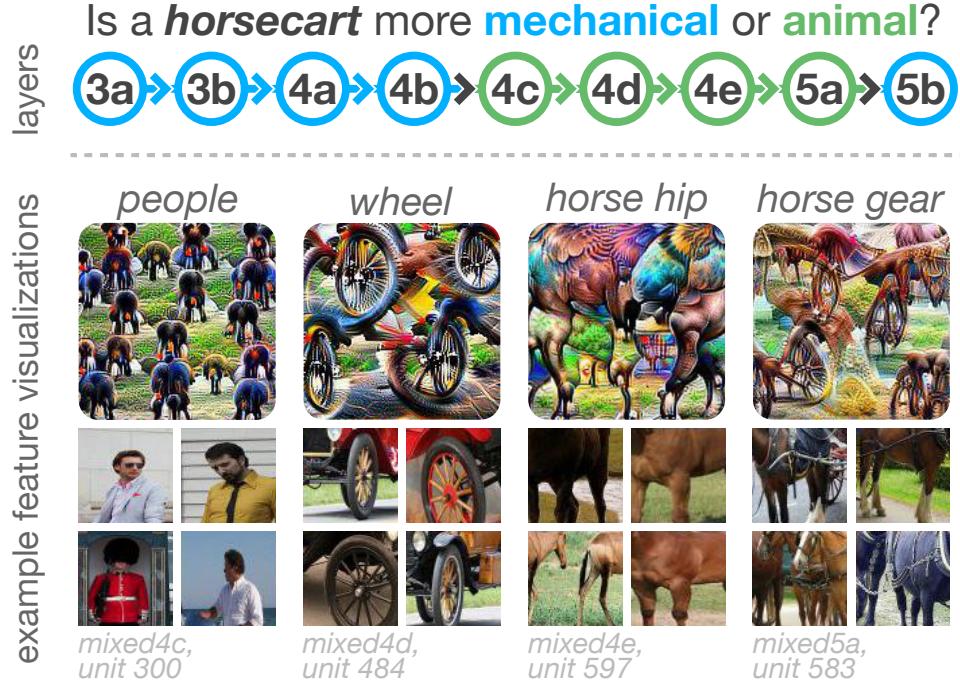


Figure 6.7: Using SUMMIT we can find classes with mixed semantics that shift their primary associations throughout the network layers. For example, early in the network, **horsecart** is most similar to **mechanical** classes (e.g., harvester, thresher, snowplow), towards the middle it shifts to be nearer to **animal** classes (e.g., bison, wild boar, ox), but ultimately returns to have a stronger **mechanical** association at the network output.

unexpected combination of features: there are few fish part detectors while there are many texture features, e.g., stripes and quills. It is not until the final layers of the network where a highly activated channel detects orange fish in water, which uses the stripe and quill detectors. Therefore we deduce that the **lionfish** class is composed of a striped body in the water with long, thin quills. Whereas the **tench** had unexpected people features, the **lionfish** lacked fish features. Regardless, findings such as these can help people more confidently deploy models when they know what composition of features results in a prediction.

6.7.2 Mixed Class Association Throughout Layers

While inspecting the Embedding View, we noticed some classes' embedding positions shift greatly between adjacent layers. This cross-layer embedding comparison is possible since each layer's embedding uses the previous layer's embedding as an initialization. Upon inspection, the classes that changed the most were classes that were either a combination of existing classes or had *mixed primary associations*.

For example, consider the **horsecart** class. For each layer, we can inspect the nearest neighbors of **horsecart** to check its similarity to other classes. We find that **horsecart**

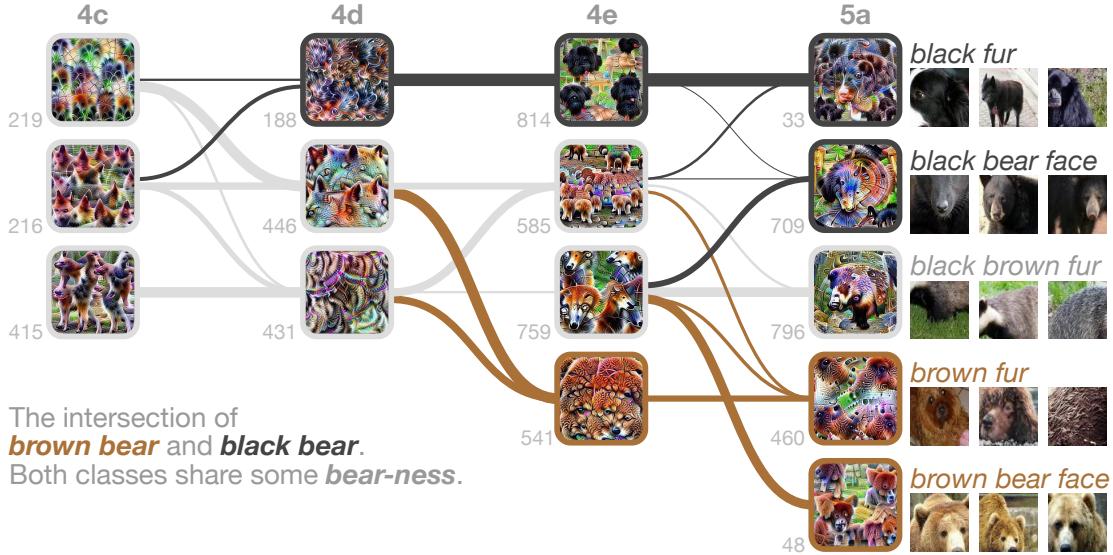


Figure 6.8: With attribution graphs, we can compare classes throughout layers of a network. Here we compare two similar classes: **black bear** and **brown bear**. From the intersection of their attribution graphs, we see both classes share features related to **bear-ness**, but diverge towards the end of the network using fur color and face color as discriminable features. This feature discrimination aligns with how humans might classify bears.

in the early layers is similar to other **mechanical** classes, e.g., harvester, thresher, and snowplow. This association shifts in the middle layers where **horsecart** moves to be near **animal** classes, e.g., bison, wild boar, and ox. However, **horsecart** flips back at the final convolutional layer, returning to a **mechanical** association (Figure 6.7, top). To better understand what features compose a **horsecart**, we inspect its attribution graph and find multiple features throughout all the layers that contain people, spoke wheels, horse hips, and eventually horse bodies with saddles and mechanical gear (Figure 6.7, bottom). Mixed semantic classes like **horsecart** allow us to test if certain classes are semantic combinations of others and probe deeper into understanding how neural networks build hierarchical representations.

6.7.3 Discriminable Features in Similar Classes

Since neural networks are loosely inspired by the human brain, in the broader machine learning literature there is great interest to understand if decision rationale in neural networks is similar to that of humans. With attribution graphs, we can further answer this question by comparing classes throughout layers of a network.

For example, consider the **black bear** and **brown bear** classes. A human would likely say that color is the discriminating difference between these classes. By taking the *inter-*



Figure 6.9: Using SUMMIT on INCEPTIONV1 we found non-semantic channels that detect irrelevant features, regardless of the input image, e.g., in layer mixed3a, channel 67 is activated by the frame of an image.

section of their attribution graphs, we can see what features are shared between the classes, as well as any discriminable features and connections. In Figure 6.8, we see in earlier layers (mixed4c) that both *black bear* and *brown bear* share many features, but as we move towards the output, we see multiple diverging paths and channels that distinguish features for each class. Ultimately, we see individual black and brown fur and bear face detectors, while some channels represent general *bear-ness*. Therefore, it appears INCEPTIONV1 classifies *black bear* and *brown bear* based on color, which may be the primary feature humans may classify by. This is only one example, and it is likely that these discriminable features do not always align with what we would expect; however, attribution graphs give us a mechanism to test hypotheses like these.

6.7.4 Finding Non-semantic Channels

Using SUMMIT, we quickly found several channels that detected non-semantic, irrelevant features, regardless of input image or class (verified manually with 100+ classes, computationally with all). For example, in layer mixed3a, channel 67 activates to the image frame, as seen in Figure 6.9. We found 5 total non-semantic channels, including mixed3a 67, mixed3a 190, mixed3b 390, mixed3b 399, and mixed3b 412. Upon finding these, we reran our algorithm for aggregating activations and influences, and generated all attribution graphs with these channels excluded from the computation, since they consistently produced high activation values but were incorrectly indicating important features in many classes. Although SUMMIT leverages recent feature visualization research [62] to visualize channels, it does not provide an automated way to measure the semantic quality of channels. We point readers to the appendix of [62] to explore this important future research direction.

6.7.5 Informing Future Algorithm Design

We noticed that some classes (e.g., *zebra*, *green mamba*) have only a few important channels in the middle layers of the network, indicating that these channels could have enough information to act as a predictor for the given class. This observation implies that it may be prudent to make classification decisions at different points in the network, as opposed to after a single softmax layer at the output. More specifically, per the A^l matrices, we can easily find these channels (in all layers) that maximally activates for each class. We could then perform a MaxPooling operation at each of these channels, followed by a Dense layer classifier to form a new “model” that only uses the most relevant features for each class to make a decision.

The inspiration for this proposed algorithm is a direct result of the observations made possible by SUMMIT. Furthermore, our proposed methodology makes it easy to test whether the motivating observation holds true for other networks besides INCEPTIONV1. It could be the case that single important channels for certain classes are a result of the training with multiple softmax ‘heads’ used by INCEPTIONV1; however, without SUMMIT, checking this would be difficult.

6.8 Conclusion

As deep learning is increasingly used in decision-making tasks, it is important to understand how neural networks learn their internal representations of large datasets. In this work, we present SUMMIT, an interactive system that scalably and systematically summarizes and visualizes what features a deep learning model has learned and how those features interact to make predictions. The SUMMIT visualization runs in modern web browsers and is open-sourced. We believe our summarization approach that builds entire class representations is an important step for developing higher-level explanations for neural networks. We hope our work will inspire deeper engagement from both the information visualization and machine learning communities to further develop human-centered tools for artificial intelligence [45, 171].

PART III

COMMUNICATING INTERPRETABILITY

WITH INTERACTIVE ARTICLES

Overview

The previous chapters have presented interactive interfaces for interpretability designed for data literate people with machine learning expertise. However, machine learning is now everywhere, and it should not require a technical background to know how to use it, identify when it is wrong, and correct it. The challenge here is how to represent and **communicate interpretability** and explanations for everyone: how can we teach people the capabilities and limitations of machine learning?

Part III begins by presenting multiple interactive articles, a new medium for communication leveraging the dynamic capabilities of the web, authored to educate broad and non-technical audiences about machine learning interpretability, fairness, data bias, and common machine learning techniques such as dimensionality reduction. These articles appear in a new open-source **publishing initiative (Chapter 7)** we launched to test their interactive techniques in the wild. This chapter is adapted from work that was published and appeared in *VisComm 2019* [13] and *VISxAI 2018* [12].

Launching the PARAMETRIC PRESS. ↗ Matthew Conlen, Fred Hohman.
Visualization for Communication at IEEE VIS (VisComm), 2019.

The Myth of the Impartial Machine. ↗ Alice Feng, Shuyan Wu, Fred Hohman, Matthew Conlen, Victoria Uren. *The Parametric Press, Issue 01, 2019.*

The Beginner’s Guide to Dimensionality Reduction. ↗ Matthew Conlen, Fred Hohman. *Workshop on Visualization for AI Explainability at IEEE VIS (VISxAI), 2018.*

Since authoring and publishing interactive content is new and highly flexible, there is little previous work for why they are useful and how they can benefit readers. After the viral success of our articles, we generalize and detail the **affordances of interactive articles (Chapter 8)** alongside atomic examples to connect the dots between interactive articles in practice and the techniques, theories, and empirical evaluations put forth by researchers across the fields of education, human-computer interaction, information visualization, and digital journalism. We also provide critical reflections from our own experience with open-source, interactive publishing at scale, and conclude with practical challenges and open research directions for authoring, designing, and publishing interactive articles. This chapter is adapted from work that was published and appeared in *Distill 2020*.

Communicating with Interactive Articles. ↗ Fred Hohman, Matthew Conlen, Jeffrey Heer, Duen Horng (Polo) Chau. *Distill, 2020.*

CHAPTER 7

MACHINE LEARNING LITERACY: INTERACTIVE ARTICLES IN PRACTICE

In contrast to traditional static media such as books and pictures, and moving media such as movies and animations, *interactive articles* are a new medium for communication that leverages a computational runtime to dynamically respond to reader input. These articles are characterized by interleaving text and interactive widgets – often utilizing animations, data visualizations, or simulations – that guide a reader through a primarily linear narrative. Interactive articles are becoming popular on the web: newspapers such as the New York Times have published interactive articles that include dynamic graphics and visualizations; educators and technical communicators enrich text with interactions and multimedia in an effort to further engage their students and readers. Such interactive content often engages a wide audience [207], and digital publishers understand that articles which utilize the rich capabilities of the web often bring both acclaim and a broad readership [208]. *Explorable explanations* [209] are a notable type of interactive article that promote active reading and inquiry into the details of a specific subject.

Interactive articles have been used within the domain of machine learning, many of which were created as supplementary material to traditional research papers [1, 210] or companion pieces to online courses. The data visualization community has also seen value in using interactive articles to make machine learning more accessible. In 2018 and 2019, the Workshop on Visualization for AI Explainability focused on “creating visual narratives to bring new insight into the often obfuscated complexity of AI systems” [211].

7.1 PARAMETRIC PRESS

Building on this momentum, we believe interactive articles are an excellent fit for communicating interpretability, and more broadly machine learning’s capabilities and limitations, to a large audience. Unfortunately, creating and publishing interactive articles takes significantly more time and effort than traditional media and research publications, and requires authors to have both an eye for design and programming experience. Furthermore, in academia specifically there is no formal incentive structure for non-traditional research artifacts. To meet this challenge, we sought to create a platform where we could publish interactive content and experiment with interfaces that use interactivity, visualizations, and



Figure 7.1: PARAMETRIC PRESS is our interactive publication, a born-digital magazine dedicated to showcasing the expository power that's possible when the audio, visual, and interactive capabilities of dynamic media are effectively combined.

simulations to teach people about complex topics, such as machine learning.

PARAMETRIC PRESS is our answer: a born-digital magazine dedicated to showcasing the expository power that is possible when the audio, visual, and interactive capabilities of dynamic media are effectively combined. PARAMETRIC PRESS is an entirely open-source publishing initiative where we can test interactive article techniques in the wild—while empowering authors to tell data-driven stories and create explorable explanations. We push the boundaries of interactive publishing by open-sourcing all code and visualization components, giving articles DOIs, and provide web archival to ensure the content can be read given in the future. Our articles went viral, which allowed us to analyze thousands of reader patterns to evaluate how this new medium is read and used in practice, a critical yet under-examined aspect of publishing interactive content [212]. Our first issue, *Issue 01: Science + Society*, focuses on examining scientific and technological phenomena that stand to shape society at large, now or in the near future. The issue covers topics that would benefit from using the interactive or otherwise dynamic capabilities of the web.

PARAMETRIC PRESS- *Issue 01: Science + Society*

- “**“Unraveling the JPEG”** [213]: JPEG images are everywhere in our digital lives, but behind the veil of familiarity lie algorithms that remove details that are imperceptible to the human eye. This produces the highest visual quality with the smallest file

size—but what does that look like? Let’s see what our eyes can’t see!

- “**The Myth of the Impartial Machine**” [11]: Wide-ranging applications of data science bring utopian proposals of a world free from bias, but in reality, machine learning models reproduce the inequalities that shape the data they’re fed. Can programmers free their models from prejudice?
- “**Data Science for Fair Housing**” [214]: Cities across America covertly exclude racial minorities from majority-white residential neighborhoods, while gentrification drives people of color out of their homes. In Atlanta, a new nonprofit seeks to resist displacement by supporting the city’s most vulnerable residents—but how effective is their project?
- “**Flatland Follies: An Adjunct Simulator**” [215]: This college used to be one of the best in the country. Fifty years later the campus is destitute, they can’t pay professors, and it’s filled with dusty, decaying art.
- “**On Particle Physics**” [216]: A CERN particle physicist walks through the history and science of particle physics, and why you should care about it—even outside of the laboratory.
- “**Anything That Flies, On Anything That Moves**” [217]: The US covertly launched over two million bombing missions over Southeast Asian countries in the 1960s and 70s. Dig into the data behind the assault.

In the following sections, we summarize two interactive articles written to communicate topics specifically within machine learning to a broad audience.

7.2 The Myth of The Impartial Machine

“Wide-ranging applications of data science bring utopian proposals of a world free from bias, but in reality, machine learning models reproduce the inequalities that shape the data they are fed. Can programmers free their models from prejudice?” reads the subheader for this interactive article. The Myth of The Impartial Machine explores and explains the consequences of using machine learning blindly on problems that impact people. It uses static graphics, data visualizations, animation, and interactive simulations and models to teach readers about the basic machine learning process, types of biases that can sneak into this process, their effect on model predictions and feedback loops, and ultimately potential solutions researchers are developing to prevent the spread and amplification of bias in decision-making tasks.

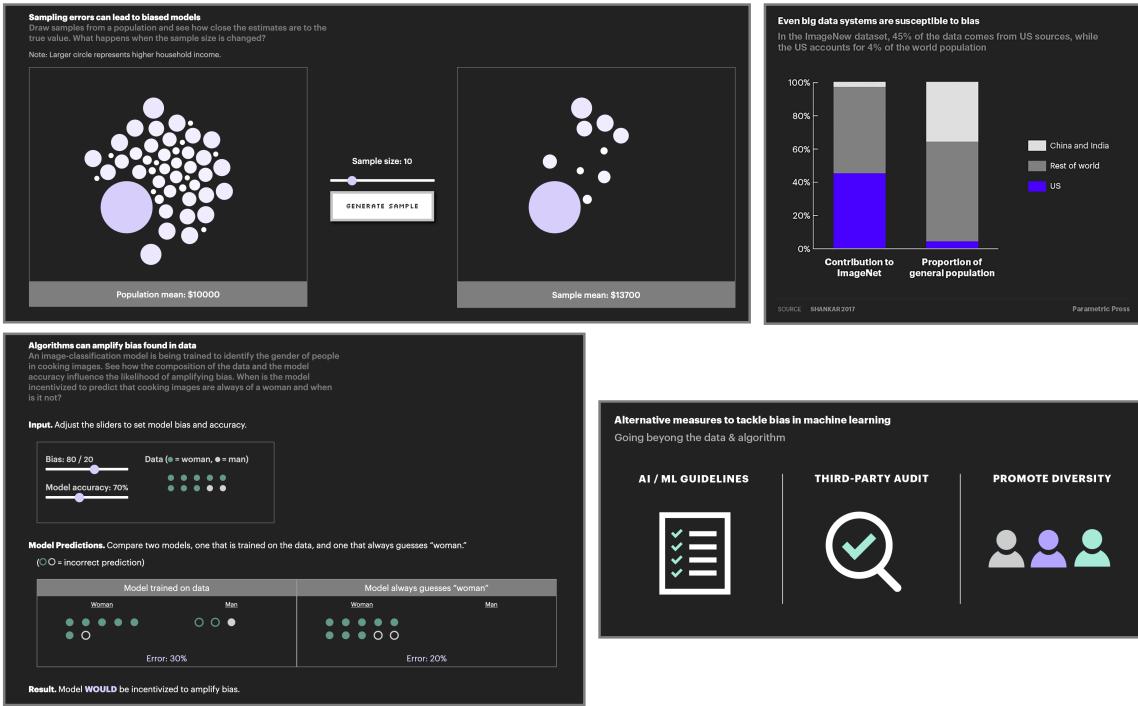


Figure 7.2: A subset of the interactive graphics from The Myth of The Impartial Machine, demonstrating the trajectory and effect of sampling bias in data collection and model building.

