

**DEMOCRATIZING HUMAN-CENTERED AI WITH VISUAL EXPLANATION
AND INTERACTIVE GUIDANCE**

A Dissertation
Presented to
The Academic Faculty

By

Zijie J. Wang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Machine Learning

Georgia Institute of Technology

December 2024

© Zijie J. Wang 2024

**DEMOCRATIZING HUMAN-CENTERED AI WITH VISUAL EXPLANATION
AND INTERACTIVE GUIDANCE**

Thesis committee:

Dr. Duen Horng Chau
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Lauren Wilcox
Responsible AI & School of Interactive Computing
eBay & Georgia Institute of Technology

Dr. Judy Hoffman
School of Interactive Computing
Georgia Institute of Technology

Dr. Jenn Wortman Vaughan
Fairness, Accountability, Transparency,
and Ethics in AI
Microsoft Research

Dr. Munmun De Choudhury
School of Interactive Computing
Georgia Institute of Technology

Dr. Rich Caruana
Deep Learning Foundations Group
Microsoft Research

Date approved: July 26, 2024

The medium is the message.

Marshall McLuhan

To my parents, Huanqiong and Xiaogang,
for everything.

ACKNOWLEDGMENTS

Doing a PhD truly takes a village. For me to have reached this point, it takes a significant amount of luck and unwavering support from a multitude of incredible mentors, colleagues, friends, and family. I am deeply grateful to each and every one of you.

First and foremost, I want to thank my advisor, Polo Chau, for his relentless support and guidance throughout my PhD journey. I could not ask for a better advisor. Without his mentorship, I would not be where I am today. Polo's cheerful attitude, compassion, attention to detail, and steadfast belief in my potential have greatly influenced my growth as a researcher, designer, and writer. His mentorship has not only taught me how to tackle challenging questions but also how to identify important ones to address and effectively communicate my work. I am forever grateful to Polo for his continued support, valuable advice, and countless hours he has dedicated to me.

I would like to thank the members of my thesis and qualifier exam committee, Judy Hoffman, Munmun De Choudhury, Lauren Wilcox, Rich Caruana, Jenn Wortman Vaughan, and Diyi Yang for their invaluable advice in shaping my dissertation. I am also incredibly grateful to Anthony Gitter, Michael Gleicher, and Yu Hen Hu for taking a chance on me when I was an undergraduate student with no research experience. They showed me how fun and rewarding conducting research can be.

I feel incredibly fortunate to have had the opportunity to be mentored by some of the smartest and kindest people during four different internships in the industry.

Liang Gou hosted me for my first-ever internship in industry at Bosch Research. Despite the chaos of the early months of the COVID pandemic, Liang provided me with a wonderful learning experience. Thank you for showing me how to apply my research skills to address critical questions across various industrial sectors.

Rich Caruana, Jenn Wortman Vaughan, and Mihaela Vorvoreanu at Microsoft Research provided me with the freedom and unwavering support to select research problems that are both important and foundational to my thesis. I am sincerely grateful for their mentorship and friendship beyond the internship. Their life and career advice have had a profound impact on my development as a researcher.

Fred Hohman, Mary Beth Kery, and Dominik Moritz at Apple were incredibly supportive and eager to share their advice on conducting research in industry settings. They also showed me the importance of networking and various ways to make an impact. I am grateful for their guidance, which has given me confidence in my research endeavors.

Finally, at Google Research, Mike Terry, Michael Madaio, Lauren Wilcox, Chinmay Kulkarni taught me how to effectively collaborate with people with diverse backgrounds and roles in a large organization. Mike reminded me of the significance of rapid feedback and socializing ideas. Micahel taught me the importance of deep thinking and maintaining a focus on the bigger picture.

In addition to mentors, I must also express my gratitude to the friends and collaborators I have had the pleasure of working with over the years. They have provided invaluable support and have been strong advocates for me.




I am grateful to be a part of the exceptional research group Polo Club, who have offered invaluable feedback for my work and always reminded me of the fun of research. I would like to especially thank my incredible academic siblings: Haekyu Park, Austin Wright, Fred Hohman, Scott Freitas, Nilaksh Das, Rahul Duggal, Seongmin Lee, Ben Hoover, Anthony Peng, Matthew Hull, Alec Helbling, and Mansi Phute. In addition, I want to extend a thank you to the undergraduate students who collaborated closely with me, especially David Munechika, Robert Turko, and Aishwarya Chakravarthy. You have shown me the joy of mentoring students, and I am looking forward to seeing what you all do in the future. I also want to thank the Georgia Tech Visualization Lab for the warm and supportive community.

My PhD journey has been remarkably smooth, all thanks to the amazing staff in the School of Computational Science and Engineering and the Machine Learning PhD program. I want to give special recognition to Bryant Wine, Stephanie Niebuhr, Nirvana Edwards, and Holly Rush, who go above and beyond to enhance my PhD experience.

I would like to express my gratitude to the fantastic friends I have made on this journey. Kaan Sancak, thank you for always being there for me. Muhammed Fatih Balin, Gaurav Verma, Yu Fu, Sichen Jin, Shengyu Xu, Hannah Kim, Arpit Narechania, Grace Guo, Adam Coscia, and Alexander Bendeck, thank you for bringing joy to my everyday life. And to Zhi Chen, Alex Kale, Sam Robertson, Luis Morales-Navarro, and Jaemarie Solyst, thank you for your companionship and for being a source of inspiration.

Finally, I could never be where I am without the sacrifices, love, and support of my parents. This PhD is dedicated to my mother, Huanqiong, and my father, Xiaogang.

TABLE OF CONTENTS

Acknowledgments	v
List of Tables	xiv
List of Figures	xv
Summary	xxiii
Chapter 1: Introduction	1
1.1 Thesis Overview	2
1.1.1 Part I: Explain AI to Everyone  Explain	2
1.1.2 Part II: Guide AI with Human Values  Guide	5
1.1.3 Part III: Democratize Human-Centered AI  Democratize	6
1.2 Thesis Statement	8
1.3 Research Contributions	8
1.4 Impact	10
Chapter 2: Background and Related Work	11
2.1 Explainable AI	11
2.1.1 Explaining AI to Experts	11
2.1.2 Explaining AI to Non-experts	12
2.1.3 Explaining AI Models and Data with Embeddings	12
2.1.4 Explaining AI Usage	13

2.2	Human Guidance in AI	13
2.2.1	Model Editing	13
2.2.2	Algorithmic Recourse	13
2.3	Democratizing Human-centered AI	14
2.3.1	Existing Responsible AI Tools and Practices	14
2.3.2	Anticipating Technology’s Negative Impacts	14
2.3.3	Identifying and Mitigating LLM Harms	15
2.3.4	<i>In Situ</i> Interfaces	15
I	EXPLAIN AI TO EVERYONE	17
	Chapter 3: CNN EXPLAINER: Explain Convolutional Neural Networks to AI	
	Novices	19
3.1	Introduction	19
3.2	Formative Research & Design Challenges	22
3.3	Design Goals	24
3.4	Visualization Interface of CNN EXPLAINER	25
3.4.1	Overview	26
3.4.2	Elastic Explanation View	27
3.4.3	Interactive Formula View	27
3.4.4	Transitions Between Views	29
3.4.5	Visualizations with Explanations	29
3.4.6	Customizable Visualizations	29
3.4.7	Web-based, Open-sourced Implementation	30
3.5	Usage Scenarios	31
3.5.1	Beginner Learning Layer Connectivity	31
3.5.2	Teaching Through Interactive Experimentation	32

3.6	Observational Study	32
3.6.1	Participants	32
3.6.2	Procedure	33
3.6.3	Results and Design Lessons	34
3.7	Discussion and Future Work	37
3.8	Conclusion	38
3.9	Impact	38
Chapter 4: WIZMAP: Explain AI Data and Embeddings to Practitioners		39
4.1	Introduction	39
4.2	Multi-scale Embedding Summarization	41
4.3	User Interface	42
4.3.1	Map View	42
4.3.2	Control Panel	43
4.3.3	Search Panel	44
4.3.4	Scalable & Open-source Implementation	44
4.4	Usage Scenarios	44
4.4.1	Exploring ACL Research Topic Trends	44
4.4.2	Investigating Text-to-Image Model Usage	45
4.5	Conclusion	46
Chapter 5: DIFFUSIONDB: Explain AI Usage to Researchers and Policymakers		47
5.1	Introduction	47
5.2	Constructing DIFFUSIONDB	49
5.2.1	Collecting User Generated Images	49
5.2.2	Extracting Image Metadata	49

5.2.3	Identifying NSFW Content	50
5.2.4	Organizing DIFFUSIONDB	51
5.2.5	Distributing DIFFUSIONDB	51
5.3	Data Analysis	51
5.3.1	Prompt Length	52
5.3.2	Prompt Language	52
5.3.3	Characterizing Prompts	52
5.3.4	Characterizing Images	55
5.3.5	Stable Diffusion Error Analysis	56
5.3.6	Potentially Harmful Uses	58
5.4	Enabling New Research Directions	58
5.5	Limitations	59
5.6	Conclusion	60

II GUIDE AI WITH HUMAN VALUES 61

Chapter 6: GAM CHANGER: Align AI Models through Model Editing 63

6.1	Introduction	63
6.2	Novel User Experience	66
6.2.1	Intuitive and Flexible Editing	66
6.2.2	Safe and Responsible Editing	69
6.2.3	Scalable, Open-source Implementation	70
6.3	Impacts to physicians	70
6.3.1	Fixing Sepsis Risk Prediction	71
6.3.2	Repairing Pneumonia Risk Prediction	74
6.4	Impacts beyond healthcare	76

6.4.1	Study Design	76
6.4.2	Benefits to Data Scientists	77
6.5	Discussion and Future Work	81
6.6	Conclusion	82
6.7	Impact	82
Chapter 7: GAM COACH: Helping People Alter Unfavorable AI Decisions . . .		83
7.1	Introduction	83
7.2	Design Goals	86
7.3	Techniques for Customizable Recourse Generation	87
7.3.1	Model Choice	87
7.3.2	CF Generation: Integer Linear Programming	88
7.3.3	Recourse Customization	90
7.4	User Interface	90
7.4.1	Coach Menu	90
7.4.2	Feature Panel	91
7.4.3	Bookmarks and Receipt	93
7.4.4	Usage Scenarios	93
7.4.5	Open-source & Generalizable Tool	95
7.5	User Study	96
7.5.1	Participants	96
7.5.2	Study Design	96
7.5.3	Results	98
7.6	Limitations	103
7.7	Discussion	105
7.8	Conclusion	106

III GUIDE AI WITH HUMAN VALUES 107

Chapter 8: FARSIGHT: Fostering Responsible AI Awareness During AI Prototyping	108
8.1 Introduction	108
8.2 Formative Study & Design Goals	111
8.2.1 Co-design Study	112
8.2.2 Design Goals	113
8.3 User Interface	115
8.3.1 Alert Symbol	115
8.3.2 Awareness Sidebar	117
8.3.3 Harm Envisioner	118
8.3.4 Open-source and Reusable Implementation	120
8.4 Usage Scenario	121
8.5 Evaluation User Study	122
8.5.1 Participants	122
8.5.2 Study Design	124
8.5.3 Data Analysis	128
8.5.4 Findings: Changes in Users' Envisioning Ability and Approach (RQ1)	130
8.5.5 Findings: FARSIGHT's Effectiveness in Assisting Harm Envisioning (RQ2)	134
8.5.6 Findings: FARSIGHT's Role in Overcoming Harm Envisioning Challenges (RQ3)	138
8.5.7 Limitations of Study Design	141
8.6 Discussion	142
8.6.1 Motivation & Engagement in Responsible AI	142
8.6.2 Subjectivity in Harm Envisioning	143
8.6.3 Mitigating Harms during AI Prototyping	144

8.7	Conclusion	145
8.8	Research Contributions	146
8.9	Impact	147
8.10	Future Directions	148
8.10.1	Human-Centered AI for All	148
8.10.2	Interactive AI Alignment	151
8.10.3	On-device Computing for Human-centered AI	152
8.11	Conclusion	154

LIST OF TABLES


8.3 We identified six non-exclusive common patterns in independent harm envisioning by analyzing transcripts of participants' think-aloud process during H1 and H3. 131

LIST OF FIGURES

1.1	In this thesis, we democratize human-centered AI by innovating techniques and tools that explain AI to humans and empower humans to guide AI with their values.	1
1.2	My thesis includes three complementary parts. Each part addresses one research question with research answers and example works mapped to seven chapters of the thesis.	2
1.3	CNN EXPLAINER is an interactive visualization tool that empowers AI novices to easily learn how a convolutional neural network (CNN) transforms an input image into a category prediction. Leveraging smooth transitions and animations, the tool integrates multiple views with different levels of abstraction that explain both high-level model structures and low-level mathematical operations.	3
1.4	An overview of the interface of WIZMAP, a scalable visual analytics tool that enables AI practitioners and researchers to easily explore and interpret <i>millions</i> of embedding vectors across different levels of granularity. To help users quickly make sense of the embedding space, WIZMAP automatically generates multi-resolution embedding summaries across different neighborhoods.	4
1.5	An example image in DIFFUSIONDB generated by a real user.	5
1.6	GAM CHANGER empowers AI practitioners and domain experts to easily and responsibly align AI model’s behaviors with their knowledge and values, via direct manipulation of the weights of Generalized Additive Models (GAMs). In addition to offering (A) easy-to-use editing interfaces, our tool actively promotes responsible editing by providing users with continuous feedback regarding (B1) the impacts of their edits, (B2) feature correlations, and (B3) a comprehensive edit history.	6

1.7	GAM COACH is the first interactive tool enabling people impacted by AI-based decision-making systems to iteratively generate algorithmic recourse plans that reflect their preferences. In this example, after a user specifies the difficulties and acceptable ranges of changing different features, the tool suggests increasing the FICO score and decreasing credit utilization to get loan approval.	7
1.8	FARSIGHT is a collection of <i>in situ</i> interfaces and novel techniques that empower AI prototypers to envision potential harms that may arise from their large language model-powered AI applications during early prototyping. In this example, an AI prototyper is crafting prompts for an English-to-French translator, and FARSIGHT alerts the user with potential harms by highlighting news articles relevant to the user’s prompt and LLM-generated potential user cases and harms.	8
3.1	In CNN EXPLAINER, tightly integrated views with different levels of abstractions work together to help users more easily learn about the intricate interplay between a CNN’s high-level structure and low-level mathematical operations. (A) the <i>Overview</i> summarizes connections of all neurons; (B) the <i>Elastic View</i> animates the intermediate convolutional computation of the user-selected neuron in the <i>Overview</i> ; and (C) <i>Interactive Formula</i> interactively demonstrates the detailed calculation on the selected input in the <i>Elastic View</i>	19
3.2	CNN EXPLAINER empowers AI novices to easily learn how CNNs transform an input image into a category prediction. (A) The <i>Overview</i> visualizes a CNN architecture where each neuron is encoded as a square with a heatmap representing its output. (B) Clicking a neuron reveals how its activations are computed from the previous layer through animations of sliding kernels. (C) <i>Convolutional Interactive Formula View</i> explains underlying mathematics of convolutions.	21
3.3	Survey results from 19 past CNN learners.	22

3.4	CNN EXPLAINER helps users learn about the connection between the output layer and its previous layer via three tightly integrated views. Users can smoothly transition between these views to gain a more holistic understanding of the output layer’s <code>lifeboat</code> prediction computation. (A) The <i>Overview</i> summarizes neurons and their connections. (B) The <i>Flatten Elastic Explanation View</i> visualizes the often-overlooked flatten layer, helping users more easily understand how a high-dimensional <code>max_pool_2</code> layer is connected to the 1-dimensional output layer. (C) The <i>Softmax Interactive Formula View</i> further explains how the softmax function that precedes the output layer normalizes the penultimate computation results (i.e., logits) into class probabilities by linking the (C1) numbers from the formula to (C2) their visual representations within the model structure.	25
3.5	Illustration of <i>Tiny VGG</i> model used in CNN EXPLAINER: this model uses the same, but fewer, convolutional layers as the VGGNet model [174]. We trained it to classify 10 classes of images.	26
3.6	Diverging color scales in CNN EXPLAINER.	26
3.7	The <i>Interactive Formula Views</i> explain the underlying mathematical operations of a CNN. (A) shows the element-wise dot-product occurring in a convolutional neuron, (B) visualizes the activation function ReLU, and (C) illustrates how max pooling works. Users can hover over heatmaps to display an operation’s input-to-output mapping. (D) interactively explains the softmax function, helping users connect numbers from the formula to their visual representations. Users can click the info button ⓘ to scroll to the corresponding section in the tutorial article, and the play button ▶ to start the window sliding animation in (A)-(C).	28
3.8	The <i>Hyperparameter Widget</i> , a component of the accompanying interactive article, allows users to adjust hyperparameters and observe in real time how the kernel’s sliding pattern changes.	30
3.9	Average ratings from 16 participants regarding the usability and usefulness of CNN EXPLAINER. Top: Participants thought CNN EXPLAINER was easy to use, enjoyable, and helped them learn about CNNs. Bottom: All features, especially animations, were rated favorably.	33
4.1	WIZMAP enables users to explore embeddings at different levels of detail. (A) The contour plot with automatically-generated embedding summaries provides an overview. (B) Embedding summaries adjust in resolution as users zoom in. (C) The scatter plot enables the investigation of individual embeddings.	39

4.2	WIZMAP empowers AI researchers and practitioners to easily explore and interpret <i>millions</i> of embedding vectors across different levels of granularity. Consider the task of investigating the embeddings of all 63k natural language processing paper abstracts indexed in ACL Anthology from 1980 to 2022. (A) The Map View tightly integrates a contour layer, a scatter plot, and automatically generated multi-resolution embedding summaries to help users navigate through the large embedding space. (B) The Search Panel enables users to rapidly test their hypotheses through a fast full-text embedding search. (C) The Control Panel allows users to customize embedding visualizations, compare multiple embedding groups, and observe how embeddings evolve over time.	40
4.3	(A) A quadtree recursively partitions a 2D space into four equally-sized squares, (B) and each square is represented as a tree node. WIZMAP efficiently aggregates information from the leaves to the root, summarizing embeddings at different levels of granularity.	41
4.4	The <i>Map View</i> provides an overview via a contour plot and auto-generated multi-resolution embedding labels placed around high-density areas.	42
4.5	WIZMAP allows users to observe how embeddings change over time. For example, when exploring 63k ACL paper abstracts, clicking the play button  in the <i>Control Panel</i> animates the visualizations to show embeddings of papers published in each year in purple and the distribution of all papers in blue . This animation highlights changes in ACL research topics over time, such as the decline in popularity of grammar and the rise of question-answering.	43
4.6	WIZMAP enables users to compare multiple embeddings by visualization superposition. For instance, comparing the CLIP embeddings of 1.8 million Stable Diffusion prompts and 1.8 million generated images reveals key differences between two distributions.	46
5.1	DIFFUSIONDB is the first large-scale dataset featuring 6.5TB data including 1.8 million unique Stable Diffusion prompts and 14 million generated images with accompanying hyperparameters. It provides exciting research opportunities in prompt engineering, deepfake detection, and understanding large generative models.	47
5.2	DIFFUSIONDB contains 14 million Stable Diffusion images, 1.8 million unique text prompts, and all model hyperparameters. Each image also has a unique filename, a hash of its creator’s identifier, a creation timestamp, and an NSFW score computed by state-of-the-art models.	48

5.3	To help researchers filter out potentially unsafe data in DIFFUSIONDB, we apply NSFW detectors to predict the probability that an image-prompt pair contains NSFW content. For images, a score of 2.0 indicates the image has been blurred by Stable Diffusion.	50
5.4	The distribution of token counts for all 1.8 million unique prompts in DIFFUSIONDB. It is worth noting that Stable Diffusion truncates prompts at 75 tokens.	52
5.5	We identify and group popular phrases in prompts through named entity recognition and dependency parsing. Our interactive circle-packing visualization highlights the distribution and hierarchy of these phrases. (A) The <i>Overview</i> visualizes each phrase as a circle, with its size representing the phrase’s frequency. In this example, a viewer clicks a circle to zoom into the “painting” phrase. (B1) The <i>Detail View</i> shows all noun phrases that use “painting” as their root. (B2) Similarly, it shows all phrases that include “oil painting” when the viewer zooms into “oil painting.”	53
5.6	An interactive plot of 1.8M prompts’ CLIP embeddings, created with UMAP and kernel density estimation. Text labels show the top keywords of prompts in a grid tile. It reveals popular prompt topics.	54
5.7	CLIP embeddings of 2M randomly selected images, with text labels being keywords of prompts in the grid tiles. It shows images have a different embedding distribution from prompts.	55
5.8	Example generated image that is semantically different from its prompt. . .	56
5.9	Example generated image that is semantically different from its prompt. . .	57
5.10	Example generated image that is semantically different from its prompt. . .	57
5.11	Example generated image that is semantically different from its prompt. . .	57
6.1	(A) Domain experts such as physicians often hesitate to trust ML models as they cannot understand how the models make predictions. (B) Interpretability reveals models can learn potentially harmful patterns. (C) Model editing turns interpretability into action—enabling domain experts to align model behaviors with their knowledge and values.	63

- 6.2 GAM CHANGER empowers domain experts and data scientists to easily and responsibly align model behaviors with their knowledge and values, via direct manipulation of GAM model weights. Take a healthcare model for example. (A) The *GAM Canvas* enables physicians to interpolate the predicted risk of dying from pneumonia to match their clinical knowledge of a gradual risk increase from age 81 to age 87. (B1) The *Metric Panel* provides real-time feedback on model performance. (B2) The *Feature Panel* helps users identify characteristics of affected samples and promotes awareness of fairness issues. (B3) The *History Panel* allows users to compare and revert changes, as well as document their motivations and editing contexts. . . . 64
- 6.3 The *GAM Canvas* employs different designs to visualize shape functions on different feature types. We use @line charts for continuous variables, @bar charts for categorical variables, @heatmaps for interaction effects of two continuous variables, @vertical bar charts for interaction effects between continuous and categorical variables, and @scatter plots for interaction effect of two categorical variables. For univariate features, the x-axis encodes the input feature x_j , and the y-axis represents the output of the shape function $f_j(x_j)$. We also use light-blue bands and error bars to represent the prediction confidence. For pair-wise interactions, the axes encode two features, and we use a diverging color scale to represent the contribution scores. . . . 67
- 6.4 The *Context Toolbar* enables users to edit GAMs with a variety of editing tools. Users can use the move tool ↻ to adjust the contribution scores of selected bins by dragging bins up and down. Users can apply the interpolate tool ↻ to linearly interpolate the scores of an interval of bins from the start to the end. Alternatively, users can interpolate scores with an arbitrary number of equal bins ↻, or by fitting a linear regression ↻. With minimal changes, the monotonicity tool transforms the selected scores into a monotonically increasing function ↻ or a monotonically decreasing function ↻. With align tools, users can unify the selected scores as the score of the left bin ↻, the right bin ↻, or the average score weighted by the training sample counts ↻. 67
- 6.5 On a GAM trained to predict house price, a user selects bins representing high-quality houses in the *GAM Canvas*. B1 For categorical variables, the *Feature Panel* shows that selected houses disproportionately have better exterior and kitchen quality and locate in certain neighborhoods. B2 For continuous variables, the year built and garage area are also highly correlated with the house quality. . . . 68
- 6.6 A GAM learns a few strange patterns between patients' temperature and sepsis risk that need to be fixed. B1 We smooth out the sudden increase of risk ↻ around 96°F, B2 flatten the risk ↻ to reflect a treatment effect, and B3 smooth out risk fluctuations ↻ at high temperature. . . . 71

6.7	<p>AContrary to clinical knowledge, a GAM predicts sepsis risk decreases when the respiratory rate decreases (left), and the risk score fluctuates when the rate increases (right). We align the model behaviors by B1raising risk scores and B2removing risk fluctuations with monotonicity.</p>	72
6.8	<p>AAgainst physicians’ expectations, a GAM predicts that patients with lower blood pressure have lower sepsis risk (left), and the risk abruptly increases at high blood pressure (right). To create a safer model, B1 we raise the risk scores, and B1 smooth out the sudden risk increase.</p>	74
6.9	<p>AContrary to physicians’ knowledge, a GAM predicts an abrupt increase of risk from age 86 to 87 (left), and that patients above 100 years old have lower pneumonia risk than patients 20 years younger (right). B1With the interpolation tool, we smooth out the abrupt increase of risk. B2We use the align tool to raise the risk score for older patients.</p>	75
6.10	<p>AA GAM predicts having asthma lowers the risk of dying from pneumonia. BWe address this problematic pattern by removing the predictive effect of having asthma.</p>	76
6.11	<p>Average ratings from 7 participants for GAM CHANGER’s usability and usefulness. (A) All participants enjoyed using the tool; they found it highly usable and it meets their editing needs. (B) All features, especially enforcing monotonicity and removing effects, were rated favorably.</p>	78
6.12	<p>Shape function of debt to income ration on the loan approval prediction.</p>	80
7.1	<p>GAM Coach enables end users to iteratively finetune recourse plans. (A) If a user finds the initial generic plan less actionable, (B) they can specify their recourse preferences through simple interactions. (C) Our tool will then generate tailored plans that reflect the user’s preferences.</p>	83
7.2	<p>GAM COACH enables people impacted by AI-based decision-making systems to iteratively generate algorithmic recourse plans that reflect their preferences. Take the loan application as an example. (A) The Coach Menu helps a rejected loan applicant browse diverse recourse plans that would lead to loan approval. After the user selects a plan, (B) the Feature Panel visualizes all feature information with progressive disclosure, enabling users to explore how hypothetical inputs affect the model’s decision and specify recourse preferences—such as (B1) the difficulty of changing a feature and (B2) its acceptable range of values—guiding GAM Coach to generate actionable plans. (C) The Bookmarks window allows users to compare bookmarked plans and save a verifiable receipt.</p>	84

7.3 A bar chart visualizes the model’s decision score of a recourse plan: the bar is marked with the user’s original score (shorter vertical line on the left) and the threshold needed to obtain the desired decision (longer vertical line on the right). 90

8.1 (A) AI prototypers from diverse backgrounds and roles use (B) prompting tools to prototype AI applications. FARSIGHT provides a range of *in situ* widgets for these tools, helping AI prototypers envision the potential harms of their AI applications during an early prototyping stage. 108

8.4 Average ratings on our design ideas from 10 AI prototypers. Features marked with ♣ were presented to participants as early-stage prototypes, while other features were presented as sketches (see details in Fig. 8.6). 113

8.17 Average ratings of envisioning tool features. 137

SUMMARY

While artificial intelligence (AI) systems have been increasingly integrated into our everyday lives, how they make predictions often remains obscure to both their developers and the people they impact. The opacity of AI models contributes to their perception as “mysterious”—rendering both developers and those impacted by these models powerless when it comes to aligning AI models with their values.

My dissertation aims to address these challenges with a human-centered approach, by designing and developing novel techniques and easy-to-adopt interactive tools that explain and guide AI models. Specifically, this thesis focuses on three complementary thrusts:

1. **Explain AI to Everyone.** We pioneer easy-to-access interactive visualization systems that help AI novices and experts understand AI models (e.g., WIZMAP and CNN EXPLAINER used by 360k+ novices worldwide). We also present first-of-its-kind resources (e.g., 6.5TB DIFFUSIONDB with 14 million prompt-image pairs) to help AI developers and policymakers understand the impacts of large generative AI models.
2. **Guide AI with Human Values.** To harness the potential of AI, gaining a better understanding of it is not enough. We empower AI developers to vet and fix problematic model behaviors (e.g., GAM CHANGER deployed by Microsoft) and those impacted by AI to receive customizable suggestions to alter unfavorable AI decisions (e.g., GAM COACH).
3. **Democratize Human-Centered AI.** Human-centered AI practices are maximally valuable when they find practical adoption. To lower the barrier to applying these practices, we introduce *in situ* tools (e.g., FARSIGHT) to foster responsible AI awareness among practitioners during the prototyping stage within their current workflows.

Our work is making significant impacts on academia, industry, and society: CNN EXPLAINER has helped 360k+ novices learn about CNNs worldwide, and it has been integrated into deep learning courses (Carnegie Mellon, Georgia Tech, Duke University, University of Tokyo and more). It has also been highlighted as a top visualization publication (top 1%) invited to SIGGRAPH. FARSIGHT has received a CHI Best Paper Honorable Mention award. DIFFUSIONDB has received an ACL Best Paper Honorable Mention award. GAM CHANGER has received the Best Paper Award at NeurIPS Workshop on Bridging the Gap: From ML Research to Clinical Practice, and the tool is now deployed by Microsoft and integrated into their inheritability library. Our work has been recognized by an Apple Scholars in AI/ML PhD fellowship and a J.P. Morgan AI PhD Fellowship.

CHAPTER 1

INTRODUCTION

As AI models have grown increasingly complex, how they make predictions is often unknown to both their developers and the people they impact. The “black-box” nature of AI models presents challenges for developers in **understanding their behaviors and impacts**, making it difficult to anticipate and prevent harms that may arise from deploying these models until it is too late. Examples include representing gender bias in the AI-powered hiring process [1], discriminating racial minorities in recidivism predictions [2], and being vulnerable to human-imperceptible adversarial attacks[3].

Also, the opacity of AI models contributes to their perception as “mysterious” and “unpredictable” [4]—rendering both developers and those impacted by these models *powerless* when it comes to **exercising human agency** for guiding AI models or seeking remedies for unfavorable AI predictions. For instance, in the algorithmic hiring example, due to a lack of understanding of job screening models and techniques to fix problematic model behaviors, developers struggle to mitigate model biases [5]. Similarly, due to the opacity surrounding these models and a lack of familiarity with AI, job applicants find themselves with limited recourse options when their applications are rejected by AI models [6].

To develop and deploy trustworthy AI systems that benefit everyone, there is an urgent need to have the capability to thoroughly *vet and rectify* AI models. First, we need to explain what AI models have learned and how they make predictions. After gaining an understanding of these models and their potential impacts, it is essential to ensure that they have acquired the correct knowledge and that their behaviors align with human values. As these solutions to **AI explainability** and **human agency** emerge, ensuring their **accessibility and ease of adoption** by AI developers is of paramount importance. After all, responsible AI techniques are maximally valuable when AI developers actively embrace them.