The first interactive graphic teaches readers about sampling bias. It allows readers to see ground truth data about a population, and then drag a slider to adjust how many points to sample; the result is compared with the original graphic side by side (Figure 7.2, top left). In the data there is one large data point where the article explains how outliers such as this point can influence the data that is gathered and ultimately used to train models. A static visualization shows sampling bias in the real world, comparing the country composition of a popular academic computer vision dataset (Figure 7.2, top right).

Taking this further, another interactive diagram in the article shows what happens to a model when it is trained on biased data. The reader has interactive control over how biased the data is how accurate the model is, introducing readers to model evaluation metrics (Figure 7.2, bottom left). The article further uses simulations to show feedback loops in biased models, and ultimately concludes with potential solutions and considerations for appropriately applying machine learning in practice (Figure 7.2, bottom right).

7.3 The Beginner’s Guide to Dimensionality Reduction

Dimensionality reduction is a powerful technique used by data scientists to look for hidden structure in data. The method is useful in a number of domains, for example document



Figure 7.3: The dataset and embeddings shown in The Beginner’s Guide to Dimensionality Reduction as a reader progresses through the interactive article.

categorization, protein disorder prediction, and machine learning model debugging. The results of a dimensionality reduction algorithm can be visualized to reveal patterns and clusters of similar or dissimilar data. Even though the data is displayed in only two or three dimensions, structures roughly present in higher dimensions are maintained. This article teaches readers how to think about these embeddings, and provides a comparison of some of the most popular dimensionality reduction algorithms used today.

The article begins with a non-technical, motivating use case. A dataset of 800 artworks from the Metropolitan Museum of Art is automatically loaded in the background while the article introduces readers to the concept of features within machine learning (Figure 7.3A). Next it introduces a 1-dimensional embedding based on a feature the reader likely knows: image brightness (Figure 7.3B). The article then builds on this embedding and projects the data to 2 dimensions, including brightness and artwork age; however, the images can not be manipulated with an interactive slider where a reader can emphasize which feature they care more about and watch as the embedding updates in real time (Figure 7.3C). Lastly, the article extends this embedding one more time and presents a reduction using real-world algorithms such as PCA, t-SNE, and UMAP (Figure 7.3D-F). The reader can toggle between the three embeddings to compare their output, and also read short pros and cons for each algorithm. This article uses the dame dataset throughout every example to build familiarity

within the reader, teaches that not all embeddings are useful, and ends with applications of real-world dimensionality reduction algorithms.

7.4 Summary

These interactive articles are best read on the platforms they were authored for, and thus the full text has not been included in this thesis. Regardless, these interactive articles were authored, designed, developed, and published as an experimental collaboration among many different people, and as a result has had tremendous impact. PARAMETRIC PRESS and our other interactive articles *went viral*, have been *read by 250,000+ people* within their first year, helped students learn about machine learning concepts, and have gathered acclaim for their mission and execution (e.g., multiple Hacker News front page appearances, featured on Stack Overflow Blog, FastCompany review). The viral success of PARAMETRIC PRESS exemplifies the power of the web as a substrate for communicating complex ideas with dynamic media; however, throughout the development of this work we were met with many challenges unique to interactive publishing. In the next chapter, we critically reflect on our experience creating PARAMETRIC PRESS and publishing interactive content at scale.

CHAPTER 8

COMMUNICATING WITH INTERACTIVE ARTICLES

8.1 Introduction

Computing has changed how people communicate. The transmission of news, messages, and ideas is instant. Anyone’s voice can be heard. In fact, access to digital communication technologies such as the Internet is so fundamental to daily life that their disruption by government is condemned by the United Nations Human Rights Council [218]. But while the technology to distribute our ideas has grown in leaps and bounds, the interfaces have remained largely the same.

Parallel to the development of the internet, researchers like Alan Kay and Douglas Engelbart worked to build technology that would empower individuals and enhance cognition. Kay imagined the Dynabook [219] in the hands of children across the world. Engelbart, while best remembered for his “mother of all demos,” was more interested in the ability of computation to augment human intellect [220]. Neal Stephenson wrote speculative fiction that imagined interactive paper that could display videos and interfaces, and books that could teach and respond to their readers [221].

More recent designs (though still historical by personal computing standards) point to a future where computers are connected and assist people in decision-making and communicating using rich graphics and interactive user interfaces [222]. While some technologies have seen mainstream adoption, such as Hypertext [223], unfortunately, many others have not. The most popular publishing platforms, for example WordPress and Medium, choose to prioritize social features and ease-of-use while limiting the ability for authors to communicate using the dynamic features of the web.

In the spirit of previous computer-assisted cognition technologies, a new type of computational communication medium has emerged that leverages active reading techniques to make ideas more accessible to a broad range of people. These interactive articles build on a long history, from Plato [224] to PHeT [225] to explorable explanations [209]. They have been shown to be more engaging, can help improve recall and learning, and attract broad readership and acclaim,¹ yet we do not know that much about them.

In this work, for the first time, we connect the dots between interactive articles

¹For example, some of the New York Times [226, 227] and the Washington Post’s [228] most read articles are interactive stories.

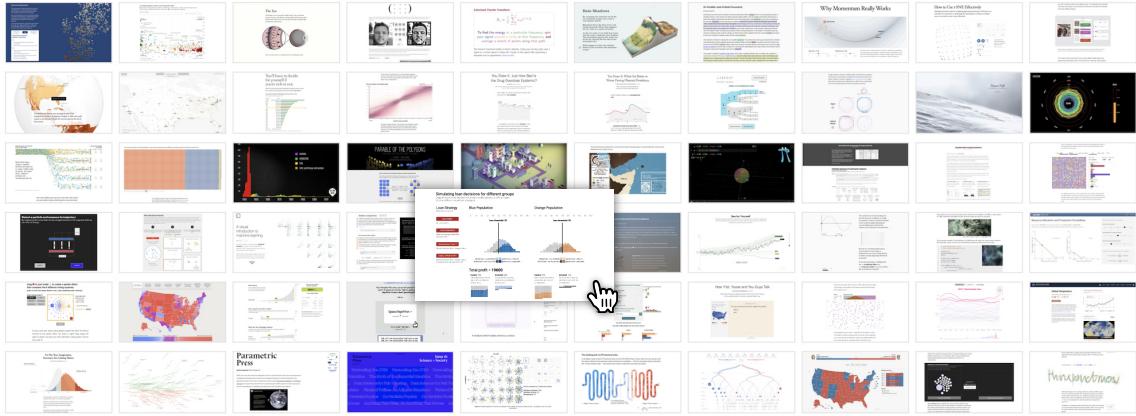


Figure 8.1: Exemplary interactive articles from around the web. In the interactive version of this figure, readers can hover over an article to enlarge its thumbnail and see more information.

such as those featured in this journal and publications like the New York Times and the techniques, theories, and empirical evaluations put forth by academic researchers across the fields of education, human-computer interaction, information visualization, and digital journalism. We show how digital designers are operationalizing these ideas to create interactive articles that help boost learning and engagement for their readers compared to static alternatives.

Today there is a growing excitement around the use of interactive articles for communication since they offer unique capabilities to help people learn and engage with complex ideas that traditional media lacks. After describing the affordances of interactive articles, we provide critical reflections from our own experience with open-source, interactive publishing at scale. We conclude with discussing practical challenges and open research directions for authoring, designing, and publishing interactive articles.

This style of communication—and the platforms which support it—are still in their infancy. When choosing where to publish this work, we wanted the medium to reflect the message. Journals like Distill are not only pushing the boundaries of machine learning research but also offer a space to put forth new interfaces for dissemination. This work ties together the theory and practice of authoring and publishing interactive articles. It demonstrates the power that the medium has for providing new representations and interactions to make systems and ideas more accessible to broad audiences.

8.2 Interactive Articles: Theory and Practice

Interactive articles draw from and connect many types of media, from static text and images to movies and animations. But in contrast to these existing forms, they also leverage

Figure 8.2: Interactive articles are applicable to variety of domains, such as research dissemination, journalism, education, and policy and decision making. In the interactive version of this figure, readers can select the tabs to view different domains.

interaction techniques such as details on demand, belief elicitation, play, and models and simulations to enhance communication.

While the space of possible designs is far too broad to be solved with one-size-fits-all guidelines, by connecting the techniques used in these articles back to underlying theories presented across disparate fields of research we provide a missing foundation for designers to use when considering the broad space of interactions that could be added to a born-digital article.

We draw from a corpus of over fifty interactive articles to highlight the breadth of techniques available and analyze how their authors took advantage of a digital medium to improve the reading experience along one or more dimensions, for example, by reducing the overall cognitive load, instilling positive affect, or improving information recall.

Because diverse communities create interactive content, this medium goes by many different names and has not yet settled on a standardized format nor definition.² Researchers have proposed artifacts such as explorable multiverse analyses [230], explainables [211], and exploranations [231] to more effectively disseminate their work, communicate their results to the public, and remove research debt [171]. In newsrooms, data journalists, developers, and designers work together to make complex news and investigative reporting

²However, one is taking shape [229].

Title ↓	Publication (or author)	Tags	Year
A UI That Lets Readers Control How Much Information They See	Kayce Basques	Reducing Cognitive Load	2018
A Visual Introduction To Machine Learning	R2D3	Personalizing Reading	2015
Are You Rich? This Income-Rank Quiz Might Change How You See Yourself	The New York Times	Personalizing Reading	2019
Attacking Discrimination With Smarter Machine Learning	Google Research	Research Dissemination	2016
Booze Calculator: What's Your Drinking Nationality?	BBC	Personalizing Reading	2017
Climate Spirals	Climate Lab Book	Connecting People and Data	2016
Coeffects: Context-Aware Programming Languages	Tomas Petricek	Research Dissemination	2017
Colorized Math Equations	Better Explained	Reducing Cognitive Load	2017
Complexity Explained	Manlio De Domenico, Hiroki Sayama	Research Dissemination	2019
Cutthroat Capitalism: The Game	Wired	Connecting People and Data	2009
Earth Primer	Chaim Gingold	Reducing Cognitive Load	2015
Earth's Relentless Warming Sets A Brutal New Record In 2017	Bloomberg	Connecting People and Data	2018
EconGraphs	Christopher Makler	Education	2017

Figure 8.3: In the interactive version of this table, readers can sort a list of the interactive articles we discuss in this work.

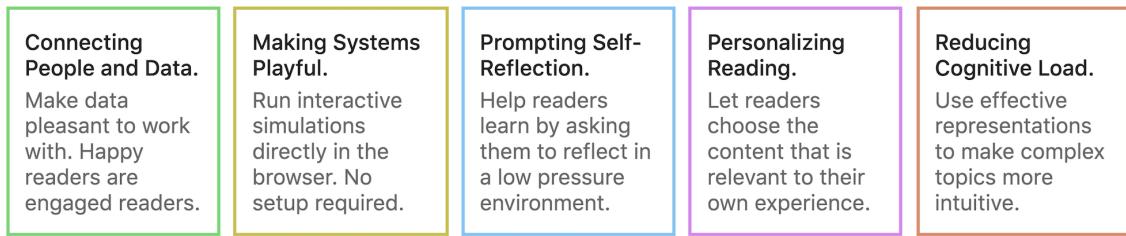


Figure 8.4: The five affordances of interactive articles we discuss.

clear and engaging using interactive stories [232]. Educators use interactive textbooks as an alternative learning format to give students hands-on experience with learning material [233].

Besides these groups, others such as academics, game developers, web developers, and designers blend editorial, design, and programming skills to create and publish explorable explanations [209], interactive fiction [234], interactive non-fiction [235], active essays [236], and interactive games [237]. While these all slightly differ in their technical approach and target audience, they all largely leverage the interactivity of the modern web.

In the original work, in-line videos and example interactive graphics are presented alongside the discussion to demonstrate specific interaction techniques. In this work, these have been replaced with static figures.

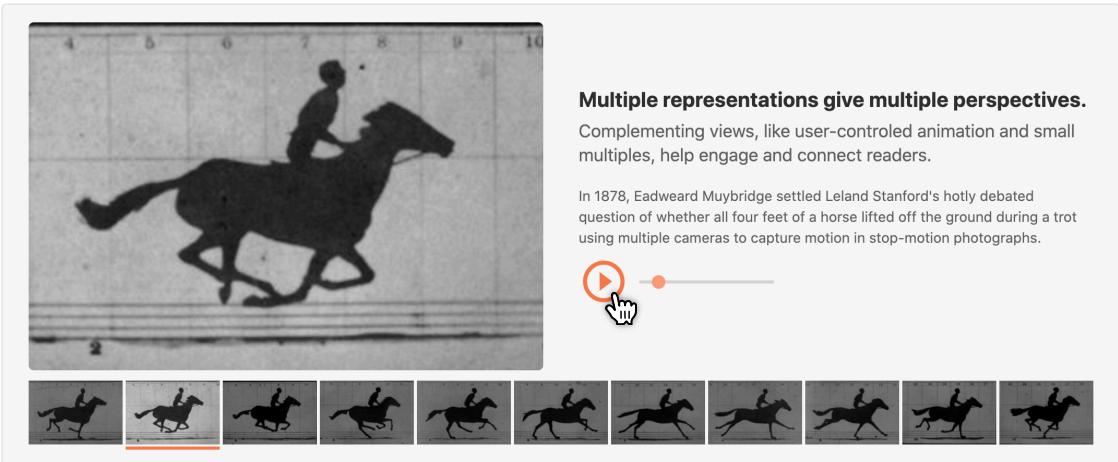


Figure 8.5: In the interactive version of this figure, readers can click the play button or scrub over the video frames to watch and control the animation.

8.2.1 Connecting People and Data

As visual designers are well aware, and as journalism researchers have confirmed empirically [207], an audience which finds content to be aesthetically pleasing is more likely to have a positive attitude towards it. This in turn means people will spend more time engaging with content and ultimately lead to improved learning outcomes. While engagement itself may not be an end goal of most research communications, the ability to influence both audience attitude and the amount of time that is spent is a useful lever to improve learning: we know from education research that both time spent [238] and emotion [239] are predictive of learning outcomes.

Animations can also be used to improve engagement [240]. While there is debate amongst researchers if animations in general are able to more effectively convey the same information compared to a well designed static graphic [241], animation has been shown to be effective specifically for communicating state transitions [242], uncertainty [243], causality [244], and constructing narratives [245]. A classic example of this is Muybridge's motion study [246] that can be seen in Figure 8.5: while the series of still images may be more effective for answering specific questions like, "Does a horse lift all four of its feet off the ground when it runs?" watching the animation in slow motion gives the viewer a much more visceral sense of how it runs. A more modern example can be found in OpenAI's reporting on their hide-and-seek agents [247]. The animations here instantly give the viewer a sense of how the agents are operating in their environment.

Passively, animation can be used to add drama to a graphic displaying important information, but which readers may otherwise find dry. Scientific data which is inherently time

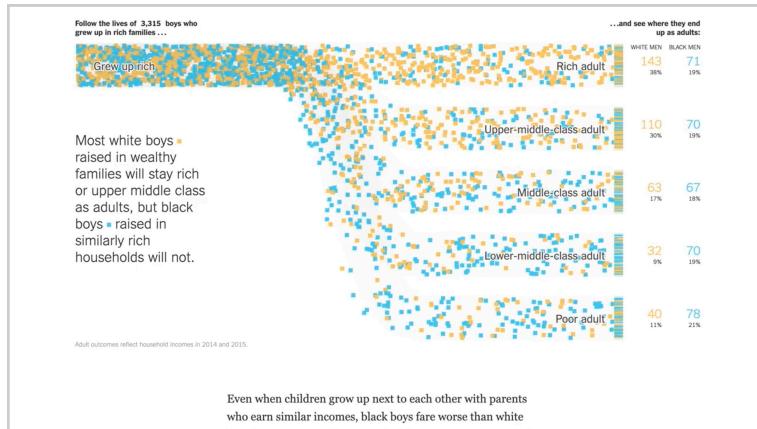


Figure 8.6: In the example, “Extensive Data Shows Punishing Reach of Racism for Black Boys,” [253] the use of unit animation carries the main visualization of the story to highlight real people’s lives changing over time.

varying may be shown using an animation to connect viewers more closely with the original data, as compared to seeing an abstracted static view. For example, Ed Hawkins designed “Climate Spirals,” which shows the average global temperature change over time [248]. This presentation of the data resonated with a large public audience, so much so that it was displayed at the opening ceremony at the 2016 Rio Olympics. In fact, many other climate change visualizations of this same dataset use animation to build suspense and highlight the recent spike in global temperatures [249, 250, 251, 252].

By adding variation over time, authors have access to a new dimension to encode information and an even wider design space to work in. Consider the animated graphic in the New York Times story “Extensive Data Shows Punishing Reach of Racism for Black Boys,” which shows economic outcomes for 10,000 men who grew up in rich families [253]. While there are many ways in which the same data could have been communicated more succinctly using a static visualization [254], by utilizing animation, it became possible for the authors to design a unit visualization in which each data point shown represented an individual, reminding readers that the data in this story was about real peoples’ lives.

Unit visualizations have also been used to evoke empathy in readers in other works covering grim topics such as gun deaths [255] and soldier deaths in war [256]. Using person-shaped glyphs (as opposed to abstract symbols like circles or squares) has been shown not to produce additional empathic responses [257], but including actual photographs of people helps readers connect with and gain interest in, remember [258, 259], and communicate complex phenomena [260] using visualizations. Correll argues that much of the power of visualization comes from abstraction, but quantization stymies empathy [261]. He instead suggests anthropomorphizing data, borrowing journalistic and rhetoric tech-

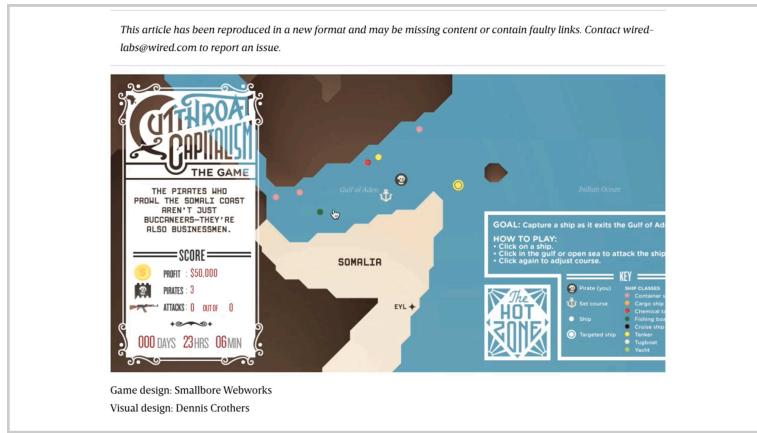


Figure 8.7: In the example, “Cutthroat Capitalism: The Game,” [265] readers play the role of a pirate commander, giving them a unique look at the economics that led to rise in piracy off the coast of Somalia.

niques to create novel designs or interventions to foster empathy in readers when viewing visualizations [261, 262].

Regarding the format of interactive articles, an ongoing debate within the data journalism community has been whether articles which utilize scroll-based graphics (scrollytelling) are more effective than those which use step-based graphics (slideshows). McKenna et al. [263] found that their study participants largely preferred content to be displayed with a step- or scroll-based navigation as opposed to traditional static articles, but did not find a significant difference in engagement between the two layouts. In related work, Zhi et al. found that performance on comprehension tasks was better in slideshow layouts than in vertical scroll-based layouts [264]. Both studies focused on people using desktop (rather than mobile) devices. More work is needed to evaluate the effectiveness of various layouts on mobile devices, however the interviews conducted by MckEnna et al. suggest that additional features, such as supporting navigation through swipe gestures, may be necessary to facilitate the mobile reading experience.

The use of games to convey information has been explored in the domains of journalism [237] and education [266]. Designers of newsgames use them to help readers build empathy with their subject, for example in The Financial Times’s “Uber Game [267],” and explain complex systems consisting of multiple parts, for example in Wired’s “Cutthroat Capitalism: The Game [265]”). In educational settings the use of games has been shown to motivate students while maintaining or improving learning outcomes [268].

As text moves away from author-guided narratives towards more reader-driven ones [269], the reading experience becomes closer to that of playing a game. For example, the critically acclaimed explorable explanation “Parable of the Polygons” puts play at the center of

the story, letting a reader manually run an algorithm that is later simulated in the article to demonstrate how a population of people with slight personal biases against diversity leads to social segregation [270].

8.2.2 Making Systems Playful

Interactive articles utilize an underlying computational infrastructure, allowing authors editorial control over the computational processes happening on a page. This access to computation allows interactive articles to engage readers in an experience they could not have with traditional media. For example, in “Drawing Dynamic Visualizations”, Victor demonstrates how an interactive visualization can allow readers to build an intuition about the behavior of a system, leading to a fundamentally different understanding of an underlying system compared to looking at a set of static equations [271]. These articles leverage active learning and reading, combined with critical thinking [272] to help diverse sets of people learn and explore using sandboxed models and simulations [209].

Complex systems often require extensive setup to allow for proper study: conducting scientific experiments, training machine learning models, modeling social phenomenon, digesting advanced mathematics, and researching recent political events, all require the configuration of sophisticated software packages before a user can interact with a system at all, even just to tweak a single parameter. This barrier to entry can deter people from engaging with complex topics, or explicitly prevent people who do not have the necessary resources, for example, computer hardware for intense machine learning tasks. Interactive articles drastically lower these barriers.

Science that utilizes physical and computational experiments requires systematically controlling and changing parameters to observe their effect on the modeled system. In research, dissemination is typically done through static documents, where various figures show and compare the effect of varying particular parameters. However, efforts have been made to leverage interactivity in academic publishing, summarized in [230]. Reimagining the research paper with interactive graphics [273], as explorations [231], or as explorable multiverse analyses [230], gives readers control over the reporting of the research findings and shows great promise in helping readers both digest new ideas and learn about existing fields that are built upon piles of research debt [171].

Beyond reporting statistics, interactive articles are extremely powerful when the studied systems can be modeled or simulated in real-time with interactive parameters without setup, e.g., in-browser sandboxes. Consider the example in Figure 8.8 of a Boids simulation that models how birds flock together. Complex systems such as these have many different parameters that change the resulting simulation. These sandbox simulations allow readers

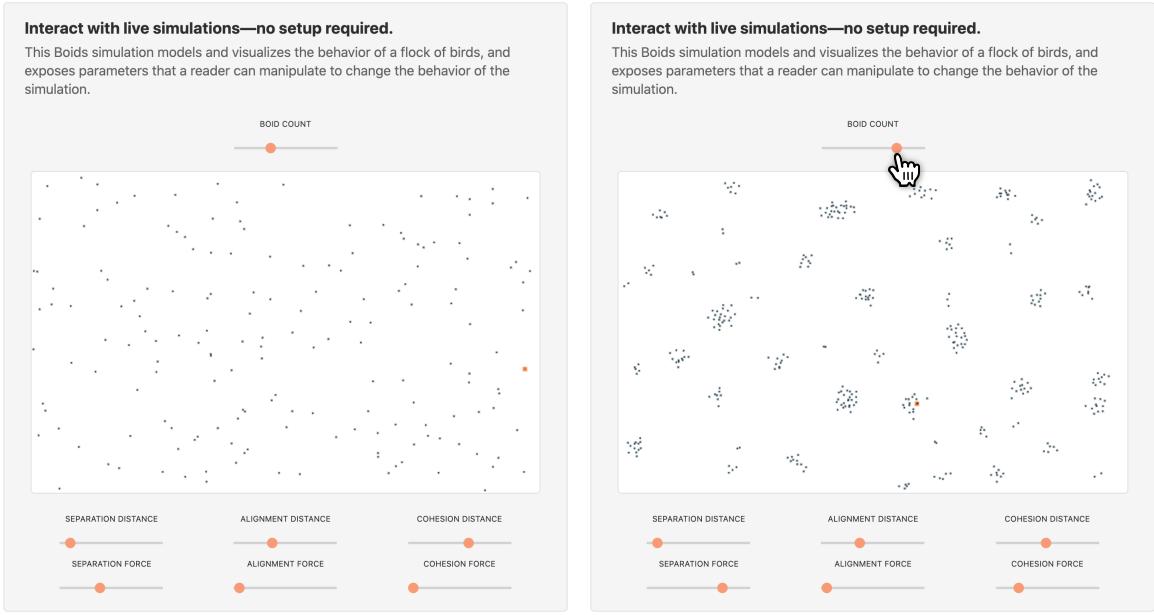


Figure 8.8: In the interactive version of this figure, readers can drag a slider to change the number of boids in the simulation. Underneath the visualization, readers can also adjust the different parameters to find interesting configurations, for example comparing the left and right views above.

to play with parameters to see their effect without worrying about technical overhead or other experimental consequences.

This is a standout design pattern within interactive articles, and many examples exist ranging in complexity. “How You Will Die” visually simulates the average life expectancy of different groups of people, where a reader can choose the gender, race, and age of a person [275]. “On Particle Physics” allows readers to experiment with accelerating different particles through electric and magnetic fields to build intuition behind electromagnetism foundations such as the Lorentz force and Maxwell’s equations—the experiments backing these simulations cannot be done without multi-million dollar machinery [216]. “Should Prison Sentences Be Based On Crimes That Haven’t Been Committed Yet?” shows the outcome of calculating risk assessments for recidivism where readers adjust the thresholds for determining who gets parole [276].

The dissemination of modern machine learning techniques has been bolstered by interactive models and simulations. Three articles, “How to Use t-SNE Effectively [277]” and “The Beginner’s Guide to Dimensionality Reduction [12],” and “Understanding UMA [278]” show the effect that hyperparameters and different dimensionality reduction techniques have on creating low-dimensional embeddings of high-dimensional data. A popular approach is to demonstrate how machine learning models work with in-browser models [279], for example, letting readers use their own video camera as inputs to an image classifica-

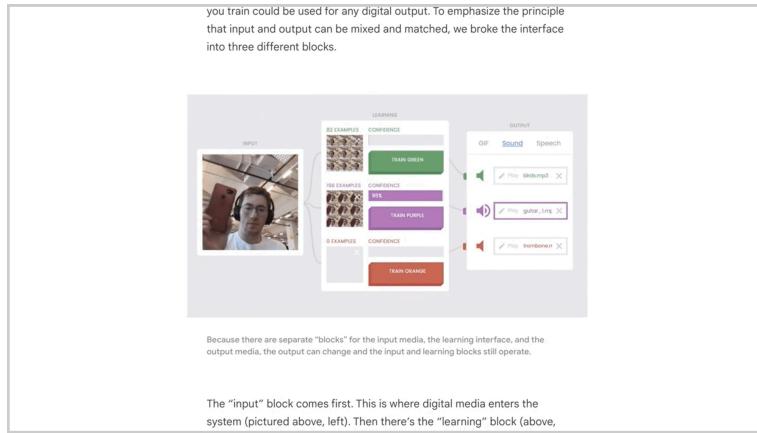


Figure 8.9: In the example, “Teachable Machines,” [274] a reader uses their own live video camera to train a machine learning image classifier in-browser without any extra computational resources.

tion model [274]. Other examples are aimed at technical readers who wish to learn about specific concepts within deep learning. Here, interfaces allow readers to choose different model hyperparameters, datasets, and training procedures that, once selected, visualize the training process and model internals to inspect the effect of varying the model configuration [75, 280].

Interactive articles commonly communicate a single idea or concept using multiple representations. The same information represented in different forms can have different impact. For example, in mathematics often a single object has both an algebraic and a geometric representation. A clear example of this is the definition of a circle [187]. Both are useful, inform one another, and lead to different ways of thinking. Examples of interactive articles that demonstrate this include various media publications’ political election coverage that break down the same outcome in multiple ways, for example, by voter demographics, geographical location, and historical perspective [281, 282, 283].

The Multimedia Principle states that people learn better from words and pictures rather than words or pictures alone [284], as people can process information through both a visual channel and auditory channel simultaneously. Popular video creators such as 3Blue1Brown [285] and Primer [286] exemplify these principles by using rich animation and simultaneous narration to break down complex topics. These videos additionally take advantage of the Redundancy Principle by including complementary information in the narration and in the graphics rather than repeating the same information in both channels [287].

While these videos are praised for their approachability and rich exposition, they are not interactive. One radical extension from traditional video content is also incorporating user input into the video while narration plays. A series of these interactive videos on “Visualizing Quaternions” lets a reader listen to narration of a live animation on screen,

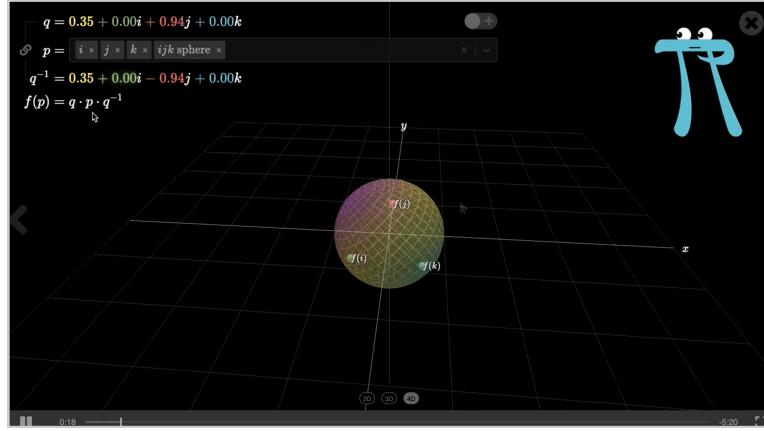


Figure 8.10: In the example, “Visualizing Quaternions,” [288] a viewer can take control of an interactive video while narration continues in the background.

but at any time the viewer can take control of the video and manipulate the animation and graphics while simultaneously listening to the narration [288].

Utilizing multiple representations allows a reader to see different abstractions of a single idea. Once these are familiar and known, an author can build interfaces from multiple representations and let readers interact with them simultaneously, ultimately leading to interactive experiences that demonstrate the power of computational communication mediums. Next, we discuss such experiences where interactive articles have transformed communication and learning by making live models and simulations of complex systems and phenomena accessible.

8.2.3 Prompting Self-Reflection

Asking a student to reflect on material that they are studying and explain it back to themselves—a learning technique called self-explanation—is known to have a positive impact on learning outcomes [289]. By generating explanations and refining them as new information is obtained, it is hypothesized that a student will be more engaged with the processes which they are studying [290]. When writing for an interactive environment, components can be included which prompt readers to make a prediction or reflection about the material and cause them to engage in self-explanation [291, 292].

While these prompts may take the form of text entry or other standard input widgets, one of the most prominent examples of this technique used in practice comes from the New York Times “You Draw It” visualizations [293, 294, 295]. In these visualizations, readers are prompted to complete a trendline on a chart, causing them to generate an explanation based on their current beliefs for why they think the trend may move in a certain direction. Only after readers make their prediction are they shown the actual data. Kim et al. showed

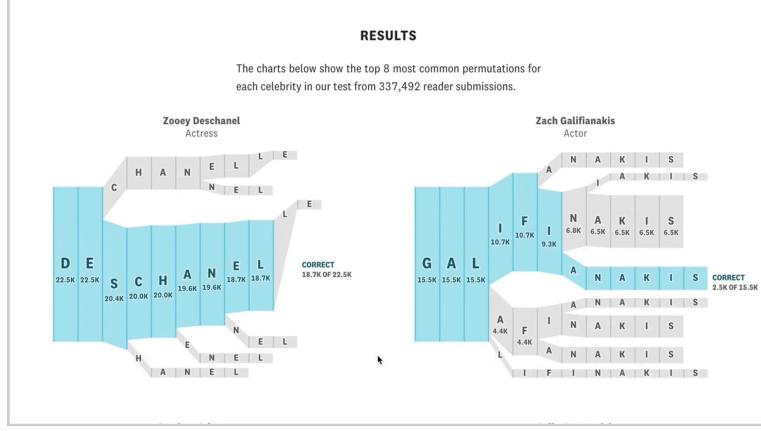


Figure 8.11: In the example, “The Gyllenhaal Experiment,” [296] readers are tasked to type the names of celebrities with challenging spellings. After submitting a guess, a visualization shows the reader’s entry against everyone else’s, scaled by the frequency of different spellings.

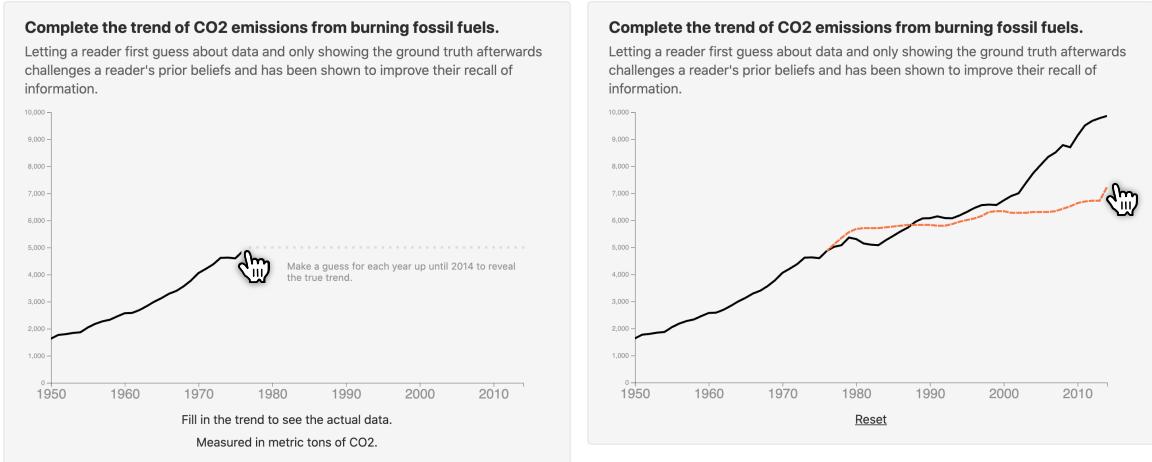


Figure 8.12: In the interactive version of this figure, readers can click and drag to make your guess of the data’s trend over time. Afterward, the real data is revealed.

that using visualizations as a prompt is an effective way to encourage readers to engage in self explanation and improve their recall of the information [291]. Figure 8.12 shows one these visualizations for CO₂ emissions from burning fossil fuels. After clicking and dragging to guess the trend, your guess will be compared against the actual data.

In the case of “You Draw It,” readers were also shown the predictions that others made, adding a social comparison element to the experience. This additional social information was not shown to necessarily be effective for improving recall [297]. However, one might hypothesize that this social aspect may have other benefits such as improving engagement, due to the popularity of recent visual stories using this technique, for example in The Pudding’s “Gyllenhaal Experiment” [296] and Quartz’s “How do you draw a circle?” [298].

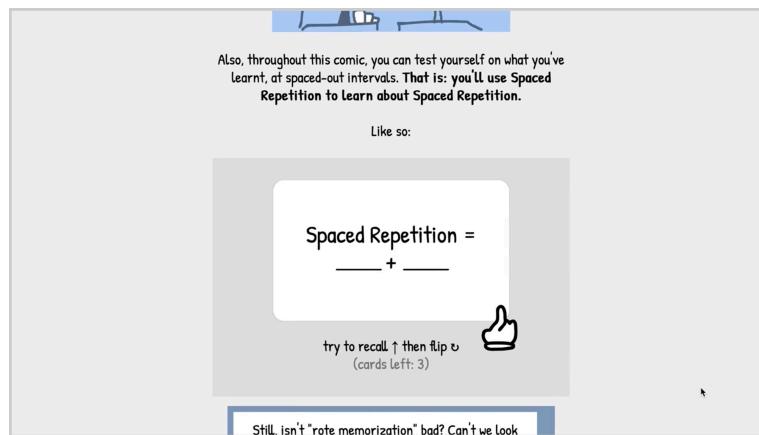


Figure 8.13: In the example, “How To Remember Anything Forever-ish,” [303] readers use spaced repetition to learn about spaced repetition.

Prompting readers to remember previously presented material, for example through the use of quizzes, can be an effective way to improve their ability to recall it in the future [299]. This result from cognitive psychology, known as the testing effect [300], can be utilized by authors writing for an interactive medium [301]. While testing may call to mind stressful educational experiences for many, quizzes included in web articles can be low stakes: there is no need to record the results or grade readers. The effect is enhanced if feedback is given to the quiz-takers, for example by providing the correct answer after the user has recorded their response [302].

The benefits of the testing effect can be further enhanced if the testing is repeated over a period of time [304], assuming readers are willing to participate in the process. The idea of spaced repetition has been a popular foundation for memory building applications, for example in the Anki flash card system. More recently, authors have experimented with building spaced repetition directly into their web-based writing [303, 305], giving motivated readers the opportunity to easily opt-in to a repeated testing program over the relevant material.