This thesis (Fig. 1.1) aims to address these critical challenges by developing new

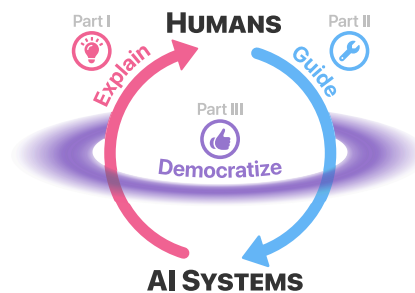


Figure 1.1: In this thesis, we **democratize** human-centered AI by innovating techniques and tools that **explain** AI to humans and empower humans to **guide** AI with their values.

Democratizing Human-Centered AI with Visual Explanation and Interactive Guidance

Part I



Explain AI to everyone

How to help people with different AI backgrounds and needs understand AI models, data, and their impacts?

Part II



Guide AI with human values

How to empower AI stakeholders exercise their human agency when interacting with AI?

Part III



Democratize human-centered AI

How to make human-centered AI design and development practices accessible to all?

Chapter 3 VIS'20

CNN EXPLAINER Explain AI model to novices

Chapter 4 ACL'23

WIZMAP Explain AI data and embeddings to practitioners

Chapter 5 ACL'23

DIFFUSIONDB Explain AI usage and impacts

Chapter 6 KDD'22

GAM CHANGER Edit AI models to reflect human values

Chapter 7 CHI'23

GAM COACH Alter unfavorable AI decisions




Chapter 8 CHI'24

FARSIGHT In situ responsible AI recommendation

Figure 1.2: My thesis includes three complementary parts. Each part addresses one research question with research answers and example works mapped to seven chapters of the thesis.

paradigms, techniques, and interactive tools that empower people with diverse ML backgrounds to **gain an understanding** of AI models, enable AI developers and those impacted by AI systems to **guide AI**, and **democratize human-centered AI** by making it accessible and readily adoptable in AI researchers and practitioners' workflows.

1.1 Thesis Overview

To *foster understanding, agency, and adoption in human-centered AI*, this thesis studies how to explain AI technologies to people with different AI backgrounds and needs (**Part I** ) , how to enable human agency in AI (**Part II** ) , and how to make human-centered AI accessible to all (**Part III** ) . This thesis addresses three complementary research questions (Fig. 1.2) with answers and example works in seven chapters.

1.1.1 Part I: Explain AI to Everyone Explain

There has been an increasing body of research that aims to help *AI experts* interpret *AI model weights*. However, AI now impacts everyone—it is crucial that everyone knows how AI works and how to use it. Moreover, good AI models require high-quality training data. To avoid the classic “garbage in, garbage out” problem, AI practitioners have a pressing need to make sense of the relationship between data and models. Yet understanding the model and data is not enough, to develop and use AI in responsible ways, everyone must learn about different potential use cases of AI technologies and their societal impacts.

In the first part of the thesis, we develop interactive visualization tools that explain

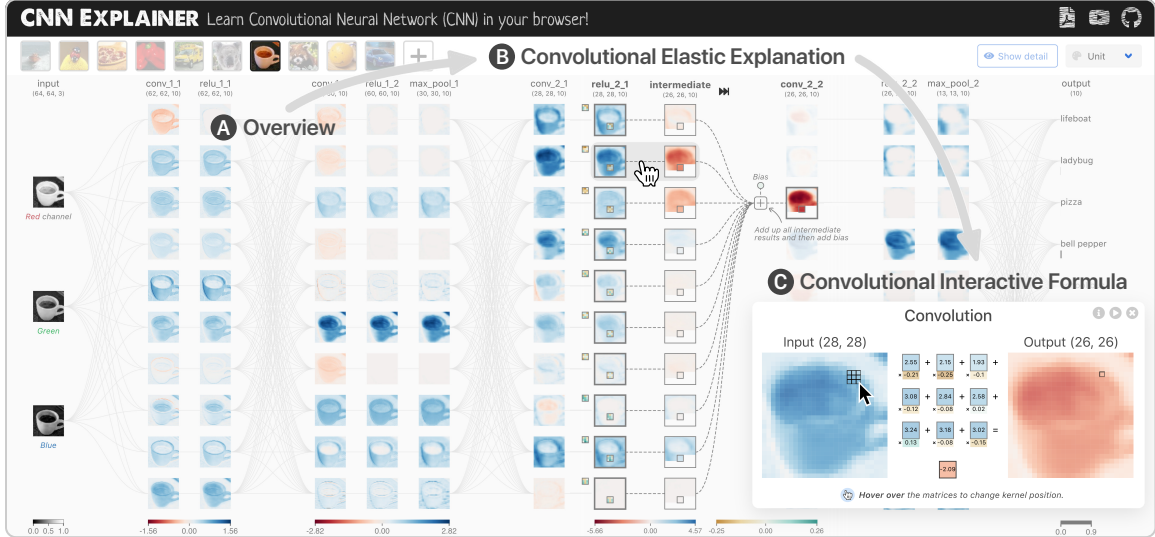


Figure 1.3: CNN EXPLAINER is an interactive visualization tool that empowers AI novices to easily learn how a convolutional neural network (CNN) transforms an input image into a category prediction. Leveraging smooth transitions and animations, the tool integrates multiple views with different levels of abstraction that explain both high-level model structures and low-level mathematical operations.

complex AI models to people without technical expertise (CNN EXPLAINER, Chapter 3) and elucidate large AI datasets (WIZMAP, Chapter 4). To help people learn about the impacts of AI, we introduce a first-of-its-kind dataset documenting how real users engage with generative AI models (DIFFUSIONDB, Chapter 5).

CNN EXPLAINER: *Explaining Convolutional Neural Networks to AI Novices* Chapter 3

Through a human-centered iterative design process, we design and develop CNN EXPLAINER (Fig. 1.3), an interactive visualization tool that helps AI novices learn about the inner workings of convolutional neural networks (CNNs). CNN EXPLAINER addresses key learning challenges that we identified through interviews with instructors and a survey with past students. Our tool tightly integrates a model overview (Fig. 1.3A) summarizing a CNN’s high-level structures, and on-demand, dynamic visual explanation views (Fig. 1.3-BC) that elucidate the low-level transformation mechanisms. Leveraging animation and smooth transitions across levels of abstraction, CNN EXPLAINER enables users to connect CNN’s high-level structures to its low-level mathematical operations. An observation user study highlights that our tool helps non-experts learn about the inner mechanisms. CNN EXPLAINER has transformed AI education: its open-source demo has been integrated into deep learning courses (Carnegie Mellon, Georgia Tech, Duke University, University of Tokyo, UC Santa Barbara, Texas A&M and more), helping 340k+ novices from 200+ countries learn about seemingly complex ML concepts, and it has received 6.9k+ stars on GitHub.

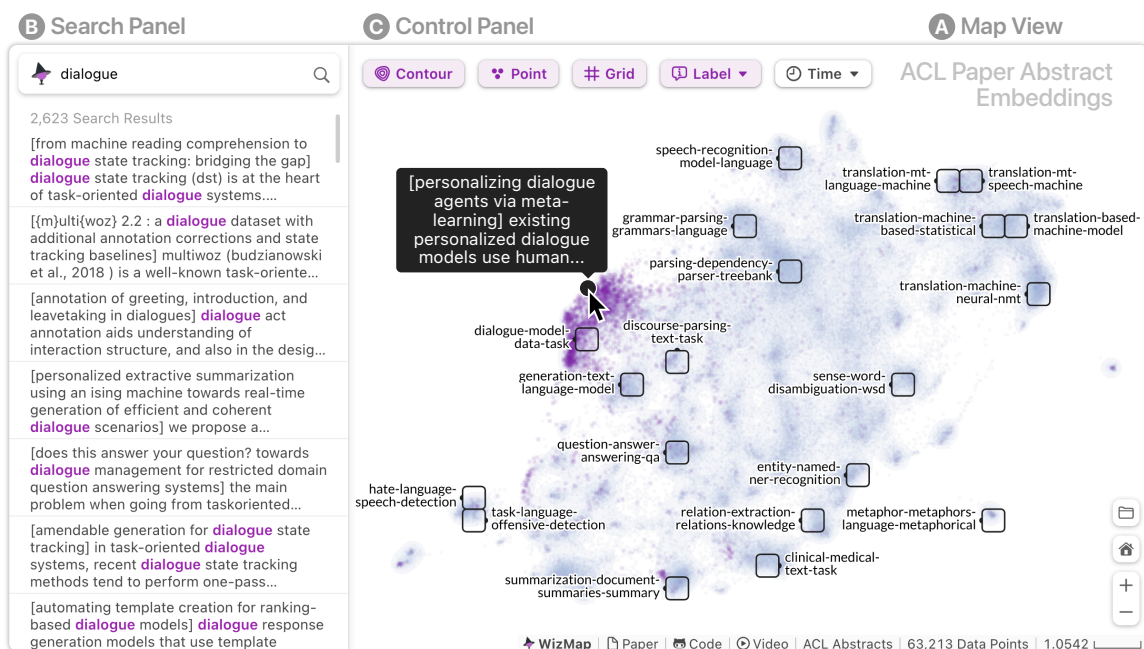


Figure 1.4: An overview of the interface of WIZMAP, a scalable visual analytics tool that enables AI practitioners and researchers to easily explore and interpret *millions* of embedding vectors across different levels of granularity. To help users quickly make sense of the embedding space, WIZMAP automatically generates multi-resolution embedding summaries across different neighborhoods.

WIZMAP: Interpreting and Exploring Large AI Embeddings Chapter 4

AI non-experts face challenges in learning complex AI models, and AI experts also struggle to make sense of their trained models. To help AI practitioners interpret AI models and training data, we developed WIZMAP (Fig. 1.4), a scalable visual analytics tool that empowers users to easily explore large embeddings of AI models. An embedding is a latent representation of what an AI model has learned from its training data—especially valuable for interpreting the model, building new models, and analyzing datasets. We introduce a novel multi-resolution embedding summarization method that guides users to dynamically decipher different neighborhoods in the embedding space. WIZMAP leverages modern web technologies such as WebGL and Web Workers to scale to millions of embedding points directly in users’ web browsers without the need for dedicated backend servers.

DIFFUSIONDB: Understanding How Real Users Use Generative AI Models Chapter 5

Besides the challenges of understanding AI faced by both AI novices and experts during development, practitioners and researchers also encounter difficulties in anticipating how real users would use a deployed AI model. Given the rapidly growing prevalence of AI in every aspect of our daily lives, it has never been more critical to understand how a model is used—the understanding is essential for practitioners and policymakers to assess its societal impact and mitigate potential harms.



Figure 1.5: An example image in DIFFUSIONDB generated by a real user.

To help address this challenge, we introduce DIFFUSIONDB, the first large-scale dataset logging real user interactions with Stable Diffusion, a popular text-to-image generation model. DIFFUSIONDB totals 6.5TB, containing 14 million images generated by real users, accompanied by 1.8 million unique text prompts and rich metadata. Our dataset enables exciting research opportunities in prompt engineering, regulation, AI interpretability, and deepfake detection.

1.1.2 Part II: Guide AI with Human Values [Guide](#)

Gaining a better understanding of AI (Part I) is not enough. To harness AI’s potential for enhancing people’s lives and preventing potential harms, it is crucial to translate our understanding of AI into actions that align AI models’ behaviors with human knowledge and values. To do that, we introduce novel interaction techniques (e.g., GAM CHANGER) that empower AI practitioners to vet and fix problematic model behaviors through model editing (Chapter 6). Moreover, we present new algorithms and tools (e.g., GAM COACH) that provide individuals impacted by AI with personalized and customizable suggestions that can alter unfavorable AI decisions (Chapter 7).

GAM CHANGER: *Editing AI Models to Reflect Human Knowledge and Values* [Chapter 6](#)

Through a collaboration between AI and human-computer interaction researchers, physicians, and data scientists across Georgia Tech, Microsoft Research, and NYU Langone Hospital, we design and develop GAM CHANGER (Fig. 1.6), the first interactive tool that enables AI practitioners and domain experts to easily and responsibly edit Generalized Additive Models (GAMs) and fix undesired behaviors. GAMs are a widely-used model class known for their predictive performance, which rivals that of complex black-box models, while remaining simple enough for humans to understand their decision-making process. With GAM CHANGER’s user-friendly interactive interfaces, even users without any programming backgrounds can easily modify the model weights of their trained GAMs. To guard against potentially harmful edits, GAM CHANGER offers users continuous feedback about feature correlations and the impacts of their edits on different subgroups. Furthermore, our tool allows users to document and undo any edits. GAM CHANGER has been deployed at Microsoft and integrated into their open-source library InterpretML [7].

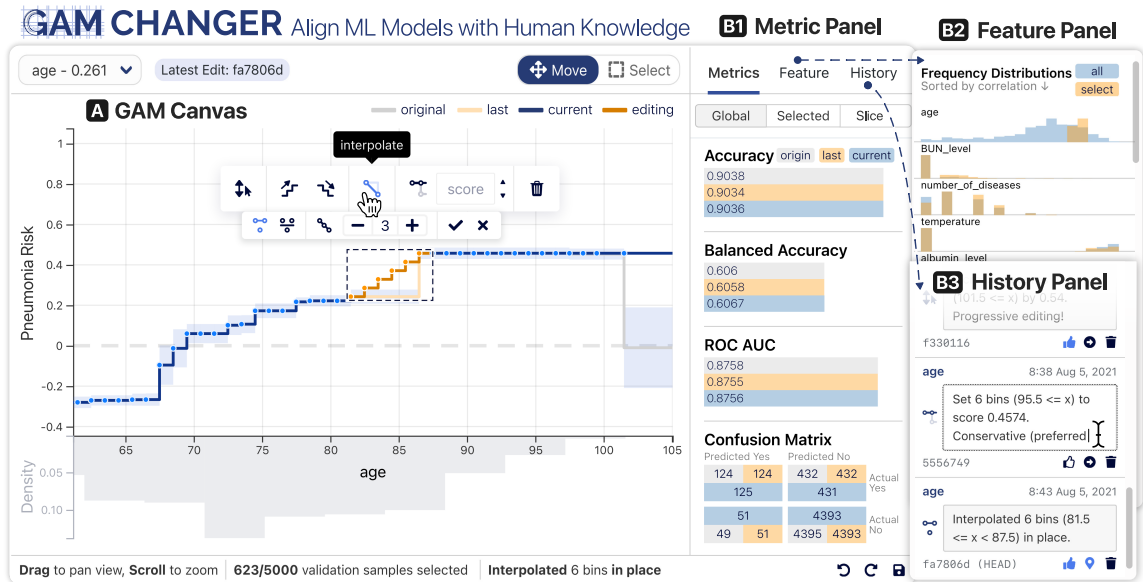


Figure 1.6: GAM CHANGER empowers AI practitioners and domain experts to easily and responsibly align AI model’s behaviors with their knowledge and values, via direct manipulation of the weights of Generalized Additive Models (GAMs). In addition to offering (A) easy-to-use editing interfaces, our tool actively promotes responsible editing by providing users with continuous feedback regarding (B1) the impacts of their edits, (B2) feature correlations, and (B3) a comprehensive edit history.

GAM COACH: *Alternating Unfavorable AI Decisions with Interactive Recourse* Chapter 7

Helping AI developers align their models with human values is a significant step, but it is not enough. Even a perfectly aligned AI model can make predictions unfavorable to certain individuals. For example, a model can rightfully deny a loan application due to the applicant’s lack of a credit history. The challenge lies in empowering individuals impacted by such unfavorable predictions to influence and potentially change the models’ decisions.

To address this challenge, we develop GAM COACH (Fig. 1.7), the first interactive tool enabling people impacted by AI-based decision-making systems to iteratively generate algorithmic recourse plans that *respect their preferences*. A recourse plan consists of minimal changes in a few features that would have led to the desired decision outcome, such as increasing the FICO score by 10 points to get loan approval. With GAM COACH’s novel adaptation of integer linear programming and simple interfaces, users can iteratively customize recourse plans. A quantitative user study with 41 participants highlights our tool is usable and useful, and users prefer personalized recourse plans over generic plans.

1.1.3 Part III: Democratize Human-Centered AI Democratize

So far we have developed novel techniques and tools that explain AI to a wide range of stakeholders (Part I) and empower individuals to exert human agency and guide AI systems (Part II). Nonetheless, these endeavors are maximally useful only if they are adopted in practice. Furthermore, within the context of an ever-expanding body of research on human-

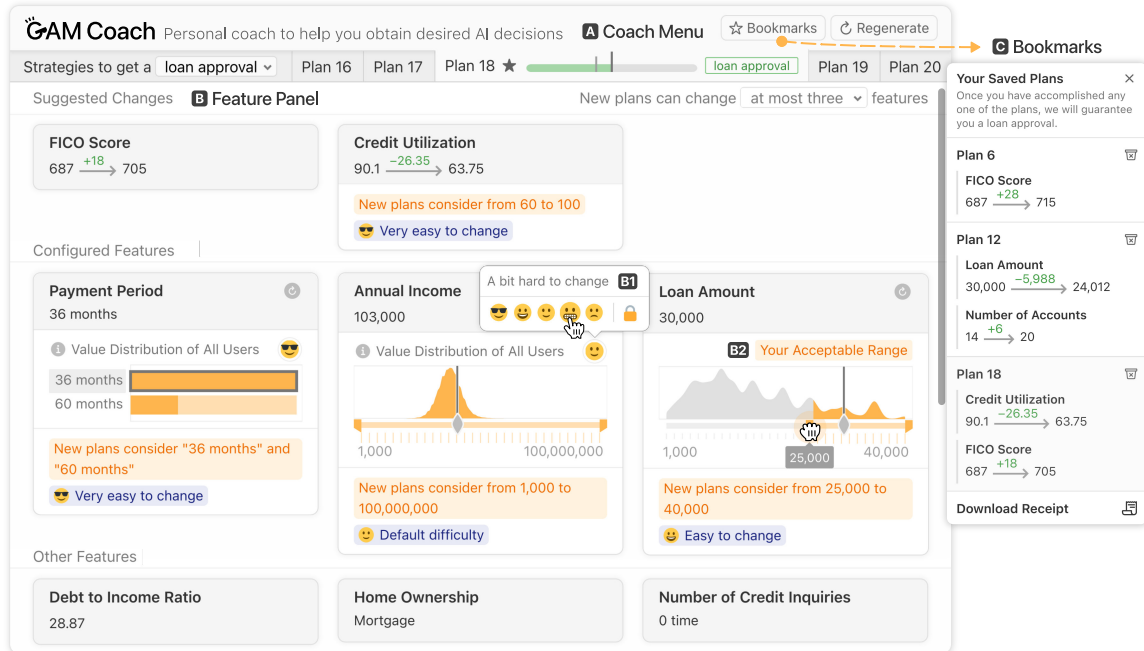


Figure 1.7: GAM COACH is the first interactive tool enabling people impacted by AI-based decision-making systems to iteratively generate algorithmic recourse plans that reflect their preferences. In this example, after a user specifies the difficulties and acceptable ranges of changing different features, the tool suggests increasing the FICO score and decreasing credit utilization to get loan approval.

centered and responsible AI, a critical question arises: How can we democratize access to human-centered AI techniques and promote its broad adoption? Our work addresses this challenge by integrating human-centered AI practices into AI practitioners’ existing workflows, such as popular prompting interfaces (FARSIGHT, Chapter 8).

FARSIGHT: Fostering Responsible AI Awareness during AI Prototyping (Chapter 8)

Modern large language models excel in various NLP tasks ranging from classification to translation. With a growing number of accessible LLMs and prompting tools such as GPT Playground and MakerSuite, we see an expanding group of “AI prototypers”. For example, many designers, writers, lawyers, and everyday users start to prototype their AI programs by writing prompts. Many of these prototypers do not have training in AI or computer science, and they face challenges in anticipating potential societal harms that might arise from their AI programs. To foster the awareness of responsible AI among AI prototypers, we introduce FARSIGHT (Fig. 1.8), an *in situ* tool that helps users envision potential use cases, stakeholders, and harms based on the prompts they are writing (Chapter 8). To lower the barrier to learning and adopting human-centered AI practices, we design FARSIGHT to integrate into practitioners’ workflows. For example, when the practitioner is crafting prompts in Google AI Studio or computational notebooks, FARSIGHT highlights related AI incident reports and AI-generated potential use cases, stakeholders, and harms.

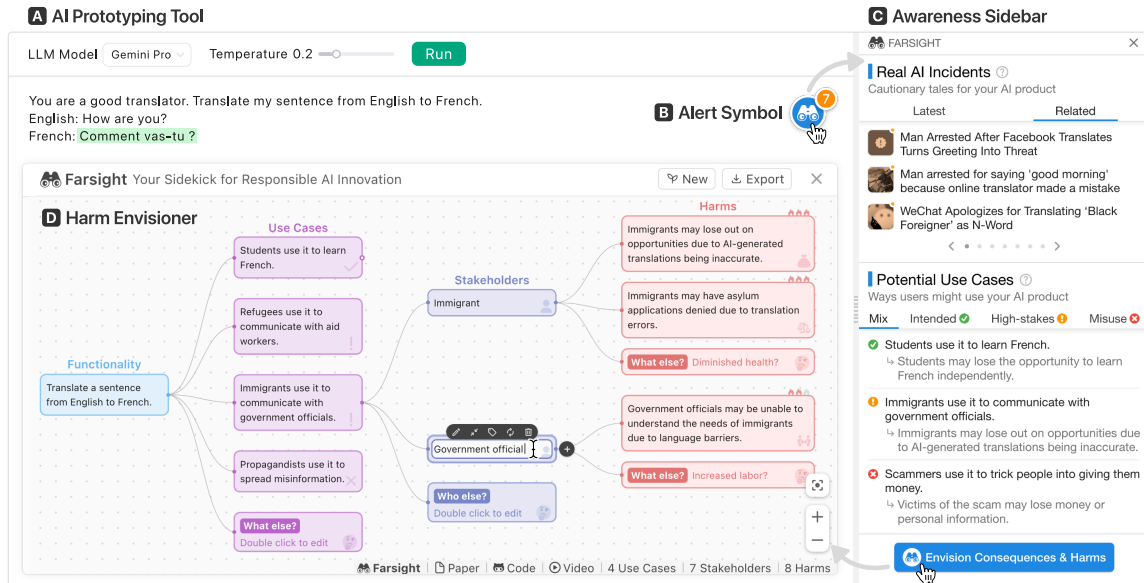


Figure 1.8: FARSIGHT is a collection of *in situ* interfaces and novel techniques that empower AI prototypers to envision potential harms that may arise from their large language model-powered AI applications during early prototyping. In this example, an AI prototyper is crafting prompts for an English-to-French translator, and FARSIGHT alerts the user with potential harms by highlighting news articles relevant to the user’s prompt and LLM-generated potential user cases and harms.

1.2 Thesis Statement

Human-centered solutions to empower novices, practitioners, and domain experts to interact with AI systems with ease, trust, and joy, through the design and development of interactive tools that aim to:

1. **Explain** AI with interactive and scalable visualizations,
2. **Guide** AI with human knowledge and values, and
3. **Democratize** human-centered AI practices within people’s workflows.

1.3 Research Contributions

My thesis makes research contributions across several major fronts, including human-computer interaction, machine learning, interactive visualization, and, importantly, their intersection to **explain** AI (Part I), **guide** AI (Part II), and **democratize** human-centered AI practices (Part III).

Transformative visual AI explanation: worldwide deployment and scalable insight

- The *viral success* of CNN EXPLAINER exemplifies the effectiveness of our proposed *dynamic explanation* in explaining complex AI models across various levels of abstraction (Chapter 3). Widely used by over 360,000 novices from more than 200

countries, CNN EXPLAINER has been integrated into deep learning courses across top universities including Carnegie Mellon, Georgia Tech, Duke University, and the University of Tokyo.

- Used by data scientists and researchers at *Apple* and *Google Deepmind*, WIZMAP is *the first system* that smoothly visualizes and summarize over 1,000,000 embedding points with novel algorithm-enabled *dynamic annotations* entirely in browsers (Chapter 4).
- We pioneer *on-device computing techniques* to accelerate scalable interactive visualization for complex AI models and large datasets. For example, CNN EXPLAINER *explains a live convolutional neural network* entirely in the user’s browser, without the need for installation or dedicated servers—broadening the public’s access to cutting-edge AI technologies (Chapter 3).

First-of-its-kind algorithms that enable actionable AI explainability

- Integrated into *Microsoft’s interpretability library*, GAM CHANGER empowers *millions of developers* to use simple clicks and drags to align the model behaviors with their knowledge and values. GAM CHANGER puts AI explanations into action by introducing *the first model-editing tool* that enables practitioners and domain experts to easily modify the weights in AI models (Chapter 6). It has been recognized with the *Best Paper* award at the NeurIPS workshop on ML for clinical practice.
- GAM COACH is the *first interactive algorithmic recourse tool* that empowers end users to specify their recourse preferences and iteratively fine-tune actionable recourse plans that can alter unfavorable AI decisions, enabled by *a novel algorithm* that adapts integer linear programming (Chapter 7).

Transformative paradigms to leapfrog responsible AI adoption

- Developed in collaboration with *Google Deepmind* researchers, FARSIGHT introduces a new paradigm for designing and developing easy-to-adopt tools that can be *directly integrated into AI practitioners’ existing workflows*. FARSIGHT helps practitioners envision the potential harms of their AI product when they write prompts within their favorite prompting interfaces (Chapter 8). This new paradigm has been recognized with the *Best Paper, Honorable Mention* award at CHI’24.
- Our research is easily accessible to AI researchers, practitioners, and the general public. For example, our tools can be used *directly in computational notebooks* (Chapter 4, Chapter 6), the most popular AI development environment. Additionally, by providing *publicly accessible web-based deployments* of CNN EXPLAINER, WIZMAP, GAM CHANGER, GAM COACH, and FARSIGHT that require no installation, we lower the barrier to learning and applying cutting-edge human-centered AI techniques.

Deployed and open-source systems and resources that accelerate AI innovation

- DIFFUSIONDB introduces *the first large-scale open-access* prompt dataset for text-to-image generative models with 14,000,000 image-prompt pairs, totaling 6.5 TB in size. With over 2,000,000 total data requests through the APIs to date, DIFFUSIONDB is instrumental in enabling researchers to study the real-world usage and impacts of generative AI models (Chapter 5). The impact of this dataset is recognized with the *Best Paper, Honorable Mention* award at ACL'23.
- This PhD thesis has introduced a suite of 6 paradigm-shifting *open-source* tools that empower and inspire researchers and practitioners to adopt our design and implementations in their human-centered AI research. Collectively, they have **received over 10,000 stars** on GitHub, the most popular platform for collaborative software development, demonstrating their significant impact and widespread adoption within the community.

1.4 Impact

My research is already making a significant impact on society and industry.

- CNN EXPLAINER has transformed AI education: its public demo has been integrated into deep learning courses (Carnegie Mellon, Georgia Tech, Duke University, University of Tokyo and more), helping **360,000 novices** from 200+ countries learn about seemingly complex ML concepts, and it has received **7,000 stars** on GitHub.
- DIFFUSIONDB has received over **2,000,000** data requests through the HuggingFace APIs. It is also among the **top 20 most-liked datasets** on HuggingFace out of 70,000 datasets. It has been integrated into official AI tutorials from Amazon AWS and Google Cloud.
- GAM CHANGER is **deployed in Microsoft** and integrated into their open-source library InterpretML. The tool is used by physicians in NYU hospitals on real-life hospital admission prediction models.
- WIZMAP is **used in Apple and Google** to explore large text datasets.
- My works have been recognized by three best-paper-type awards across top-tier HCI, NLP, and AI venues: FARSIGHT received the **Best Paper Honorable Mention Award** at CHI'24; DIFFUSIONDB received the **Best Paper Honorable Mention Award** at ACL'23; GAM CHANGER received the **Best Paper Award** at the NeurIPS Workshop on Bridging the Gap: From ML Research to Clinical Practice. CNN EXPLAINER was highlighted as a top visualization publication (**top 1%**) invited to present in SIGGRAPH.
- My research on democratizing human-centered AI has been invested in and recognized by an **Apple Scholars in AI/ML PhD fellowship** and a **J.P. Morgan AI PhD Fellowship**.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter briefly reviews related work. I focus on three related topics from which my thesis will contribute to: (1) explaining AI technologies; (2) exercising human agency in AI; and (3) democratizing human-centered AI.

2.1 Explainable AI

2.1.1 Explaining AI to Experts

With the growing complexity of AI models, there is a rapidly increasing body of research on AI explainability, where researchers aim to understand how AI models make predictions. There are two overall directions in explainability research: developing intelligible AI models and designing post-hoc explanation techniques [8].

Intelligible Models. First, researchers have made strides in developing models that are not only simple enough for humans to understand but also maintain high accuracy. These “glass-box” models include rule sets [9], sparse decision trees [10], and generalized additive models (GAMs) [11]. In particular, GAMs have emerged as a popular model class among the data science community due to their simplicity and high performance. Given an **input** $x \in \mathbb{R}^M$ with M **features** and a **target** $y \in \mathbb{R}$, a GAM with a **link function** g and **shape function** f_j for each feature $j \in \{1, 2, \dots, M\}$ can be written as:

$$g(y) = \beta_0 + f_1(x_1) + f_2(x_2) + \dots + f_M(x_M) \quad (2.1)$$

The **link function** is determined by the task. For example, in binary classification, g is **logit**. In Equation 2.1, β_0 represents the intercept constant. There are many options for the **shape functions** f_j , such as **splines** [12] and **gradient-boosted trees** [11]. Some GAMs also support pair-wise interaction terms $f_{ij}(x_i, x_j)$. Different GAM variants come with different training methods, but once trained, they all have the same form.

Post-hoc Explanations. Besides intelligible models, researchers have also proposed post-hoc techniques that can explain complex AI models. For example, LIME [13] leverages local linear approximation to compute feature importance in a model. SHAP [14] applies a game theory framework to attribute a model’s prediction to all input features. In addition to feature-based explanations, researchers also directly study what an AI model has learned from the training data. For example, researchers interpret models by visualizing and analyzing their embeddings—latent representations of input data [15]. Some researchers also try to understand the trained weights in models by studying their activation patterns [16].