8.2.4 Personalizing Reading

Content personalization—automatically modifying text and multimedia based on a reader’s individual features or input (e.g., demographics or location)—is a technique that has been shown to increase engagement and learning within readers [306] and support behavioral change [307]. The PersaLog system gives developers tools to build personalized content and presents guidelines for personalization based on user research from practicing journalists [308]. Other work has shown that “personalized spatial analogies,” presenting distance measurements in regions where readers are geographically familiar with, help people more

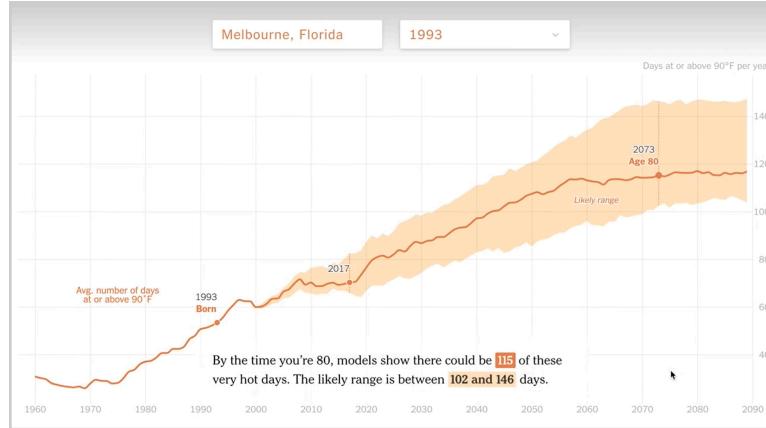


Figure 8.14: In the example, “How Much Hotter Is Your Hometown Than When You Were Born?,” [310] a reader enters their birthplace and birth year and is shown multiple visualizations describing the impact of climate on their hometown.

concretely understand new distance measurements within news stories [309].

Personalization alone has also been used as the center of interactive articles. Both “How Much Hotter Is Your Hometown Than When You Were Born?” [310] and “Human Terrain” [311] use location to drive stories relating to climate change and population densities respectively. Other examples ask for explicit reader input, such as a story that visualizes a reader’s net worth to challenge a reader’s assumptions if they are wealthy or not (relative to the greater population) [312], or predicting a reader’s political party affiliation [313]. Another example is the interactive scatterplot featured in “Find Out If Your Job Will Be Automated” [314]. Here, professions are plotted to inspect their likelihood of being automated against their average annual wage. The article encourages readers to use the search bar to type in their own profession to highlight it against the others.

An interactive medium has the potential to offer readers an experience other than static, linear text. Non-linear stories, where a reader can choose their own path through the content, have the potential to provide a more personalized experience and focus on areas of user interest [235]. For example, the BBC has used this technique in both online articles [315] and in a recent episode of Click [316], a technology focused news television program. Non-linear stories present challenges for authors, as they must consider the myriad possible paths through the content, and consider the different possible experiences that the audience would have when pursuing different branches.

Another technique interactive articles often use is segmenting content into small pieces to be read in-between or alongside other graphics. While we have already discussed cognitive load theory, the Segmenting Theory, the idea that complex lessons are broken into smaller, bit-sized parts [317], also supports personalization within interactive articles. Pro-

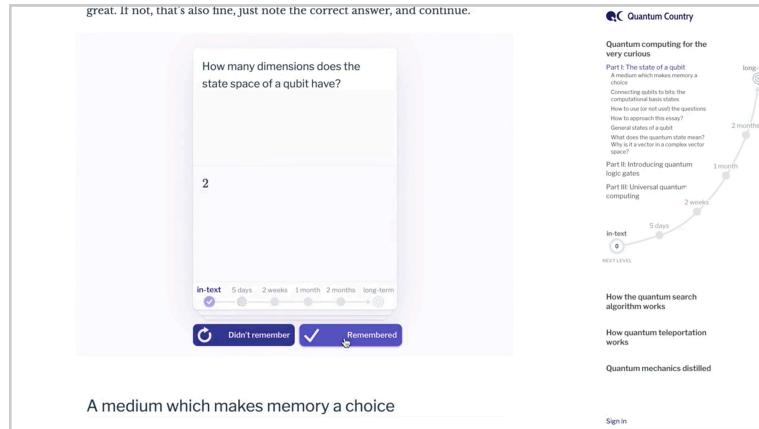


Figure 8.15: In the example, “Quantum Country,” [305] the interactive textbook uses spaced repetition and allows a reader to opt-in and save their progress while reading through dense material and mathematical notion over time..

viding a reader the ability to play, pause, and scrub content allows the reader to move at their own speed, comprehending the information at a speed that works best for them. Segmenting also engages a reader’s essential processing without overloading their cognitive system [317].

Multiple studies have been conducted showing that learners perform better when information is segmented, whether it be only within an animation [318] or within an interface with textual descriptions [319]. One excellent example of using segmentation and animation to personalize content delivery is “A Visual Introduction to Machine Learning,” which introduces fundamental concepts within machine learning in bite-sized pieces, while transforming a single dataset into a trained machine learning model [320]. Extending this idea, in “Quantum Country,” an interactive textbook covering quantum computing, the authors implemented a user account system, allowing readers to save their position in the text and consume the content at their own pace [305]. This book further utilizes the interactive medium by utilizing spaced repetition that helps improve recall.

8.2.5 Reducing Cognitive Load

Authors must calibrate the detail at which to discuss ideas and content to their readers expertise and interest to not overload them. When topics become multifaceted and complex, a balance must be struck between a high-level overview of a topic and its lower-level details. One interaction technique used to prevent a cognitive overload within a reader is “details-on-demand.”

Details-on-demand has become an ubiquitous design pattern. For example, modern operating systems offer to fetch dictionary definitions when a word is highlighted. When

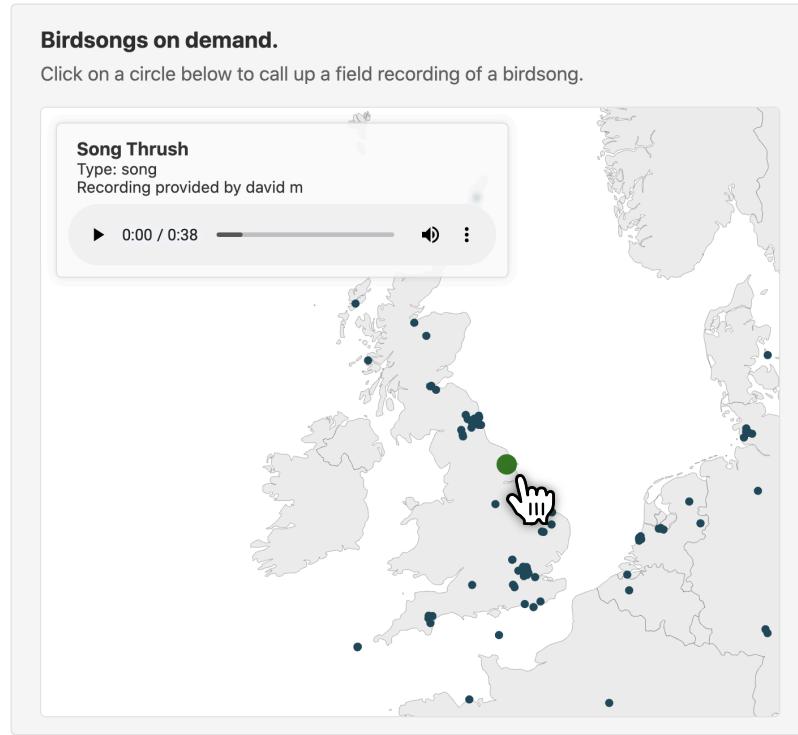


Figure 8.16: In the interactive version of this figure, readers can click any point to listen to a different bird's chirp.

applied to visualization, this technique allows users to select parts of a dataset to be shown in more detail while maintaining a broad overview. This is particularly useful when a change of view is not required, so that users can inspect elements of interest on a point-by-point basis in the context of the whole [321]. Below we highlight areas where details-on-demand has been successfully applied to reduce the amount of information present within an interface at once.

Data Visualization Details-on-demand is core to information visualization, and concludes the seminal Visual Information-Seeking Mantra: “Overview first, zoom and filter, then details-on-demand” [322]. Successful visualizations not only provide the base representations and techniques for these three steps, but also bridge the gaps between them [323]. In practice, the solidified standard for details-on-demand in data visualization manifests as a tooltip, typically summoned on a cursor mouseover, that presents extra information in an overlay. Given that datasets often contain multiple attributes, tooltips can show the other attributes that are not currently encoded visually [324], for example, the map in Figure 8.16 that shows where different types of birdsongs were recorded and what they sound like.

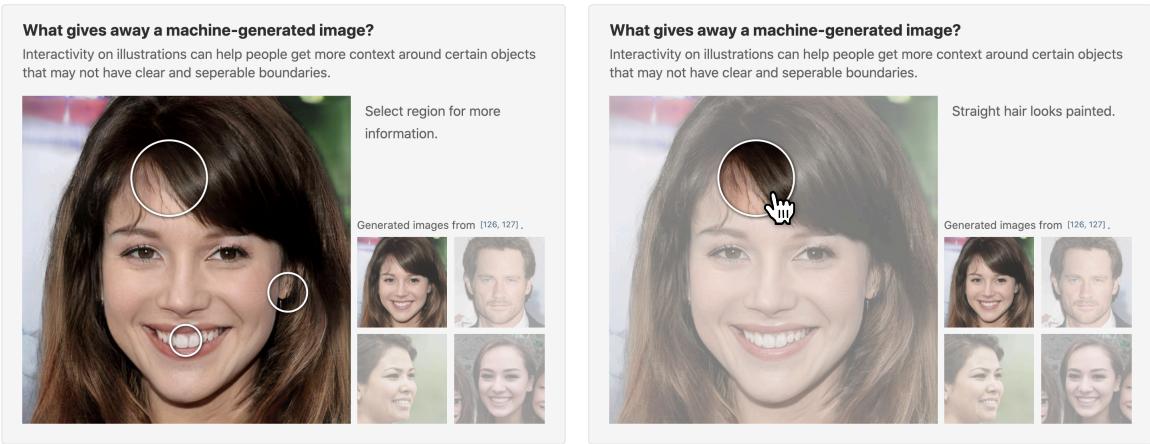


Figure 8.17: In the interactive version of this figure, readers can choose between 1 of 4 machine-generated images and brush over the circle callouts to display a short message about each region.

Illustration Details-on-demand is also used in illustrations, interactive textbooks, and museum exhibits, where highlighted segments of a figure can be selected to display additional information about the particular segment. For example, in “How does the eye work?” readers can select segments of an anatomical diagram of the human eye to learn more about specific regions, e.g., rods and cones [325]. Another example is “Earth Primer,” an interactive textbook on tablets that allows readers to inspect the Earth’s interior, surface, and biomes [326]. Each illustration contains segments the reader can tap to learn and explore in depth. Figure 8.17 demonstrates this by pointing out specific regions in machine-generated imagery to help people spot fake images.

Mathematical Notation Formal mathematics, a historically static medium, can benefit from details-on-demand, for example, to elucidate a reader with intuition about a particular algebraic term, present a geometric interpretation of an equation, or to help a reader retain high-level context while digesting technical details.³ For example, in “Why Momentum Really Works,” equation layout is done using Gestalt principles plus annotation to help a reader easily identify specific terms [327]. In “Colorized Math Equations,” the Fourier transform equation is presented in both mathematical notation and plain text, but the two are linked through a mouseover that highlights which term in the equation corresponds to which word in the text [328]. Another example that visualizes mathematics and computation is the “Image Kernels” tutorial where a reader can mouseover a real image and observe the effect and exact computation for applying a filter over the image [329].

³See this list of examples that experiment with applying new design techniques to various mathematical notation <https://github.com/fredhohman/awesome-mathematical-notation-design>.

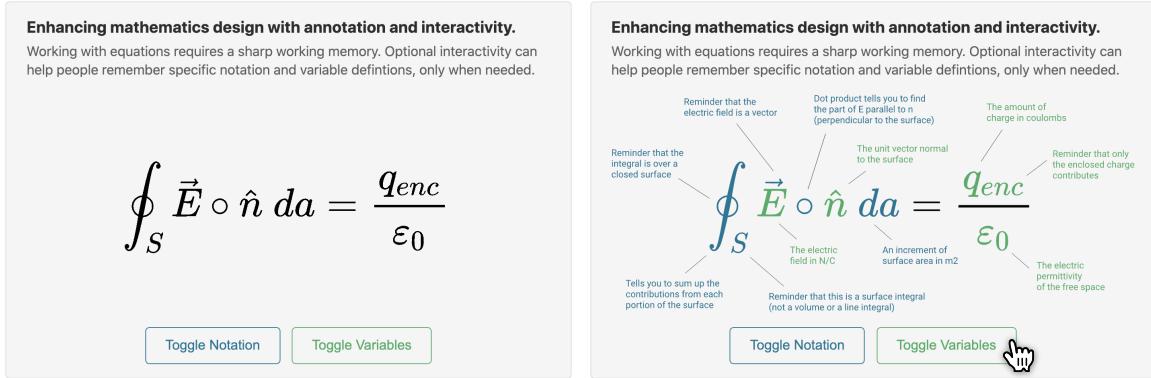


Figure 8.18: In the interactive version of this figure, readers can click to reveal, or remind oneself, what each mark of notation or variable represents in the equation.

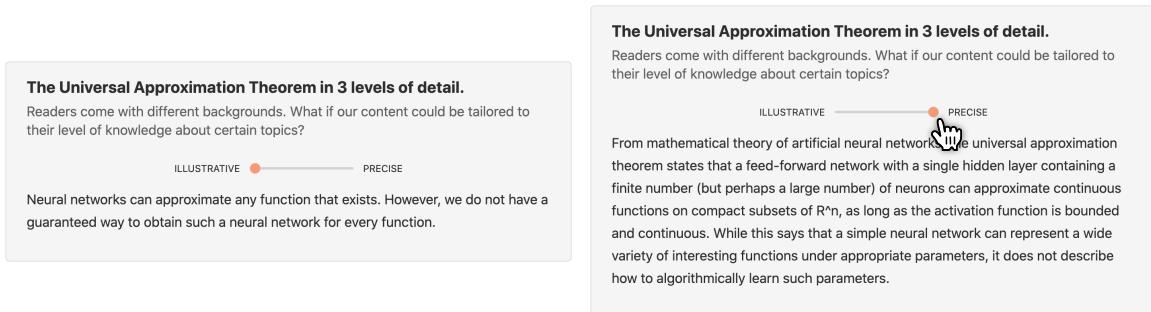


Figure 8.19: In the interactive version of this figure, readers can drag the slider to display the theorem's statement in increasing levels of detail.

Instead of writing down long arithmetic sums, the interface allows readers to quickly see the summation operation's terms and output. In Figure 8.18, one of Maxwell's equations is shown. Click the two buttons to reveal, or remind yourself, what each notation mark and variable represent.

Text While not as pervasive, text documents and other longform textual mediums have also experimented with letting readers choose a variable level of detail to read. This idea was explored as early as the 1960s in StretchText, a hypertext feature that allows a reader to reveal a more descriptive or exhaustive explanation of something by expanding or contracting the content in place [330]. The idea has resurfaced in more recent examples, including “On Variable Level-of-detail Documents” [331], a PhD thesis turned interactive article [332], and the call for proposals of the PARAMETRIC PRESS [333]. One challenge that has limited this technique’s adoption is the burden it places on authors to write multiple versions of their content. For example, drag the slider in Figure 8.19 to read descriptions of the Universal Approximation Theorem in increasing levels of detail. For other examples

of details-on-demand for text, such as application in code documentation, see this small collection of examples [334].

Previewing Content Details-on-demand can also be used as a method for previewing content without committing to another interaction or change of view. For example, when hovering over a hyperlink on Wikipedia, a preview card is shown that can contain an image and brief description; this gives readers a quick preview of the topic without clicking through and loading a new page [335]. This idea is also not new: work from human-computer interaction explored fluid links [336, 337] within hypertext that present information about a particular topic in a location that does not obscure the source material. Both older and modern preview techniques use perceptually-based animation and simple tooltips to ensure their interactions are natural and lightweight feeling to readers [336].

8.3 Challenges for Authoring Interactives

If interactive articles provide clear benefits over other mediums for communicating complex ideas, then why aren't they more prevalent?

Unfortunately, creating interactive articles today is difficult. Domain-specific diagrams, the main attraction of many interactive articles, must be individually designed and implemented, often from scratch. Interactions need to be intuitive and performant to achieve a nice reading experience. Needless to say, the text must also be well-written, and, ideally, seamlessly integrated with the graphics.

The act of creating a successful interactive article is closer to building a website than writing a blog post, often taking significantly more time and effort than a static article, or even an academic publication.⁴ Most interactive articles are created using general purpose web-development frameworks which, while expressive, can be difficult to work with for authors who are not also web developers. Even for expert web developers, current tools offer lower levels of abstraction than may be desired to prototype and iterate on designs.

While there are some tools that help with alleviating this problem [338, 339, 340, 341, 342], they are relatively immature and mainly help with reducing the necessary programming tedium. Tools like Idyll [338] can help authors start writing quickly and even enable rapid iteration through various designs (for example, letting an author quickly compare between sequencing content using a “scroller” or “stepper” based layout). However, Idyll does not offer any design guidance, help authors think through where interactivity would be most effectively applied, nor highlight how their content could be improved to increase

⁴As a proxy, see the number of commits on an example Distill article [70].

its readability and memorability. For example, Idyll encodes no knowledge of the positive impact of self-explanation, instead it requires authors to be familiar with this research and how to operationalize it.

To design an interactive article successfully requires a diverse set of editorial, design, and programming skills. While some individuals are able to author these articles on their own, many interactive articles are created by a collective team consisting of multiple members with specialized skills, for example, data analysts, scripters, editors, journalists, graphic designers, and typesetters, as outlined in [232]. The current generation of authoring tools do not acknowledge this collaboration. For example, to edit only the text of the Distill article requires one to clone its source code using git, install project-specific dependencies using a terminal, and be comfortable editing HTML files. All of this complexity is incidental to task of editing text.

Publishing to the web brings its own challenges: while interactive articles are available to anyone with a browser, they are burdened by rapidly changing web technologies that could break interactive content after just a few years. For this reason, easy and accessible interactive article archival is important for authors to know their work can be confidently preserved indefinitely to support continued readership.⁵ Authoring interactive articles also requires designing for a diverse set of devices, for example, ensuring bespoke content can be adapted for desktop and mobile screen sizes with varying connection speeds, since accessing interactive content demands more bandwidth.

There are other non-technical limitations for publishing interactive articles. For example, in non-journalism domains, there is a mis-aligned incentive structure for authoring and publishing interactive content: why should a researcher spend time on an “extra” interactive exposition of their work when they could instead publish more papers, a metric by which their career depends on? While different groups of people seek to maximize their work’s impact, legitimizing interactive artifacts requires buy-in from a collective of communities.

Making interactive articles accessible to people with disabilities is an open challenge. The dynamic medium exacerbates this problem compared to traditional static writing, especially when articles combine multiple formats like audio, video, and text. Therefore, ensuring interactive articles are accessible to everyone will require alternative modes of presenting content (e.g. text-to-speech, video captioning, data physicalization, data sonification) and careful interaction design.

It is also important to remember that not everything needs to be interactive. Authors should consider the audience and context of their work when deciding if use of interactivity

⁵This challenge has been pointed out by the community: <https://twitter.com/redblobgames/status/1168520452634865665>

would be valuable. In the worst case, interactivity may be distracting to readers or the functionality may go unused, the author having wasted their time implementing it. However, even in a domain where the potential communication improvement is incremental,⁶ at scale (e.g., delivering via the web), interactive articles can still have impact [343].

8.4 Critical Reflections

We write this article not as media theorists, but as practitioners, researchers, and tool builders. While it has never been easier for writers to share their ideas online, current publishing tools largely support only static authoring and do not take full advantage of the fact that the web is a dynamic medium. We want that to change, and we are not alone. Others from the explorable explanations community have identified design patterns that help share complex ideas through play [344, 345, 269, 346, 263].

To explore these ideas further, two of this work’s authors created the PARAMETRIC PRESS [13]: an annually published digital magazine that showcases the expository power that interactive dynamic media can have when effectively combined. In late 2018, we invited writers to respond to a call for proposals for our first issue focusing on exploring scientific and technological phenomena that stand to shape society at large. We sought to cover topics that would benefit from using the interactive or otherwise dynamic capabilities of the web. Given the challenges of authoring interactive articles, we did not ask authors to submit fully developed pieces. Instead, we accepted idea submissions, and collaborated with the authors over the course of four months to develop the issue, offering technical, design, and editorial assistance collectively to the authors that lacked experience in one of these areas. For example, we helped a writer implement visualizations, a student frame a cohesive narrative, and a scientist recap history and disseminate to the public. Multiple views from one article are shown in Figure 8.20.

We see the PARAMETRIC PRESS as a crucial connection between the often distinct worlds of research and practice. The project serves as a platform through which to operationalize the theories put forth by education, journalism, and human-computer interaction researchers. Tools like Idyll which are designed in a research setting need to be validated and tested to ensure that they are of practical use; the PARAMETRIC PRESS facilitates this by allowing us to study its use in a real-world setting, by authors who are personally motivated to complete their task of constructing a high-quality interactive article, and only have secondary concerns and care about the tooling being used, if at all.

⁶In reality, multimedia studies show large effect sizes for improvement of transfer learning in many cases, see Chapter 12 of [284].



Figure 8.20: The Myth of the Impartial Machine was one of five articles published in PARAMETRIC PRESS. The article used techniques like animation, data visualizations, explanatory diagrams, margin notes, and interactive simulations to explain how biases occur in machine learning systems.

Through the PARAMETRIC PRESS, we saw the many challenges of authoring, designing, and publishing first hand, dually as researchers and practitioners.

As researchers we can treat the project as a series of case studies, where we were observers of the motivation and workflows which were used to craft the stories, from their initial conception to their publication. Motivation to contribute to the project varied by author. Where some authors had personal investment in an issue or dataset they wanted to highlight and raise awareness to broadly, others were drawn towards the medium, recognizing its potential but not having the expertise or support to communicate interactively. We also observed how research software packages like Apparatus [339], Idyll [338], and D3 [100] fit into the production of interactive articles, and how authors must combine these disparate tools to create an engaging experience for readers. In one article, “On Particle Physics,” an author combined two tools in a way that allowed him to create and embed dynamic graphics directly into his article without writing any code beyond basic markup. One of the creators of Apparatus had not considered this type of integration before, and upon seeing the finished article commented, “That’s fantastic! Reading that article, I had no idea that Apparatus was used. This is a very exciting proof-of-concept for unconventional exploratory-explanation workflows.”⁷

We were able to provide editorial guidance to the authors drawing on our knowledge of empirical studies done in the multimedia learning and information visualization communities to recommend graphical structures and page layouts, helping each article’s message be communicated most effectively. One of the most exciting outcomes of the project is that we saw authors develop interactive communication skills like any other skill: through continued practice, feedback, and iteration. We also observed the challenges that are inherent in publishing dynamic content on the web and identified the need for improved tooling in this area, specifically around the archiving of interactive articles. Will an article’s code still run a year from now? Ten years from now? To address interactive content archival, we set up a system to publish a digital archive of all of our articles at the time that they

⁷[https://twitter.com/qualmист/status/1128157840672051200?s=20](https://twitter.com/qualmיסט/status/1128157840672051200?s=20)

	Research	Practice
Authoring	Next generation tooling	Evaluate in production setting, identify bugs
Designing	Developing theory, conducting laboratory studies	Evaluate specific design decisions in the wild, understand constraints
Publishing	Tools, guidelines, and best practices	Concrete examples for others to follow, available source code, accessible archives, DOI, branding

Figure 8.21: Interactive communication opportunities from both research and practice.

are first published to the site. At the top of each PARAMETRIC PRESS article is an archive link that allows readers to download a WARC (Web ARChive) file that can “played back” without requiring any web infrastructure. While our first iteration of the project relied on ad-hoc solutions to these problems, we hope to show how digital works such as ours can be published confidently knowing that they will be preserved indefinitely.

As practitioners we pushed the boundaries of the current generation of tools designed to support the creation of interactive articles on the web. We found bugs and limitations in Idyll, a tool which was originally designed to support the creation of one-off articles that we used as a content management system to power an entire magazine issue. We were forced to write patches and plugins to work around the limitations and achieve our desired publication.⁸ We were also forced to craft designs under a more realistic set of constraints than academics usually deal with: when creating a visualization it is not enough to choose the most effective visual encodings, the graphics also had to be aesthetically appealing, adhere to a house style, have minimal impact on page load time and runtime performance, be legible on both mobile and desktop devices, and not be overly burdensome to implement. Any extra hour spent implementing one graphic was an hour that was not spent improving some other part of the issue, such as the clarity of the text, or the overall site design.

There are relatively few outlets that have the skills, technology, and desire to publish interactive articles. From its inception, one of the objectives of the PARAMETRIC PRESS is to showcase the new forms of media and publishing that are possible with tools available today, and inspire others to create their own dynamic writings. For example, Omar Shehata, one of the authors of a Parametric article “Unraveling the JPEG [213],” told us he had wanted to write this interactive article for years yet never had the opportunity, support, or incentive to create it. His article drew wide interest and critical acclaim.

⁸Many of these patches have since been merged to Idyll itself. This is the power of modular open-source tooling in action.

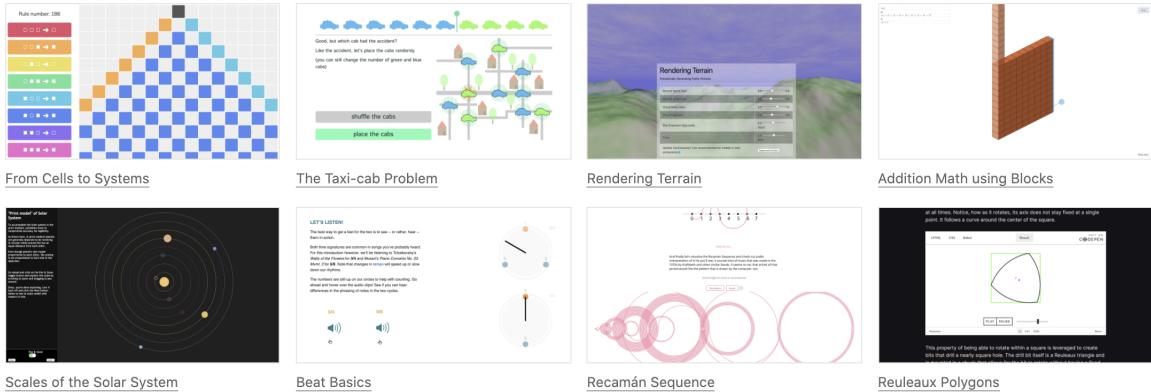


Figure 8.22: Example explorable explanations made in three weeks during the Explorables Jam covering topics from math, astronomy, computer graphics, and music.

We also wanted to take the opportunity as an independent publication to serve as a concrete example for others to follow, to represent a set of best practices for publishing interactive content. To that end, we made available all of the software that runs the site, including reusable components, custom data visualizations, and the publishing engine itself.

8.5 Looking Forward

A diverse community has emerged to meet these challenges, exploring and experimenting with what interactive articles could be. The Explorable Explanations community is a “disorganized ‘movement’ of artists, coders and educators who want to reunite play and learning.” Their online hub contains 170+ interactive articles on topics ranging from art, natural sciences, social sciences, journalism, and civics. The curious can also find tools, tutorials, and meta-discussion around learning, play, and representations. Explorables also hosted a mixed in-person and online Jam: a community-based sprint focused on creating new explorable explanations.⁹ Figure 8.22 highlights a subset of the interactive articles created during the Jam.

Many interactive articles are self-published due to a lack of platforms that support interactive publishing. Creating more outlets that allow authors to publish interactive content will help promote their development and legitimization. The few existing examples, including newer journals such as Distill, academic workshops like VISxAI [211], open-source publications like PARAMETRIC PRESS [13], and live programming notebooks like Observable [340] help, but currently target a narrow group of authors, namely those who have programming skills. Such platforms should also provide clear paths to submission, quality and editorial standards, and authoring guidelines. For example, news outlets have

⁹<https://explorables.es/jam/>

clear instructions for pitching written pieces, yet this is under-developed for interactive articles. Lastly, there is little funding available to support the development of interactive articles and the tools that support them. Researchers do not receive grants to communicate their work, and practitioners outside of the largest news outlets are not able to afford the time and implementation investment. Providing more funding for enabling interactive articles incentivizes their creation and can contribute to a culture where readers expect digital communications to better utilize the dynamic medium.

We have already discussed the breadth of skills required to author an interactive article. Can we help lower the barrier to entry? While there have been great, practical strides in this direction [338, 339, 340, 341, 342], there is still opportunity for creating tools to design, develop, and evaluate interactive articles in the wild. Specific features should include supporting mobile-friendly adaptations of interactive graphics (for example [347, 348, 349]), creating content for different platforms besides just the web, and tools that allow people to create interactive content without code.

The usefulness of interactive articles is predicated on the assumption that these interactive articles actually facilitate communication and learning. There is limited empirical evaluation of the effectiveness of interactive articles. The problem is exacerbated by the fact that large publishers are unwilling to share internal metrics, and laboratory studies may not generalize to real world reading trends. The New York Times provided one of the few available data points, stating that only a fraction of readers interact with non-static content, and suggested that designers should move away from interactivity [350]. However, other research found that many readers, even those on mobile devices, are interested in utilizing interactivity when it is a core part of the article's message [212]. This statement from the New York Times has solidified as a rule-of-thumb for designers, and many choose not to utilize interactivity because of it, despite follow-up discussion that contextualizes the original point and highlights scenarios where interactivity can be beneficial [351]. This means designers are potentially choosing a suboptimal presentation of their story due to this anecdote. More research is needed in order to identify the cases in which interactivity is worth the cost of creation.

We believe in the power and untapped potential of interactive articles for sparking reader's desire to learn and making complex ideas accessible and understandable to all.

PART IV

CONCLUSIONS

CHAPTER 9

CONCLUSIONS AND FUTURE DIRECTIONS

This dissertation contributes novel interactive interfaces, scalable algorithms, and pushes the boundary for representing, distilling, and communicating large data and complex machine learning explanations. My research advances our technical understanding of data-driven decision making, helps people uncover what machine learning models learn, and more importantly reinforces machine learning as a technique to empower people.

9.1 Research Contributions

This thesis makes research contributions to multiple fields, including interactive data visualization, machine learning, and more importantly their intersection to **enable** (Part I), **scale** (Part II), and **communicate** (Part III) machine learning interpretability.

First formulation of machine learning interpretability system design.

- Through user research with practitioners, our work represents the *first operationalization of interpretability that defines a set of unique capabilities* interactive interpretability systems should support, and establishes a *model for future investigations* on understanding how people use interpretability tools in practice (Chapter 3).
- We *design, develop, and deploy a cohesive collection of interactive systems*, GAMUT (Chapter 3), TELEGAM (Chapter 4), and SUMMIT (Chapter 6), that showcases how our operationalization helps people understand models and their predictions across multiple modalities, data types, and explanation mediums.

New interactive and scalable techniques for global model understanding.

- GAMUT (Chapter 3) and TELEGAM (Chapter 4) *interactively combine global and local explanations*, commonly done separately, which not only give users the best of both worlds but we show is essential to effectively enabling interpretability.
- Our INTERROGATIVE SURVEY (Chapter 5), the *first comprehensive survey for visual analytics in deep learning*, helps practitioners quickly learn key aspects of this young and rapidly growing field.
- SUMMIT (Chapter 6) introduces *two new aggregation algorithms to create attribution*

graphs, the first scalable graph representation for understanding neural networks, and combines feature visualization, graph visualization, and graph mining techniques to interactively explore neural network feature representations for *millions of images*.

Future models for research dissemination and interactive communication.

- SUMMIT’s (Chapter 6) *live demo and article provide a model for amplifying research dissemination* that engages people with state-of-the-art computing research while reducing the barrier to entry.
- The *viral success* of PARAMETRIC PRESS (Chapter 7) exemplifies the power of the web as a substrate for communicating complex ideas with dynamic media.
- Our work that connects the theory and practice of INTERACTIVE ARTICLES (Chapter 8) is itself *authored as an interactive article, the first work of its kind* that demonstrates interactive techniques alongside its discussion inline.

Open-source systems that broadens people’s access to interpretability.

- SUMMIT (Chapter 6) and TELEGAM (Chapter 4) are both *open-sourced* and accessible without any installation via *interactive web demos*.
- PARAMETRIC PRESS (Chapter 7) and our INTERACTIVE ARTICLES (Chapter 8) are also *open-sourced, including every article, visualization component, and the publishing engine itself* to allow authors to reuse templates for interactive articles.

9.2 Impact

Beyond the visualization and machine learning research communities, this thesis work has made significant broader impact to industry and society:

- GAMUT (Chapter 3) has been *deployed at Microsoft*, was *demoed for executive leadership* at their internal TechFest, and has been *incorporated into their open-source interpretability toolkit InterpretML* (2,900+ stars on Github).
- PARAMETRIC PRESS (Chapter 7) and our other interactive articles *went viral*, have been *read by 250,000+ people*, *helped students learn about machine learning concepts*, and have *gathered acclaim for their mission and execution* (e.g., multiple Hacker News front page appearances, featured on Stack Overflow Blog, FastCompany review).
- The designed and developed interactive interfaces for interpretability, with a focus on

SUMMIT (Chapter 6) have been invested in and recognized by a *NASA Space Technology Research PhD Fellowship at the Jet Propulsion Lab*, as well as a *Microsoft AI for Earth Award*.

9.3 Future Directions

While this thesis work makes a number of important contributions and has had impact in industry and society, it also unlocks multiple important future research directions and practical applications for applying interactive interfaces for interpretability.

9.3.1 Multi-model Interpretability Interfaces

This work focuses on contributing methods and interfaces for helping people interpret a single machine learning model. These scenarios provide a simplistic environment where a user can have control and governance over both the inputs and outputs of a model. However, in practice large machine learning systems are often composed of multiple models. An analogy for these systems may represent a model chain, or perhaps a model soup, where the outputs of one model become the inputs of another downstream. Here, a user or developer of one model likely does not have full control over the other models included in the overall system, which presents a unique design constraint for interpreting multi-model systems. Another challenge with multi-model systems is the effect of updating a single model, such as retraining and updating weights, has over performance of the overall system. Future interactive interfaces and tools for multi-model interpretability remains an open problem but will only become increasingly important as machine learning applications grow in scope, tackling problems that require multiple models and data dependencies.

Visual and Computational Scalability for Interpretability Systems

Multi-model systems also introduce scalability challenges for interpretability interfaces and their visual encodings. Visual scalability challenges arise when dealing with large data, e.g., large number of hyperparameters and millions of parameters in deep neural networks. Some research has started to address this, by simplifying complex dataflow graphs and network weights for better model explanations [74, 73, 143]. However, when considering activations and embeddings, a popular technique to visualizing machine learning data, dimensionality reduction techniques have bounded utility when datasets contain too many points to discern in 2D [162]. Another example can be seen in GAMUT’s current design, where the shape function charts scale well with the number of data points, but not with the

number of features; the waterfall charts become harder to read as the number of features grows. This is an important research direction, especially given that the information visualization community has developed techniques to visualize large, high-dimensional data that could potentially be applicable to interpreting multiple models simultaneously [352].

Aside from visual scalability, some tools also suffer from system scalability. While some of these problems may require more engineering effort than research, for visual analytics systems to adopted in practice, they must handle state-of-the-art models without penalizing user-performance. Furthermore, these systems, which are often web-based, will greatly benefit from optimized computations to support real-time, rich user interactions [146].

9.3.2 Understanding Adversarial Attacks

Regardless of the benefits machine learning systems are bringing to society, it is remiss to immediately trust them; like most technologies, machine learning has security faults. Identified and studied in a handful of seminal works, it has been shown that deep learning models such as image classifiers can be easily fooled by perturbing an input image [353, 4, 354]. Most alarming, some perturbations are so subtle that they are untraceable by the human eye, yet can completely fool a model into misclassification [4]. This has sparked great interest and focus in the machine learning communities where researchers are trying to understand model fragility by identifying in what ways models can break and constructing methods to protect them. Norton et al. [173] demonstrate these adversarial attacks in an interactive tool where users can tweak the type of attack and its intensity and observe the resulting (mis)classification on small image-based models. While other work has proposed computational techniques to protect models from attacks, such as identifying adversarial examples before classification [355], modifying the network architecture [356], modifying the training process [357, 4], and performing pre-processing steps before classification [196, 358], interpretability tools could further help detect and explain how an attack works and ultimately suggest defenses for protecting machine learning systems.

9.3.3 Making Interpretability Common Practice

Knowing how to represent and communicate machine learning explanations requires a deep understanding of the target users. Since the design and development of interpretability tools is still in its infancy, little work exists on evaluating such tools in practice.

Evaluating Interpretability

Through our survey of visual analytics tools for machine learning, we observed many tools contain multiple-coordinated views with multiple visual representations. Displaying this much information at once can be overwhelming, and when interpretability is the primary focus, it is critical for these systems to have superior usability. While existing literature on machine learning visualization tools emphasize the importance of the “user” [74, 39, 143, 73, 152, 35], most research only reports on design studies conducted with machine learning experts to understand the users and their needs before building a tool instead of evaluating the success of the tool after development. Instead, it is common to see example use cases or illustrative usage scenarios that demonstrate the capabilities of the interactive systems. This trend is because while visualization evaluation is already challenging, evaluating interactive visualization systems for improving interpretability requires precise definitions and measures for interpretability, an open problem discussed earlier. Therefore, investigating better evaluations of interpretability tools in practice is an open and important area of future research.

While work has identified potential strategies for evaluating explanations [19], one practical approach could be to conduct longitudinal evaluations of the impact of deployed interpretability tools in practice. Interpretability tools, such as ones presented in this work like GAMUT, TELEGAM, and SUMMIT, could be deployed to a machine learning development team and then observed how the team uses these tools over long periods of time, through interaction logs or contextual inquiry. These studies could also lead to other insights such as informing algorithmic model design, prompting data collection for ill-represented data subsets, and discovering latent properties of large models.