2.1.2 Explaining AI to Non-experts

Education and gaining user trust are two main goals for explaining AI models to non-experts. To promote AI literacy, researchers introduce interactive tools to help AI novices learn about different AI technologies. For example, TEACHABLE MACHINE [17] explains the training process for an image classifier. TENSORFLOW PLAYGROUND [18] and GAN LAB [19] use interactive visualizations to help beginners learn about the underlying mechanisms of neural networks and generative adversarial networks, respectively. To promote trust, researchers and practitioners propose situational and text-based explanations that help end users understand how their data is being used and how recommendations are generated [20].

2.1.3 Explaining AI Models and Data with Embeddings

Researchers and domain experts are increasingly using expressive embedding representations to interpret trained models [21], develop models for new domains [22] and modalities [23], as well as analyze and synthesize new datasets [24]. People extract a data point’s embeddings by collecting its corresponding layer activations in neural networks trained for tasks like classification and generation [15]. Additionally, researchers have developed task-agnostic models, such as word2vec [25], ELMo [26], and CLIP [27] that generate transferable embeddings directly. These embeddings have been shown to outperform task-specific, state-of-the-art models in downstream tasks [27, 28].

Dimensionality Reduction. Embeddings are often high-dimensional, such as 300-dimensions for word2vec, or 768-dimensions for CLIP and BERT Base [29]. Therefore, to make these embeddings easier to visualize, researchers often apply dimensionality reduction techniques to project them into 2D or 3D space. Some popular dimensionality reduction techniques include UMAP [30], t-SNE [31], and PCA [32]. Each of these techniques has its own strengths and weaknesses in terms of how well it preserves the embeddings’ global structure, its stochasticity, interpretability, and scalability. Despite these differences, all dimensionality reduction techniques produce data in the same structure.

Interactive Embedding Visualization. Researchers have introduced interactive visualization tools to help users explore embeddings [e.g., 33, 34, 35]. For example, Embedding Projector [36] allows users to zoom, rotate, and pan 2D or 3D projected embeddings to explore and inspect data point features. Similarly, Deepscatter [37] and regl-scatterplot [38] empowers users to explore billion-scale 2D embeddings in their browsers. Latent Space Cartography [39] helps users find and refine meaningful semantic dimensions within the embedding space. In addition, researchers have designed visualizations to aid users in comparing embeddings, such as embComp [40] visualizing local and global similarities between two embeddings, Emblaze [41] tracing the changes in the position of data points across two embeddings, and Embedding Comparator [42] highlighting the neighborhoods around points that change the most across embeddings.

2.1.4 Explaining AI Usage

With the increasing popularity of large generative models, researchers have proposed prompt datasets and studied how people write prompts to generate text and images. For example, Researchers have been studying prompt engineering for text-to-text generation [e.g., 43, 44, 45]. To facilitate this line of research, researchers develop PromptSource [46], a dataset of 2k text prompts along with a framework to create and share prompts. There is also a growing interest in text-to-image prompt engineering research from NLP, Computer Vision, and HCI communities [e.g., 47, 48]. For example, Oppenlaender [49] identifies six types of prompt modifiers through an ethnographic study, and Liu and Chilton [50] proposes design guidelines for text-to-image prompt engineering by experimenting with 1,296 prompts. Lexica [51] allows users to search over 5 million Stable Diffusion images with their prompts, but it does not release its internal database.

2.2 Human Guidance in AI

More recently, researchers put AI explainability into action and help AI stakeholders exercise their human agency, such as enabling AI practitioners to edit model weights and provide recourse to people impacted by AI-powered decision systems.

2.2.1 Model Editing

Prior research has highlighted that being able to modify AI models can lead to greater trust and better human-AI team performance [52]. To enable practitioners and domain experts to guide AI model behaviors, researchers have proposed model editing techniques that modify model behaviors by changing the learned weights. For example, practitioners can modify important neurons in a neural network to change semantic concepts in generated images [53], control text translation styles [54], and induce basic concepts in text generation [55]. More recently, researchers have also studied model editing in large language models, such as editing model weights to update the model’s knowledge [56] and unlearn certain knowledge [57], and amplifying stored facts [58].

2.2.2 Algorithmic Recourse

Algorithmic recourse aims to design techniques that provide people impacted by AI systems with actionable feedback about how to alter AI predictions. Take AI-assisted loan application approval as an example, to help a rejected applicant get approval, an algorithmic recourse plan can be “increasing the annual income by \$5k.” Popularized by Wachter *et al.* [59], researchers typically generate recourse plans by creating *counterfactual examples*. These counterfactual examples suggest minimal changes in a few features that would have led to a different AI prediction outcome. There are many different methods to generate

counterfactual examples, such as casting it as an optimization problem [60], searching through similar samples [61], and using generative models [62].

2.3 Democratizing Human-centered AI

2.3.1 Existing Responsible AI Tools and Practices

Despite the recent advancements in human-centered AI and responsible AI, incorporating these practices into AI product development remains a challenge, in part due to practitioners’ insufficient training [63] and organizational culture [64]. To address these challenges, researchers have proposed several approaches to democratize human-centered AI, such as integrating them into AI education [65], providing engaging playbooks or design activities [66], and cultivating ethical norms in AI research and development [67]. More recently, researchers have also designed and developed easy-to-use interactive tools to operationalize human-centered AI practices. These tools cover a wide range of dimensions in human-centered AI, such as helping users assess and improve fairness in AI models [68], explaining AI predictions [7], testing and error analysis [69], as well as documenting data and model development [70]. These tools enable practitioners and domain experts with less experience in human-centered AI to prioritize humans in AI development.

2.3.2 Anticipating Technology’s Negative Impacts

Various design methods and approaches have been developed to support ideation about potential downstream impacts of technology, including anticipatory tech ethics [71, 72], speculative design [73, 74, 75], and value-sensitive design [76, 77, 78] among others. To support designers with this, prior research has developed design toolkits [e.g., 79] and resources, such as Envisioning Cards [80], Value Cards [81], Timelines [82], and the Black Mirror Writers’ Room [83], among others [e.g., 84, 85]. Such resources are intended to be used by designers of technology early in the design process, but they may not fit neatly into existing product design and development processes, particularly for AI-powered application design paradigms, where large pre-trained models are used for many downstream tasks [86].

In addition to technology designers, computing researchers have called for the computer science field to consider the negative impacts of their work in addition to the positive impacts [87]. In AI research, conferences such as NeurIPS have begun requiring that researchers articulate potential negative broader impacts of their work in statements at the ends of their papers [88] to avoid the “failures of imagination” [89] that may lead to downstream harms. Prior work analyzed these broader impacts statements, finding convergence around a set of topics such as risks to privacy and bias, but often lacking concrete specifics or strategies for mitigation [90, 91, 67, 92]. However, prior work suggests that many CS researchers may not have the training, resources, or inclination to engage in this type of anticipatory

work [93, 94], suggesting that new tools, training, and processes, are needed to support researchers and developers in engaging in anticipatory work in ways that are integrated into their research practices. More recently, researchers have proposed a framework that uses LLMs to anticipate harms for classifiers by generating stakeholders and vignettes for a given scenario [95], evaluating this framework through interviews with responsible AI researchers.

2.3.3 Identifying and Mitigating LLM Harms

More recently, there has been a growing body of research that specifically focuses on identifying and mitigating the harms of LLMs. Researchers have introduced harm taxonomies specifically for LLMs, which identify known risks (i.e., informed by observed instances of harm) [96, 97, 98] and emerging risks of LLMs (anticipated risks based on foreseeable capabilities of LLMs) [99, 100]. Since LLMs can be used for a wide range of tasks associated with many different categories of harms, researchers have presented frameworks and evaluation methods to assess a particular type of LLM harm, including misinformation [101, 102], representation and toxicity [103, 104], human autonomy [105, 106], malicious use [107, 108], and data privacy [109, 110]. The popular methods to identify these harms include benchmarking [111, 112], user research [113, 114], and adversarial testing [115, 116]. Based on existing benchmarks and harm taxonomies of LLM risks, Weidinger *et al.* [117] introduce a sociotechnical evaluation framework that identifies three AI actors with LLM safety responsibilities: AI model developers, AI application developers, and third-party stakeholders. The mitigation strategies for these harms depend on the use cases and context. Popular strategies include algorithmic and sociotechnical approaches [118], such as improving the training data to mitigate social stereotypes and biases [119]; fine-tuning LLM models on curated datasets [103]; filtering LLM outputs [120, 121]; employing special decoding techniques [122, 123], adding instructions in prompts [124], monitoring the use of LLMs [118]; as well as inclusive product design and development from the beginning [125, 126, 127, 128].

2.3.4 *In Situ* Interfaces

Although *in situ* responsible AI tools are relatively nascent, there is a large body of research in designing in-context warning tools and interfaces. For example, security and HCI researchers study how to best present warnings to raise people’s online security awareness [e.g., 129, 130, 131] and protect people from malware and phishing attacks [e.g., 132, 133, 134]. The key challenges when designing effective warning interfaces include the presentation of comprehensible messages and supporting evidence [135, 136], engaging users [137, 138], and preventing alert fatigue and habituation [139, 140]. To address these challenges, researchers recommend designing simple interfaces [141, 142], considering the trade-off between blocking and non-blocking warnings [138], varying interfaces [139], and requiring user input [143]. Using in-context warnings to improve users’ safety awareness and encour-

age users to take protection measures can be considered a form of “digital nudging” [144, 145]. More recently, researchers have also adapted in-context security warnings to nudge social media users to recognize and avoid online disinformation [146, 147] and reflect before posting potentially harmful content [148, 149, 150]. Beyond platform-initiated integration of warnings, end-users also voluntarily seek in-context alert interfaces for productivity improvement. For example, writers use grammar checker tools like Grammarly [151], which offer in-context warnings and scores to improve their writing. Similarly, software developers use accessibility developer tools [152, 153] to detect potential accessibility issues during the development process. However, there has been little work in designing and evaluating *in situ* warnings for developing AI applications, particularly for responsible AI.

Part I

EXPLAIN AI TO EVERYONE

Overview

AI models have grown increasingly complex and ubiquitous in our daily lives. There are dire needs from different stakeholders to **understand AI models**. For example, AI novices desire to learn about developing and applying AI tools to improve their productivity and lives. AI experts aim to interpret trained AI models to debug them and build trust with end users. Policymakers seek to understand the usage and impacts of AI models to develop better regulations. Part I of this thesis focuses on a fundamental question: how can we explain AI to people with diverse AI backgrounds?

To answer this question, we start by investigating interactive visualizations, as they are powerful techniques to help users explore and understand complex systems and concepts. We first describe **CNN EXPLAINER (Chapter 3)**, a novel interactive visualization system that helps AI novices learn about the inner workings of convolutional neural networks (CNNs), the most foundational deep learning models. This chapter is adapted from work that was published and appeared at IEEE VIS 2020 [154].

Chapter 3

CNN EXPLAINER: Learning Convolutional Neural Networks with Interactive Visualization. Zijie J. Wang, Robert Turko, Omar Shaikh, Haekyu Park, Nilaksh Das, Fred Hohman, Minsuk Kahng, and Duen Horng Chau. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2020. [PDF](#)

Through an observational user study, we find that *interactivity* and *progressive disclosure* are powerful techniques to help users learn about AI models. Based on the observation, we extend these design techniques to help AI practitioners interpret their AI models with **WIZMAP (Chapter 4)**, a scalable interactive visualization tool to explain AI embeddings. This chapter is adapted from work published and appeared at ACL 2023 [155].


Chapter 4

WIZMAP: Scalable interactive visualization for exploring large machine learning embeddings. Zijie J. Wang, Fred Hohman, and Duen Horng Chau. *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 3: System demonstrations)*, 2023. [PDF](#)

The scalability of WIZMAP reveals previously unknown insights from million-scale datasets. With the recent surge in popularity of large generative AI models, there is a dearth of large-scale datasets documenting how users use these models. To help policymakers comprehend the usage and impacts of AI models *at scale*, we introduce **DIFFUSIONDB (Chapter 5)**, the

first dataset of 14 million prompt-image pairs generated by real users of large text-to-image generative models. This chapter is adapted from work published at ACL 2023 [156].

Chapter 5

DIFFUSIONDB: A large-scale prompt gallery dataset for text-to-image generative models. Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, 2023. 

CHAPTER 3

CNN EXPLAINER: EXPLAIN CONVOLUTIONAL NEURAL NETWORKS TO AI NOVICES

Powered by deep learning models, AI has transformed our everyday technologies, and it has attracted immense interest from students and practitioners who wish to learn and apply this technology. However, beginners find it challenging to take the first step in understanding deep learning concepts, such as convolutional neural networks (CNNs), the foundational deep learning model architecture. A key challenge in learning about CNNs is the intricate interplay between low-level mathematical operations and high-level integration of such operations within the neural network. We present CNN EXPLAINER, an interactive visualization tool designed for non-experts to learn about both CNN’s high-level model structure and low-level mathematical operations. We conducted a user study with 16 students to evaluate the usefulness and usability of CNN EXPLAINER. The study highlights that CNN EXPLAINER is easy to use, enjoyable, and helps participants learn about CNNs. Through a qualitative analysis, we distill design lessons for future visualization tools designed to explain AI concepts to novices.

3.1 Introduction

Deep learning enables many of our everyday technologies. Its continued success and potential application in diverse domains has attracted immense interest from students and practitioners who wish to learn and apply this technology. However, many beginners find it challenging to take the first step in studying and understanding deep learning concepts. For example, convolutional neural networks (CNNs), a foundational deep learning model

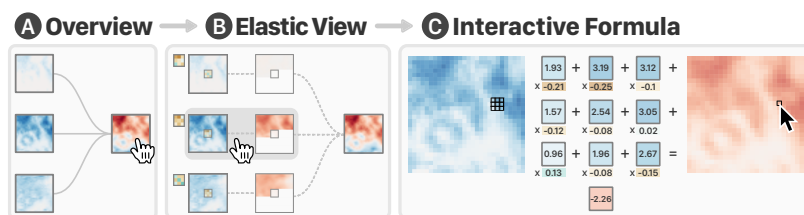


Figure 3.1: In CNN EXPLAINER, tightly integrated views with different levels of abstractions work together to help users more easily learn about the intricate interplay between a CNN’s high-level structure and low-level mathematical operations. (A) the *Overview* summarizes connections of all neurons; (B) the *Elastic View* animates the intermediate convolutional computation of the user-selected neuron in the *Overview*; and (C) *Interactive Formula* interactively demonstrates the detailed calculation on the selected input in the *Elastic View*.

architecture, is often one of the first and most widely used models that students learn. CNNs are often used in image classification, achieving state-of-the-art performance [157]. However, through interviews with deep learning instructors and a survey of past students, we found that even for this “introductory” model, it can be challenging for beginners to understand how inputs (e.g., image data) are transformed into class predictions. This steep learning curve stems from CNN’s complexity, which typically leverages many computational layers to reach a final decision. Within a CNN, there are many types of network layers (e.g., fully-connected, convolution, and activation), each with a different structure and underlying mathematical operations. Thus, a student needs to develop a mental model of not only how each layer operates, but also how to choose different layers that work together to transform data. Therefore, a key challenge in learning about CNNs is the intricate interplay between *low-level mathematical operations* and *high-level integration* of such operations within the network.

Key challenges in designing learning tools for CNNs. There is a growing body of research that uses interactive visualization to explain the complex mechanisms of modern machine learning algorithms, such as TensorFlow Playground [18] and GAN Lab [158], which help students learn about dense neural networks and generative adversarial networks (GANs) respectively. Regarding CNNs, some existing visualization tools focus on demonstrating the high-level model structure and connections between layers (e.g., Harley’s Node-Link Visualization [159]), while others focus on explaining the low-level mathematical operations (e.g., Karpathy’s interactive CNN demo [160]). There is no visual learning tool that explains and connects CNN concepts from both levels of abstraction. This interplay between global model structure and local layer operations has been identified as one of the main obstacles to learning deep learning models, as discussed in [18] and corroborated from our interviews with instructors and student survey. CNN EXPLAINER aims to bridge this critical gap.

Contributions. In this work, we contribute:

- **CNN EXPLAINER, an interactive visualization tool designed for non-experts** to learn about both CNN’s high-level model structure and low-level mathematical operations, addressing learners’ key challenge in connecting unfamiliar layer mechanisms with complex model structures. Our tool advances over prior work [159, 160], overcoming unique design challenges identified from a literature review, instructor interviews and a survey with past students (§ 3.2).
- **Novel interactive system design** of CNN EXPLAINER (Fig. 3.2), which adapts familiar techniques such as *overview + detail* and *animation* to simultaneously summarize intricate model structure, while providing context for users to inspect detailed mathematical operations. CNN EXPLAINER’s visualization techniques work together through fluid transitions between different abstraction levels (Fig. 3.1), helping users gain a more comprehensive understanding of complex concepts within CNNs (§ 3.4).
- **Design lessons distilled from user studies** on an interactive visualization tool for

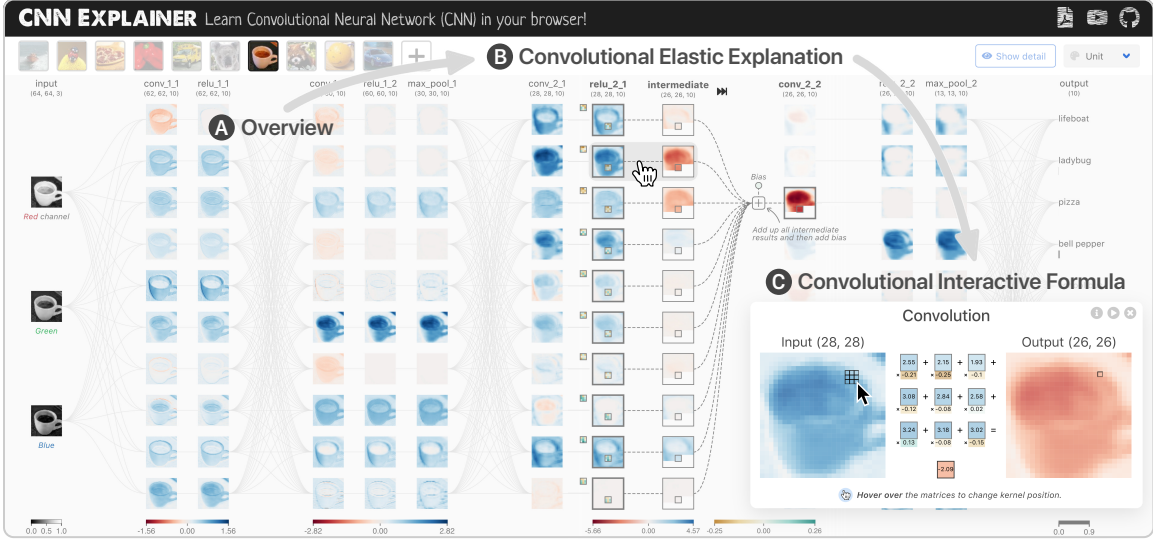


Figure 3.2: CNN EXPLAINER empowers AI novices to easily learn how CNNs transform an input image into a category prediction. **(A)** The *Overview* visualizes a CNN architecture where each neuron is encoded as a square with a heatmap representing its output. **(B)** Clicking a neuron reveals how its activations are computed from the previous layer through animations of sliding kernels. **(C)** *Convolutional Interactive Formula View* explains underlying mathematics of convolutions.

machine learning education. While visual and interactive approaches have been gaining popularity in explaining machine learning concepts to non-experts, little work has been done to evaluate such tools [161, 162]. We interviewed four instructors who have taught CNNs and conducted a survey with 19 students who have previously learned about CNNs to identify the needs and challenges for a deep learning educational tool (§ 3.2). In addition, we conducted an observational study with 16 students to evaluate the usability of CNN EXPLAINER, and investigated how our tool could help students better understand CNN concepts (§ 3.6). Based on these studies, we discuss the advantages and limitations of interactive visual educational tools for machine learning.

- **An open-source, web-based implementation** that broadens the public’s education access to modern deep learning techniques without the need for advanced computational resources. Deploying deep learning models conventionally requires significant computing resources, e.g., servers with powerful hardware. In addition, even with a dedicated backend server, it is challenging to support a large number of concurrent users. Instead, CNN EXPLAINER is developed using modern web technologies, where all results are directly and efficiently computed in users’ web browsers (§ 3.4.7). Therefore, anyone can access CNN EXPLAINER using their web browser without the need for installation.

Broadening impact of visualization for AI. In recent years, many visualization systems have been developed for deep learning, but very few are designed for non-experts [159, 158, 163, 18], as surveyed in [164]. CNN EXPLAINER joins visualization research that introduces beginners to modern machine learning concepts. Applying visualization techniques to

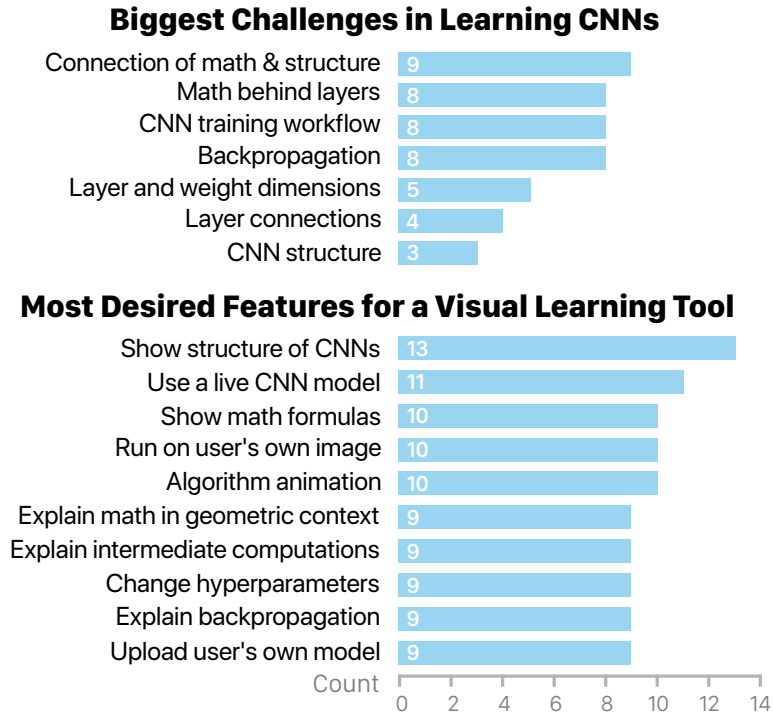


Figure 3.3: Survey results from 19 past CNN learners.

explain the inner workings of complex models has great potential. We hope our work will inspire further research and development of visual learning tools that help democratize and lower the barrier to understanding and applying artificial intelligent technologies.

3.2 Formative Research & Design Challenges

Our goal is to build an interactive visual learning tool to help students gain an understanding of key CNN concepts to design their own models. To identify the learning challenges faced by the students, we conducted interviews with deep learning instructors and surveyed past students.

Instructor interviews. To inform our tool’s design, we recruited 4 instructors (2 female, 2 male) who have taught CNNs in a large university. We refer to them as T1-T4 throughout our discussion. One instructor teaches computer vision, and the others teach deep learning. We interviewed them one-on-one in a conference room (3/4) and via a video-conferencing software (1/4); each interview lasted around 30 minutes. Through these semi-structured interviews, we learned that (1) instructors currently rely on simple illustrations with toy examples to explain CNN concepts, and an interactive tool like TensorFlow Playground with real image inputs would be highly appreciated; and (2) key challenges exist for instructors teaching and students learning about CNNs, which informed us to design a student survey.

Student survey. After the interviews, we recruited students from a large university who have previously studied CNNs to fill out an online survey. We received 43 responses, and 19

of them (4 female, 15 male) met the criteria. Among these 19 participants, 10 were Ph.D. students, 3 were M.S. students, 5 were undergraduates, and 1 was a faculty member. We asked participants what were “the biggest challenges in studying CNNs” and “the most helpful features if there was a visualization tool for explaining CNNs to beginners”. We provided pre-selected options based on the prior instructor interviews, but participants could write down their own responses if it was not included in the options. The aggregated results of this survey are shown in Fig. 3.3.

Together with a literature review, we synthesized our findings from these two studies into the following five design challenges (C1-C5).

C1. Intricate model structure. CNN models consist of many layers, each having a different structure and underlying mathematical functions [157]. Fewer past students listed CNN structure as their biggest challenge, but most of them believe a visual learning tool should explain the structure (Fig. 3.3), as the complex construction of CNNs can be overwhelming, especially for beginners who just started learning. T2 said “*It can be very hard for them [students with less knowledge of neural networks] to understand the structure of CNNs, you know, the connections between layers.*”

C2. Complex layer operations. Different layers serve different purposes in CNNs [165]. For example, convolutional layers exploit the spatially local correlations in inputs—each convolutional neuron connects to only a small region of its input; whereas max pooling layers introduce regularization to prevent overfitting. T1 said, “*The most challenging part is learning the math behind it [CNN model].*” Many students also reported that CNN layer computations are the most challenging learning objective (Fig. 3.3). To make CNNs perform better than other models in tasks like image classification, these models have complex and unique mathematical operations that many beginners may not have seen elsewhere.

C3. Connection between model structure and layer operation. Based on instructor interviews and the survey results from past students (Fig. 3.3), one of the cruxes to understand CNNs is understanding the interplay between low-level mathematical operations (C2) and the high-level model structure (C1). Smilkov et al., creators of the popular dense neural network learning tool Tensorflow Playground [18], also found this challenge key to learning about deep learning models: “*It’s not trivial to translate the equations defining a deep network into a mental model of the underlying geometric transformations [change of feature representations].*” In other words, in addition to comprehending the mathematical formulas behind different layers, students are also required to understand how each operation works within the complex, layered model structure.

C4. Effective algorithm visualization (AV). The success of applying visualization to explain machine learning algorithms to beginners [166, 18, 158] suggests that an AV tool is a promising approach to help people more easily learn about CNNs. However, AV

tools need to be carefully designed to be effective in helping learners gain an understanding of algorithms [167]. In particular, AV systems need to clearly explain the mapping between the algorithm and its visual encoding [168], and actively engage learners [169].

C5. Challenge in deploying interactive learning tools. Most neural networks are written in deep learning frameworks, such as TensorFlow [170] and PyTorch [171]. Although these libraries have made it much easier to create AI models, they require users to understand key concepts of deep learning in the first place [172]. Can we make understanding CNNs more accessible without installation and coding, so that everyone has the opportunity to learn and interact with deep learning models?

The above design challenges cover most of the desired features (Fig. 3.3). We assessed the feasibility to also support explaining backpropagation in the same tool, and we concluded that its effective explanation will necessitate designs that are hard to be unified (e.g., backpropagation Algorithm [173]). Indeed, T1 commented that “*Deriving backpropagation is applying a series chain rules [...] It doesn’t really make sense to visualize the gradients [in our tool].*” Supporting the training process would require client-side in-browser computation on many data examples, which incur both high amount of data download and slow convergence ([160, 158]). Therefore, as the first prototype, we decided for CNN EXPLAINER to focus on explaining inference after a model has been trained. We plan to support the explanation for backpropagation and training process as future work (§ 3.7).

3.3 Design Goals

Based on the identified design challenges (§ 3.2), we distill the following key design goals (**G1–G5**) for CNN EXPLAINER, an interactive visualization tool to help students more easily learn about CNNs.

G1. Visual summary of CNN models and data flow. Based on the survey results, showing the structure of CNNs is the most desired feature for a visual learning tool (Fig. 3.3). Therefore, to give users an overview of the structure of CNNs, we aim to create a visual summary of a CNN model by visualizing all layer outputs and connections in one view. This could help users to visually track how input image data are transformed to final class predictions through a series of layer operations (**C1**). (§ 3.4.1)

G2. Interactive interface for mathematical formulas. Since CNNs employ various complex mathematical functions to achieve high classification performance, it is important for users to understand each mathematical operation in detail (**C2**). In response, we would like to design an interactive interface for each mathematical formula, enabling users to examine and better understand the inner-workings of layers. (§ 3.4.3)

G3. Fluid transition between different levels of abstraction. To help users connect low-level layer mathematical mechanisms to high-level model structure (**C3**), we would like to design a focus + context display of different views, and provide smooth transitions

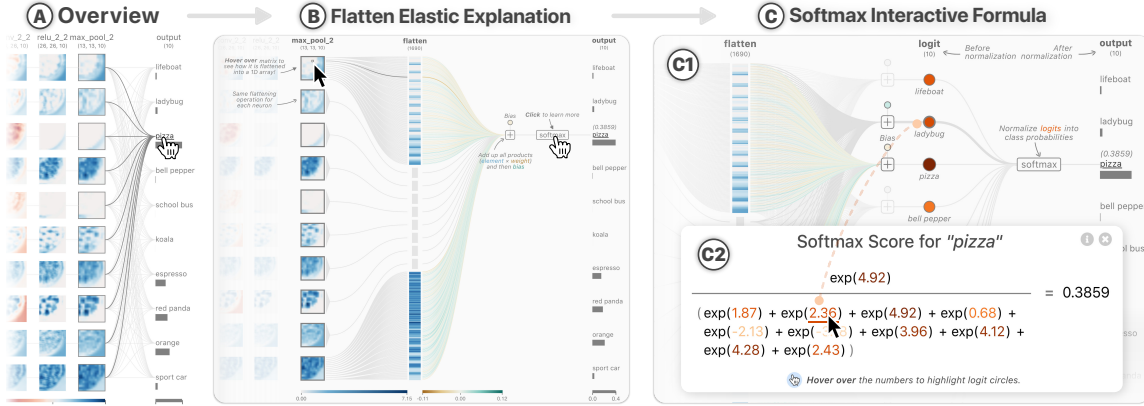


Figure 3.4: CNN EXPLAINER helps users learn about the connection between the **output** layer and its previous layer via three tightly integrated views. Users can smoothly transition between these views to gain a more holistic understanding of the output layer’s `lifeboat` prediction computation. **(A)** The *Overview* summarizes neurons and their connections. **(B)** The *Flatten Elastic Explanation View* visualizes the often-overlooked flatten layer, helping users more easily understand how a high-dimensional `max_pool_2` layer is connected to the 1-dimensional output layer. **(C)** The *Softmax Interactive Formula View* further explains how the softmax function that precedes the output layer normalizes the penultimate computation results (i.e., logits) into class probabilities by linking the **(C1)** numbers from the formula to **(C2)** their visual representations within the model structure.

between them. By easily navigating through different levels of CNN model abstraction, users can get a holistic picture of how CNN works. (§ 3.4.4)

- G4. Clear communication and engagement.** Our goal is to design an interactive system that is easy to understand and engaging to use so that it can help people to more easily learn about CNNs **(C4)**. We aim to accompany our visualizations with explanations to help users to interpret the graphical representation of the CNN model (§ 3.4.5), and we wish to actively engage learners through visualization customizations. (§ 3.4.6)
- G5. Web-based implementation.** To develop an interactive visual learning tool that is accessible for users without installation and coding **(C5)**, we would like to use modern web browsers as the platform to explain the inner-workings of a CNN model, where users can access directly on their laptops or tablets. We also open-source our code to support future research and development of deep learning educational tools. (§ 3.4.7)

3.4 Visualization Interface of CNN EXPLAINER

CNN EXPLAINER’s interface is built on our prior prototype [175]. We visualize the forward propagation, i.e., transforming an input image into a class prediction, of a trained model (Fig. 3.5). Users can explore a CNN at different levels of abstraction through the tightly integrated *Overview* (§ 3.4.1), *Elastic Explanation View* (§ 3.4.2), and the *Interactive Formula View* (§ 3.4.3). Our tool allows users to smoothly transition between these views (§ 3.4.4), provides text annotations and a tutorial article to help users interpret the visual-

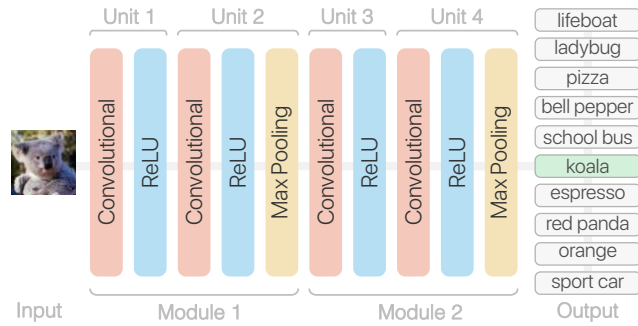


Figure 3.5: Illustration of *Tiny VGG* model used in CNN EXPLAINER: this model uses the same, but fewer, convolutional layers as the VGGNet model [174]. We trained it to classify 10 classes of images.