Understanding How People Use Interpretability Systems

Beyond usability studies, evaluating interpretability tools could provide a better understanding for how people interpret and read explanations. For example, during our GAMUT evaluation, the six datasets used could be considered small by modern machine learning standards. Preliminary work has shown that as scale increases, interpretability and satisfaction decreases [56]. Therefore, it would be useful to see similar studies to ours use larger datasets to see how interpretability is affected by both the number of data points and the number of features. Our studies in GAMUT also revealed that practitioners are eager to trust explanations, neglecting their typical healthy skepticism about their data and models. Interpretability in practice must instill confidence in a user while they are taking their next action, but not mislead. Future studies that integrate interpretability tools on deployed

models to understand how practitioners incorporate interpretability into their workflow will help make interpretability common practice.

Lastly, in the machine learning communities, most works do not include human-subject experiments to evaluate new explanations. For those that do, they greatly benefit from showing why their proposed methods are superior to the ones being tested against [138, 359, 174, 18]. Taking this idea to the quantifiable extreme, a related avenue of evaluating these techniques is the notion of quantifying interpretability, but requires more materials than standard supervised learning approaches [137, 360]. Other domains have recognized that interpretable deep learning research may require evaluation techniques for their explanations, and argue that there is a large body of work from fields such as philosophy, cognitive science, and social psychology that could be utilized [21, 361]. Evaluating interpretability tools is challenging for a number of different reasons, but understanding how people use them can better inform their design, increase their effectiveness, and ultimately help people produce better models.

9.3.4 Responsible Data-driven Decision Making

The democratization of artificial intelligence has led to major breakthroughs in multiple domains, but has also amplified the need for ensuring that data-driven applications remain ethical, fair, safe, transparent, and ultimately benefit society [125]. An important consideration for future research is detecting and mitigating data and model bias. This has been identified as a major problem in deep learning [362, 2], where a number of works are using visualization to understand why a model may be biased [210].

Detecting and Mitigating Machine Bias

One example that aims to detect data bias is Google’s Facets tool [363], a visual analytics system designed specifically to preview and visualize machine learning datasets before training. This allows one to inspect large datasets by exploring the different classes or data instances, to see if there are any high-level imbalances in the class or data distribution. Other works have begun to explore if the mathematical algorithms themselves can be biased towards particular decisions. An example of this is an interactive article titled “Attacking discrimination with smarter machine learning” [210], which explores how one can create both fair and unfair threshold classifiers in an example task such as loan granting scenarios where a bank may grant or deny a loan based on a single, automatically computed number such as a credit score. The article aims to highlight that *equal opportunity* [364] is not preserved by machine learning algorithms, and that as AI-powered systems continue to

make important decisions across core social domains, it is critical to ensure decisions are not discriminatory.

Detecting and Mitigating Human Bias

Finally, aside from data and model bias, we know that humans are inherently biased decision makers. There is a growing area of research into detecting and understanding bias in visual analytics and its affect on the decision making process [365]. Some work has developed metrics to detect types of bias to present to a user during data analysis [365] which could also be applied to visual tools for deep learning in the future. Some work has employed developmental and cognitive psychology analysis techniques to understand how humans learn, focusing on uncovering how human bias is developed and influences learning, to ultimately influence artificial neural network design [361].

It's important to remember that at every stage of the machine learning development process biases can be introduced through data, models, or people. Interpretability tools should be designed with these inevitable pitfalls in mind as they play an important role in evaluating models and discovering various sources of biases.

9.4 Conclusion

I believe that data-driven technology should *empower people, augmenting human intelligence and decision-making*. My continued mission is to create close collaborations between diverse disciplines to both advance our understanding of human-machine collaboration and create practical tools so that people can confidently interact with, responsibly apply, and trust machine learning. This dissertation is an initial step towards this mission, and addresses the fundamental and practical challenges for understanding what machine learning interpretability is and enabling it at scale for everyone.

REFERENCES

- [1] J. Buolamwini and T. Gebru, “Gender shades: Intersectional accuracy disparities in commercial gender classification,” in *ACM Conference on Fairness, Accountability and Transparency*, 2018, pp. 77–91.
- [2] A. Caliskan, J. J. Bryson, and A. Narayanan, “Semantics derived automatically from language corpora contain human-like biases,” *Science*, vol. 356, no. 6334, pp. 183–186, 2017.
- [3] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, “Machine bias,” *ProPublica, May*, vol. 23, 2016.
- [4] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *ICLR*, 2014.
- [5] D. S. Weld and G. Bansal, “Intelligible artificial intelligence,” *arXiv:1803.04263*, 2018.
- [6] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann, “Software engineering for machine learning: A case study,” in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, IEEE, 2019, pp. 291–300.
- [7] F. Hohman, A. Head, R. Caruana, R. DeLine, and S. M. Drucker, “Gamut: A design probe to understand how data scientists understand machine learning models,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, ACM, 2019, p. 579.
- [8] F. Hohman, A. Srinivasan, and S. M. Drucker, “Telegam: Combining visualization and verbalization for interpretable machine learning,” *IEEE Visualization Conference (VIS)*, 2019.
- [9] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, “Visual analytics in deep learning: An interrogative survey for the next frontiers,” *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2018.
- [10] F. Hohman, H. Park, C. Robinson, and D. H. Chau, “Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations,” *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2020.

- [11] A. Feng, S. Wu, F. Hohman, M. Conlen, and S. Stalla, “The myth of the impartial machine,” *The Parametric Press, Issue 01*, 2019.
- [12] M. Conlen and F. Hohman, “The beginner’s guide to dimensionality reduction,” *Workshop on Visualization for AI Explainability (VISxAI) at IEEE VIS*, 2018.
- [13] M. Conlen and F. Hohman, “Launching the parametric press,” *Visualization for Communication (VisComm) at IEEE VIS*, 2019.
- [14] F. Hohman, M. Conlen, J. Heer, and D. H. P. Chau, “Communicating with interactive articles,” *Distill*, 2020.
- [15] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, “Explaining explanations: An approach to evaluating interpretability of machine learning,” *arXiv:1806.00069*, 2018.
- [16] O. Biran and C. Cotton, “Explanation and justification in machine learning: A survey,” in *IJCAI Workshop on Explainable AI*, 2017.
- [17] G. Montavon, W. Samek, and K.-R. Müller, “Methods for interpreting and understanding deep neural networks,” *Digital Signal Processing*, 2017.
- [18] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should i trust you?: Explaining the predictions of any classifier,” in *ACM International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 1135–1144.
- [19] F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” *arXiv:1702.08608*, 2017.
- [20] Z. C. Lipton, “The mythos of model interpretability,” *ICML Workshop on Human Interpretability in Machine Learning*, 2016.
- [21] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *arXiv:1706.07269*, 2017.
- [22] C. Molnar, *Interpretable machine learning*. <https://christophm.github.io/interpretable-ml-book/>, 2018.
- [23] Parliament and C. of the European Union, “General data protection regulation,” 2016.
- [24] B. Goodman and S. Flaxman, “European union regulations on algorithmic decision-making and a “right to explanation”,” *ICML Workshop on Human Interpretability in Machine Learning*, 2016.

- [25] F. Doshi-Velez, M. Kortz, R. Budish, C. Bavitz, S. Gershman, D. O'Brien, S. Schieber, J. Waldo, D. Weinberger, and A. Wood, “Accountability of ai under the law: The role of explanation,” *arXiv:1711.01134*, 2017.
- [26] S. Wachter, B. Mittelstadt, and C. Russell, “Counterfactual explanations without opening the black box: Automated decisions and the gdpr,” *arXiv:1711.00399*, 2017.
- [27] S. Amershi, D. Weld, M. Vorvoreanu, A. Journey, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, J. Teevan, R. Kikin-Gil, and E. Horvitz, “Guidelines for human-ai interaction,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–13.
- [28] “People + ai guidebook: Designing human-centered ai products,” *Google People + AI Research (PAIR)*, 2019.
- [29] “Machine learning human interface guidelines,” *Apple Human Interface Guidelines*, 2019.
- [30] S. Stumpf, V. Rajaram, L. Li, W.-K. Wong, M. Burnett, T. Dietterich, E. Sullivan, and J. Herlocker, “Interacting meaningfully with machine learning systems: Three experiments,” *International Journal of Human-Computer Studies*, vol. 67, no. 8, pp. 639–662, 2009.
- [31] Y. Lu, R. Garcia, B. Hansen, M. Gleicher, and R. Maciejewski, “The state-of-the-art in predictive visual analytics,” in *Computer Graphics Forum*, Wiley Online Library, vol. 36, 2017, pp. 539–562.
- [32] J. Lu, W. Chen, Y. Ma, J. Ke, Z. Li, F. Zhang, and R. Maciejewski, “Recent progress and trends in predictive visual analytics,” *Frontiers of Computer Science*, vol. 11, no. 2, pp. 192–207, 2017.
- [33] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, “What you see is what you can change: Human-centered machine learning by interactive visualization,” *Neurocomputing*, vol. 268, pp. 164–175, 2017.
- [34] S. Amershi, M. Chickering, S. M. Drucker, B. Lee, P. Simard, and J. Suh, “Modeltracker: Redesigning performance analysis tools for machine learning,” in *ACM Conference on Human Factors in Computing Systems*, ACM, 2015, pp. 337–346.
- [35] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams, “Squares: Supporting interactive performance analysis for multiclass classifiers,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 61–70, 2017.

- [36] S. McGregor, H. Buckingham, T. G. Dietterich, R. Houtman, C. Montgomery, and R. Metoyer, “Interactive visualization for testing markov decision processes: Md-pvis,” *Journal of Visual Languages & Computing*, vol. 39, pp. 93–106, 2017.
- [37] M. Brooks, S. Amershi, B. Lee, S. M. Drucker, A. Kapoor, and P. Simard, “Featureinsight: Visual support for error-driven feature ideation in text classification,” in *IEEE Conference on Visual Analytics Science and Technology*, IEEE, 2015, pp. 105–112.
- [38] J. Krause, A. Perer, and E. Bertini, “Infuse: Interactive feature selection for predictive modeling of high dimensional data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1614–1623, 2014.
- [39] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. P. Chau, “Activis: Visual exploration of industry-scale deep neural network models,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 88–97, 2018.
- [40] M. Kahng, D. Fang, and D. H. P. Chau, “Visual exploration of machine learning results using data cube analysis,” in *Workshop on Human-In-the-Loop Data Analytics*, ACM, 2016.
- [41] J. Zhang, Y. Wang, P. Molino, L. Li, and D. S. Ebert, “Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 364–373, 2019.
- [42] J. Chuang, D. Ramage, C. Manning, and J. Heer, “Interpretation and trust: Designing model-driven visualizations for text analysis,” in *ACM Conference on Human Factors in Computing Systems*, ACM, 2012, pp. 443–452.
- [43] D. Gunning, “Explainable artificial intelligence (xai),” *Defense Advanced Research Projects Agency (DARPA)*, 2017.
- [44] M. Card, *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.
- [45] A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli, “Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda,” in *ACM Conference on Human Factors in Computing Systems*, ACM, 2018, p. 582.
- [46] K. A. Cook and J. J. Thomas, “Illuminating the path: The research and development agenda for visual analytics,” Pacific Northwest National Lab. Richland, WA, USA, Tech. Rep., 2005.

- [47] J. Krause, A. Perer, and K. Ng, “Interacting with predictions: Visual inspection of black-box machine learning models,” in *ACM Conference on Human Factors in Computing Systems*, ACM, 2016, pp. 5686–5697.
- [48] 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*, 2019.
- [49] J. Krause, A. Perer, and E. Bertini, “A user study on the effect of aggregating explanations for interpreting machine learning models,” *ACM KDD Workshop on Interactive Data Exploration and Analytics*, 2018.
- [50] R. Sevastjanova, F. Beck, B. Ell, C. Turkay, R. Henkin, M. Butt, D. A. Keim, and M. El-Assady, “Going beyond visualization: Verbalization as complementary medium to explain machine learning models,” in *Workshop on Visualization for AI Explainability at IEEE VIS*, 2018.
- [51] G. Montavon, W. Samek, and K.-R. Müller, “Methods for interpreting and understanding deep neural networks,” *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.
- [52] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati, “Foresight: Recommending visual insights,” *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1937–1940, 2017.
- [53] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko, “Augmenting visualizations with interactive data facts to facilitate interpretation and communication,” *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 672–681, 2018.
- [54] K. Eckelt, P. Adelberger, T. Zichner, A. Wernitznig, and M. Streit, “Tourdino: A support view for confirming patterns in tabular data,” *EuroVis Workshop on Visual Analytics*, 2019.
- [55] M. Gillies, R. Fiebrink, A. Tanaka, J. Garcia, F. Bevilacqua, A. Heloir, F. Nunnari, W. Mackay, S. Amershi, B. Lee, *et al.*, “Human-centred machine learning,” in *ACM Conference Extended Abstracts on Human Factors in Computing Systems*, ACM, 2016, pp. 3558–3565.
- [56] M. Narayanan, E. Chen, J. He, B. Kim, S. Gershman, and F. Doshi-Velez, “How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation,” *arXiv:1802.00682*, 2018.
- [57] F. Poursabzi-Sangdeh, D. G. Goldstein, J. M. Hofman, J. W. Vaughan, and H. Wallach, “Manipulating and measuring model interpretability,” *NIPS Women in Machine Learning Workshop*, 2017.

- [58] D. Collaris, L. M. Vink, and J. J. van Wijk, “Instance-level explanations for fraud detection: A case study,” *ICML Workshop on Human Interpretability in Machine Learning*, 2018.
- [59] B. Kim, W. M., J. Gilmer, C. C., W. J., F. Viegas, and R. Sayres, “Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV),” *ICML*, 2018. arXiv: 1711.11279.
- [60] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, “Network dissection: Quantifying interpretability of deep visual representations,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6541–6549.
- [61] R. Fong and A. Vedaldi, “Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8730–8738.
- [62] C. Olah, A. Mordvintsev, and L. Schubert, “Feature visualization,” *Distill*, 2017.
- [63] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, “Plug & play generative networks: Conditional iterative generation of images in latent space,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4467–4477.
- [64] A. Mordvintsev, C. Olah, and M. Tyka, “Inceptionism: Going deeper into neural networks,” *Google Research Blog*, 2015.
- [65] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *arXiv:1312.6034*, 2013.
- [66] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, “Visualizing higher-layer features of a deep network,” *University of Montreal*, vol. 1341, 2009.
- [67] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, and C. Olah, “Activation atlas,” *Distill*, vol. 4, no. 3, e15, 2019.
- [68] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *ECCV*, Springer, 2014.
- [69] M. Liu, J. Shi, K. Cao, J. Zhu, and S. Liu, “Analyzing the training processes of deep generative models,” *IEEE TVCG*, vol. 24, no. 1, 2018.
- [70] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, “The building blocks of interpretability,” *Distill*, 2018.

- [71] A. Bilal, A. Jourabloo, M. Ye, X. Liu, and L. Ren, “Do convolutional neural networks learn class hierarchy?” *IEEE TVCG*, vol. 24, no. 1, pp. 152–162, 2018.
- [72] R. R. Selvaraju, P. Chattopadhyay, M. Elhoseiny, T. Sharma, D. Batra, D. Parikh, and S. Lee, “Choose your neuron: Incorporating domain knowledge through neuron-importance,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 526–541.
- [73] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, “Towards better analysis of deep convolutional neural networks,” *IEEE TVCG*, vol. 23, no. 1, 2017.
- [74] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mané, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg, “Visualizing dataflow graphs of deep learning models in TensorFlow,” *IEEE TVCG*, vol. 24, no. 1, 2018.
- [75] D. Smilkov, S. Carter, D Sculley, F. B. Viegas, and M. Wattenberg, “Direct-manipulation visualization of deep networks,” in *ICML Workshop on Vis for Deep Learning*, 2016.
- [76] M. Kahng, N. Thorat, D. H. P. Chau, F. B. Viégas, and M. Wattenberg, “Gan lab: Understanding complex deep generative models using interactive visual experimentation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 310–320, 2019.
- [77] A. W. Harley, “An interactive node-link visualization of convolutional neural networks,” in *ISVC*, 2015, pp. 867–877.
- [78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
- [79] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” *arXiv:1609.08144*, 2016.
- [80] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, “Mastering the game of go without human knowledge,” *Nature*, vol. 550, no. 7676, p. 354, 2017.
- [81] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, 2015.
- [82] J. A. Cruz and D. S. Wishart, “Applications of machine learning in cancer prediction and prognosis,” *Cancer Informatics*, vol. 2, p. 117 693 510 600 200 030, 2006.

- [83] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, “Machine learning applications in cancer prognosis and prediction,” *Computational and Structural Biotechnology Journal*, vol. 13, pp. 8–17, 2015.
- [84] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, “Combining satellite imagery and machine learning to predict poverty,” *Science*, vol. 353, no. 6301, pp. 790–794, 2016.
- [85] B. Singh Walia, Q. Hu, J. Chen, F. Chen, J. Lee, N. Kuo, P. Narang, J. Batts, G. Arnold, and M. Madaio, “A dynamic pipeline for spatio-temporal fire risk prediction,” in *ACM International Conference on Knowledge Discovery & Data Mining*, ACM, 2018, pp. 764–773.
- [86] M. Madaio, S.-T. Chen, O. L. Haimson, W. Zhang, X. Cheng, M. Hinds-Aldrich, D. H. Chau, and B. Dilkina, “Firebird: Predicting fire risk and prioritizing fire inspections in atlanta,” in *ACM International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 185–194.
- [87] Q. Yang, J. Suh, N.-C. Chen, and G. Ramos, “Grounding interactive machine learning tool design in how non-experts actually build models,” in *Designing Interactive Systems Conference*, ACM, 2018, pp. 573–584.
- [88] T. J. Hastie and R. Tibshirani, “Generalized additive models,” in *Chapman & Hall/CRC*, 1990.
- [89] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission,” in *ACM International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 1721–1730.
- [90] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker, “Accurate intelligible models with pairwise interactions,” in *ACM International Conference on Knowledge Discovery and Data Mining*, ACM, 2013, pp. 623–631.
- [91] Y. Lou, R. Caruana, and J. Gehrke, “Intelligible models for classification and regression,” in *ACM International Conference on Knowledge Discovery and Data Mining*, ACM, 2012, pp. 150–158.
- [92] H. Hutchinson, W. Mackay, B. Westerlund, B. B. Bederson, A. Druin, C. Plaisant, M. Beaudouin-Lafon, S. Conversy, H. Evans, H. Hansen, *et al.*, “Technology probes: Inspiring design for and with families,” in *ACM Conference on Human Factors in Computing Systems*, ACM, 2003, pp. 17–24.
- [93] C. Graham and M. Rouncefield, “Probes and participation,” in *Conference on Participatory Design*, Indiana University, 2008, pp. 194–197.