Figure 3.6: Diverging color scales in CNN EXPLAINER.

izations (§ 3.4.5), and engages them to test hypotheses through visualization customizations (§ 3.4.6). The system is targeted towards beginners and describes all mathematical operations necessary for a CNN to classify an image.

Color scales are used throughout the visualization to show the impact of weight, bias, and activation map values. Consistently in the interface, a **red** to **blue** color scale is used to visualize neuron activation maps as heatmaps, and a **yellow** to **green** color scale represents weights and biases (Fig. 3.6). A persistent color scale legend is present across all views, so the user always has context for the displayed colors. We chose these distinct, diverging color scales with white representing zero, so that a user can easily differentiate positive and negative values. We group layers in the *Tiny VGG* model, our CNN architecture, into four units and two modules (Fig. 3.5). Each unit starts with one convolutional layer. Both modules are identical and contain the same sequence of operations and hyperparameters. To analyze neuron activations throughout the network with varying contexts, users can alter the range of the heatmap color scale (§ 3.4.6).

3.4.1 Overview

The *Overview* (Fig. 3.2A, Fig. 3.4A) is the opening view of CNN EXPLAINER. This view represents the high-level structure of a CNN: neurons grouped into layers with distinct, sequential operations. It shows neuron activation maps for all layers represented as heatmaps with a diverging **red** to **blue** color scale. Neurons in consecutive layers are connected with edges, which connect each neuron to its inputs; to see these edges, users simply can hover over any activation map. In the model, neurons in convolutional layers and the **output** layer are fully connected to the previous layer, while all other neurons are only connected to one neuron in the previous layer.

3.4.2 Elastic Explanation View

The *Elastic Explanation Views* visualize the computations that leads to an intermediate result without overwhelming users with low-level mathematical operations. CNN EXPLAINER enters two elastic views after a user clicks a convolutional or an output neuron from the *Overview*. After the transition, far-away heatmaps and edges fade out to help users focus on the selected layers while providing CNN structural context in the background (Fig. 3.2A).

Explaining the Convolutional Layer (Fig. 3.2B). The *Convolutional Elastic Explanation View* applies a convolution on each input node of the selected neuron, visualized by a kernel sliding across the input neurons, which yields an intermediate result for each input neuron. This sliding kernel forms the output heatmap during the animation, which imitates the internal process during a convolution operation. While the sliding kernel animation is in progress, the edges in this view are represented as flowing-dashed lines; upon the animations completion, the edges transition to solid lines.

Explaining the Flatten Layer (Fig. 3.4B). The *Flatten Elastic Explanation View* visualizes the operation of transforming an n-dimensional tensor into a 1-dimensional tensor by traversing pixels in row-major order. This flattening operation is often necessary in a CNN prior to classification so that the fully-connected **output** layer can make classification decisions. The view represents each neuron in the **flatten** layer as a short line whose color is the same as its source pixel in the previous layer. Then, edges connect these neurons with their source components and intermediate results. These edges are colored by the model’s weight value. Users can hover over any connection to highlight the associated edges as well as the **flatten** layer’s neuron and the pixel value from the previous layer.

3.4.3 Interactive Formula View

The *Interactive Formula View* consists of four variations for convolutional, ReLU activation, pooling, and softmax layers. After users have built up a mental model of the CNN model structure from the previous *Overview* and *Elastic Explanation Views*, these four views demonstrate the detailed mathematics occurring in each layer.

Explaining Convolution, ReLU Activation, and Pooling (Fig. 3.7A, B, C). Each view animates the window-sliding operation on the input and output matrices over an interval, so that the user can understand how each element in the input is connected to the output, and vice versa. The user can interact with these matrices by hovering over the heatmaps to control the position of the sliding window. For example, in the *Convolutional Interactive Formula View* (§ 3.4.3A), as the user controls the window (kernel) position in either the input or the output matrix, this view visualizes the dot-product formula with input numbers and kernel weights directly extracted from the current kernel. This synchronization between the input, the output and the mathematical function enables the user to better understand how the kernel convolves a matrix in convolutional layers.

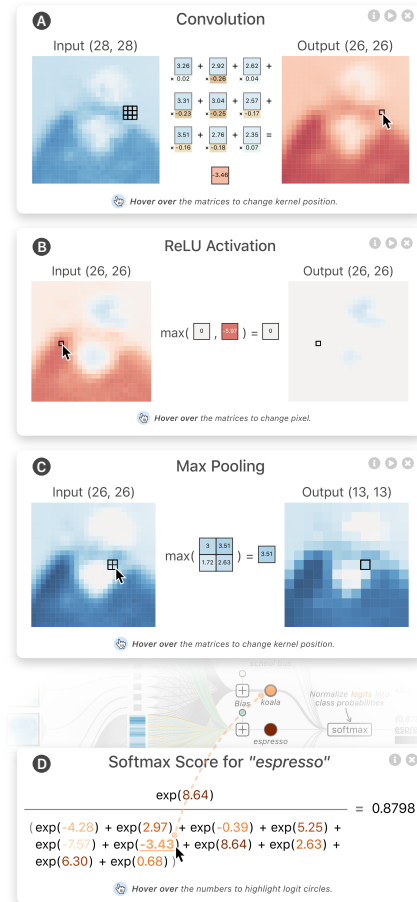




Figure 3.7: The *Interactive Formula Views* explain the underlying mathematical operations of a CNN. (A) shows the element-wise dot-product occurring in a convolutional neuron, (B) visualizes the activation function ReLU, and (C) illustrates how max pooling works. Users can hover over heatmaps to display an operation’s input-to-output mapping. (D) interactively explains the softmax function, helping users connect numbers from the formula to their visual representations. Users can click the info button  to scroll to the corresponding section in the tutorial article, and the play button  to start the window sliding animation in (A)-(C).

Explaining the Softmax Activation (Fig. 3.7D). This view outlines the operations necessary to calculate the classification score. It is accessible from the *Flatten Elastic Explanation View* to explain how the results (logits) from the previous view lead to the final classification. The view consists of logit values encoded as circles and colored with a **light orange** to **dark orange** color scale, which provides users with a visual cue of the importance of every class. This view also includes a corresponding equation, which explains how the classification score is computed. When users enter this view, pairs of each logit circle and its corresponding value in the equation appear sequentially with animations. As a user hovers over a logit circle, its value will be highlighted in the equation along with the logit circle itself, so the user can understand how each logit contributes to the softmax

function. Hovering over numbers in the equation will also highlight the appropriate logit circles. Interacting with logit circles and the mathematical equation in combination allows a user to discern the impact that every logit has on the classification score in the **output** layer.

3.4.4 Transitions Between Views

The *Overview* is the starting state of CNN EXPLAINER and shows the model architecture. From this high-level view, the user can begin inspecting layers, connectivity, classifications, and tracing activations of neurons through the model. When a user is interested in more detail, they can click on neuron activation maps in the visualization. Neurons in a layer that have simple one-to-one connections to a neuron in the previous layer do not require an auxiliary *Elastic Explanation View*, so upon clicking one of these neurons, a user will be able to enter the *Interactive Formula View* to understand the low-level operation that a tensor undergoes at that layer. If a neuron has more complex connectivity, then the user will enter an *Elastic Explanation View* first. In this view, CNN EXPLAINER uses visualizations and annotations before displaying mathematics. Through further interaction, a user can hover and click on parts of the *Elastic Explanation View* to uncover the mathematical operations as well as examine the values of weights and biases. The low-level *Interactive Formula Views* are only shown after transitioning from the previous two views, so that users can learn about the underlying mathematical operations after having a mental model of the complex and layered CNN model structure.


3.4.5 Visualizations with Explanations

CNN EXPLAINER is accompanied by an interactive tutorial article beneath the interface that explains CNN layer functions, hyperparameters, and outlines CNN EXPLAINER's interactive features. Learners can read freely, or jump to specific sections by clicking layer names or the info buttons (Fig. 3.7) from the main visualization. The article provides beginner users detailed information regarding CNNs complementary to the visualization.

Additionally, text annotations are placed throughout the visualization, which further guide users and explain concepts that are not easily discernible from the visualization alone. These annotations help users map the underlying algorithm to its visual encoding.

3.4.6 Customizable Visualizations

The *Control Panel* located at the top of the visualization (Fig. 3.2) allows users to alter the CNN input image and edit overall representations of the network. The *Hyperparameter Widget* (Fig. 3.8) enables the user to experiment with different convolution hyperparameters.

Change input image. Users can choose between (1) preloaded input  images for each output class, or (2) upload their own custom image. Preloaded images allow

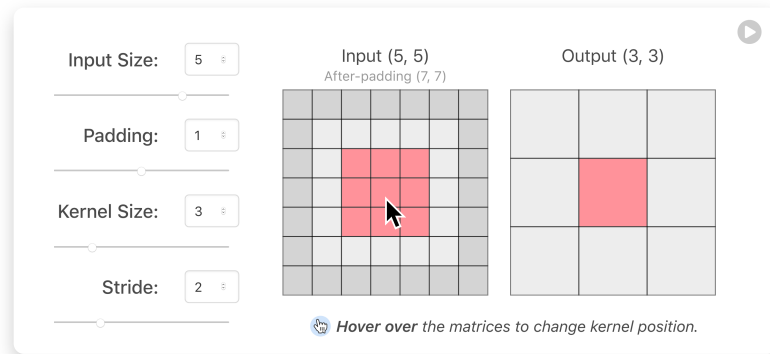


Figure 3.8: The *Hyperparameter Widget*, a component of the accompanying interactive article, allows users to adjust hyperparameters and observe in real time how the kernel’s sliding pattern changes.

a user to easily access data from the classes the model was trained on. User can also freely upload any image for classification into the ten classes the network was trained on. CNN EXPLAINER resizes a user’s image while preserving the aspect ratio to fit one dimension of the model input size, and then crop the central region if the other dimensions do not match. The fourth of six AV tool engagement levels allows users to change the AV tool’s input [176]. Supporting custom images engages users, by allowing them to analyze the network’s classification decisions and interactively test hypotheses on diverse image inputs.

Show network details. A user can toggle the “Show detail” button, which [Show detail](#) displays additional network specifications in the *Overview*. When toggled on, the *Overview* will reveal layer dimensions and show color scale legends. Additionally, a user can vary the activation map color scale range. The CNN architecture presented by CNN EXPLAINER is grouped into four units and two modules (Fig. 3.5). By modifying the drop-down menu in the *Control Panel*, a user can adjust the color scale range used by the network to investigate activations with different groupings.

Explore hyperparameter impact. The tutorial article (§ 3.4.5) includes an interactive *Hyperparameter Widget* that allows users to experiment with convolutional hyperparameters (Fig. 3.8). Users can adjust the input and hyperparameters of the stand-alone visualization to test how different hyperparameters change the sliding convolutional kernel and the output’s dimensions. This interactive element emphasizes learning through experimentation by supplementing knowledge gained from reading the article and using the main visualization.

3.4.7 Web-based, Open-sourced Implementation

CNN EXPLAINER is a web-based, open-sourced visualization tool to teach students the foundations of CNNs. A new user only needs a modern web-browser to access our tool, no installation required. Additionally, other datasets and linear models can be quickly applied to our visualization system due to our robust implementation.

Model Training. The CNN architecture, Tiny VGG (Fig. 3.5), presented by CNN EXPLAINER for image classification is inspired by both the popular deep learning architecture, VGGNet [174], and Stanford’s CS231n course notes [177]. It is trained on the Tiny ImageNet dataset [178]. The training dataset consists of 200 image classes and contains 100,000 64×64 RGB images, while the validation dataset contains 10,000 images across the 200 image classes. The model is trained using *TensorFlow* [170] on 10 handpicked, everyday classes: `lifeboat`, `ladybug`, `bell pepper`, `pizza`, `school bus`, `koala`, `espresso`, `red panda`, `orange`, and `sport car`. During the training process, the batch size and learning rate are fine-tuned using a 5-fold-cross-validation scheme. This simple model achieves a 70.8% top-1 accuracy on the validation dataset.

Front-end Visualization. CNN EXPLAINER loads the pre-trained Tiny VGG model and computes forward propagation results in real time in a user’s web browser using *TensorFlow.js* [179]. These results are visualized using *D3.js* [180] throughout all views.

3.5 Usage Scenarios

3.5.1 Beginner Learning Layer Connectivity

Janis is a virology researcher using CNNs in a current project. Through an online deep learning course she has a general understanding of the goals of applying CNNs, and some basic knowledge of different types of CNN layers, but she needs help filling in some gaps in knowledge. Interested in learning how a 3-dimensional input (RGB image) leads to a 1-dimensional output (vector of class probabilities) in a CNN, Janis begins exploring the architecture from the *Overview* (Fig. 3.4A).

After clicking the “Show detail” button, Janis notices that the `output` layer is a 1-dimensional tensor of size 10, while `max_pool_2`, the previous layer, is a 3-dimensional ($13 \times 13 \times 10$) tensor. Confused, she hovers over a neuron in the `output` layer to inspect connections between the final two layers of the architecture: the `max_pool_2` layer has 10 neurons; the `output` layer has 10 neurons each representing a class label, and the `output` layer is fully-connected to the `max_pool_2` layer. She clicks that `output` neuron, which transitions the *Overview* (Fig. 3.4A) to the *Flatten Elastic Explanation View* (Fig. 3.4B). She notices that edges between these two layers intersect a 1-dimensional `flatten` layer and pass through a softmax function. By hovering over pixels from the activation map, Janis understands how the 2-dimensional matrix is “unwrapped” to yield a portion of the 1-dimensional `flatten` layer. As she continues to follow the edge after the `flatten` layer, she clicks the softmax button which leads her to the *Softmax Interactive Formula View* (Fig. 3.4C). She learns how the outputs of the `flatten` layer are normalized by observing the equation linked with logits through animations. Janis recognizes that her previous coursework has not taught these “hidden” operations prior to the `output` layer, which flatten and normalize the output of the `max_pool_2` layer. Instead of searching through lecture

videos and textbooks, CNN EXPLAINER enables Janis to learn these often-overlooked operations through a hierarchy of interactive views in a stand-alone website. She now feels more equipped to apply CNNs to her virology research.

3.5.2 Teaching Through Interactive Experimentation

A university professor, Damian, is currently teaching a computer vision class which covers CNNs. Damian begins his lecture with standard slides. After describing the theory of convolutions, he opens CNN EXPLAINER to demonstrate the convolution operation working inside a full CNN for image classification. With CNN EXPLAINER projected to the class, Damian transitions from the *Overview* (Fig. 3.2A) to the *Convolutional Elastic Explanation View* (Fig. 3.2B). Damian encourages the class to interpret the sliding window animation (Fig. 3.1B) as it generates several intermediate results. He then asks the class to predict kernel weights in a specific neuron. To test student’s hypotheses, Damian enters the *Convolutional Interactive Formula View* (Fig. 3.2C), to display the convolution operation with the true kernel weights. In this view, he can hover over the input and output matrices to answer questions from the class, and display computations behind the operation.

Recalled from theory, a student asks a question regarding the impact of altering the stride hyperparameter on the animated sliding window in convolutional layers. To illustrate the impact of alternative hyperparameters, Damian scrolls down to the “Convolutional Layer” section of the complementary article, where he experiments by adjusting stride and other hyperparameters with the *Hyperparameter Widget* (Fig. 3.8) in front of the class. CNN EXPLAINER is the first software that allows Damian to explain convolutional operations and hyperparameters with real image inputs, and quickly answer students’ questions in class. Previously, Damian had to draw illustrations with simple matrix inputs on slides or a chalkboard. Finally, to reinforce the concepts and encourage individual experimentation, Damian provides the class with a URL to the web-based CNN EXPLAINER for students to return to in the future.

3.6 Observational Study

We conducted an observational study to investigate how CNN EXPLAINER’s target users (e.g., aspiring deep learning students) would use this tool to learn about CNNs, and also to test the tool’s usability.

3.6.1 Participants

CNN EXPLAINER is designed for deep learning beginners who are interested in learning CNNs. In this study, we aimed to recruit participants who aspire to learn about CNNs and have some knowledge of basic machine learning concepts (e.g., knowing what an image

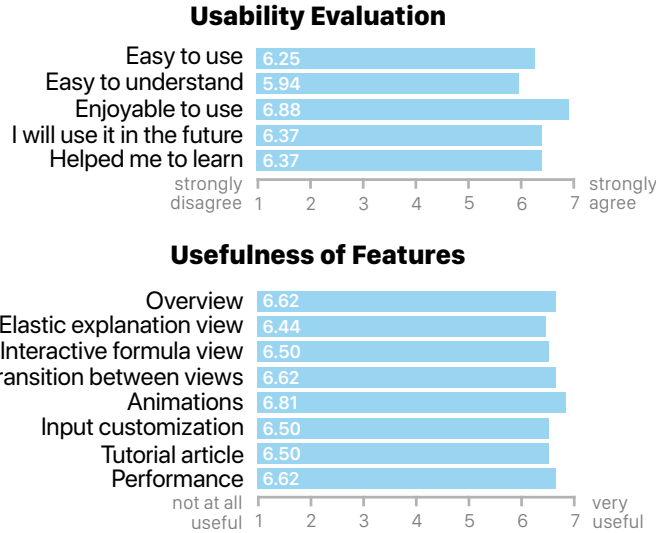


Figure 3.9: Average ratings from 16 participants regarding the usability and usefulness of CNN EXPLAINER. **Top:** Participants thought CNN EXPLAINER was easy to use, enjoyable, and helped them learn about CNNs. **Bottom:** All features, especially animations, were rated favorably.

classifier is). We recruited 16 student participants from a large university (4 female, 12 male) through internal mailing lists (e.g., machine learning and computer science Ph.D., M.S., and undergraduate students). Seven participants were Ph.D. students, seven were M.S. students, and the other two were undergraduates. All participants were interested in learning CNNs, and none of them had known CNN EXPLAINER before. Participants self-reported their level of knowledge on non-neural network machine learning techniques, with an average score of 3.26 on a scale of 0 to 5 (0 being “no knowledge” and 5 being “expert”); and an average score of 2.06 on CNNs (on the same scale). No participant self-reported a score of 5 for their knowledge on CNNs, and one participant had a score of 0. To help better organize our discussion, we refer to participants with CNN knowledge score of 0, 1 or 2 as B1-B11, where “B” stands for “Beginner”; and those with score of 3 or 4 as K1-K5, where “K” stands for “Knowledgeable.”

3.6.2 Procedure

We conducted this study with participants one-on-one via video-conferencing software. With the permission of all participants, we recorded the participants’ audio and computer screen for subsequent analysis. After participants signed consent forms, we provided them a 5-minute overview of CNNs, followed by a 3-minute tutorial of CNN EXPLAINER. Participants then freely explored our tool in their computer’s web browser. We also provided a feature checklist, which outlined the main features of our tool and encouraged participants to try as many features as they could. During the study, participants were asked to think aloud and share their computer screen with us; they were encouraged to ask questions when necessary. Each session ended with a usability questionnaire coupled with an exit interview

that asked participants about their process of using CNN EXPLAINER, and if this tool could be helpful for them. Each study lasted around 50 minutes, and we compensated each participant with a \$10 Amazon Gift card.

3.6.3 Results and Design Lessons

The exit questionnaire included a series of 7-point Likert-scale questions about the utility and usefulness of different views in CNN EXPLAINER (Fig. 3.9). All average Likert ratings were above 6 except the rating of “easy to understand”. From the high ratings and our observations, participants found our tool easy to use and understand, retained a high engagement level during their session, and eventually gained a better understanding of CNN concepts. Our observations also reflect key findings in previous AV research [167, 169]. This section describes design lessons and limitations of our tool distilled from this study.

3.6.3.1 Transitions between different views

Transitions help users link CNN operations and structures. Several participants (9/16) commented that they liked how our tool transitions between high-level CNN structure views and low-level mathematical explanations. It helps them better understand the interplay between layer computations and the overall CNN data transformation—one of the key challenges for understanding CNN concepts, as we identified from our instructor interviews and our student survey. For example, initially K4 was confused to see the *Convolutional Elastic Explanation View*, but after reading the annotation text, he remarked, “*Oh, I understand what an intermediate layer is now—you run the convolution on the image, then you add all those results to get this.*” After exploring the *Convolutional Interactive Formula View*, he immediately noted, “*Every single aspect of the convolution layer is shown here. [This] is super helpful.*” Similarly, B5 commented, “*Good to see the big picture at once and the transition to different views [...] I like that I can hide details of a unit in a compact way and expand it when [needed].*”

CNN EXPLAINER employs the fisheye view technique for presenting the *Elastic Explanation Views* (Fig. 3.2B, Fig. 3.4B): after transitioning from the *Overview* to a specific layer, neighboring layers are still shown while further layers (lower degree-of-interest) have lower opacity. Participants found this transition design helpful for them to learn layer-specific details while having CNN structural context in the background. For instance, K5 said “*I can focus on the current layer but still know the same operation goes on for other layers.*” Our observations suggest that our fluid transition design between different level of abstraction can help users to better connect unfamiliar layer mechanisms to the complex model structure.

3.6.3.2 Animations for enjoyable learning experience

Another favorite feature of CNN EXPLAINER that participants mentioned was the use of animations, which received the highest rating in the exit questionnaire (Fig. 3.9). In our tool, animations serve two purposes: to assimilate the relationship between different visual components and to help illustrate the model’s underlying operations.

Transition animations help navigating. Layer movement is animated during view transitions. We noticed it helped participants to be aware of different views, and all participants navigated through the views naturally. In addition to assisting with understanding the relationship between distinct views, animation also helped them discover the linking between different visualization elements. For example, B8 quickly found that the logit circle is linked to its corresponding value in the formula, when she saw the circle-number pair appear one-by-one with animation in the *Softmax Interactive Formula View* (Fig. 3.4C).

Algorithm animations contribute to understanding. Animations that simulate the model’s inner-workings helped participants learn underlying operations by validating their hypotheses. In the *Convolutional Elastic Explanation View* (Fig. 3.1B), we animate a small rectangle sliding through one matrix to mimic the CNN’s internal sliding window. We noticed many participants had their attention drawn to this animation when they first transitioned into the *Convolutional Elastic Explanation View*. However, they did not report that they understood the convolution operation until interacting with other features, such as reading the annotation text or transitioning to the *Convolutional Interactive Formula View* (Fig. 3.1C). Some participants went back to watch the animation multiple times and commented that it made sense, for example, K5 said “*Very helpful to see how the image builds as the window slides through,*” but others, such as B9 remarked, “*It is not easy to understand [convolution] using only animation.*” Therefore, we hypothesize that this animation can indirectly help users to learn about the convolution algorithm by validating their newly formed mental models of how specific operation behave. To test this hypothesis, a rigorous controlled experiment would be needed. Related research work on the effect of animation in computer science education also found that algorithm animation does not automatically improve learning, but it may lead learners to make predictions of the algorithm behavior which in turn helps learning [181].

Animations improve learning engagement and enjoyment. We found animations helped to increase participants’ engagement level (e.g., spending more time and effort) and made CNN EXPLAINER more enjoyable to use. In the study, many participants repeatedly played and viewed different animations. For example, K2 replayed the window sliding animation multiple times: “*The is very well-animated [...] I always love smooth animations.*” B7 also attributed animations to his enjoyable experience with our tool: “*[The tool is] enjoyable to use [...] I especially like the lovely animation.*”

3.6.3.3 Engaging learning through visualization customization

CNN EXPLAINER allows users to modify the visualization. For example, users can change the input image or upload their own image for classification; CNN EXPLAINER visualizes the new prediction with the new activation maps in every layer. Similarly, users can interactively explore how hyperparameters affect the convolution operation (Fig. 3.8).

Customization enables hypothesis testing. Many participants used visualization customization to test their predictions of model behaviors. For example, through inspecting the input layer in the *Overview*, B4 learned that the input layer comprised multiple different image channels (e.g., red, green, and blue). He changed the input image to a red bell pepper from Tiny Imagenet and expected to see high values in the input red channel: “*If I click the red image, I would see...*” After the updated visualization showed what he predicted, he said “*Right, it makes sense.*” We found the *Hyperparameter Widget* also allowed participants to test their hypotheses. While reading the description of convolution hyperparameters in the tutorial article, K3 noted “*Wait, then sometimes they won’t work*”. He then modified the hyperparameters in the *Hyperparameter Widget* and noticed some combinations indeed did not yield a valid operation output: “*It won’t be able to slide, because the stride and kernel size don’t fit the matrix*”.

Customization facilitates engagement. Participants were intrigued to modify the visualization, and their engagement sparked further interest in learning CNNs. In the study, B6 spent a large amount of time on testing the CNN’s behavior on edge cases by finding “difficult” images online. He searched with keywords “koala”, “koala in a car”, “bell pepper pizza”, and eventually found a bell pepper pizza photo. Our CNN model predicted the image as `bell pepper` with a probability of 0.71 and `ladybug` with a probability of 0.2. He commented, “*The model is not robust [...] oh, the ladybug [’s high softmax score] might come from the red dot.*” Another participant B5 uploaded his own photo as a new input image for the CNN model. After seeing his picture being classified as `espresso`, B5 started to use our tool to explore the reason of such classification by tracking back activation maps. He also asked how do experts interpret CNNs and said he would be interested in learning more about deep learning interpretability. This observation reflects previous findings that customizable visualization makes learning more engaging [167, 176].

3.6.3.4 Limitations

While we found CNN EXPLAINER provided participants with an engaging and enjoyable learning experience and helped them to more easily learn about CNNs, we also noticed some potential improvements to our current system design from this study.

Beginners need more guidance. We found that participants with less knowledge of CNNs needed more instructions to begin using CNN EXPLAINER. Some participants reported that the visual representation of the CNN and animation initially were not easy

to understand, but the tutorial article and text annotations greatly helped them to interpret the visualizations. B8 skimmed through the tutorial article before interacting with the main visualization. She said, “*After going through the article, I think I will be able to use the tool better [...] I think the article is good, for beginner users especially.*” B2 appreciated the ability to jump to a certain section in the article by clicking the layer name in the visualization, and he suggested us to “*include a step-by-step tutorial for first time users [...] There was too much information, and I didn’t know where to click at the beginning*”. Therefore, we believe adding more text annotation and having a step-by-step tutorial mode could help beginners better understand the relations between CNN operations and their visual representations.

Limited explanation of why CNN works. Some participants, especially those less experienced with CNNs, were interested in learning *why* the CNN architecture works in addition to learning *how* a CNN model makes predictions. For example, B7 asked “*Why do we need ReLU?*” when he was learning the formula of the ReLU function. B5 understood what a Max Pooling layer’s operation does but was unclear why it contributes to CNN’s performance: “*It is counter-intuitive that Max Pooling reduces the [representation] size but makes the model better.*” Similarly, B6 commented on the Max Pooling layer: “*Why not take the minimum value? [...] I know how to compute them [layers], but I don’t know why we compute them.*” Even though it is still an open question why CNNs work so well for various applications [165, 182], there are some commonly accepted “intuitions” of how different layers help this model class succeed. We briefly explain them in the tutorial article: for example, ReLU function is used to introduce non-linearity in the model. However, we believe it is worth designing visualizations that help users to learn about these concepts. For example, allowing users to change the ReLU activation function to a linear function, and then visualizing the new model predictions may help users gain understanding of *why* non-linear activation functions are needed in CNNs.

3.7 Discussion and Future Work

Explaining training process and backpropagation. CNN EXPLAINER helps users to learn how a pre-trained CNN model transforms the input image data into a class prediction. As we identified from two preliminary studies and an observational study, students are also interested in learning about the training process and backpropagation of CNNs. We plan to work with instructors and students to design and develop new visualizations to help beginners gain understanding of the training process and backpropagation in detail.

Generalizing to other layer types and neural network models. Our observational study demonstrated that CNN EXPLAINER helps users more easily understand low-level layer operations, high-level model structure, and their connections. We can adapt the *Interactive Formula Views* to explain other layer types (e.g., Leaky ReLU [183]) or a combination of layers (e.g. Residual Block [184]). Similarly, the transition between

different levels of abstraction can be generalized to other neural networks, such as long short-term memory networks [185] and Transformer models [186] that require learners to understand the intricate layer operations in the context of a complex network structure.

Integrating algorithm visualization best practices. Existing work has studied how to design effective visualizations to help students learn algorithms. CNN EXPLAINER applies two key design principles from AV—visualizations with explanations and customizable visualizations (**G4**). However, there are many other AV design practices that future researchers can integrate in educational deep learning tools, such as giving interactive “quizzes” during the visualization process [187] and encouraging users to build their own visualizations [188].

Quantitative evaluation of educational effectiveness. We conducted a qualitative observational study to evaluate the usefulness and usability of CNN EXPLAINER. Further quantitative user studies would help us investigate how visualization tools help users gain understanding of deep learning concepts. We will draw inspiration from recent research [161, 189] to assess users’ engagement level and content understanding through analysis of interaction logs.

3.8 Conclusion

As deep learning is used throughout our everyday life, it is important to help learners take the first step toward understanding this promising yet complex technology. In this work, we present CNN EXPLAINER, an interactive visualization system designed for non-experts to more easily learn about CNNs. Our tool runs in modern web browsers and is open-sourced, broadening the public’s education access to modern AI techniques. We discussed design lessons learned from our iterative design process and an observational user study. We hope our work will inspire further research and development of visualization tools that help democratize and lower the barrier to understanding and appropriately applying AI technologies.

3.9 Impact

To broaden the public’s education access to modern AI technologies, we release CNN EXPLAINER as an open-source web-based tool. The public demo has transformed AI education: it has been integrated into deep learning courses (Carnegie Mellon, Georgia Tech, University of Tokyo, UC Santa Barbara, and more), helping **360k+ novices** from 200+ countries learn about CNNs, and it has received **7k+ starts** on GitHub. It has also been highlighted as a top visualization publication (**top 1%**) invited to present in SIGGRAPH.

CHAPTER 4

WIZMAP: EXPLAIN AI DATA AND EMBEDDINGS TO PRACTITIONERS

Powered by neural networks, modern AI models can learn high-dimensional embedding representations that capture the domain semantics and relationships in the training data. These embeddings are extremely useful for AI researchers and practitioners to probe what the AI models have learned [15]. However, it can be difficult to interpret embeddings in practice, as these high-dimensional representations are often opaque, complex, and can contain unpredictable structures [190]. Moreover, practitioners also face scalability challenges as large training datasets can require them to study millions of embeddings holistically [191]. To help AI practitioners explore and interpret large embeddings in their AI models, we design and develop WIZMAP, a scalable interactive visualization tool for AI embeddings. We leverage dimensionality reduction and a familiar map-like interaction design to visualize any AI embedding models (Fig. 4.2). In addition, we introduce a novel technique to generate multi-resolution embedding summaries to help users interpret large-scale embedding data.

4.1 Introduction

Modern ML models learn high-dimensional embedding representations to capture the domain semantics and relationships in the training data [15]. ML researchers and domain experts are increasingly using expressive embedding representations to interpret trained models [21], develop models for new domains [22] and modalities [23], as well as analyze and synthesize new datasets [24]. However, it can be difficult to interpret and use embeddings in practice, as these high-dimensional representations are often opaque, complex, and can contain unpredictable structures [192]. Furthermore, analysts face scalability challenges as large datasets can require them to study millions of embeddings holistically [191].

To tackle these challenges, researchers have proposed several interactive visualization tools to help users explore embedding spaces [e.g., 36, 39]. These tools often visualize embeddings in a low-dimensional scatter plot where users can browse, filter, and compare

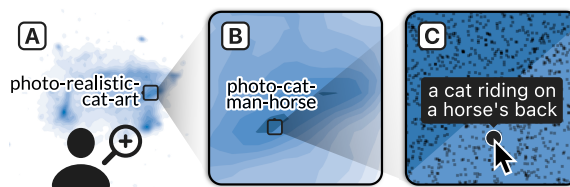


Figure 4.1: WIZMAP enables users to explore embeddings at different levels of detail. (A) The contour plot with automatically-generated embedding summaries provides an overview. (B) Embedding summaries adjust in resolution as users zoom in. (C) The scatter plot enables the investigation of individual embeddings.

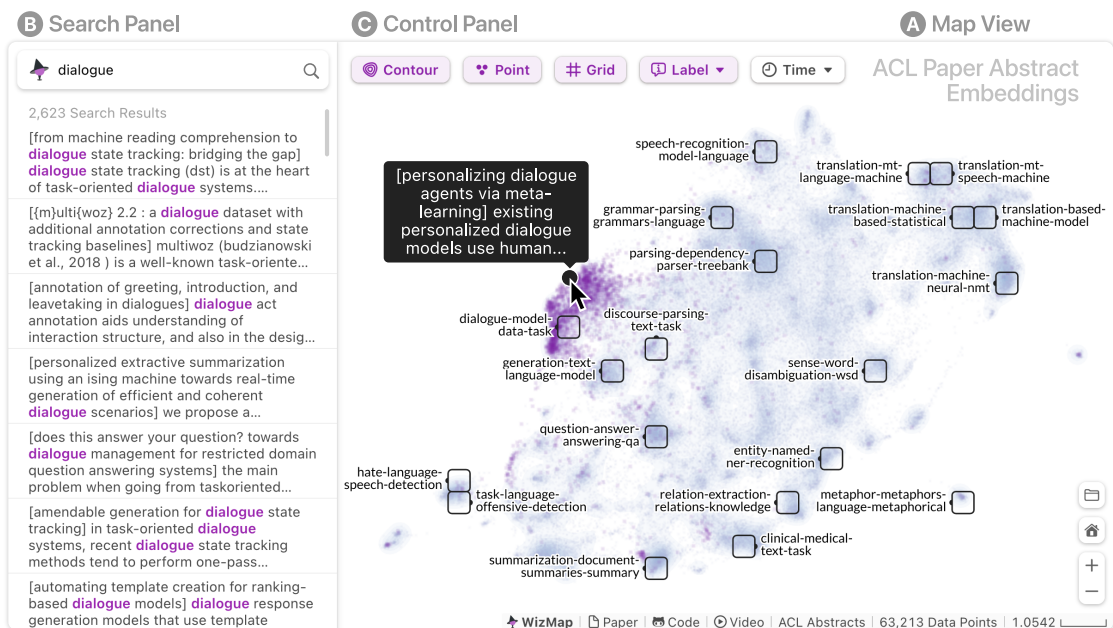


Figure 4.2: WIZMAP empowers AI researchers and practitioners to easily explore and interpret *millions* of embedding vectors across different levels of granularity. Consider the task of investigating the embeddings of all 63k natural language processing paper abstracts indexed in ACL Anthology from 1980 to 2022. (A) **The Map View** tightly integrates a contour layer, a scatter plot, and automatically generated multi-resolution embedding summaries to help users navigate through the large embedding space. (B) **The Search Panel** enables users to rapidly test their hypotheses through a fast full-text embedding search. (C) **The Control Panel** allows users to customize embedding visualizations, compare multiple embedding groups, and observe how embeddings evolve over time.

embedding points. However, for large datasets, it is taxing or even implausible to inspect embedded data point by point to make sense of the *global structure* of an embedding space. Alternatively, recent research explores using contour plots to summarize embeddings [193, 194]. Although contour abstractions enable users to obtain an overview of the embedding space and compare multiple embeddings through superposition, a user study reveals that contour plots restrict users’ exploration of an embedding’s *local structures*, where users would prefer to have more visual context [194]. To bridge this critical gap between two visualization approaches and provide users with a holistic view, we design and develop WIZMAP (Fig. 4.2). Our work makes the following **major contributions**:

- **WIZMAP, a scalable interactive visualization tool** that empowers ML researchers and domain experts to explore and interpret embeddings with *millions* of points. Our tool employs a familiar map-like interaction design and fluidly presents adaptive visual summaries of embeddings across different levels of granularity (Fig. 4.1, § 4.3).
- **Novel and efficient method to generate multi-resolution embedding summaries.** To automatically summarize embedding neighborhoods with different degrees of granularity, we construct a quadtree [195] from embedding points and extract keywords (text data) or exemplar points (other data types) from tree nodes with efficient branch aggregation (§ 4.2).

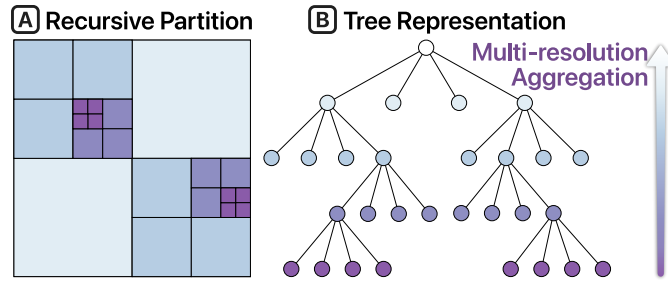


Figure 4.3: (A) A quadtree recursively partitions a 2D space into four equally-sized squares, (B) and each square is represented as a tree node. WIZMAP efficiently aggregates information from the leaves to the root, summarizing embeddings at different levels of granularity.

- **An open-source¹ and web-based implementation** that lowers the barrier to interpreting and using embeddings. We develop WIZMAP with modern web technologies such as WebGL and Web Workers so that anyone can access the tool directly in both their web browsers and computational notebooks without a need for dedicated backend servers (§ 4.3.4). For a demo video of WIZMAP, visit <https://youtu.be/8fJG87QVceQ>.

4.2 Multi-scale Embedding Summarization

Researchers have highlighted users’ desire for embedding visualizations to provide visual contexts and embedding summaries to facilitate exploration of various regions within the embedding space [194]. However, generating embedding summaries is challenging for two reasons. First, efficiently summarizing millions of data points in larger datasets can be a formidable task. Second, selecting the embedding regions to summarize is difficult, as users possess varying interests in regions of different sizes and levels of granularity. To tackle this challenge, we propose a novel method to automatically generate multi-resolution embedding summaries at scale.

Multi-resolution Quadtree Aggregation. First, we apply a dimensionality reduction technique such as UMAP to project high-dimensional embedding vectors into 2D points. From these points, we construct a quadtree [195], a tree data structure that recursively partitions a 2D space into four equally-sized squares, each represented as a node. Each data point exists in a unique leaf node. To summarize embeddings across different levels of granularity, we traverse the tree bottom up. In each iteration, we first extract summaries of embeddings in each leaf node, and then merge the leaf nodes at the lowest level with their parent node. This process continues recursively, with larger and larger leaf nodes being formed until the entire tree is merged into a single node at the root. Finally, we map pre-computed embedding summaries to a suitable granularity level and dynamically show them as users zoom in or out in WIZMAP (Fig. 4.3.1).

Scalable Leaf-level Summarization. When performing quadtree aggregation, re-

¹WIZMAP code: <https://github.com/poloclub/wizmap>

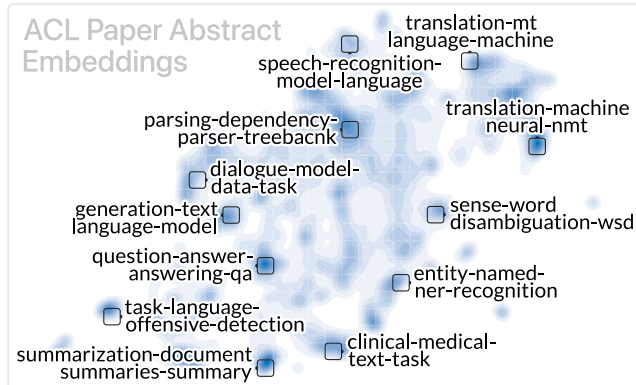


Figure 4.4: The *Map View* provides an overview via a contour plot and auto-generated multi-resolution embedding labels placed around high-density areas.

searchers have the flexibility to choose any suitable method for summarizing embedding from leaf nodes. For text embeddings, we propose t-TF-IDF (tile-based TF-IDF) that adapts TF-IDF (term frequency-inverse document frequency) to extract keywords from leaf nodes [196]. Our approach is similar to c-TF-IDF (classed-based TF-IDF) that combines documents in a cluster into a meta-document before computing TF-IDF scores [197]. Here, we merge all documents in each leaf node (i.e., a tile in the quadtree partition) as a meta-document and compute TF-IDF scores across all leaf nodes. Finally, we extract keywords with the highest t-TF-IDF scores to summarize embeddings in a leaf node. This approach is scalable and complementary to quadtree aggregation. Because our document merging is hierarchical, we only construct the n-gram count matrix once and update it in each aggregation iteration with just one matrix multiplication. Summarizing 1.8 million text embeddings across three granularity levels takes only about 55 seconds on a MacBook Pro. For non-text data, we summarize embeddings by finding points closest to the embedding centroid in a leaf node.

4.3 User Interface

Leveraging pre-computed multi-resolution embedding summarization (§ 4.2), WIZMAP tightly integrates three interface components (Fig. 4.2A–C).

4.3.1 Map View

The *Map View* (Fig. 4.2A) is the primary view of WIZMAP. It provides a familiar map-like interface that allows users to pan and zoom to explore different embedding regions with varying sizes. To help users easily investigate both the global structure and local neighborhoods of their embeddings, the *Map View* integrates three layers of visualization.

Distribution Contour. To provide users with a quick overview of the global structure of their embeddings, we use Kernel Density Estimation (KDE) [198] to estimate the distribution

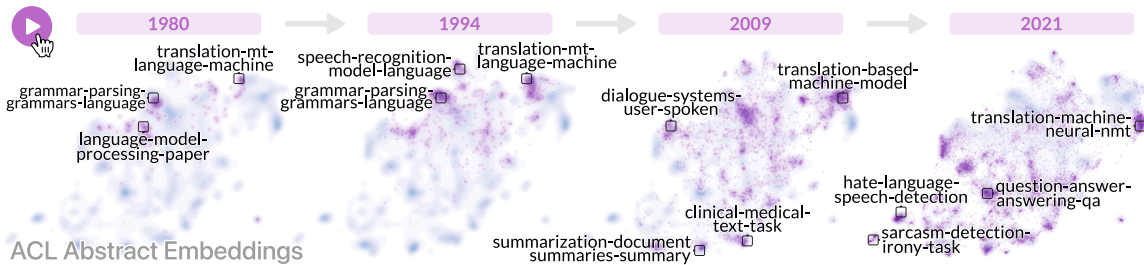



Figure 4.5: WIZMAP allows users to observe how embeddings change over time. For example, when exploring 63k ACL paper abstracts, clicking the play button  in the *Control Panel* animates the visualizations to show embeddings of papers published in each year in purple and the distribution of all papers in blue. This animation highlights changes in ACL research topics over time, such as the decline in popularity of grammar and the rise of question-answering.

of 2D embedding points. We use a standard multivariate Gaussian kernel with a Silverman bandwidth for the KDE model [199]. Next, we compute the distribution likelihoods over a 200×200 2D grid whose size is determined by the range of all embedding points. Finally, we visualize the likelihoods over the grid as a contour plot (Fig. 4.4), highlighting the high-level density distribution of users’ embeddings. Researchers can adjust the grid density, and we tune it by balancing the computation time and the contour resolution.

Multi-resolution Labels. The *Map View* helps users interpret embeddings across various levels of granularity by dynamically providing pre-computed contextual labels. It overlays summaries generated via quadtree aggregation (§ 4.2) onto the distribution contour and scatter plot. Users can hover over to see the summary from a quadtree tile closest to the cursor. Our tool adjusts the label’s tile size based on the user’s current zoom level. For example, when a user zooms into a small region, the *Map View* shows summaries computed at a lower level in the quadtree. In addition to on-demand embedding summaries, this view also automatically labels high-density regions (Fig. 4.4) by showing summaries from quadtree tiles near the geometric centers of high-probability contour polygons.

Scatter Plot. To help users pinpoint embeddings within their local neighborhoods, the *Map View* visualizes all embedding points in a scatter plot with their 2D positions. Users can specify the color of each embedding point to encode additional features, such as the class of embeddings. Also, users can hover over an embedding point to reveal its original data, such as ACL paper abstracts (§ 4.4.1).

4.3.2 Control Panel

The *Map View* shows all three visualization layers by default, and users can customize them to fit their needs by clicking buttons in the *Control Panel* (Fig. 4.2C). In addition, WIZMAP allows users to compare multiple embedding groups in the same embedding space by superimposing them in the *Map View* [200]. In the case of embeddings that include times, users can use a slider in the *Control Panel* to observe changes in the embeddings over time (Fig. 4.5).

4.3.3 Search Panel

Searching and filtering help users discover interesting embedding patterns and test hypothesis regarding embedding structures [201]. In WIZMAP, users can use the *Search Panel* (Fig. 4.2B) to search text embeddings including specified words in the original data. The panel shows search results, and the *Map View* highlights their corresponding points.

4.3.4 Scalable & Open-source Implementation

WIZMAP is scalable to *millions* of embedding points, providing a seamless user experience with zooming and animations, all within web browsers without backend servers. To achieve this, we leverage modern web technologies, especially WebGL to render embedding points with the `regl` API [202]. We also use Web Workers and Streams API to enable the streaming of large embedding files in parallel with rendering. To enable fast full-time search, we apply a contextual index scoring algorithm with FlexSearch [203]. We use D3 [180] for other visualizations and `scikit-learn` [204] for KDE. To ensure that our tool can be easily incorporated into users' current workflows [205], we apply NOVA [206] to make WIZMAP available within computational notebooks. Users can also share their embedding maps with collaborators through unique URLs. We provide detailed tutorials to help users use our tool with their embeddings. We have open-sourced our implementation to support future research and development of embedding exploration tools.

4.4 Usage Scenarios

We present two hypothetical scenarios, each with real embedding data, to demonstrate how WIZMAP can help ML researchers and domain experts easily explore embeddings and gain a better understanding of ML model behaviors and dataset patterns.


4.4.1 Exploring ACL Research Topic Trends

Helen, a science historian, is interested in exploring the evolution of computational linguistic and natural language processing (NLP) research since its inception. She downloads the Bibtext files of all papers indexed in ACL Anthology [207]. and extracts the paper title and abstract from 63k papers that have abstracts available. Then, Helen applies MPNet, a state-of-the-art embedding model [208], to transform the concatenation of each paper's title and abstract into a 768-dimensional embedding vector. She then trains a UMAP model to project extracted embeddings into a 2D space. She tunes the UMAP's hyperparameter `n_neighbors` to ensure projected points are spread out [209].

Helen uses a Python function provided by WIZMAP to generate three JSON files containing embedding summaries (§ 4.2), the KDE distributions (Fig. 4.3.1), and the original data in a streamable format [210]. Helen configures the function to use the dataset's

year feature as the embedding’s time—the function computes the KDE distribution of embeddings for each year slice. She provides the files to WIZMAP and sees a visualization of all ACL abstract embeddings (Fig. 4.4A).

Embedding Exploration. In the *Map View*, Helen explores embeddings with zoom and pan. She also uses the *Search Panel* to find papers with specific keywords, such as “dialogue”, and Helen is pleased to see all related papers are grouped in a cluster (Fig. 4.2B). With the help of multi-resolution embedding summaries, Helen quickly gains an understanding of the structure of her embedding space. For example, she finds that the top right cluster features translation papers while the lower clusters feature summarization and medical NLP.

Embedding Evolution. To examine how ACL research topics change over time, Helen clicks the play button clicking the play button  in the *Control Panel* to animate the visualizations. The *Map View* shows embeddings of papers published in each year from 1980 to 2022 in purple, while the distribution of all papers is shown as a blue background (Fig. 4.5). As Helen observes the animation, she identifies several interesting trends. For example, she observes a decline in the popularity of grammar research, while question-answering has become increasingly popular. She also notes the emergence of some small clusters in recent years, featuring relatively new topics, such as sarcasm, humor, and hate speech. Satisfied with the findings using WIZMAP, Helen decides to write an essay on the trend of NLP research over four decades.

4.4.2 Investigating Text-to-Image Model Usage

Bob, an ML researcher, works on improving text-to-image generative models. Recent advancements in diffusion models, such as Stable Diffusion [211], have attracted an increasing number of users to generate photorealistic images by writing text prompts. To understand these models’ behaviors and identify potential weaknesses for improvement, Bob decides to study how users use these models in the wild by analyzing DiffusionDB, a dataset containing 14 million images generated by Stable Diffusion with 1.8 million unique text prompts [212].

Bob’s analysis goal is to study the relationship between the text prompts and their generated images. Thus, he chooses to use CLIP [27] to encode both prompts and images into a 768-dimensional multimodal embedding before projecting them to a 2D space with UMAP. He uses prompts to generate embedding summaries for the CLIP space. After creating all JSON files, WIZMAP visualizes all 3.6 million embeddings (Fig. 4.6).

Embedding Exploration. Bob starts by hiding image embeddings and scatter plots, focusing on the global structure of embeddings with the contour plot and embedding summaries. He discovers two dominant prompt categories: art-related prompts and photography-related prompts. Two categories appear far from each other, which is not surprising as they are expected to have distinct semantic representations. Bob notices two smaller clusters within the photography region, prompting him to zoom in and turn on the scatter plot to

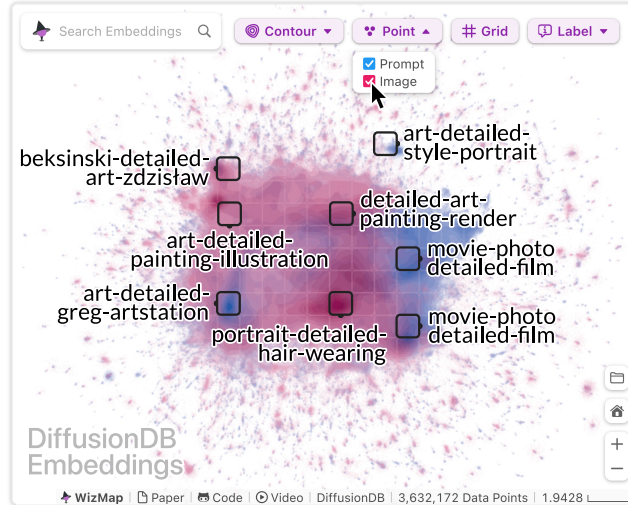


Figure 4.6: WIZMAP enables users to compare multiple embeddings by visualization superposition. For instance, comparing the CLIP embeddings of **1.8 million** Stable Diffusion **prompts** and **1.8 million** generated **images** reveals key differences between two distributions.

further investigate these local regions (Fig. 4.1). After hovering over a few points, he realizes one cluster is mostly about non-human objects while the other is about celebrities.

Embedding Comparison. To investigate the relationship between text prompts and their generated images, Bob clicks a button in the *Control Panel* to superimpose the contour and scatter plot of image embeddings **in red** onto the text embedding visualizations **in blue** (Fig. 4.6). Bob quickly identifies areas where two distributions overlap and differ. He notes that the “movie” cluster in the text embeddings has a lower density in the image embeddings, whereas a high-density “art portrait” cluster emerges in image embeddings. Bob hypothesizes that Stable Diffusion may have limited capability to generate photorealistic human faces [213]. After exploring embedding with WIZMAP, Bob is pleased with his findings, and he will apply his insights to improve the curation of his training data.

4.5 Conclusion

WIZMAP integrates a novel quadtree-based embedding summarization technique that enables users to easily explore and interpret large embeddings across different levels of granularity. Our usage scenarios showcase our tool’s potential for providing ML researchers and domain experts with a holistic view of their embeddings. Future researchers can use WIZMAP as a research instrument to conduct observational user studies to test how practitioners interpret embedding data, study more robust methods for embedding summarization [214], and integrate more effective embedding comparison techniques [200].

CHAPTER 5

DIFFUSIONDB: EXPLAIN AI USAGE TO RESEARCHERS AND POLICYMAKERS

Recent breakthroughs in large text-to-image generative models (e.g., Stable Diffusion, DALL-E, and Midjourney) and easy access to these models have attracted millions of users to use them to create award-winning artworks, synthetic radiology images, and even hyper-realistic videos. There are growing concerns from researchers and policymakers about the potential misuse of these models, such as generating misinformation and scams [215]. However, as these models possess a wide range of capabilities and are relatively new, it is difficult for researchers and policymakers to assess their impacts and potential harms. To help researchers and policymakers easily investigate the real usage of large generative models and assess their impacts, we present DIFFUSIONDB, the first large-scale usage log dataset of large text-to-image generative models. DIFFUSIONDB contains 14 million images generated by Stable Diffusion using 1.8 million unique prompts and hyperparameters specified by real users (Fig. 5.2). We release DIFFUSIONDB with a CC0 1.0 license, allowing anyone to flexibly share and adapt the dataset for their use. Finally, we open source our code that collects, processes, and analyzes the images and prompts.

5.1 Introduction

Recent diffusion models have gained immense popularity by enabling high-quality and controllable image generation based on text prompts written in natural language [211, 216, 217]. Since the release of these models, people from different domains have quickly applied them to create award-winning artworks [218], synthetic radiology images [219], and even

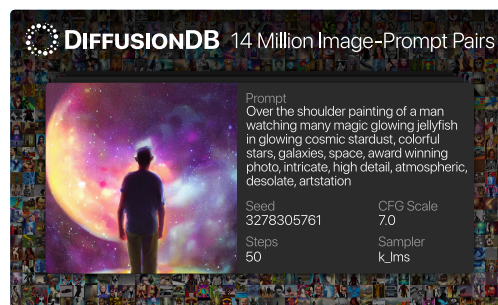


Figure 5.1: DIFFUSIONDB is the first large-scale dataset featuring 6.5TB data including 1.8 million unique Stable Diffusion prompts and 14 million generated images with accompanying hyperparameters. It provides exciting research opportunities in prompt engineering, deepfake detection, and understanding large generative models.

	Prompt a keeshond puppy, watercolor painting by jean - baptiste monge, muted colors	Filename 9dba5021-cd9b- 43a3-ac0a- b0f8ed4afeeb.webp	User Hash 481089cb827f2 63b26445dc0f1 81e08dcfd4ad2e a212abcf29f3fdf 7ec3c11cf	Seed 856498039 Timestamp 2022-08-14 21:51:00+0000	Step 100 Sampler k_lms	CFG Scale 11.0 Image Size (512, 512)	Prompt NSFW 0.15525 Image NSFW 0.04811
	Prompt poignant portrait black and white photo of an old couple smiling at each other, nostalgia, love	Filename fa5c8b9f-3789- 46a4-8d8a- 6cbe5f104acf.webp	User Hash 9e1ee59715df53 70f703859a2b0 8619783e31f55 c0582398ccf71 9d9f7c68d58	Seed 1596176968 Timestamp 2022-08-20 08:12:00+0000	Step 50 Sampler k_lms	CFG Scale 7.0 Image Size (512, 512)	Prompt NSFW 0.01437 Image NSFW 0.02996

Figure 5.2: DIFFUSIONDB contains 14 million Stable Diffusion images, 1.8 million unique text prompts, and all model hyperparameters. Each image also has a unique filename, a hash of its creator’s identifier, a creation timestamp, and an NSFW score computed by state-of-the-art models.

hyper-realistic videos [220].

However, generating images with desired details is difficult, as it requires users to write proper prompts specifying the exact expected results. Developing such prompts requires trial and error, and can often feel random and unprincipled [50]. Willison *et al.* analogize writing prompts to wizards learning “magical spells”: users do not understand why some prompts work, but they will add these prompts to their “spell book.” For example, to generate highly-detailed images, it has become a common practice to add special keywords such as “trending on artstation” and “unreal engine” in the prompt.

Prompt engineering has become a field of study in the context of text-to-text generation, where researchers systematically investigate how to construct prompts to effectively solve different downstream tasks [222, 223]. As large text-to-image models are relatively new, there is a pressing need to understand how these models react to prompts, how to write effective prompts, and how to design tools to help users generate images [50]. Our work helps researchers tackle these critical challenges, through three major **contributions**:

- **DIFFUSIONDB (Fig. 5.1), the first large-scale prompt dataset totaling 6.5TB**, containing 14 million images generated by Stable Diffusion [211] using 1.8 million unique prompts and hyperparameters specified by real users. We construct this dataset by collecting images shared on the Stable Diffusion public Discord server (§ 5.2). We release DIFFUSIONDB with a CC0 1.0 license, allowing users to flexibly share and adapt the dataset for their use. In addition, we open-source our code¹ that collects, processes, and analyzes the images and prompts.
- **Revealing prompt patterns and model errors.** The unprecedented scale of DIFFUSIONDB paves the path for researchers to systematically investigate diverse prompts and associated images that were previously not possible. By characterizing prompts and images, we discover common prompt patterns and find different distributions of the semantic representations of prompts and images. Our error analysis highlights particular hyperparameters and prompt styles can lead to model errors. Finally, we provide evidence

¹Code: <https://github.com/poloclub/diffusiondb>

of image generative models being used for potentially harmful purposes such as generating misinformation and nonconsensual pornography (§ 5.3).

- **Highlighting new research directions.** As the first-of-its-kind text-to-image prompt dataset, DIFFUSIONDB opens up unique opportunities for researchers from natural language processing (NLP), computer vision, and human-computer interaction (HCI) communities. The scale and diversity of this human-actuated dataset will provide new research opportunities in better tooling for prompt engineering, explaining large generative models, and detecting deepfakes (§ 5.4).

We believe DIFFUSIONDB will serve as an important resource to study the roles of prompts in text-to-image generation and design next-generation human-AI interaction tools.

5.2 Constructing DIFFUSIONDB

We construct DIFFUSIONDB (Fig. 5.2) by scraping user-generated images from the official Stable Diffusion Discord server. We choose Stable Diffusion as it is currently the only open-source large text-to-image generative model, and all generated images have a CC0 1.0 license that allows uses for any purpose [224]. We choose the official public Discord server as it has strict rules against generating illegal, hateful, or NSFW (not suitable for work, such as sexual and violent content) images, and it prohibits sharing prompts with personal information [225]. Our construction process includes collecting images (§ 5.2.1), linking them to prompts and hyperparameters (§ 5.2.2), applying NSFW detectors (§ 5.2.3), creating a flexible file structure (§ 5.2.4), and distributing the dataset (§ 5.2.5). We discuss DIFFUSIONDB’s limitations in § 5.5.

5.2.1 Collecting User Generated Images

We download chat messages from the Stable Diffusion Discord channels with Discord-ChatExporter [226], saving them as HTML files. We focus on channels where users can command a bot to run Stable Diffusion Version 1 to generate images by typing a prompt, hyperparameters, and the number of images. The bot then replies with the generated images and used random seeds.

5.2.2 Extracting Image Metadata

We use Beautiful Soup [227] to parse HTML files, mapping generated images with their prompts, hyperparameters, seeds, timestamps, and the requester’s usernames. Some images are collages, where the bot combines n generated images as a grid (e.g., a 3×3 grid of $n = 9$ images); these images have the same prompt and hyperparameters but different seeds. We use Pillow [228] to split a collage into n images and assign them with the correct metadata and unique filenames. We compress all images using lossless WebP [229].