- [94] B. Gaver, T. Dunne, and E. Pacenti, “Design: Cultural probes,” *Interactions*, vol. 6, no. 1, pp. 21–29, 1999.
- [95] M. Schmid and T. Hothorn, “Boosting additive models using component-wise p-splines,” *Computational Statistics & Data Analysis*, vol. 53, no. 2, pp. 298–311, 2008.
- [96] K. Jones and S. Almond, “Moving out of the linear rut: The possibilities of generalized additive models,” *Transactions of the Institute of British Geographers*, pp. 434–447, 1992.
- [97] S. N. Wood, *Generalized additive models: an introduction with R*. Chapman and Hall / CRC, 2006.
- [98] D. Servén and C. Brummitt, *Pygam: Generalized additive models in python*, Mar. 2018.
- [99] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Annals of Statistics*, pp. 1189–1232, 2001.
- [100] M. Bostock, V. Ogievetsky, and J. Heer, “D³ data-driven documents,” *IEEE Transactions on Visualization and Computer Graphics*, no. 12, pp. 2301–2309, 2011.
- [101] J. Krause, A. Dasgupta, J. Swartz, Y. Aphinyanaphongs, and E. Bertini, “A workflow for visual diagnostics of binary classifiers using instance-level explanations,” *IEEE Conference on Visual Analytics Science and Technology*, 2017.
- [102] M. T. Ribeiro, S. Singh, and C. Guestrin, “Anchors: High-precision model-agnostic explanations,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [103] N. Kong, M. A. Hearst, and M. Agrawala, “Extracting references between text and charts via crowdsourcing,” in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM, 2014, pp. 31–40.
- [104] B. C. Kwon, F. Stoffel, D. Jäckle, B. Lee, and D. Keim, “Visjockey: Enriching data stories through orchestrated interactive visualization,” in *Poster Compendium of the Computation+ Journalism Symposium*, vol. 3, 2014.
- [105] S. Latif, D. Liu, and F. Beck, “Exploring interactive linking between text and visualization,” in *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers*, Eurographics Association, 2018, pp. 91–94.
- [106] H. Nori, S. Jenkins, P. Koch, and R. Caruana, “Interpretml: A unified framework for machine learning interpretability,” *arXiv preprint arXiv:1909.09223*, 2019.

- [107] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature Machine Intelligence*, vol. 1, no. 5, p. 206, 2019.
- [108] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, no. 4, 1943.
- [109] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” *Neural computation*, vol. 29, no. 9, 2017.
- [110] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *CVPR*, 2015.
- [111] A. Karpathy, *What I learned from competing against a convnet on ImageNet*, 2014.
- [112] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [113] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *CVPR*, 2009.
- [114] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, “Imagenet large scale visual recognition challenge,” *IJCV*, vol. 115, no. 3, 2015.
- [115] M. W. Craven and J. W. Shavlik, “Visualizing learning and computation in artificial neural networks,” *International Journal on Artificial Intelligence Tools*, vol. 1, no. 03, 1992.
- [116] M. J. Streeter, M. O. Ward, and S. A. Alvarez, “NVIS: An interactive visualization tool for neural networks,” in *Visual Data Exploration and Analysis VIII*, International Society for Optics and Photonics, vol. 4302, 2001.
- [117] F.-Y. Tzeng and K.-L. Ma, “Opening the black box: Data driven visualization of neural networks,” in *IEEE Visualization*, 2005.
- [118] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, “Power to the people: The role of humans in interactive machine learning,” *AI Magazine*, vol. 35, no. 4, 2014.
- [119] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, D. Weiskopf, S. North, and D. Keim, “Human-centered machine learning through interactive visualization,” in *ESANN*, 2016.

- [120] C. Seifert, A. Aamir, A. Balagopalan, D. Jain, A. Sharma, S. Grottel, and S. Gumhold, “Visualizations of deep neural networks in computer vision: A survey,” in *Transparent Data Mining for Big and Small Data*, Springer, 2017.
- [121] H. Zeng, “Towards better understanding of deep learning with visualization,” *The Hong Kong University of Science and Technology*, 2016.
- [122] S. Liu, X. Wang, M. Liu, and J. Zhu, “Towards better analysis of machine learning models: A visual analytics perspective,” *Visual Informatics*, vol. 1, no. 1, 2017.
- [123] J. Choo and S. Liu, “Visual analytics for explainable deep learning,” *IEEE Computer Graphics and Applications*, 2018.
- [124] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. 2016.
- [125] A. Weller, “Challenges for transparency,” *ICML Workshop on Human Interpretability in ML*, 2017.
- [126] T. Zahavy, N. Ben-Zrihem, and S. Mannor, “Graying the black box: Understanding DQNs,” in *ICML*, 2016.
- [127] F. Offert, “”i know it when I see it”. visualization and intuitive interpretability,” *NIPS Symposium on Interpretable ML*, 2017.
- [128] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, *et al.*, “Google’s multilingual neural machine translation system: Enabling zero-shot translation,” *arXiv:1611.04558*, 2016.
- [129] C. Robinson, F. Hohman, and B. Dilkina, “A deep learning approach for population estimation from satellite imagery,” in *SIGSPATIAL Workshop on Geospatial Humanities*, Redondo Beach, CA, USA, 2017.
- [130] M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, U. Muller, and K. Zieba, “Visualbackprop: Visualizing cnns for autonomous driving,” *arXiv:1611.05418*, 2016.
- [131] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, “Visualizing deep neural network decisions: Prediction difference analysis,” *arXiv:1702.04595*, 2017.
- [132] L. Li, J. Tompkin, P. Michalatos, and H. Pfister, “Hierarchical visual feature analysis for city street view datasets,” in *Workshop on Visual Analytics for Deep Learning*, 2017.

- [133] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg, “Embedding Projector: Interactive visualization and interpretation of embeddings,” in *NIPS Workshop on Interpretable ML in Complex Systems*, 2016.
- [134] J. Li, X. Chen, E. Hovy, and D. Jurafsky, “Visualizing and understanding neural models in nlp,” *arXiv:1506.01066*, 2015.
- [135] A. Karpathy, J. Johnson, and L. Fei-Fei, “Visualizing and understanding recurrent networks,” *arXiv:1506.02078*, 2015.
- [136] S. Carter, D. Ha, I. Johnson, and C. Olah, “Experiments in handwriting with a neural network,” *Distill*, 2016.
- [137] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, “Network Dissection: Quantifying interpretability of deep visual representations,” in *CVPR*, 2017.
- [138] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, “Grad-CAM: Why did you say that? visual explanations from deep networks via gradient-based localization,” *arXiv:1610.02391*, 2016.
- [139] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, “Synthesizing the preferred inputs for neurons in neural networks via deep generator networks,” in *NIPS*, 2016.
- [140] K. Patel, J. Fogarty, J. A. Landay, and B. Harrison, “Investigating statistical machine learning as a tool for software development,” in *CHI*, 2008.
- [141] T. Kulesza, M. Burnett, W.-K. Wong, and S. Stumpf, “Principles of explanatory debugging to personalize interactive machine learning,” in *IUI*, 2015.
- [142] B. Nushi, E. Kamar, E. Horvitz, and D. Kossmann, “On human intellect and machine failures: Troubleshooting integrative machine learning systems,” in *AAAI*, 2017.
- [143] H. Strobelt, S. Gehrman, H. Pfister, and A. M. Rush, “LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks,” *IEEE TVCG*, vol. 24, no. 1, 2018.
- [144] N. Pezzotti, T. Höllt, J. Van Gemert, B. P. Lelieveldt, E. Eisemann, and A. Vilanova, “DeepEyes: Progressive visual analytics for designing deep neural networks,” *IEEE TVCG*, vol. 24, no. 1, 2018.
- [145] W. Zhong, C. Xie, Y. Zhong, Y. Wang, W. Xu, S. Cheng, and K. Mueller, “Evolutionary visual analysis of deep neural networks,” in *ICML Workshop on Vis for Deep Learning*, 2017.

- [146] X. Rong and E. Adar, “Visual tools for debugging neural language models,” in *ICML Workshop on Vis for Deep Learning*, 2016.
- [147] J. Chae, S. Gao, A. Ramanthan, C. Steed, and G. D. Tourassi, “Visualization for classification in deep neural networks,” in *Workshop on Visual Analytics for Deep Learning*, 2017.
- [148] E. Alexander and M. Gleicher, “Task-driven comparison of topic models,” *IEEE TVCG*, vol. 22, no. 1, 2016.
- [149] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hralfkelsson, T. Boulos, and J. Kubica, “Ad click prediction: A view from the trenches,” in *KDD*, 2013.
- [150] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, “TensorFlow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv:1603.04467*, 2016.
- [151] W. Yu, K. Yang, Y. Bai, H. Yao, and Y. Rui, “Visualizing and comparing convolutional neural networks,” *arXiv:1412.6631*, 2014.
- [152] Y. Ming, S. Cao, R. Zhang, Z. Li, and Y. Chen, “Understanding hidden memories of recurrent neural networks,” *VAST*, 2017.
- [153] H. Zeng, H. Haleem, X. Plantaz, N. Cao, and H. Qu, “CNNComparator: Comparative analytics of convolutional neural networks,” in *Workshop on Visual Analytics for Deep Learning*, 2017.
- [154] B. Webster, “Now anyone can explore machine learning, no coding required,” *Google Official Blog*, 2017.
- [155] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, “Understanding neural networks through deep visualization,” in *ICML Deep Learning Workshop*, 2015.
- [156] M. Wattenberg, F. Viégas, and I. Johnson, “How to use t-SNE effectively,” *Distill*, 2016.
- [157] C. Olah, “Visualizing MNIST,” *Olah’s Blog*, 2014.
- [158] D. Bruckner, “MI-o-scope: A diagnostic visualization system for deep machine learning pipelines,” Master’s thesis, EECS Department, University of California, Berkeley, 2014.

- [159] F. Hohman, N. Hodas, and D. H. Chau, “ShapeShop: Towards understanding deep learning representations via interactive experimentation,” in *CHI, Extended Abstracts*, 2017.
- [160] S. Chung, C. Park, S. Suh, K. Kang, J. Choo, and B. C. Kwon, “ReVACNN: Steering convolutional neural network via real-time visual analytics,” in *NIPS Workshop on Future of Interactive Learning Machines*, 2016.
- [161] D. Cashman, G. Patterson, A. Mosca, and R. Chang, “RNNbow: Visualizing learning via backpropagation gradients in recurrent neural networks,” in *Workshop on Visual Analytics for Deep Learning*, 2017.
- [162] P. E. Rauber, S. G. Fadel, A. X. Falcao, and A. C. Telea, “Visualizing the hidden activity of artificial neural networks,” *IEEE TVCG*, vol. 23, no. 1, 2017.
- [163] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE,” *JMLR*, vol. 9, no. Nov, 2008.
- [164] A. Nguyen, J. Yosinski, and J. Clune, “Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks,” in *ICML Workshop on Vis for Deep Learning*, 2016.
- [165] P. E. Rauber, A. X. Falcão, and A. C. Telea, “Visualizing time-dependent data using dynamic t-sne,” *EuroVis*, vol. 2, no. 5, 2016.
- [166] S. Liu, P.-T. Bremer, J. J. Thiagarajan, V. Srikanth, B. Wang, Y. Livnat, and V. Pasucci, “Visual exploration of semantic relationships in neural word embeddings,” *IEEE TVCG*, vol. 24, no. 1, 2018.
- [167] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, “Hoggles: Visualizing object detection features,” in *ICCV*, 2013.
- [168] Y. Goyal, A. Mohapatra, D. Parikh, and D. Batra, “Towards transparent ai systems: Interpreting visual question answering models,” *arXiv:1608.08974*, 2016.
- [169] X. Rong, “Word2vec parameter learning explained,” *arXiv:1411.2738*, 2014.
- [170] D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, and N. Elmquist, “ConceptVector: Text visual analytics via interactive lexicon building using word embedding,” *IEEE TVCG*, vol. 24, no. 1, 2018.
- [171] C. Olah and S. Carter, “Research debt,” *Distill*, 2017.
- [172] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, “Generative visual manipulation on the natural image manifold,” in *ECCV*, Springer, 2016.

- [173] A. P. Norton and Y. Qi, “Adversarial-Playground: A visualization suite showing how adversarial examples fool deep learning,” in *VizSec*, IEEE, 2017.
- [174] A. Mahendran and A. Vedaldi, “Visualizing deep convolutional neural networks using natural pre-images,” *IJCV*, vol. 120, no. 3, 2016.
- [175] F. Grün, C. Rupprecht, N. Navab, and F. Tombari, “A taxonomy and library for visualizing learned features in convolutional neural networks,” *ICML Workshop on Vis for Deep Learning*, 2016.
- [176] P.-J. Kindermans, K. T. Schütt, M. Alber, K.-R. Müller, and S. Dähne, “Learning how to explain neural networks: Patternnet and patternattribution,” *arXiv:1705.05598*, 2017.
- [177] H. Li, K. Mueller, and X. Chen, “Beyond saliency: Understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation,” *arXiv:1712.08268*, 2017.
- [178] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, “SmoothGrad: Removing noise by adding noise,” in *ICML Workshop on Vis for Deep Learning*, 2017.
- [179] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *CVPR*, 2016.
- [180] A. Dosovitskiy and T. Brox, “Inverting visual representations with convolutional networks,” in *CVPR*, 2016.
- [181] A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” in *CVPR*, 2015.
- [182] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Object detectors emerge in deep scene CNNs,” *arXiv:1412.6856*, 2014.
- [183] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for simplicity: The all convolutional net,” *arXiv:1412.6806*, 2014.
- [184] D. Wei, B. Zhou, A. Torrabla, and W. Freeman, “Understanding intra-class knowledge inside CNN,” *arXiv:1507.02379*, 2015.
- [185] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune, “Plug & play generative networks: Conditional iterative generation of images in latent space,” *arXiv:1612.00005*, 2016.
- [186] Y. Bengio *et al.*, “Learning deep architectures for ai,” *Foundations and trends in Machine Learning*, vol. 2, no. 1, 2009.

- [187] S. Carter and M. Nielsen, “Using artificial intelligence to augment human intelligence,” *Distill*, 2017.
- [188] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” in *CVPR*, 2015.
- [189] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C Lawrence Zitnick, and D. Parikh, “Vqa: Visual question answering,” in *ICCV*, 2015.
- [190] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NIPS*, 2014.
- [191] I. Goodfellow, “NIPS 2016 tutorial: Generative adversarial networks,” *arXiv:1701.00160*, 2016.
- [192] J. Wang, L. Gou, H. Yang, and H.-W. Shen, “GANViz: A visual analytics approach to understand the adversarial game,” *IEEE TVCG*, 2018.
- [193] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.
- [194] Y Liu, T Kohlberger, M Norouzi, G. Dahl, J. Smith, A Mohtashamian, N Olson, L. Peng, J. Hipp, and M. Stumpe, “Artificial intelligence-based breast cancer nodal metastasis detection.,” *Archives of Pathology & Laboratory Medicine*, vol. 143, no. 7, pp. 859–868, 2019.
- [195] D. F. Steiner, R. MacDonald, Y. Liu, P. Truszkowski, J. D. Hipp, C. Gammage, F. Thng, L. Peng, and M. C. Stumpe, “Impact of deep learning assistance on the histopathologic review of lymph nodes for metastatic breast cancer,” *The American Journal of Surgical Pathology*, vol. 42, no. 12, pp. 1636–1646, 2018.
- [196] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, S. Li, L. Chen, M. E. Kounavis, and D. H. Chau, “Shield: Fast, practical defense and vaccination for deep learning using jpeg compression,” *arXiv:1802.06816*, 2018.
- [197] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, “Taskonomy: Disentangling task transfer learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3712–3722.
- [198] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

- [199] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423.
- [200] A. N. Langville and C. D. Meyer, “A survey of eigenvector methods for web information retrieval,” *SIAM Review*, vol. 47, no. 1, pp. 135–161, 2005.
- [201] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web,” Stanford InfoLab, Tech. Rep., 1999.
- [202] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction,” *ArXiv e-prints*, Feb. 2018. arXiv: 1802.03426 [stat.ML].
- [203] G. W. Furnas, *Generalized fisheye views*, 4. Bell Communications Research. Morris Research and Engineering Center . . ., 1986, vol. 17.
- [204] F. Van Ham and A. Perer, ““search, show context, expand on demand”: Supporting large graph exploration with degree-of-interest,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 953–960, 2009.
- [205] T. Crnovrsanin, I. Liao, Y. Wu, and K.-L. Ma, “Visual recommendations for network navigation,” in *Computer Graphics Forum*, Wiley Online Library, vol. 30, 2011, pp. 1081–1090.
- [206] S. Kairam, N. H. Riche, S. Drucker, R. Fernandez, and J. Heer, “Refinery: Visual exploration of large, heterogeneous networks through associative browsing,” in *Computer Graphics Forum*, Wiley Online Library, vol. 34, 2015, pp. 301–310.
- [207] E. Greussing and H. G. Boomgaarden, “Simply bells and whistles? cognitive effects of visual aesthetics in digital longforms,” *Digital Journalism*, vol. 7, no. 2, pp. 273–293, 2019.
- [208] R. K. Hernandez and J. Rue, *The principles of multimedia journalism: Packaging digital news*. Routledge, 2015.
- [209] B. Victor, “Explorable explanations,” *Bret Victor*, vol. 10, 2011.
- [210] M. Wattenberg, F. Viegas, and M. Hardt, “Attacking discrimination with smarter machine learning,” *Google Research Website*, 2016.
- [211] “Workshop on visualization for ai explainability,” *IEEE Visualization*, 2018.

- [212] M. Conlen, A. Kale, and J. Heer, “Capture & analysis of active reading behaviors for interactive articles on the web,” in *Computer Graphics Forum*, Wiley Online Library, vol. 38, 2019, pp. 687–698.
- [213] O. Shehata and M. Conlen, “Unraveling the jpeg,” *The Parametric Press, Issue 01*, 2019.
- [214] A. P. Key, F. Hohman, M. Conlen, and S. Stalla, “Data science for fair housing,” *The Parametric Press, Issue 01*, 2019.
- [215] J. McGirk and M. Conlen, “Flatland follies: An adjunct simulator,” *The Parametric Press, Issue 01*, 2019.
- [216] R. M. Bianchi, F. Hohman, and M. Conlen, “On particle physics,” *The Parametric Press, Issue 01*, 2019.
- [217] M. Vo, M. Conlen, and V. Uren, “Anything that flies, on anything that moves,” *The Parametric Press, Issue 01*, 2019.
- [218] “The special rapporteur’s 2017 report to the united nations human rights council is now online,” *United Nations Human Rights Office of the High Commissioner*, 2017.
- [219] A. C. Kay, “A personal computer for children of all ages,” in *Proceedings of the ACM Annual Conference*, ACM, 1972.
- [220] D. C. Engelbart, “Augmenting human intellect: A conceptual framework,” *Menlo Park, CA*, 1962.
- [221] N. Stephenson, *The diamond age*. Penguin UK, 1998.
- [222] “Knowledge navigator,” *Apple Inc*, 1987.
- [223] T. H. Nelson, “Getting it out of our system,” *Information Retrieval: A Critical Review*, pp. 191–210, 1967.
- [224] “Plato,” *University of Illinois*, 1960.
- [225] “Phet interactive simulations,” *University of Colorado Boulder*, 2002.
- [226] J. Katz and W. Andrews, “How y’all, youse and you guys talk,” *The New York Times*, 2013.
- [227] J. Branch, *Snow fall: The avalanche at tunnel creek*. 2014.