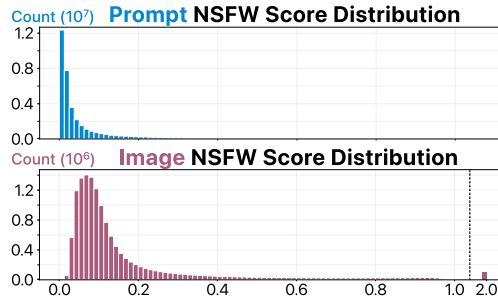


Figure 5.3: To help researchers filter out potentially unsafe data in DIFFUSIONDB, we apply NSFW detectors to predict the probability that an image-prompt pair contains NSFW content. For images, a score of 2.0 indicates the image has been blurred by Stable Diffusion.

5.2.3 Identifying NSFW Content

The Stable Diffusion Discord server prohibits generating NSFW images [225]. Also, Stable Diffusion has a built-in NSFW filter that automatically blurs generated images if it detects NSFW content. However, we find DIFFUSIONDB still includes NSFW images that were not detected by the built-in filter or removed by server moderators. To help researchers filter these images, we apply state-of-the-art NSFW classifiers to compute NSFW scores for each prompt and image. Researchers can determine a suitable threshold to filter out potentially unsafe data for their tasks.

NSFW Prompts. We use a pre-trained multilingual toxicity prediction model to detect unsafe prompts [230]. This model outputs the probabilities of a sentence being toxic, obscene, threat, insult, identity attack, and sexually explicit. We compute the text NSFW score by taking the maximum of the probabilities of being toxic and sexually explicit (Fig. 5.3 Top).

NSFW Images. We use a pre-trained EfficientNet classifier to detect images with sexual content [231]. This model predicts the probabilities of five image types: drawing, hentai, neutral, sexual, or porn. We compute the image NSFW score by summing the probabilities of hentai, sexual, and porn. We use a Laplacian convolution kernel with a threshold of 10 to detect images that have already been blurred by Stable Diffusion and assign them a score of 2.0 (Fig. 5.3 Bottom). As Stable Diffusion’s blur effect is strong, our blurred image detector has high precision and recall (both 100% on 50k randomly sampled images).

NSFW Detector Accuracy. To access the accuracy of these two pre-trained state-of-the-art NSFW detectors, we randomly sample 5k images and 2k prompt texts and manually annotate them with two binary NSFW labels (one for image and one for prompt) and analyze the results. As the percentage of samples predicted as NSFW (score > 0.5) is small, we up-sample positive samples for annotation, where we have an equal number of positive and negative examples in our annotation sample. After annotation, we compute the precisions and recalls. Because we have up-sampled positive predictions, we adjust the recalls by multiplying false negatives by a scalar to adjust the sampling bias. The up-sampling does not

affect precisions. Finally, the precisions, recalls and adjusted recalls are 0.3604, 0.9565, and 0.6661 for the prompt NSFW detector, and 0.315, 0.9722, and 0.3037 for the image NSFW detector. Our results suggest two detectors are progressive classifiers. The lower adjusted recall of the prompt NSFW detector can be attributed to several potential factors, including the use of a fixed binary threshold and the potential discrepancy in the definition of NSFW prompts between the detector and our annotation process.

5.2.4 Organizing DIFFUSIONDB

We organize DIFFUSIONDB using a flexible file structure. We first give each image a unique filename using Universally Unique Identifier (UUID, Version 4) [232]. Then, we organize images into 14,000 sub-folders—each includes 1,000 images. Each sub-folder also includes a JSON file that contains 1,000 key-value pairs mapping an image name to its metadata. An example of this image-prompt pair can be seen in Fig. 5.2. This modular file structure enables researchers to flexibly use a subset of DIFFUSIONDB.

We create a metadata table in Apache Parquet format [233] with 13 columns: unique image name, image path, prompt, seed, CFG scale, sampler, width, height, username hash, timestamp, image NSFW score, and prompt NSFW score. We store the table in a column-based format for efficient querying of individual columns.

5.2.5 Distributing DIFFUSIONDB

We distribute DIFFUSIONDB by bundling each image sub-folder as a Zip file. We collect Discord usernames of image creators (§ 5.2.2), but only include their SHA256 hashes in the distribution—as some prompts may include sensitive information, and explicitly linking them to their creators can cause harm. We host our dataset on a publicly accessible repository² under a CC0 1.0 license. We provide scripts that allow users to download and load DIFFUSIONDB by writing two lines of code. We discuss the limitations in § 5.5. To mitigate the potential harms, we provide a form for people to report harmful content for removal. Image creators can also use this form to remove their images.

5.3 Data Analysis

To gain a comprehensive understanding of the dataset, we analyze it from different perspectives. We examine prompt length (§ 5.3.1), language (§ 5.3.2), characteristics of both prompts (§ 5.3.3) and images (§ 5.3.4). We conduct an error analysis on misaligned prompt-image pairs (§ 5.3.5) and provide empirical evidence of potentially harmful uses of image generative models (§ 5.3.6).

²Public dataset repository: <https://huggingface.co/datasets/poloclub/diffusiondb>

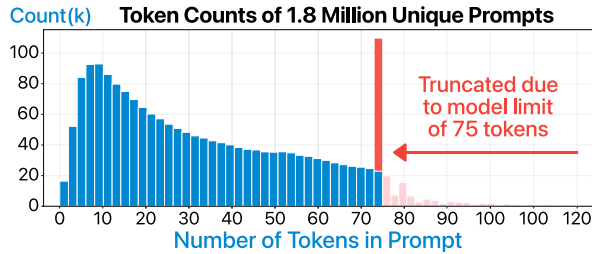


Figure 5.4: The distribution of token counts for all 1.8 million unique prompts in DIFFUSIONDB. It is worth noting that Stable Diffusion truncates prompts at 75 tokens.

5.3.1 Prompt Length

We collect prompts from Discord, where users can submit one prompt to generate multiple images and experiment with different hyperparameters. Our dataset contains 1,819,808 unique prompts. We tokenize prompts using the same tokenizer as used in Stable Diffusion [234]. This tokenizer truncates tokenized prompts at 75 tokens, excluding special tokens `<|startoftext|>` and `<|endoftext|>`. We measure the length of prompts by their tokenized length. The prompt length distribution (Fig. 5.4) indicates that shorter prompts (e.g., around 6 to 12 tokens) are the most popular. The spike at 75 suggests many users submitted prompts longer than the model’s limit, highlighting the need for user interfaces guiding users to write prompts within the token limit.

5.3.2 Prompt Language

We use a pre-trained language detector [235] to identify the languages used in prompts. 98.3% of the unique prompts in our dataset are written in English. However, we also find a large number of non-English languages, with the top four being German (5.2k unique prompts), French (4.6k), Italian (3.2k), and Spanish (3k). The language detector identifies 34 languages with at least 100 unique prompts in total. Stable Diffusion is trained on LAION-2B(en) [231] that primarily includes images with English descriptions, thus our findings suggest that expanding the training data’s language coverage to improve the user experience for non-English communities.

5.3.3 Characterizing Prompts

In this section, we explore the characteristics of prompts in DIFFUSIONDB. We examine the syntactic (§ 5.3.3.1) and semantic (§ 5.3.3.2) features of prompt text via interactive data visualizations. Lastly, We discuss the implications of our findings and suggest future research directions.

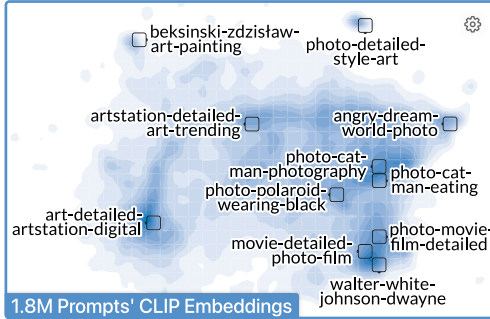


Figure 5.6: An interactive plot of 1.8M prompts’ CLIP embeddings, created with UMAP and kernel density estimation. Text labels show the top keywords of prompts in a grid tile. It reveals popular prompt topics.

to explore the distribution and relationships between different phrases (Fig. 5.5). Circle packing [237] is a technique to visualize hierarchical data, and each phrase is represented as a circle whose size encodes the phrase’s frequency in the dataset. We position sibling noun phrases (e.g., phrases sharing the same NP root) inside their parent phrase’s circle through a front-chain packing algorithm [237]. Viewers can hover over a circle to see the corresponding phrase and its frequency. Viewers can click a circle (Fig. 5.5A) to zoom into that sub-tree to see more details about a phrase (Fig. 5.5-B1) or a sub-phrase (Fig. 5.5-B2).

Insights and implications. Our interactive visualization reveals that key phrases such as “highly detailed,” “intricate,” and “greg rutkowski” are commonly used in prompts (Fig. 5.5A). The hierarchical visualization also surfaces popular image styles specified by users, such “digital painting,” “oil painting,” and “portrait painting” for painting styles (Fig. 5.5-B1) and “studio lighting,” “volumetric lighting”, and “atmospheric lighting” for lighting. These phrases can be unfamiliar to Stable Diffusion users, especially beginners, which highlights the importance of helping users develop prompting vocabularies. Researchers can leverage DIFFUSIONDB and our visualization to design tutorials and user interfaces that integrate exemplar prompts to guide users in describing their desired images.

5.3.3.2 Prompt Semantic Features

In addition to analyzing the syntactic characteristics of prompts, we also analyze their semantic features. We use a pre-trained CLIP model [27] to extract semantic features [216]. We use a frozen CLIP ViT-L/14 text encoder (the same model used in Stable Diffusion) to convert prompts into 768-dimension vectors.

Visualizing Prompt Embeddings.

To study the distribution of prompts in high-dimensional space, we use UMAP [30] to project 768-dimensional vectors into 2-D vectors for easy visualization. UMAP is a popular dimensionality reduction technique that is better at preserving the global structure of data

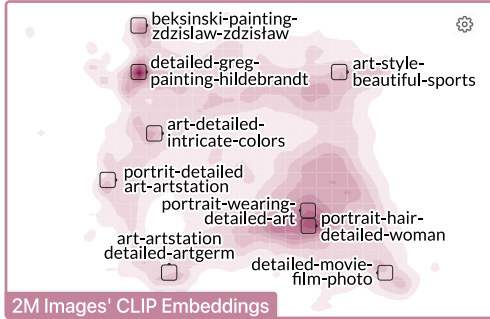


Figure 5.7: CLIP embeddings of 2M randomly selected images, with text labels being keywords of prompts in the grid tiles. It shows images have a different embedding distribution from prompts.

and more scalable to large datasets compared to t-SNE [31] and PCA [238]. We use grid search to fine-tune hyperparameters `n_neighbors` (60) and `min_dist` (0.1) so that prompts are more spread out in a 2-D space. We develop an interactive visualization tool⁴ to explore prompts’ semantic embeddings (Fig. 5.6). We use Kernel Density Estimation (KDE) [198] with a standard multivariate Gaussian kernel and Silverman bandwidth [199] to estimate the distribution of prompts’ UMAP representations. Then, we visualize the estimated distribution as a contour plot. To summarize prompts that are in the same region, we create four grids with varying granularity and pre-compute keywords for each grid tile, by treating all prompts in the tile as a document and selecting the top 4 keywords with the highest TF-IDF scores.

Interactions. Our visualization shows keywords of tiles that are close to high-density regions and prompt clusters by default. Viewers can hover over a tile to see its keywords, pan and zoom in to see more details of specific regions, and click a button to display each prompt as a small dot that viewers can hover over to read its prompt text.

Insights and implications. Our semantic embedding visualization (Fig. 5.6) highlights two popular prompt categories: art-related prompts (left in the plot) and photography-related prompts (dark blue regions on the right). These two groups appear distant from each other in the UMAP space, suggesting that the prompts for art and photography typically have distinct semantic representations. Interestingly, photography prompts appear to contain two clusters: one for non-human objects (top right) and another for celebrities (bottom right). Small prompt clusters outside the central area often feature artist names. Our findings suggest that future researchers can leverage the prompt usage distribution to fine-tune generative models to tailor to specific popular prompt categories.

5.3.4 Characterizing Images

We visualize⁵ the CLIP embedding distribution of 2 million unique image instances randomly sampled from DIFFUSIONDB (Fig. 5.7) by defining the unique key as the combination of

⁴Prompt embedding visualization: <https://poloclub.github.io/diffusiondb/explorer/#prompt-embedding>

⁵Image embedding visualization: <https://poloclub.github.io/diffusiondb/explorer/#image-embedding>



Figure 5.8: Example generated image that is semantically different from its prompt.

the image’s prompt and hyperparameters CFG scale, step, size, and seed. We use the UMAP model that was previously trained on the prompt embeddings to project the image embeddings into the same 2-D space. Finally, we apply the same method we used for our prompt embedding visualization (§ 5.3.3.2) to generate a contour plot and grid label overlays.

Insights and implications. Our image embedding visualization reveals that generated images have a different distribution from their prompts in the CLIP embedding space. For example, the “movie” cluster in the prompt embedding has been replaced by the “portrait” cluster in the image embedding. This suggests the semantic representations of prompts and their generated images may not be perfectly aligned. One hypothesis is that large image generative models face limitations when generating photorealistic human faces [213], and therefore some images generated with movie-related prompts appear to be closer to art and portrait regions in the embedding space.

5.3.5 Stable Diffusion Error Analysis

We leverage DIFFUSIONDB to discover Stable Diffusion generation failure cases and examine potential causes. To surface poor image generations, we compute CLIP embeddings for all prompts and images in DIFFUSIONDB. We then select prompt-image pairs with a large cosine distance (d) between their embeddings. The cosine distances have a normal distribution ($\mathcal{N}(0.7123, 0.0413^2)$). In this analysis, we focus on 13,411 “bad” prompt-image pairs (1) with a distance that is larger than 4 standard deviations from the mean and (2) the image was not blurred by Stable Diffusion (§ 5.2.3).

Impacts of hyperparameters. We conduct a logistic regression test to analyze the relationship between Stable Diffusion hyperparameter values (e.g., CFG scale, step, width, and height) and the likelihood of generating an image that is semantically different from its prompt. The results reveal that all four hyperparameters are negatively correlated with the likelihood of generating a bad image. The correlation is statistically significant with a p -value of less than 0.0001 for all four variables. Furthermore, we find the distribution of selected sampler options when generating bad images is significantly different from the overall distribution ($X^2 = 40873.11, p < 0.0001$).

CFG scale controls how much the generated image looks like the prompt. We find some users specify negative CFG scales that make images look different from their prompts (large cosine distance d). In the example shown in Fig. 5.8, a user generates an image using

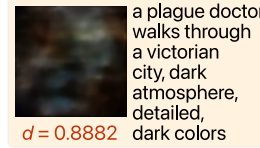


Figure 5.9: Example generated image that is semantically different from its prompt.

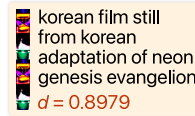


Figure 5.10: Example generated image that is semantically different from its prompt.

a prompt about “superman” with all default hyperparameters values, except for setting CFG scale to -1 . This results in an image featuring a bowl of soup instead of “superman”.

A small step could also generate under-developed images that look different from the specified prompts. As demonstrated in the example in Fig. 5.9, a user generates an image about “plague doctor” with all default hyperparameter values, except for setting step to 2, which leads to a blurry image.

Stable Diffusion struggles with generating images with a small size or large aspect ratios. The dissimilar image shown in Fig. 5.10 is generated with default hyperparameters except for a size of (64, 512).

Impacts of prompts. Despite controlling all hyperparameters to be close to default values, we still find 1.1k unique bad image-prompt pairs. Most of these instances have non-English prompts, very short prompts, or prompts consisting primarily emojis (see Fig. 5.11). The token lengths of these instances are significantly lower than the overall token length (one-tailed $t = -23.7203$, $p < 0.0001$). The English prompt frequency among these instances is also significantly lower than the overall frequency ($X^2 = 1024.56$, $p < 0.0001$). Interestingly, we also find that Stable Diffusion sometimes generates unexpected images even when prompts are meaningful English sentences. Future researchers can use our error analysis and failure cases to check potentially mislabeled training data.

Implications. Our study reveals Stable Diffusion can make mistakes when generating images with certain hyperparameter values or prompt styles. Negative CFG scales, small steps, or small sizes contributes to generating images dissimilar to prompts. Short and

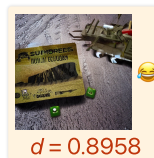


Figure 5.11: Example generated image that is semantically different from its prompt.

non-English prompts can also lead to errors. To improve the quality of generative models, researchers can expand the training data to cover these edge cases. There are opportunities for researchers to design user interfaces that can help users understand the impact of different hyperparameters and guide them in choosing values that fit their specific use cases.

5.3.6 Potentially Harmful Uses

To identify potentially malicious uses of Stable Diffusion, we use named entity recognition to analyze prompts. We find that many prompts include names of influential politicians, such as over 65k images generated with a prompt including “Donald Trump” and over 48k images with “Joe Biden.” Some prompts portray these politicians in negative lights, ranging from depicting them “as Gollum with hair” to “arrested in handcuffs.” Additionally, we find female celebrities are frequently used in prompts, with a high frequency after artists and influential politicians. Some of these prompts are presented in a sexual context that could be considered nonconsensual pornography.

Through keyword search, we discover prompts generating misinformation that could cause harm. For example, the prompt “scientists putting microchips into a vaccine” may harm public trust in medical institutions by potentially validating conspiracy theories. Similarly, the prompt “Russian soldiers in gas masks found the last surviving ukrainian after a nuclear war to liberate ukraine” depicts false images of the Russo-Ukrainian War and could lead to new forms of propaganda. Our findings highlight the crucial need for further research on the broader impacts of large generative models and ways to regulate and mitigate their harms.

5.4 Enabling New Research Directions

The unprecedented scale and diversity of DIFFUSIONDB bring new exciting research opportunities to help users generate images more effectively and efficiently, and enable researchers to improve, explain, and safeguard generative models.

Prompt Autocomplete. With DIFFUSIONDB, researchers can develop an autocomplete system to help users construct prompts. For example, one can use the prompt corpus to train an n -gram model to predict likely words following a prompt part. Alternatively, researchers can use *semantic autocomplete* [239] by categorizing prompt keywords into ontological categories such as subject, style, quality, repetition, and magic terms [49]. This allows the system to suggest related keywords from unspecified categories, for example suggesting style keyword “depth of field” and a magic keyword “award-winning” to improve the quality of generated images. Additionally, researchers can also use DIFFUSIONDB to study prompt *auto-replace* by distilling effective prompt patterns and creating a “translation” model that replaces weaker prompt keywords with more effective ones.

Generation through Search. As DIFFUSIONDB contains 14 million images, this dataset might have already included images with a user’s desired effects. Thus, a user can quickly search images in DIFFUSIONDB instead of running Stable Diffusion, which can be slow and costly. Lexica [51], an AI start-up, provides such a search engine, where users can search Stable Diffusion images by natural language or images. Researchers can also construct a structured index of images and prompts, such as building a *semantivisual image hierarchy* of images [240] or a *hierarchical topic model* of prompts [241], to help users easily discover and explore images and prompts with similar styles.

Improving Generative Models. With DIFFUSIONDB, a large and diverse collection of Stable Diffusion usage logs, researchers not only can identify weak points and failure modes of Stable Diffusion but also gain insights into user preferences. For example, we demonstrate that researchers can use joint text-image embeddings between prompts and images to detect generation misalignments (§ 5.3.5). Additionally, DIFFUSIONDB provides important metadata such as `username`, `hash` and `timestamp` for each generated image. By analyzing these metadata fields, researchers can trace the evolution chain of prompts, parameters, and images, which offers valuable insights into how users develop mental models of large generative models and their preferences of generated images. This understanding can inform future researchers to enhance generative models and design interfaces that facilitate better image-generation experiences.

Explainable Generation. As generative models have been gaining immense popularity, there is a call for explainable creativity [242]. Many explanation techniques use input permutation that computes feature attribution scores by running a model on slightly-modified input values [243]. DIFFUSIONDB contains 14 million prompt-image pairs including similar prompts with minor differences, such as “a happy dog” and “a sad dog”, allowing researchers to investigate how individual keywords affect the generation process.

Deepfake Detection. Breakthroughs in generative models raise concerns about deepfakes—fake images of real individuals for unethical purposes [244]. DIFFUSIONDB is valuable for detecting deepfakes, as it contains a large-scale collection of model-generated images and their metadata. Researchers can use this collection to train ML models to identify synthetic artifacts and train classifiers that classify synthetic images from real images [245].

5.5 Limitations

We discuss the limitations of our work: inclusion of unsafe content, potential biases, a limited measure of image quality and generalizability to different generative models.

- **Inclusion of unsafe images and prompts.** We collect images and their prompts from the Stable Diffusion Discord server (§ 5.2). The Discord server has rules against users generating or sharing harmful or NSFW (not suitable for work, such as sexual and violent

content) images. The Stable Diffusion model used in the server also has an NSFW filter that blurs the generated images if it detects NSFW content. However, we observe that DIFFUSIONDB includes some NSFW images that were not detected by the NSFW filter or removed by the server moderators. To mitigate the potential harm, we compute and share the likelihood of an image or a prompt containing unsafe content using the state-of-the-art NSFW detectors (§ 5.2.3). In addition, we provide a Google Form on the DIFFUSIONDB website where users can report harmful or inappropriate images and prompts. We will closely monitor this form and remove reported images and prompts from DIFFUSIONDB.

- **Potential biases of the data source.** The 14 million images in DIFFUSIONDB have diverse styles and categories. However, Discord can be a biased data source. Our images come from channels where early users could use a bot to use Stable Diffusion before release. As these users had started using Stable Diffusion before the model was public, we hypothesize that they are AI art enthusiasts and are likely to have experience with other text-to-image generative models. Therefore, the prompting style in DIFFUSIONDB might not represent novice users. Similarly, the prompts in DIFFUSIONDB might not generalize to domains that require specific knowledge, such as medical images [219].
- **Limited measure of image quality.** We use joint text-image CLIP embeddings between prompts and images to detect generation misalignments (§ 5.3.5). While the CLIP embedding distance can indicate the degree of alignment between the prompts and generated images, it does not provide a measure of the overall image quality. When constructing our dataset, we have considered including image properties such as entropy, variance, and the most common colors to help users gauge image qualities. However, these metrics do not provide a good measure of the overall image quality as well. To better measure image quality, future researchers can recruit annotators to rate images in DIFFUSIONDB.
- **Generalizability.** Previous research has shown a prompt that works well on one generative model might not give the optimal result when used in other models [213]. Therefore, different models can require users to write different prompts. For example, many Stable Diffusion prompts use commas to separate keywords, while this pattern is less seen in prompts for DALL-E 2 [216] or Midjourney [246]. Thus, we caution researchers that some findings from DIFFUSIONDB might not be generalizable to other text-to-image generative models.

5.6 Conclusion

We present DIFFUSIONDB, the first large-scale text-to-image prompt dataset, containing 14 million images with their prompts and hyperparameters collected from the Stable Diffusion discord server. We release the dataset with a CC0 1.0 license and open source all collection and analysis code, broadening the public’s access to cutting-edge AI technologies. We discuss findings on prompt and image patterns. We hope our work will serve as a cornerstone for the future development of large generative modes and tools that help users use these modes.

Part II


GUIDE AI WITH HUMAN VALUES

Overview

My research first explains AI to everyone, including AI non-experts, experts, and policymakers (Part I). However, gaining a better understanding of AI is not enough. To harness the full potential of AI and prevent potential harms of AI technologies, it is more important to translate our understanding of AI into *actions* that align AI models' behaviors with human knowledge and values.

Many AI users, including physicians and domain experts, are not AI experts. To empower these diverse stakeholders to exercise their human agency and easily guide AI models, we explore interactive interfaces that do not require programming. We first describe **GAM CHANGER (Chapter 6)**, a novel interactive visualization tool that enables AI practitioners and domain experts to easily and responsibly modify the behaviors of generalized additive models (GAMs) through model editing. GAMs are a popular model class among the data science community, being famous for their high intelligibility and accuracy. As modifications of high-stake models have serious consequences, GAM CHANGER promotes responsible editing by providing users with continuous feedback about the impacts of their edits. GAM CHANGER has been deployed in Microsoft and integrated into their interpretability library. Our tool also supports transparent and reversible model modifications. This chapter is adapted from work that was published and appeared at KDD 2022 [247].

Chapter 6

Interpretability, Then What? Editing Machine Learning Models to Reflect Human Knowledge and Values. Zijie J. Wang, Alex Kale, Harsha Nori, Peter Stella, Mark E. Nunnally, Duen Horng Chau, Mihaela Vorvoreanu, Jennifer Wortman Vaughan, and Rich Caruana. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*, 2022. 

In addition to *upstream* stakeholders such as AI practitioners and domain experts, we also help *downstream* stakeholders such as those impacted by AI-powered decision-making systems to exercise their human agency in guiding AI models. To help people alter unfavorable predictions, we introduce **GAM COACH (Chapter 7)**. Take AI-powered loan application approval as an example, a recourse suggestion can be “*decrease the loan amount by \$800, and you will get a loan approval.*” With a novel adaptation of integer linear programming, GAM COACH enables rejected loan applicants to interactively generate and customize diverse recourse plans that respect their preferences. An online user study with 48 participants reveals that people prefer customizable recourse plans. This chapter is adapted from work published and appeared at CHI 2023 [248].

Chapter 7

GAM COACH: Towards Interactive and User-centered Algorithmic Recourse.

Zijie J. Wang, Jennifer Wortman Vaughan, Rich Caruana, and Duen Horng Chau.

Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems,

2023.  PDF

CHAPTER 6

GAM CHANGER: ALIGN AI MODELS THROUGH MODEL EDITING

Researchers have made great efforts to make AI models interpretable [7, 155]. Interpretability reveals AI models can learn dangerous patterns from the data and rely on these patterns to make predictions, such as healthcare models predicting asthmatic patients have a lower risk of dying from pneumonia [11]. One explanation is that asthmatic patients receive care earlier, leading to better outcomes in the training data. However, if we use these flawed models to make hospital admission decisions, asthmatic patients are likely to miss out on the care they need. Interpretability helps us identify these dangerous patterns, but how can we take a step further and use explanations to align models with our knowledge and values? To help AI practitioners and domain experts improve AI models with model explanations, we design and develop GAM CHANGER, a novel interactive tool that enables users to fix problematic behaviors in their AI models through model editing. Physicians have started to use our tool to investigate and fix pneumonia and sepsis risk prediction models, and an evaluation with 7 data scientists working in diverse domains highlights that our tool is easy to use, meets their model editing needs, and fits into their current workflows.

6.1 Introduction

It is crucial to understand how machine learning (ML) models make predictions in high-stakes settings, such as finance, criminal justice, and healthcare (Fig. 6.1A). Recently, researchers have made substantial efforts to make ML models interpretable [e.g., 249, 243, 11], but there is not much research focused on how to *act on* model interpretations. In practice, data scientists and domain experts often compare model interpretations with their knowledge [8]. If a model uses expected patterns to make predictions, they feel more confident to deploy it. Interpretability can also uncover hidden relationships in the data—helping

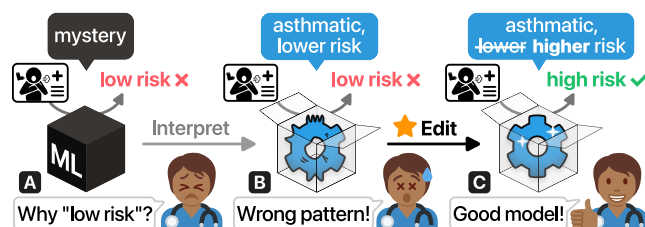


Figure 6.1: (A) Domain experts such as physicians often hesitate to trust ML models as they cannot understand how the models make predictions. (B) Interpretability reveals models can learn potentially harmful patterns. (C) Model editing turns interpretability into action—enabling domain experts to align model behaviors with their knowledge and values.

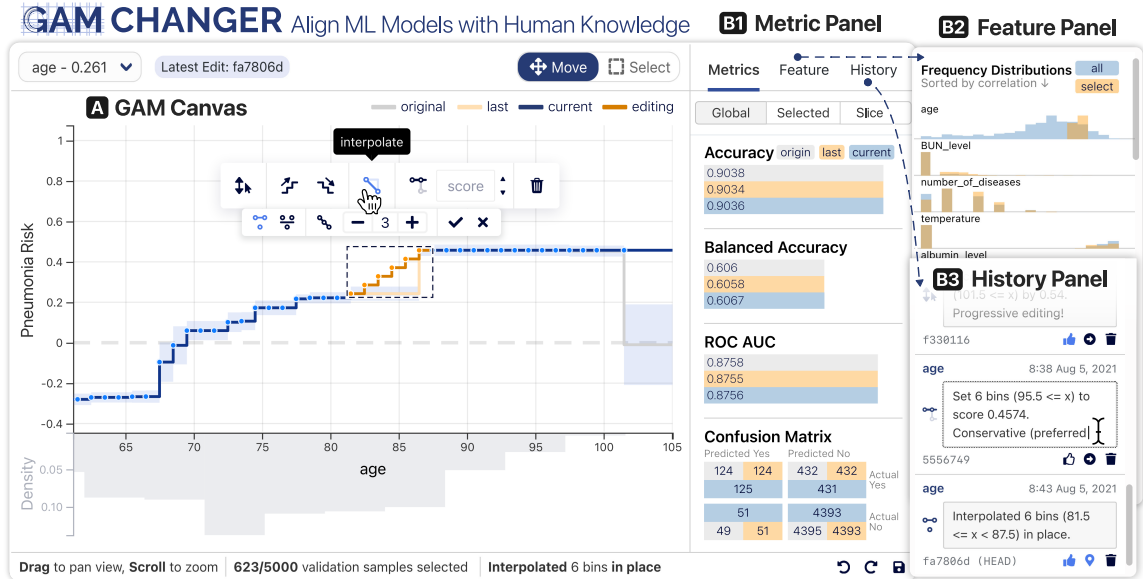


Figure 6.2: GAM CHANGER empowers domain experts and data scientists to easily and responsibly align model behaviors with their knowledge and values, via direct manipulation of GAM model weights. Take a healthcare model for example. (A) The *GAM Canvas* enables physicians to interpolate the predicted risk of dying from pneumonia to match their clinical knowledge of a gradual risk increase from age 81 to age 87. (B1) The *Metric Panel* provides real-time feedback on model performance. (B2) The *Feature Panel* helps users identify characteristics of affected samples and promotes awareness of fairness issues. (B3) The *History Panel* allows users to compare and revert changes, as well as document their motivations and editing contexts.

users gain insights into the problems they want to tackle.