- [228] H. Stevens, “Why outbreaks like coronavirus spread exponentially, and how to “flatten the curve”,” *The Washington Post*, 2020.
- [229] “Explorable explanation,” *Wikipedia*, 2019.
- [230] P. Dragicevic, Y. Jansen, A. Sarma, M. Kay, and F. Chevalier, “Increasing the transparency of research papers with explorable multiverse analyses,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ACM, 2019, p. 65.
- [231] A. Ynnerman, J. Löwgren, and L. Tibell, “Exploranation: A new science communication paradigm,” *IEEE computer graphics and applications*, vol. 38, no. 3, pp. 13–20, 2018.
- [232] B. Lee, N. H. Riche, P. Isenberg, and S. Carpendale, “More than telling a story: Transforming data into visually shared stories,” *IEEE Computer Graphics and Applications*, vol. 35, no. 5, pp. 84–90, 2015.
- [233] J. Sarama and D. H. Clements, “‘concrete’ computer manipulatives in mathematics education,” *Child Development Perspectives*, vol. 3, no. 3, pp. 145–150, 2009.
- [234] E. J. Aarseth, *Cybertext: Perspectives on ergodic literature*. JHU Press, 1997.
- [235] J. H. Sizemore and J. Zhu, “Interactive non-fiction: Towards a new approach for storytelling in digital journalism,” in *International Conference on Interactive Digital Storytelling*, Springer, 2011, pp. 313–316.
- [236] T. Yamamiya, A. Warth, and T. Kaehler, “Active essays on the web,” in *2009 Seventh International Conference on Creating, Connecting and Collaborating through Computing*, IEEE, 2009, pp. 3–10.
- [237] I. Bogost, S. Ferrari, and B. Schweizer, *Newsgames: Journalism at play*. Mit Press, 2012.
- [238] W. C. Fredrick and H. J. Walberg, “Learning as a function of time,” *The Journal of Educational Research*, vol. 73, no. 4, pp. 183–194, 1980.
- [239] E. Um, J. L. Plass, E. O. Hayward, B. D. Homer, *et al.*, “Emotional design in multimedia learning,” *Journal of Educational Psychology*, vol. 104, no. 2, p. 485, 2012.
- [240] F. Amini, N. H. Riche, B. Lee, J. Leboe-McGowan, and P. Irani, “Hooked on data videos: Assessing the effect of animation and pictographs on viewer engagement,” in *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, 2018, pp. 1–9.

- [241] B. Tversky, J. B. Morrison, and M. Betrancourt, “Animation: Can it facilitate?” *International Journal of Human-computer Studies*, vol. 57, no. 4, pp. 247–262, 2002.
- [242] J. Heer and G. Robertson, “Animated transitions in statistical data graphics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1240–1247, 2007.
- [243] J. Hullman, P. Resnick, and E. Adar, “Hypothetical outcome plots outperform error bars and violin plots for inferences about reliability of variable ordering,” *PLOS One*, vol. 10, no. 11, e0142444, 2015.
- [244] A. Michotte, “La perception de la causalité.(etudes psychol.), vol. vi,” 1946.
- [245] F. Thomas, O. Johnston, and F. Thomas, *The illusion of life: Disney animation*. Hyperion New York, 1995.
- [246] J Muybridge, “The horse in motion,” *Nature*, vol. 25, no. 652, p. 605, 1882.
- [247] B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch, “Emergent tool use from multi-agent autocurricula,” *arXiv:1909.07528*, 2019.
- [248] E. Hawkins, “Climate spirals,” *Climate Lab Book*, 2016.
- [249] E. Roston and B. Migliozzi, “What’s really warming the world,” *Bloomberg*, 2015.
- [250] T. Randall and B. Migliozzi, “Earth’s relentless warming sets a brutal new record in 2017,” *Bloomberg*, 2018.
- [251] “Global temperature,” *NASA Global Climate Change*, 2020.
- [252] N Popovich and A Pearce, “It’s not your imagination. summers are getting hotter.,” *The New York Times*, 2017.
- [253] E. Badger, C. C. Miller, A. Pearce, and K. Quealy, “Extensive data shows punishing reach of racism for black boys,” *The New York Times*, 2018.
- [254] A. Cox and K. Quealy, *Disagreements*, 2018.
- [255] B. Casselman, M. Conlen, and R. Fischer-Baum, “Gun deaths in america,” *FiveThirtyEight*, 2016.
- [256] N. Halloran, *The fallen of world war ii*, web, 2015.

- [257] J. Boy, A. V. Pandey, J. Emerson, M. Satterthwaite, O. Nov, and E. Bertini, “Showing people behind data: Does anthropomorphizing visualizations elicit more empathy for human rights data?” In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 5462–5474.
- [258] M. A. Borkin, Z. Bylinskii, N. W. Kim, C. M. Bainbridge, C. S. Yeh, D. Borkin, H. Pfister, and A. Oliva, “Beyond memorability: Visualization recognition and recall,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 519–528, 2015.
- [259] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister, “What makes a visualization memorable?” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2306–2315, 2013.
- [260] S. Slobin, “What if the data visualization is actually people,” *Source*, 2014.
- [261] M. Correll, “Ethical dimensions of visualization research,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–13.
- [262] A. Ivanov, K. Danyluk, C. Jacob, and W. Willett, “A walk among the data,” *IEEE Computer Graphics and Applications*, vol. 39, no. 3, pp. 19–28, 2019.
- [263] S. McKenna, N Henry Riche, B. Lee, J. Boy, and M. Meyer, “Visual narrative flow: Exploring factors shaping data visualization story reading experiences,” in *Computer Graphics Forum*, Wiley Online Library, vol. 36, 2017, pp. 377–387.
- [264] Q. Zhi, A. Ottley, and R. Metoyer, “Linking and layout: Exploring the integration of text and visualization in storytelling,” in *Computer Graphics Forum*, Wiley Online Library, vol. 38, 2019, pp. 675–685.
- [265] S. Webworks and D. Crothers, “Cutthroat capitalism: The game,” *Wired*, 2009.
- [266] K. Squire, “Video games and learning,” *Teaching and Participatory Culture in the Digital Age*, 2011.
- [267] D. Blood, J. S. Kao, N. Knoll, R. Kwong, C. Locke, and Æ. Rininsland, “The uber game,” *Financial Times*, 2017.
- [268] M. Virvou, G. Katsionis, and K. Manos, “Combining software games with education: Evaluation of its educational effectiveness,” *Journal of Educational Technology & Society*, vol. 8, no. 2, pp. 54–65, 2005.
- [269] E. Segel and J. Heer, “Narrative visualization: Telling stories with data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1139–1148, 2010.

- [270] V. Hart and N. Case, “Parable of the polygons,” 2016.
- [271] B. Victor, “Drawing dynamic visualizations,” 2013.
- [272] M. J. Adler and C. Van Doren, *How to read a book: The classic guide to intelligent reading*. Simon and Schuster, 2014.
- [273] B. Victor, “Scientific communication as sequential art,” 2011.
- [274] B. Webster, “Designing (and learning from) a teachable machine,” *Google Design*, 2018.
- [275] N. Yau, “How you will die,” *Flowing Data*, 2016.
- [276] A. M. Barry-Jester, B. Casselman, and D. Goldstein, “Should prison sentences be based on crimes that haven’t been committed yet?” *FiveThirtyEight*, 2015.
- [277] M. Wattenberg, F. Viégas, and I. Johnson, “How to use t-sne effectively,” *Distill*, vol. 1, no. 10, e2, 2016.
- [278] A. Coenen and A. Pearce, “Understanding umap,” *Google PAIR*, 2019.
- [279] D. Smilkov, N. Thorat, Y. Assogba, A. Yuan, N. Kreeger, P. Yu, K. Zhang, S. Cai, E. Nielsen, D. Soergel, *et al.*, “Tensorflow.js: Machine learning for the web and beyond,” *arXiv:1901.05350*, 2019.
- [280] M. Kahng, N. Thorat, D. H. P. Chau, F. B. Viégas, and M. Wattenberg, “Gan lab: Understanding complex deep generative models using interactive visual experimentation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1–11, 2018.
- [281] “Who will win the presidency,” *FiveThirtyEight*, 2016.
- [282] J. Katz, “Who will be president,” *The New York Times*, 2016.
- [283] “Live results: Presidential slection,” *The Washington Post*, 2016.
- [284] R. E. Mayer, “Multimedia learning,” in *Psychology of Learning and Motivation*, vol. 41, Elsevier, 2002, pp. 85–139.
- [285] G. Sanderson, *3blue1brown*.
- [286] J. Helps, *Primer*.

- [287] R. E. Mayer and C. I. Johnson, “Revising the redundancy principle in multimedia learning,” *Journal of Educational Psychology*, vol. 100, no. 2, p. 380, 2008.
- [288] G. Sanderson and B. Eater, “Visualizing quaternions: An explorable video series,” 2018.
- [289] M. T. Chi, M. Bassok, M. W. Lewis, P. Reimann, and R. Glaser, “Self-explanations: How students study and use examples in learning to solve problems,” *Cognitive Science*, vol. 13, no. 2, pp. 145–182, 1989.
- [290] M. T. Chi, “Self-explaining expository texts: The dual processes of generating inferences and repairing mental models,” *Advances in Instructional Psychology*, vol. 5, pp. 161–238, 2000.
- [291] Y.-S. Kim, K. Reinecke, and J. Hullman, “Explaining the gap: Visualizing one’s predictions improves recall and comprehension of data,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ACM, 2017, pp. 1375–1386.
- [292] F. Nguyen, Y.-S. Kim, J. Germuska, and J. Hullman, “They draw it!” *Midwest Uncertainty Collective and The Knight Lab.*, 2019.
- [293] G. Aisch, A. Cox, and K. Quealy, “You draw it: How family income predicts children’s college chances,” *The New York Times*, 2015.
- [294] J. Katz, “You draw it: Just how bad is the drug overdose epidemic,” *The New York Times*, 2017.
- [295] L. Buchanan, H. Park, and A. Pearce, “You draw it: What got better or worse during obama’s presidency,” *The New York Times*, 2017.
- [296] R. Goldenberg and M. Daniels, “The gyllenhaal experiment,” *The Pudding*, 2019.
- [297] Y.-S. Kim, K. Reinecke, and J. Hullman, “Data through others’ eyes: The impact of visualizing others’ expectations on visualization interpretation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 760–769, 2017.
- [298] T.-H. Ha and N. Sonnad, *How do you draw a circle? we analyzed 100,000 drawings to show how culture shapes our instincts*, web, 2017.
- [299] A. I. Gates, *Recitation as a factor in memorizing*, 40. Science Press, 1922.
- [300] H. L. Roediger III and J. D. Karpicke, “The power of testing memory: Basic research and implications for educational practice,” *Perspectives on Psychological Science*, vol. 1, no. 3, pp. 181–210, 2006.

- [301] Khan academy, 2008.
- [302] R. L. Bangert-Drowns, C.-L. C. Kulik, J. A. Kulik, and M. Morgan, “The instructional effect of feedback in test-like events,” *Review of Educational Research*, vol. 61, no. 2, pp. 213–238, 1991.
- [303] N. Case, “How to remember anything for forever-ish,” 2018.
- [304] J. D. Karpicke and H. L. Roediger, “The critical importance of retrieval for learning,” *Science*, vol. 319, no. 5865, pp. 966–968, 2008.
- [305] A. Matuschak and M. A. Nielsen, *Quantum country*. 2019.
- [306] D. I. Cordova and M. R. Lepper, “Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice.,” *Journal of Educational Psychology*, vol. 88, no. 4, p. 715, 1996.
- [307] C. Di Marco, P. Bray, H. D. Covvey, D. D. Cowan, V. Di Ciccio, E. Hovy, J. Lipa, and C. Yang, “Authoring and generation of individualized patient education materials,” in *AMIA Annual Symposium Proceedings*, American Medical Informatics Association, vol. 2006, 2006, p. 195.
- [308] E. Adar, C. Gearig, A. Balasubramanian, and J. Hullman, “Persalog: Personalization of news article content,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ACM, 2017, pp. 3188–3200.
- [309] Y.-S. Kim, J. Hullman, and M. Agrawala, “Generating personalized spatial analogies for distances and areas,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM, 2016, pp. 38–48.
- [310] N. Popovich, B. Migliozzi, R. Taylor, J. Williams, and D. Watkins, “How much hotter is your hometown than when you were born?” *The New York Times*, 2018.
- [311] M. Daniels, “Human terrain,” *The Pudding*, 2018.
- [312] K. Quealy, R. Gebeloff, and R. Taylor, “Are you rich? this income-rank quiz might change how you see yourself,” *The New York Times*, 2019.
- [313] S. Chinoy, “Quiz: Let us predict whether you’re a democrat or a republican,” *The New York Times*, 2019.
- [314] M. Whitehouse and M. Rojanasakul, “Find out if your job will be automated,” *Bloomberg*, 2017.

- [315] E. Lowther, L. Huynh, M. Bryson, and S. Connor, “Booze calculator: What’s your drinking nationality?” *BBC*, 2017.
- [316] S. Beckett, “Click 1,000: How the pick-your-own-path episode was made,” *BBC*, 2019.
- [317] R. C. Clark and R. E. Mayer, *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. John Wiley & Sons, 2016.
- [318] R. E. Mayer, G. T. Dow, and S. Mayer, “Multimedia learning in an interactive self-explaining environment: What works in the design of agent-based microworlds?” *Journal of Educational Psychology*, vol. 95, no. 4, p. 806, 2003.
- [319] R. E. Mayer, P. Mautone, and W. Prothero, “Pictorial aids for learning by doing in a multimedia geology simulation game.,” *Journal of Educational Psychology*, vol. 94, no. 1, p. 171, 2002.
- [320] S. Yee and T. Chu, “A visual introduction to machine learning,” *R2D3*, 2015.
- [321] B. Craft and P. Cairns, “Beyond guidelines: What can we learn from the visual information seeking mantra?” In *Ninth International Conference on Information Visualisation (IV’05)*, IEEE, 2005, pp. 110–118.
- [322] B. Shneiderman, “The eyes have it: A task by data type taxonomy for information visualizations,” in *Proceedings 1996 IEEE Symposium on Visual Languages*, IEEE, 1996, pp. 336–343.
- [323] D. A. Keim, “Information visualization and visual data mining,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 1–8, 2002.
- [324] J. Ashkenas and A. Parlapiano, *How the recession shaped the economy, in 255 charts*, web, 2014.
- [325] “How does the eye work?” *Explorable Explanations Game Jam*, 2018.
- [326] C. Gingold, “Earth primer,” 2015.
- [327] G. Goh, “Why momentum really works,” *Distill*, 2017.
- [328] K. Azad, “Colorized math equations,” *Better Explained*, 2017.
- [329] V. Powell, “Image kernels,” *Setosa*, 2015.
- [330] T. Nelson, “Stretchtext - hypertext note #8,” *Project Xanadu*, 1967.

- [331] W. Beecroft, “On variable level-of-detail documents,” 2016.
- [332] T. Petricek, “Coeffects: Context-aware programming languages,” 2017.
- [333] “The parametric press: Call for proposals winter/spring 2019,” *The Parametric Press*, 2019.
- [334] K. Basques, “A ui that lets readers control how much information they see,” 2018.
- [335] “Wikipedia preview card,” *Wikipedia*, 2018.
- [336] P. Zellweger, B.-W. Chang, and J. D. Mackinlay, “Fluid links for informed and incremental link transitions,” 1998.
- [337] P. T. Zellweger, A. Mangen, and P. Newman, “Reading and writing fluid hypertext narratives,” in *Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia*, ACM, 2002, pp. 45–54.
- [338] M. Conlen and J. Heer, “Idyll: A markup language for authoring and publishing interactive articles on the web,” in *ACM User Interface Software & Technology (UIST)*, 2018.
- [339] *Apparatus*, 2015.
- [340] *Observable*, 2019.
- [341] N. Case, “LOOPY: A tool for thinking in systems,” 2017.
- [342] C. N. Klokmose, J. R. Eagan, S. Baader, W. Mackay, and M. Beaudouin-Lafon, “Webstrates: Shareable dynamic media,” in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 2015, pp. 280–290.
- [343] M. A. Nielsen, *Neural networks and deep learning*. Determination Press, 2015.
- [344] N. Case, “How i make explorable explanations,” 2017.
- [345] N. Case, “Explorable explanations: 4 more design patterns,” 2018.
- [346] C. D. Stolper, B. Lee, N. H. Riche, and J. Stasko, “Emerging and recurring data-driven storytelling techniques: Analysis of a curated collection of recent stories,” *Microsoft Research*, 2016.
- [347] J. Hoffswell, W. Li, and Z. Liu, “Techniques for flexible responsive visualization design,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ACM, 2020.

- [348] M. Brehmer, B. Lee, P. Isenberg, and E. K. Choe, “A comparative evaluation of animation and small multiples for trend visualization on mobile phones,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 364–374, 2019.
- [349] M. Brehmer, B. Lee, P. Isenberg, and E. K. Choe, “Visualizing ranges over time on mobile phones: A task-based crowdsourced evaluation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 619–629, 2018.
- [350] A. Tse, “Why we are doing fewer interactives,” *Malofiej Infographics World Summit*, 2016.
- [351] G. Aisch, “In defense of interactive graphics,” 2017.
- [352] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, “Visualizing high-dimensional data: Advances in the past decade,” *IEEE TVCG*, vol. 23, no. 3, 2017.
- [353] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” *arXiv:1312.6199*, 2013.
- [354] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,” in *CVPR*, 2015.
- [355] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, “On detecting adversarial perturbations,” in *ICLR*, 2017.
- [356] S. Gu and L. Rigazio, “Towards deep neural network architectures robust to adversarial examples,” *arXiv:1412.5068*, 2014.
- [357] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in *Security and Privacy*, 2016.
- [358] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” *arXiv:1607.02533*, 2016.
- [359] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, “Evaluating the visualization of what a deep neural network has learned,” *IEEE transactions on neural networks and learning systems*, 2017.
- [360] C.-Y. Tsai and D. D. Cox, “Characterizing visual representations within convolutional neural networks: Toward a quantitative approach,” *ICML Workshop on Vis for Deep Learning*, 2016.

- [361] S. Ritter, D. G. Barrett, A. Santoro, and M. M. Botvinick, “Cognitive psychology for deep neural networks: A shape bias case study,” *arXiv:1706.08606*, 2017.
- [362] S. Barocas and A. D. Selbst, “Big data’s disparate impact,” *Calif. L. Rev.*, vol. 104, pp. 671–769, 2016.
- [363] “Facets,” *Google PAIR*, 2017.
- [364] M. Hardt, E. Price, N. Srebro, *et al.*, “Equality of opportunity in supervised learning,” in *NIPS*, 2016.
- [365] E. Wall, L. Blaha, L. Franklin, and A. Endert, “Warning, bias may occur: A proposed approach to detecting cognitive bias in interactive visual analytics,” *VAST*, 2017.