Other times, however, ML interpretability reveals that models learn dangerous patterns from the data and rely on these patterns to make predictions. These patterns might accurately reflect real phenomena, but leaving them untouched can cause serious harm in deployment. For example, with interpretability, KDD researchers [11, 250] found healthcare models predict that having asthma lowers a patient’s risk of dying from pneumonia (Fig. 6.1B). Researchers suspect this is because asthmatic patients would receive care earlier, leading to better outcomes in the training data. If we use these flawed models to make hospital admission decisions, asthmatic patients are likely to miss out on care they need. Interpretability helps us identify these dangerous patterns, but how can we take a step further and use model explanations to *improve* (Fig. 6.1C) ML models?

To answer this question, our research team—consisting of ML and human-computer interaction researchers, physicians, and data scientists—presents **GAM CHANGER** (Fig. 6.2): the first interactive system to empower domain experts and data scientists to easily and responsibly edit the weights of *generalized additive models* (GAMs) [12, 251, 252], a state-of-the-art interpretable model [253]. Model editing is already common practice for regulatory compliance (§ 6.4.2.1). We aim to tackle two critical challenges to make model editing more accessible and responsible:

Challenge 1: Enable domain experts to vet and fix models. Editing model weights to align model behavior with domain knowledge has been discussed in the KDD community [11]. It requires the “editors” to have expertise in ML engineering and write code to adjust specific weights until the model behaves as expected. However, domain experts who have less experience in ML engineering, such as physicians and legal experts, play a critical role in creating trustworthy models [8]. To democratize model editing, we develop easy-to-use and flexible user interfaces that support a wide range of editing methods—enabling stakeholders with diverse backgrounds to easily investigate and improve ML models.

Challenge 2: Promote accountable model modifications. Accessible model editing helps users exercise their human agency but demands caution, as modifications of high-stake models have serious consequences. For example, if a user only monitors edits’ effects on a metric like overall accuracy, their edits might have unfavorable effects on underrepresented groups [254]. To guard against harmful edits, we provide users with continuous feedback about impacts on different subgroups and feature correlations. We also support transparent and reversible model modifications.

Contributions & Impacts. GAM CHANGER has already begun to help users improve their models. Our major contributions include:

- **GAM CHANGER, the first interactive system** that empowers domain experts and data scientists to edit GAMs to align model behaviors with their knowledge and values. Through a participatory and iterative design process with physicians and data scientists, we adapt easy-to-use *direct manipulation* [255] interfaces to edit complex ML models. Guarding against harmful edits is our priority: we employ *continuous feedback* and *reversible actions* to elucidate editing effects and promote accountable edits (§ 6.2).
- **Impacts to physicians: GAM CHANGER in action.** Physicians have started to use our tool to vet and fix healthcare ML models. We present two examples where physicians on our team applied GAM CHANGER to align pneumonia and sepsis risk predictions with their clinical knowledge. The edited sepsis risk prediction model will be adapted for use in a large hospital (§ 6.3).
- **Impacts to data scientists: beyond healthcare.** To investigate how our tool will help ML practitioners, we further evaluate it via a user study with 7 data scientists in finance, healthcare, and media. Our study suggests GAM CHANGER is easy to understand, fits into practitioners’ workflow, and is especially enjoyable to use. We also find model editing via feature engineering and parameter tuning is a common practice for regulatory compliance. Reflecting on our study, we derive lessons and future directions for model editing and interpretability tools (§ 6.4, § 6.5).
- **An open-source,¹ web-based implementation** that broadens people’s access to creating more accountable ML models and exercising their human agency in a world penetrated

¹GAM CHANGER code: <https://github.com/interpretml/gam-changer>

by ML systems. We develop GAM CHANGER with modern web technologies such as WebAssembly.² Therefore, anyone can access our tool directly in their web browser or computational notebooks and edit ML models with their own datasets at scale (§ 6.2.3). For a demo video of GAM CHANGER, visit <https://youtu.be/D6whtfInqTc>.

We hope our work helps emphasize the importance of human agency in responsible ML research, and inspires future work in human-AI interaction and actionable ML interpretability.

6.2 Novel User Experience

To lower barriers to controlling ML model behavior (Challenge 1), GAM CHANGER (Fig. 6.2) adapts easy-to-use direct manipulation interface to edit the parameters of GAMs with a variety of editing tools (§ 6.2.1). To promote responsible edits (Challenge 2), our tool provides real-time feedback; all edits are reversible, and users can document and compare their edits (§ 6.2.2). Built with modern web technologies, our tool is accessible (§ 6.2.3).

6.2.1 Intuitive and Flexible Editing

The *GAM Canvas* (Fig. 6.2A) is the main view of GAM CHANGER, where we visualize one **input feature** x_j 's contribution to the model's prediction by plotting its **shape function** $f_j(x_j)$. Users can select a drop-down to transition across features. GAMs usually discretize continuous variables into finite bins, so that shape functions can easily capture complex non-linear relationships. Thus, the output of shape functions is a continuous piecewise constant function, where we use a dot to show the start of each bin and a line to encode the bin's constant score (Fig. 6.2A). For categorical features, we represent each bin as a bar whose height encodes the bin's score (Fig. 6.3B). Lines and bins are colored by editing status (e.g., original or edited).

Model direct manipulation. In the *GAM Canvas*, users can *zoom-and-pan* to control their viewpoint in the *move mode*, or use *marquee selection* to select a region of the shape function to edit in the *select mode* (Fig. 6.2A). Once a region is selected, the *Context Toolbar* appears: it affords a variety of editing tools represented as icon buttons. Clicking a button changes the shape function in the selected region. For example, the monotonicity tool \nearrow can transform the selected interval of the shape function into a monotonically increasing function. Internally, GAM CHANGER fits an isotonic regression [256] weighted by the bin counts to determine a monotone function with minimal changes. Other editing tools include interpolating \approx scores of selected bins, dragging \updownarrow to adjust scores, and aligning \approx scores to the most left or right bin.

GAM Canvas. In the *GAM Canvas* (Fig. 6.2A), users can inspect and direct manipulate shape functions. As GAMs support continuous and categorical features, as well as their

²WebAssembly: <https://webassembly.org>

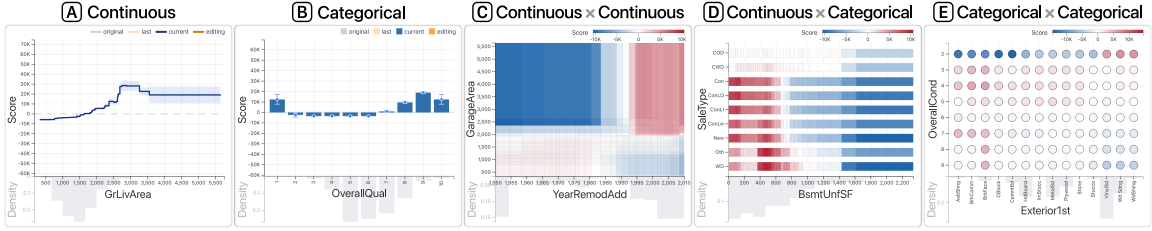


Figure 6.3: The *GAM Canvas* employs different designs to visualize shape functions on different feature types. We use **A** line charts for continuous variables, **B** bar charts for categorical variables, **C** heatmaps for interaction effects of two continuous variables, **D** vertical bar charts for interaction effects between continuous and categorical variables, and **E** scatter plots for interaction effect of two categorical variables. For univariate features, the x-axis encodes the input feature x_j , and the y-axis represents the output of the shape function $f_j(x_j)$. We also use light-blue bands and error bars to represent the prediction confidence. For pair-wise interactions, the axes encode two features, and we use a diverging color scale to represent the contribution scores.

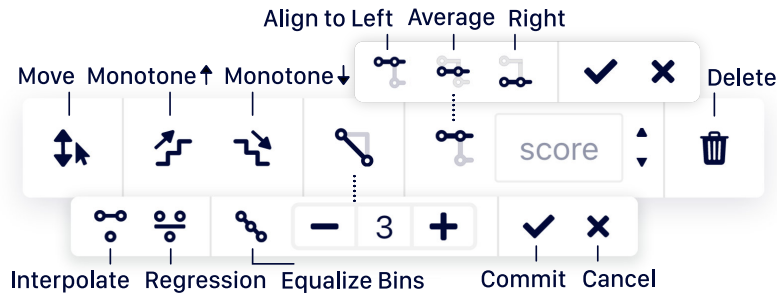


Figure 6.4: The *Context Toolbar* enables users to edit GAMs with a variety of editing tools. Users can use the move tool \updownarrow to adjust the contribution scores of selected bins by dragging bins up and down. Users can apply the interpolate tool ∞ to linearly interpolate the scores of an interval of bins from the start to the end. Alternatively, users can interpolate scores with an arbitrary number of equal bins ∞ , or by fitting a linear regression ∞ . With minimal changes, the monotonicity tool transforms the selected scores into a monotonically increasing function \nearrow or a monotonically decreasing function \searrow . With align tools, users can unify the selected scores as the score of the left bin \leftarrow , the right bin \rightarrow , or the average score weighted by the training sample counts ∞ .

two-way interactions, we design unique visualization for each variable type, featuring line chart, bar chart, heatmaps, and scatter plots (Fig. 6.3). Users can use the feature selection drop-down to transition across features. To begin, the *GAM Canvas* shows the feature with the highest importance score, computed as the weighted average of a feature’s absolute contribution scores. We re-center the contribution scores by adjusting the **intercept constant** β_0 (Equation 2.1) such that the mean prediction for each feature has a zero score across the training data. Thus, a positive score suggests the feature positively affects the prediction and vice versa. Consider a GAM trained to predict house prices (Fig. 6.3A), if the living area is larger than 2000 square feet, it increases the predicted house price, while areas lower than 2000 decrease the predicted value compared with average. We highlight the 0-baseline as a thick dashed line.

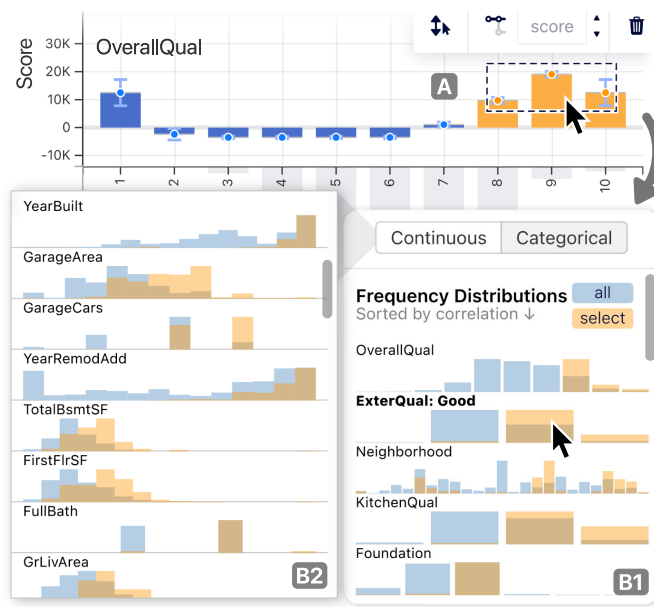


Figure 6.5: **A** On a GAM trained to predict house price, a user selects bins representing high-quality houses in the *GAM Canvas*. **B1** For categorical variables, the *Feature Panel* shows that selected houses disproportionately have better exterior and kitchen quality and locate in certain neighborhoods. **B2** For continuous variables, the year built and garage area are also highly correlated with the house quality.


Editing tools. In the *GAM Canvas*, users can switch between *move mode* and *select mode* by clicking the mode toggle button. In the *move mode*, users can use *zoom-and-pan* to control their view portion and focus on analyzing an interesting region in the GAM visualization. In the *select mode*, users can use *marquee selection* to pick a subset of bins (or bars for categorical features) to edit. Once a region of the shape function is selected, the *Context Toolbar* (Fig. 6.4) appears. In the bottom *Status Bar*, users can view the number of samples in the selected region and a description of their last edit. Users can click the check icon ✓ to “commit” (§ 6.2.2) the change if they are satisfied with this edit, or click the cross icon ✕ to discard the change.



Feature Panel. The *Feature Panel* (Fig. 6.2-B2, Fig. 6.5) helps users gain an overview of correlated features as well as their distributions and elucidate potential editing effect disparities. We develop *linking+reordering*—a novel technique to identify correlated features. Once a user selects an interval of the shape function in the *GAM Canvas* (Fig. 6.5-A), we look up affected samples and their associated bins across all features. For each feature, we compare the bin count frequency in **all training data** and the frequency in the **selected samples** by computing the ℓ_2 distance between these two frequency vectors. Then, we plot two frequency distributions in an overlaid histogram for each feature, and sort all histograms in descending order of the distance scores (Fig. 6.5-B). The intuition is that if two features x_1 and x_2 are independent, then samples selected from an interval in x_1 should have a **distribution** similar to the **training data distribution** in x_2 , and vice versa. Therefore, correlated

features will be on top of the sorted histogram list. Our *linking+reordering* technique allows users to interactively and quickly identify local correlations across features, even between continuous and categorical features. By observing correlated features, users can identify potential disparities in editing impacts. For example, editing high-quality houses would disproportionately affect newer houses (Fig. 6.5).

Metric Panel. The *Metric Panel* (Fig. 6.2-B1) provides real-time and continuous feedback on the model performance. For a binary classifier, we present a confusion matrix and use bar plots to encode the model’s accuracy, balanced accuracy, and the Area Under the Curve (AUC). For a regressor, we report root mean squared error, mean absolute error, and mean absolute percentage error. We use the same color codes of shape functions in the *GAM Canvas* to describe the performance of original model, model from the last edit, and current model.

Besides monitoring global metrics that are computed on all validation samples, users can choose a subset of validation samples to compute the metrics by switching the metric scope. For example, with the *Selected Scope*, the *Metric Panel* only computes model metrics on samples that are in the currently selected region. With the *Slice Scope*, users can choose a data slice by selecting a level of a categorical variable, e.g., the `female` level of the `gender` variable. Then, performance metrics in the *Metric Panel* are computed on the validation samples that belong to the selected subgroup.

History Panel. GAM CHANGER users can undo and redo their edits by  clicking the buttons in the bottom *Status Bar* (shown on the right) or using keyboard shortcuts. Reversible actions promote accountable model editing, as users can easily fix their mistakes.

Inspired by the version control system Git³, the *History Panel* (Fig. 6.2-B3) tracks each edit as a commit: a snapshot of the underlying GAM. Each commit has a timestamp, a unique identifier, and a commit message. Once an edit is committed, we automatically generate an initial commit message to describe the edit; users can update the message in the *History Panel* to further document their editing motivation and context. Once users finish editing, they can click the Save button  in the *Status Bar* to save the latest GAM along with all editing history, which can be used for deployment or future continuing editing. Before saving the model, GAM CHANGER requires users to examine and confirm  all edits.

6.2.2 Safe and Responsible Editing

Guarding against harmful edits is our top priority. To begin using GAM CHANGER, users provide a trained GAM (i.e., model weights) and set of validation samples (a subset of the training data or separate validation set). The *Metric Panel* (Fig. 6.2-B1) provides real-time and continuous feedback on the model’s performance on the validation samples to help users identify the effects of their edits. During a user’s editing process, our tool efficiently

³Git: <https://git-scm.com>

recomputes performance metrics on the edited model. To probe if an edit is equitable across different subgroups, users can choose which subset of samples to measure performance on: the *Global Scope* for all samples, the *Selected Scope* for samples in the selected region, and the *Slice Scope* for samples having a specific categorical value (e.g., females).

Recognize impact disparities. The *Feature Panel* (Fig. 6.2-B2) helps users gain an overview of correlated features and elucidates potential disparities in the impact of edits. For example, it can alert users of the disproportionate impact of edits addressing elder patients on females as females live longer. We develop *linking+reordering*—a novel method to identify correlated features. Once a user selects a region in the *GAM Canvas*, we look up affected samples’ associated bins across all features. For each feature, we compute the ℓ_2 distance between the bin count frequency in all training data and the frequency in affected samples. By observing overlaid histograms sorted in descending order of the distance scores, users can inspect correlated features of affected samples and identify potential editing effect disparity.

Reversible and documented modifications. To promote safe model editing, GAM CHANGER allows users to undo and redo any edits. In addition, the *History Panel* (Fig. 6.2-B3) tracks all edits and displays each edit in a list. Inspired by the version control system Git, we save each edit as a commit—a snapshot of the underlying GAM weights. Each commit has a timestamp, a unique identifier, and a commit message. Therefore, users can easily explore model evolution by checking out ◉ a previous GAM version, discard ◐ modifications, and document edit contexts and motivations in commit messages. Once satisfied with their edits, a user can save the modified model with edit history for deployment or future continuing editing. To help users identify editing mistakes and promote accountable edits, GAM CHANGER requires users to examine and confirm ◑ all edits before saving the model.

6.2.3 Scalable, Open-source Implementation

GAM CHANGER is a web-based GAM editor that users can access with any web-browsers on laptops or tablets, or directly in computational notebooks. Our tool has been integrated into the popular ML interpretability ecosystem *InterpretML* [7]: users can easily *export* models to edit and *load* modified models. Using cutting-edge WebAssembly to accelerate in-browser model inference and isotonic regression fitting, our tool is scalable: all computations are real-time with up to 5k validation samples in Firefox on a MacBook, and the sample size is only bounded by the browser’s memory limit. We open source GAM CHANGER so that future researchers can easily generalize our design to other forms of model editing.

6.3 Impacts to physicians

GAM CHANGER in action. The early prototype [257] of our tool has received overwhelmingly positive feedback in two physician-focused workshops. In addition, physicians have

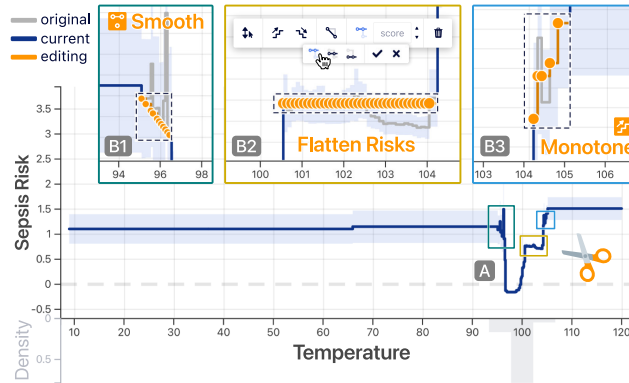


Figure 6.6: **A** A GAM learns a few strange patterns between patients’ temperature and sepsis risk that need to be fixed. **B1** We smooth out the sudden increase of risk around 96°F, **B2** flatten the risk to reflect a treatment effect, and **B3** smooth out risk fluctuations at high temperature.

begun to use our tool to interpret and edit medical models. We share examples in which two physicians in our research team apply GAM CHANGER to investigate and improve GAMs for sepsis (§ 6.3.1) and pneumonia (§ 6.3.2) risk predictions, editing the models to reflect their clinical knowledge and values such as safety. The edited sepsis risk prediction model will be adapted for use in a large hospital.

6.3.1 Fixing Sepsis Risk Prediction

A physician in our team trained a GAM with boosted-trees to predict if pediatric patients should receive sepsis treatments. This model exhibited many problematic patterns. In this section, we share our experience in applying GAM CHANGER to align this model’s behavior with the physician’s clinical knowledge and values.

The data comes from a large hospital; it includes 26,564 pediatric patients. There are 7 continuous features: `age`, `oxygen saturation`, `body temperature`, systolic and diastolic `blood pressure`, `heart rate`, and `respiratory rate`. The `blood pressure`, `heart rate`, and `respiratory rate` are normalized by taking the difference between the original value and the age-adjusted normal. The other 83 features are categorical with binary values, each indicating if a keyword—such as “pain,” “fever,” or “fall”—is present in the *chief complaint of patient*, a concise statement describing the symptom, diagnosis, and other reasons for a medical encounter. The target variable is binary: 1 if the patient received a treatment for sepsis and 0 if not. The model yields an AUC score of 0.865 on the test set (20% of all data). The physician loads GAM CHANGER in their browser with 5,000 random training samples; they share their computer screen with 3 other researchers in the team via video-conferencing software. All edits are made by the physician after discussing with other researchers on the call.

There is a plateau of risk scores from 100–104°F, with a small, but notable dip from 103–104°F. The presence of the plateau itself is physiologically plausible (due to antipyretic treat-

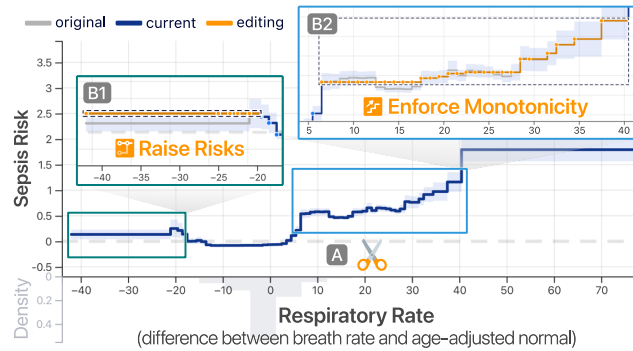


Figure 6.7: **A** Contrary to clinical knowledge, a GAM predicts sepsis risk decreases when the respiratory rate decreases (left), and the risk score fluctuates when the rate increases (right). We align the model behaviors by **B1** raising risk scores ↗ and **B2** removing risk fluctuations with monotonicity ↘.

ments), but the dip is hard to explain and suspicious, perhaps reflecting a treatment effect in which treatment is delayed outside of the model’s prediction window as physicians evaluate the child’s response to antipyretics. Because of a concern that this might artificially depress risk scores and encourage physicians to believe that children in this range are healthier than they really are, the risk curve in this region is flattened using the align tool ↗ (Fig. 6.6-B2).

The observation of many small dips of predicted risk scores around 104–105.5°F does not align with physiological knowledge. Therefore, we remove these dips by making the scores monotonically increasing ↘ in this region by fitting an isotonic regression model. The physician in our team thinks this edit is conservative and safe because it smooths out many dips in the region that might cause patients to lose necessary care. The physician comments *“Taking out unpredictable behaviors from a model to my mind is deeply safer. If this ends up being a life and death decision, and we go back, and we look that a kid died because he didn’t trigger the model by falling into one of those dips, then that is a catastrophe.”*

6.3.1.1 Editing the temperature feature.

The *GAM Canvas* first shows `temperature` (Fig. 6.6A) since this feature has the highest importance score, computed as the weighted average of a feature’s absolute contribution scores. The x-axis ranges from 10 to 120°F, where the low range is due to data errors. The y-axis encodes the predicted risk score (log odds) of dying from sepsis, ranging from -0.2 to 1.5. The shape function has a “U-shape”: the model predicts that patients with `temperature` lower and higher than the normal range (97–99°F) have a higher risk of sepsis. It matches clinical knowledge as fever (high `temperature`) and hypothermia (low `temperature` caused by cardiovascular collapse) are severe symptoms of sepsis. There is a peak of predicted risk when the `temperature` is around 96°F. However, there is no physiological reason that hypothermia with a `temperature` of 96°F has a higher risk than a `temperature` of 95°F. Therefore, we remove the risk peak at 96°F by linearly interpolating ↘ the risk scores from 95 to 96.5°F (Fig. 6.6-B1).

6.3.1.2 Editing the respiratory rate feature.

The `respiratory rate` feature measures the *difference* between the number of breaths taken in one minute and its age-adjusted normal. The “U-shape” in the *GAM Canvas* (Fig. 6.7A) suggests the model predicts that patients with high deviation from the normal respiratory rate range have a higher risk of sepsis, and higher `respiratory rate` yields a higher risk score than lower `respiratory rate`. This pattern matches the clinical knowledge. Interestingly, the center of the “U-shape” is around -5 instead of 0 . This also makes sense because the “normal range” of respiratory rate for adults is considered 12–20 times a minute, but healthy adults actually only take 12–15 breaths per minute. In other words, this left-shifted center indicates the model has learned a realistic distribution of respiratory rate.

The predicted risk decreases when `respiratory rate` is below -21 , for which there is no physiological explanation. We decide to remove this counterintuitive risk decrease by flattening \rightsquigarrow all scores below -21 (Fig. 6.7-B1). After this edit, we notice some fluctuations on the right side of the “U-shape.” Clinical knowledge suggests sepsis risk should only increase when `respiratory rate` increases for rates which are already above normal. To fix the counterintuitive pattern in the model, we make the risk scores monotonically increasing \rightsquigarrow for bins between 7 and 40 (Fig. 6.7-B2).

An alternative edit is to linearly interpolate \rightsquigarrow the scores of bins from 7 to 40. However, we prefer the former edit, because linear interpolation \rightsquigarrow would break the plateau of predicted risk when `respiratory rate` is between 8 and 28, which are values that are commonly associated with children suffering from mild to moderate obstructive lung pathologies such as bronchiolitis and asthma, neither of which are likely to require treatment for suspected sepsis. Removing this pattern might obscure a meaningful signal—there are many non-sepsis related reasons for moderately elevated respiratory rate. Compared to the linear interpolation tool \rightsquigarrow , the monotone increasing tool \rightsquigarrow is much less intrusive: it makes minimal changes to make the selected region monotone via isotonic regression.

6.3.1.3 Editing the systolic blood pressure feature.

The feature `blood pressure` measures the *difference* between the systolic blood pressure in millimeters of mercury and its age-adjusted normal. The *GAM Canvas* (Fig. 6.8A) shows that the model predicts patients with `blood pressure` from -25 to -10 to have a significantly higher risk of sepsis. Interestingly, the predicted risk score decreases when `blood pressure` decreases after peaking at -15 . The *GAM Canvas* shows only 19 patients out of 5000 patients with `blood pressure` below -20 , and 118 patients with `blood pressure` from -20 to -10 . Clinical knowledge suggests that when blood pressure readings move away from the typical range, both the odds of having a measurement artifact and the risk of sepsis increase. To create a safer model, we select all the bins below -15 and align \rightsquigarrow their risk score to the right (Fig. 6.8-B1). Although by doing so, we raise the predicted risk score of all bins below

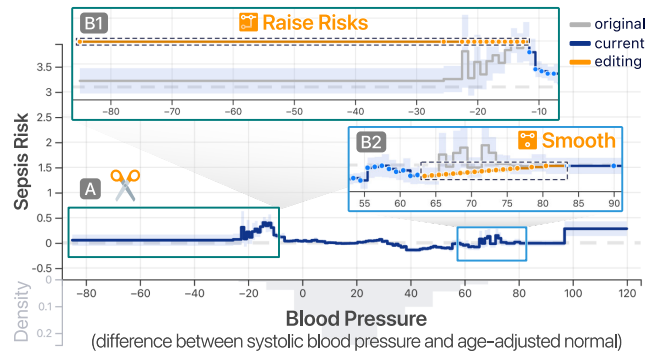


Figure 6.8: **A** Against physicians’ expectations, a GAM predicts that patients with lower blood pressure have lower sepsis risk (left), and the risk abruptly increases at high blood pressure (right). To create a safer model, **B1** we raise the risk scores ☹️, and **B1** smooth out the sudden risk increase ☹️.

-15 to 0.38, this is a conservative edit as we do not further increase the risk when `blood pressure` decreases after -15. Here `blood pressure` below -20 is most likely an error, and this edit might increase false-positive predictions on incorrect inputs. However, the physician prefers this model to predict data errors and outliers as high risk, because it is safer to have a high false-positive rate than to have a high false-negative rate when predicting sepsis risk. When editing healthcare models, physicians often consider the tradeoff between false-positive and false-negative rates, and the sweet spot for the tradeoff varies for different healthcare models (see § 6.5 for more discussion).

The risk score of sepsis fluctuates when systolic `blood pressure` is around 60–80. There is no physiological explanation for this fluctuation, so we smooth it out by linearly interpolating ☹️ these scores. Interestingly, there is a sudden increase in the predicted risk score when `blood pressure` is higher than 95, where these inputs are most likely errors. Therefore, we decide not to edit this increase because it is safer to have a high false-positive rate than to have a high false-negative rate on a sepsis risk prediction model.

6.3.2 Repairing Pneumonia Risk Prediction

KDD researchers [11] have identified problematic patterns in pneumonia risk prediction models and raised the possibility to fix these patterns via model editing. With GAM CHANGER, we operationalize this possibility by editing the same model [11] with a physician in our research team. This GAM is trained to predict a patient’s risk of dying from pneumonia. The dataset includes 14,199 pneumonia patients; it has 46 features: 19 are continuous and 27 are categorical. The outcome variable is binary: 1 if the patient died of pneumonia and 0 if they survived. The AUC score on the test set (30% of data) is 0.853. One ML researcher in our team loads GAM CHANGER in their browser with 5,000 random training samples; they share their computer screen with a physician and 2 other researchers in the team via video-conferencing software. All edits are made by the ML researcher after discussing with

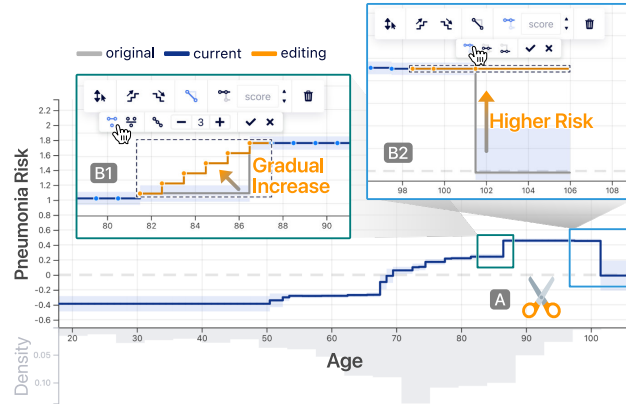





Figure 6.9: **A** Contrary to physicians’ knowledge, a GAM predicts an abrupt increase of risk from age 86 to 87 (left), and that patients above 100 years old have lower pneumonia risk than patients 20 years younger (right). **B1** With the interpolation tool , we smooth out the abrupt increase of risk. **B2** We use the align tool  to raise the risk score for older patients.

all people in the call.

6.3.2.1 Editing the age feature.

After loading GAM CHANGER, the *GAM Canvas* (Fig. 6.9A) first displays `age`, which has the highest importance score. The x-axis ranges from 18 to 106 years old. The y-axis encodes the predicted risk score (log odds) of dying from pneumonia. It ranges from a score of -0.4 for patients in their 20s to 0.5 for patients in their 90s. The model predicts that younger patients have a lower risk than older patients. However, the risk suddenly plunges when patients pass 100—leading to a similar risk score as if the patient is 30 years younger! It might be due to outliers in this `age` range, especially as this range has a small sample size, or patients who live this long might have “good genes” to recover from pneumonia.

To identify the true impact of `age` on pneumonia risk, additional causal experiments and analysis are needed. Without robust evidence that people over 100 are truly at lower risk, physicians fear that they would be injuring patients by depriving needy older people of care, and violating their primary obligation to *do no harm*. Therefore, physicians would like to fix this pattern. We apply a conservative remedy by setting  the risk of older patients to be equal to that of those slightly younger (Fig. 6.9-B2).

From the *Metric Panel*, we notice a drop of accuracy of 0.0004 in the *Global Scope*, and the confusion matrix in the *Selected Scope* shows that this edit causes the model to misclassify two negative cases as positives out of 28 patients who would be affected by the edit. To learn more about these patients, we observe the *Feature Panel*, which shows that `gender` is the second most correlated categorical feature with the selected `age` range. It means patients who are affected by this edit are disproportionately female—it makes sense because on average women live longer than men. Seeing the correlated features helps us be aware of

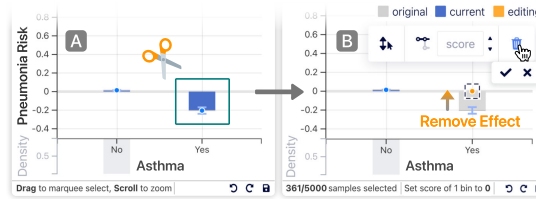


Figure 6.10: **A** A GAM predicts having asthma lowers the risk of dying from pneumonia. **B** We address this problematic pattern by removing the predictive effect of having asthma 🗑️.

potential fairness issues during model editing.

Besides the problematic drop of risk for older patients, the risk suddenly rises around 86 years old (Fig. 6.9A). After converting the risk score from log-odds to probability, the predicted likelihood of dying from pneumonia increases by 4.89% when the `age` goes from 86 to 87. This model behavior can cause 81–86 year-old patients to miss the care they need. To create a safer model, we apply the linear interpolation tool 🔄 in the region from `age` 81 to 87 to smooth out the sudden increase of pneumonia risk (Fig. 6.9-B1).

6.3.2.2 Editing the asthma feature.

The *GAM Canvas* (Fig. 6.10A) of the binary feature `asthma` suggests the model predicts asthmatic patients to have a lower risk of pneumonia than non-asthmatic patients. It could be because pneumonia patients with a history of asthma are likely to receive care earlier and receive more intensive care. However, if we use this model to make hospital admission decisions, this pattern might cause asthmatic patients to miss necessary care. Therefore, we remove 🗑️ the predictive effect of having `asthma` (Fig. 6.10B)—the new model would predict asthmatic patients to have an average risk. This is a conservative edit as one might argue asthmatic patients should have higher risk of pneumonia. Our edit is a step in the right direction, but further experiments are needed to see if we need further adjustments.

6.4 Impacts beyond healthcare

Evaluation with data scientists. We conducted a user study to further evaluate the usability and usefulness of GAM CHANGER, and also to investigate how data scientists would use our tool in practice. In the study, we chose a loan default prediction model in a lending scenario, because there is no specialized knowledge needed to interpret and potentially edit this model. The authors’ Institutional Review Board (IRB) has approved this study.

6.4.1 Study Design

Participants. The target users of GAM CHANGER are ML practitioners and domain experts who are familiar with GAM models. Therefore, we recruited 7 data scientists (P1–P7) for this

study by posting advertisements⁴ on the online issue board of a popular GAM Library [7]. The participation was voluntary, and we did not collect any personal information. All participants have developed GAMs for work: three participants use GAMs multiple times a week (P1, P5, P6), three use them a few times a month (P2, P3, P4), and one uses them about once a month (P7). Four participants work in finance (P1, P2, P3, P7), two work in healthcare (P4, P5), and one works in media (P6). Each study lasted about 1 hour, and we compensated each participant with a \$25 Amazon Gift card.

Procedure. We conducted the study with participants one-on-one through video-conferencing software. With permission from all participants, we recorded the video conference for subsequent analysis. After signing a consent form and a background questionnaire (e.g., familiarity with GAMs), each participant was given an 8-minute tutorial about GAM CHANGER. Participants then were pointed to a website consisting of GAM CHANGER with a model trained on the LendingClub dataset [258] to predict if a loan applicant can pay off the loan: the outcome variable is 1 if they can and 0 otherwise. Participants were given a list of recommended tasks to look for surprising patterns, edit 3 continuous features and 2 categorical features with different editing tools, experiment with different views, and freely explore the tool. Participants were told that the list was a guide to help them try out all features in the tool, and they were encouraged to freely edit the model as seen fit. Participants were asked to think aloud and share their computer screens with us. Each session ended with a usability survey and a semi-structured interview that asked participants about their experience of using GAM CHANGER and if this tool could fit their workflow and help them improve models in practice.

6.4.2 Benefits to Data Scientists

Below we summarize key findings from our observations and participants' feedback.

6.4.2.1 *Meet the pressing needs for model editing*

Through analyzing interviews and participants' verbalization of thoughts during the exploration task, we find there are critical needs for model editing in practice, and ML practitioners have already been editing their models with different methods. All participants have observed counterintuitive patterns when developing models in their work. For example, P6 recalled their GAM model, "*Some weights are negative, and I know by definition this cannot happen because [... of the nature of that feature].*" P7 commented "*[Strange patterns] happen a lot, mostly the direction of a certain variable. We expect the score to be increasing; however, the model shows something opposite.*"

Many participants were required to fix these strange patterns. P3 and P7 needed to remove counterintuitive patterns because of the *Adverse Action Notice Requirement*, a

⁴Participant recruitment: <https://github.com/interpretml/interpret/issues/283>

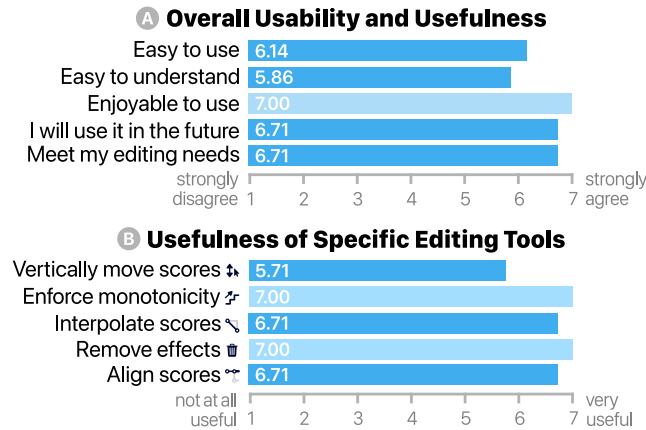


Figure 6.11: Average ratings from 7 participants for GAM CHANGER’s usability and usefulness. (A) All participants enjoyed using the tool; they found it highly usable and it meets their editing needs. (B) All features, especially enforcing monotonicity and removing effects, were rated favorably.

policy requiring lenders to provide explanations to loan applicants. If there are strange patterns, the model explanations sometimes will not make sense to loan applicants. P7 explained, “*Basically you want to make the model easier to explain in adverse action calls.*” Adverse action calls refer to situations when applicants dial in and demand real-time model explanations. On the other hand, P5 and P6 needed to edit their models on some well-understood features to align model behaviors with the expectations of knowledgeable stakeholders—physicians and business partners, respectively.

Improve and unify current editing approaches. Most participants reported using feature engineering to fix counterintuitive patterns in their own day-to-day work. For example, after discussing with domain experts, P5 removed features where they thought the shape functions were wrong or did not make sense. In P7’s case, a legal compliance team would decide which features to include and exclude after inspecting the model behaviors. P2 trained multiple models with different hyper-parameters and then chose models that not only had high accuracy but also learned expected trends. P1 had set up a sophisticated post-processing pipeline that would automatically smooth out shape functions, enforce monotonicity, and remove predictive effects on missing values. With interactivity and flexible tools, GAM CHANGER provided participants with direct control of their model behaviors and unify current editing approaches.

6.4.2.2 Usable and useful

The study survey included a series of 7-point Likert-scale questions regarding the usability and usefulness of GAM CHANGER (Fig. 6.11A). The results suggest that the tool is easy to use (average 6.14), easy to understand (average 5.86), and especially enjoyable to use (average 7.00—all participants gave the highest rating). Most participants would

like to use GAM CHANGER in their work to edit models. For example, P6 commented “*I have the dire hope that it will be a groundbreaking experience. [...] I strongly believe that this interactive model editing will please a lot of stakeholders, and increases trust and acceptance.*”

Versatile editing tools. We asked participants to rate specific editing tools in GAM CHANGER (Fig. 6.11B). All tools were rated favorably, and participants particularly liked the monotonicity tool \neq and deletion tool \boxminus (both received the highest rating from all participants). Monotonicity constraints are common across different domains, which might explain the high interest in the monotonicity tool. In particular, P4 appreciated that the monotonicity tool supported regional monotonicity: P4 gave an example from his work where the relationship between the `num of insurance claims` and people’s `age` was expected to form a “U-shape” (kids and seniors tend to have more insurance claims), and he would like to use our tool to enforce monotonicity with different directions $\neq \neq$ on the two ends of the shape function. Unlike the monotonicity tool, the deletion tool \boxminus had a much simpler functionality, and yet it was participants’ favorite. P7 liked the deletion tool because it was useful to edit categorical features, “*For missing values and neutral values [in categorical features], we don’t want to reward them, and we don’t want to punish them, so we usually just neutralize them [with the deletion tool].*” Participants’ overwhelmingly positive feedback provides evidence that GAM CHANGER is easy to use, and it can help practitioners improve their ML models through model editing.

6.4.2.3 Fit into model development workflows

Interviews with participants highlight that GAM CHANGER fits into data scientists’ workflows. Five participants used Jupyter notebooks to develop ML models, and they all appreciated that they could use GAM CHANGER directly in their notebooks. Many participants found the “git commit” style of editing history in the *History Panel* (§ 6.2.2) familiar and useful. When P6 wrote edit commit messages, they followed their company’s git commit style to include their name and commit type at the end of the message. In addition, P3 found the editing history and auto-generated messages helpful for their company’s model auditing process, “*I especially like the history panel where all the edits are tracked. You can technically use it as a reference when writing your model documentation [for auditors to review].*”

A platform for collaboration. Interestingly, many participants commented that besides model editing, GAM CHANGER would be a helpful tool to communicate and collaborate with different stakeholders. For example, P5’s work involved collaborating with physicians to interpret models, and they thought our tool would be a tangible tool to promote discussion about models: “*This work is very important because it lets people discuss about it [model behaviors].*” P1 had been building dashboards to explain models to their marketing teams, and they would like to use GAM CHANGER to facilitate the communication. Similarly,

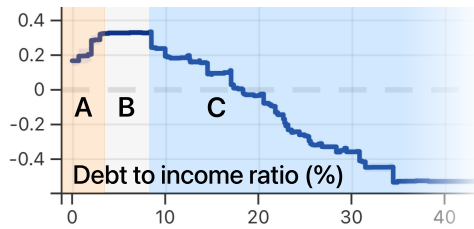


Figure 6.12: Shape function of debt to income ratio on the loan approval prediction.

P6 told us they would use our tool to communicate model insights to their stakeholders, including business partners, UX designers, and the sales team.

6.4.2.4 Diverse ways to use GAM CHANGER

Even with a relatively small sample size of 7 participants, we observed a wide spectrum of views regarding *when* and *how* to edit models. For example, P2 was more conservative about interactive model editing; they felt it was more “objective” to retrain the model until it learned expected patterns rather than manually modifying the model. P3 thought GAM CHANGER would be useful to enforce monotonicity and fix obvious errors, but they were more cautious and worried about irresponsible edits: “*Anyone behind the model can just add whatever relationship they want, rather than keep the model learn empirically whatever is in the data. I mean, it [the tool] is good, but you need to be diligent and make sure whatever changes you made make sense and are justifiable.*” On the other side of the spectrum, P5 and P6 found model editing with GAM CHANGER very natural as they had already been iterating on models with domain experts.

Multiple approaches. In addition to *whether* and *when* people should edit models, participants had different views on *how* to edit the model. For example, in the model used in this user study, debt to income ratio (dti) is a continuous feature (shown on the right): the log odds score (y-axis) of an applicant paying off their loan first increases when dti (x-axis) increases from 0% to 3% (area A); after a plateau (area B), the score then decreases when dti increases from 8% to 40% (area C). One suggested task is to increase the score for low dti in area A. Five participants (P1, P2, P3, P4, and P7) commented the trend in area A made sense—applicants in this range are likely people who have no or little loan experience and thus less likely to pay off the loan in time. Although the pattern made sense to P3 and P7, they agreed that one should fix it; P3 and P7 raised the score by aligning \approx all scores in area A to be the same as area B. P3 explained: “[*Although the pattern in area A makes sense,*] *we’ll still try to make this relationship monotonic. For the relationship that I described, like somebody is less experienced with the credit and other stuff, there are other variables that will factor in, like the number of accounts open.*” P7 made the same edit but with a different reason: “*We do not want a model to punish people with no debt.*” In contrast to P3 and P7, P4 said they were uncomfortable with raising the scores in area A, and they would need to talk to

finance experts if they were editing this model in practice. P1 also decided to keep the trend in **area A**. Additionally, P1 applied the interpolation tool \approx to smooth the score increase in **area A** and decrease in **area C**, because P1 believed small bumps in **area A** and **area C** are due to overfitting. Participants’ diverse views on *whether*, *when*, and *how* to edit models highlight that users with different backgrounds may use GAM CHANGER differently in practice.

6.5 Discussion and Future Work

Reflecting on our iterative design of GAM CHANGER with diverse stakeholders, model editing experiences with physicians, and an evaluation with data scientists in various domains, we distill lessons and future directions for model editing and interpretability research.

Promote accountable edits & develop guidelines. Our user study shows model editing via feature engineering and parameter tuning is already common practice in data scientists’ workflow (§ 6.4.2.1). As the first interactive model editing tool, GAM CHANGER lowers the barrier to modifying model behaviors to reflect users’ domain knowledge and values. We find different users could have distinct views on *whether*, *when*, and *how* to edit models (§ 6.4.2.4). Some users might raise concerns that GAM CHANGER makes model editing too easy, and that irresponsible edits could potentially cause harm (e.g., P3 in § 6.4.2.4). Guarding against harmful edits is our top priority—we provide users with continuous feedback (§ 6.2.1), as well as transparent and reversible edits (§ 6.2.2). However, they do not guarantee to prevent users from overfitting the model, injecting harmful bias, or maliciously manipulating model predictions. This potential vulnerability warrants further study on how to audit and regulate model editing.

To help “model editors” modify ML models responsibly, we see a pressing need of *guidelines* that unify best practices in model editing. However, model editing is complex—*whether*, *when*, and *how* to edit a model depends on many factors, including the data, model’s behaviors, and end-tasks in a sociotechnical context. Take our sepsis risk prediction model as an example (§ 6.3.1); we inform our edit decisions by considering treatment effects, the potential impact of edits, and physicians’ values. We make specific edits because physicians prefer false positives over false negatives when predicting sepsis risks—we will make different edits if false negatives are favored. For example, in prostate cancer screenings, false positives are much riskier than false negatives [259]. Therefore, we may prioritize lowering the predicted risk when fixing problematic patterns in a risk prediction model for prostate cancer. Using GAM CHANGER as a research instrument, we plan to develop editing guidelines by further research that engages with experts in diverse domains as well as people who would be impacted by edited models.

Measure real-life impacts. GAM CHANGER provides continuous feedback on model performance (§ 6.2.2). Due to the additive nature of GAMs, global metrics—computed on all validation samples—are not very sensitive to edits that slightly change a few bins of a single feature. An edit’s effect is more significant when we measure the accuracy locally,

such as in the *Selected Scope* or the *Slice Scope*. The *Metric Panel*'s goal is to alert users of accidental edits that might demolish the model's predictive power or disproportionately affect a subgroup in the data. However, GAM CHANGER's ultimate goal is to help users create *safer* and *more correct* models—accuracy on the train and test sets is a secondary metric. To evaluate model editing, we need to measure edited models' performance for their intended use. In high-stakes settings such as healthcare, editing would make a substantial impact if it changed a deployed model's prediction on one patient. We plan to adapt the edited sepsis risk prediction model (§ 6.3.1) in a large hospital and conduct a longitudinal study to monitor and investigate the model's performance.

Enhance collaborative editing. When using GAM CHANGER to edit healthcare models with physicians, we find the tool provides a unique *collaborative experience* for ML researchers and domain experts to discuss, interpret, and improve models together. Our user study echos this observation: (1) participants had been editing models through teaming with diverse stakeholders including domain experts, auditors, and marketing teams (§ 6.4.2.1); (2) participants appreciated GAM CHANGER as a platform to facilitate ML communication with various stakeholders (§ 6.4.2.3). Therefore, we would like to further enhance the tool's affordance for collaborations. We plan to explore interaction techniques that support multiple users to edit the same model simultaneously (e.g., Google Slides). Also, we plan to enhance our Git-inspired editing history to support users to *merge* multiple independent edit series onto one model—enabling collaborators to easily edit a model asynchronously.

6.6 Conclusion

In this work, we present GAM CHANGER, an interactive visualization tool that empowers domain experts and data scientists to not only interpret ML models, but also align model behaviors with their knowledge and values. This open-source tool runs in web browsers or computational notebooks, broadening people's access to responsible ML technologies. We discuss lessons learned from two editing examples and an evaluation user study. We hope our work helps emphasize the critical role of human agency in responsible ML research, and inspire future work in actionable ML interpretability.

6.7 Impact

GAM CHANGER is **deployed in Microsoft** and integrated into their open-source library InterpretML. The tool is used by physicians in NYU hospitals on real-life hospital admission prediction models. An early version of the publication has won the **Best Paper Award** at the NeurIPS Workshop on Bridging the Gap: From ML Research to Clinical Practice.

CHAPTER 7

GAM COACH: HELPING PEOPLE ALTER UNFAVORABLE AI DECISIONS

As AI models are increasingly used in high-stakes decision-making, such as lending, hiring, and college admissions, there has been a call for algorithmic recourse, which aims to help those impacted by AI systems not only learn about the decision rules used, but also provide suggestions for *actions* to change decision outcome in the future [59]. This often involves generating counterfactual (CF) examples, which suggest minimal changes in a few features that would lead to the desired decision outcome. There are many techniques to generate CF examples. However, the actionability of recourse is ultimately subjective and varies from one user to another, or even for a single user at different times. To realize human agency in algorithmic recourse, we introduce GAM COACH to enable an interactive algorithmic recourse paradigm. GAM COACH enables people impacted by AI to specify their recourse preferences, such as difficulty and acceptable range for changing a feature, and iteratively fine-tune recourse plans. With an exploratory interface design, our tool helps users understand the ML model behaviors by experimenting with hypothetical input values and inspecting their effects on model outcomes.

7.1 Introduction

As machine learning (ML) is increasingly used in high-stakes decision-making, such as lending [260], hiring [261], and college admissions [262], there has been a call for greater transparency and increased opportunities for algorithmic recourse [59]. Algorithmic recourse aims to help those impacted by ML systems learn about the decision rules used [263], and provide suggestions for *actions* to change decision outcome in the future [264]. This often involves generating counterfactual (CF) examples, which suggest minimal changes in a few features that would have led to the desired decision outcome [59], such as “if you had decreased your requested loan amount by \$9k and changed your home ownership from renting to mortgage, your loan application would have been approved.” (Fig. 7.1A)

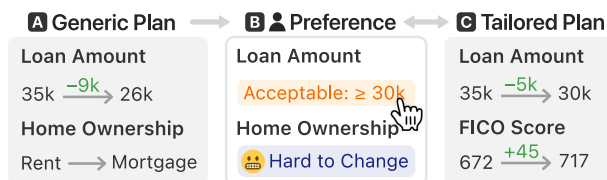


Figure 7.1: GAM Coach enables end users to iteratively finetune recourse plans. (A) If a user finds the initial generic plan less actionable, (B) they can specify their recourse preferences through simple interactions. (C) Our tool will then generate tailored plans that reflect the user’s preferences.

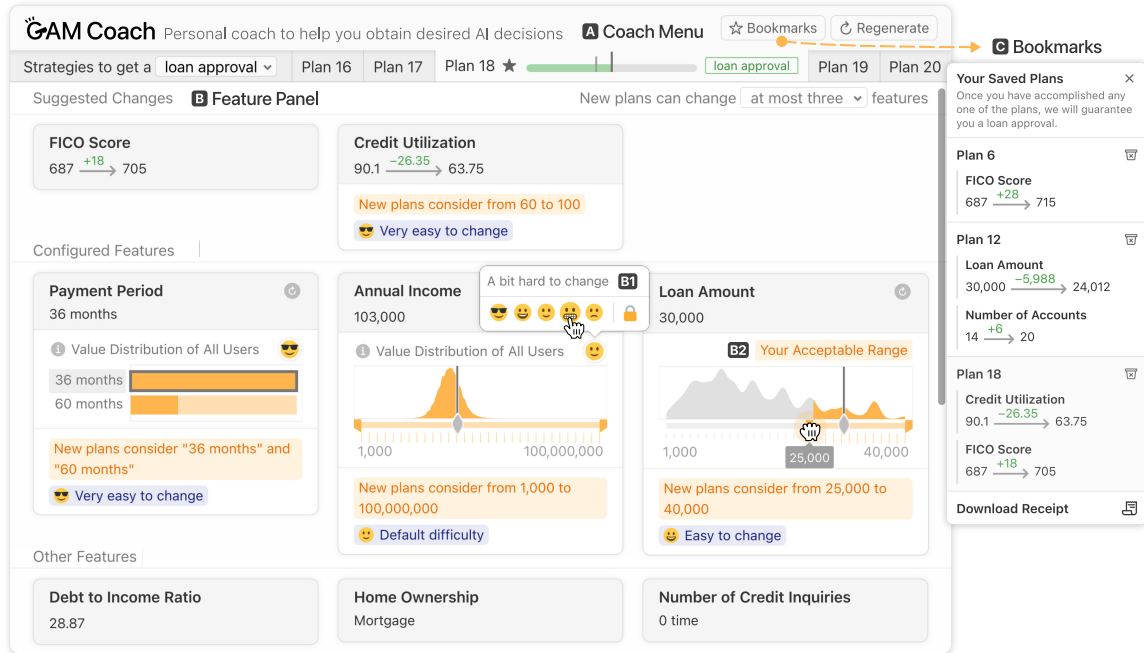


Figure 7.2: GAM COACH enables people impacted by AI-based decision-making systems to iteratively generate algorithmic recourse plans that reflect their preferences. Take the loan application as an example. (A) **The Coach Menu** helps a rejected loan applicant browse diverse recourse plans that would lead to loan approval. After the user selects a plan, (B) **the Feature Panel** visualizes all feature information with progressive disclosure, enabling users to explore how hypothetical inputs affect the model’s decision and specify recourse preferences—such as (B1) the difficulty of changing a feature and (B2) its acceptable range of values—guiding GAM Coach to generate actionable plans. (C) The Bookmarks window allows users to compare bookmarked plans and save a verifiable receipt.

For such approaches to be useful, it is necessary for the suggested actions to be *actionable*—realistic actions that users can appreciate and follow in their real-life circumstances. In the example above, changing home ownership status would arguably not be an actionable suggestion for most loan applicants. To provide actionable recourse, recent work proposes techniques such as generating concise CF examples [265], creating a diverse set of CF examples [266, 267], and grouping features into different actionability categories [268]. These approaches often rely on the underlying assumption that ML developers can measure and predict which CF examples are actionable for all users. However, the actionability of recourse is ultimately subjective and varies from one user to another [269, 270], or even for a single user at different times [271, 272]. Therefore, there is a pressing need to capture and integrate user preferences into algorithmic recourse [273, 270]. GAM COACH aims to take a user-centered approach (Fig. 7.1B–C) to fill this critical research gap. In this work, we **contribute**:

- **GAM COACH, the first interactive algorithmic recourse tool that empowers end users** to specify their recourse *preferences*, such as difficulty and acceptable range for changing a feature, and iteratively *fine-tune* actionable recourse plans (Fig. 7.2). With

an exploratory interface design [274], our tool helps users understand the ML model behaviors by experimenting with hypothetical input values and inspecting their effects on the model outcomes. Our tool advances over existing interactive ML tools [275, 276], overcoming unique design challenges identified from a literature review of recent algorithmic recourse work (§ 7.2, § 7.4).

- **Novel adaptation of integer linear programming to generate CF examples.** To operationalize interactive recourse, we ground our research in generalized additive models (GAMs) [277, 11], a popular class of models that performs competitively to other state-of-the-art models yet has a transparent and simple structure [253, 250, 278, 7]. GAMs enable end users to probe model behaviors with hypothetical inputs in real time directly in web browsers. Adapting integer linear programming, we propose an efficient and flexible method to generate optimal CF examples for GAM-based classifiers and regressors with continuous and categorical features and pairwise feature interactions [251] (§ 7.3).
- **Design lessons distilled from a user study with log analysis.** We conducted an online user study with 41 Amazon Mechanical Turk workers to evaluate GAM COACH and investigate how everyday users would use an interactive algorithmic recourse tool. Through analyzing participants' interaction logs and subjective ratings in a hypothetical lending scenario, our study highlights that GAM COACH is usable and useful, and users prefer personalized recourse plans over generic plans. We discuss the *characteristics* of users' satisfactory recourse plans, *approaches* users take to discover them, and *design lessons* for future interactive recourse tools. We also provide empirical evidence that with transparency, everyday users can discover and are often puzzled by counterintuitive patterns in ML models (§ 7.5).
- **An open-source, web-based implementation** that broadens people's access to developing and using interactive algorithmic recourse tools. We implement our CF generation method in both Python and JavaScript, enabling future researchers to use it on diverse platforms. We develop GAM COACH with modern web technologies such as WebAssembly, so that anyone can access our tool using their web browsers without the need for installation or a dedicated backend server. We open-source¹ our CF generation library and GAM COACH system with comprehensive documentation² (§ 7.4.5). For a demo video of GAM COACH, visit <https://youtu.be/ubacP34H9XE>.

To design and evaluate a prospective interface [274] for interactive algorithmic recourse, we situate GAM COACH in loan application scenarios. However, we caution that adapting GAM COACH for real lending settings would require further research with financial and legal experts as well as people who would be impacted by the system. Our goal is for this work to serve as a foundation for designing future user-centered recourse tools.

¹GAM COACH code: <https://github.com/poloclub/gam-coach>

²GAM COACH documentation: <https://poloclub.github.io/gam-coach/docs>

7.2 Design Goals

Our goal is to design and develop an interactive, visual experimentation tool that respects end users’ autonomy in algorithmic recourse, helping them discover and fine-tune recourse plans that reflect their preferences and needs. We identify five main design goals of GAM COACH through synthesizing the trends and limitations of traditional algorithmic recourse systems [e.g., 270, 279, 280, 281, 274, 59, 282].

- G1. Visual summary of diverse algorithmic recourse plans.** To help end users find actionable recourse plans, researchers suggest presenting diverse CF options that users can pick from [266, 270]. Thus, GAM COACH should efficiently generate diverse recourse plans (§ 7.3.2) and present a visual summary of each plan as well as display multiple plans at the same time (§ 7.4.1). This could help users compare different strategies and inform interactions to generate better recourse plans.
- G2. Easy ways to specify recourse preferences.** What makes a recourse plan actionable varies from one user to another—it is crucial for a recourse tool to enable users to specify a wide range of recourse preferences [270, 281, 273]. Therefore, we would like to allow users to easily configure (1) the *difficulty* of changing a feature, (2) the *acceptable range* within which a feature can change, and (2) the *maximum number of features* that a recourse plan can change (§ 7.4.2), and GAM COACH should generate plans reflecting users’ specified preferences (§ 7.3.3). This interactive recourse design would empower users to iteratively customize recourse plans until they find satisfactory plans.
- G3. Exploratory interface to experiment with hypothetical inputs.** The goal of algorithmic recourse is not only to help users identify actions to alter unfavorable model decisions, but also to help them understand how a model makes decisions [59, 279]. When explaining a model’s decision-making, research shows that interfaces allowing users to probe an ML model with different inputs help users understand model behaviors and lead to greater satisfaction with the model [283, 284, 274, 276]. Therefore, we would like GAM COACH to enable users to experiment with different hypothetical inputs and inspect how these changes affect the model’s decision (§ 7.4.2).
- G4. Clear communication and engagement.** The target users of GAM COACH are everyday people who are usually less knowledgeable about ML and domain-specific concepts [285]. Our goal is to design and develop an interactive system that is easy to understand and engaging to use, requiring the tool to communicate and explain recourse plans and domain-specific information to end users (§ 7.4.2, § 7.4.3).
- G5. Open-source and model-agnostic implementation.** We aim to develop an interactive recourse tool that is easily accessible to users, with no installation required. By using web browsers as the platform, users can directly access GAM COACH through their

laptops or tablets. Additionally, we aim to make our interface model-agnostic so that future researchers can use it with different ML models and recourse techniques. Finally, we would like to open-source our implementation and provide documentation to support future design, research, and development of interactive algorithmic recourse (§ 7.4.5).

7.3 Techniques for Customizable Recourse Generation

Given our design goals (G1–G5), it is crucial for GAM COACH to generate customizable recourse plans interactively with a short response time. Therefore, we base our design on GAMs, a family of ML models that perform competitively to state-of-the-art models yet have a transparent and simple structure—enabling end users to probe model behaviors in real-time with hypothetical inputs. In addition, with a novel adaptation of integer linear programming (§ 7.3.2), GAMs allow us to efficiently generate recourse plans that respect users’ preferences and thus achieve our design goals (§ 7.3.3).

7.3.1 Model Choice

To operationalize our design of interactive algorithmic recourse, we ground our research in GAMs [12]. More specifically, we make use of a type of GAMs called *Explainable Boosting Machines*, (EBMs) [11, 7], which perform competitively to the state-of-the-art black-box models yet have a transparent and simple structure [253, 250, 278, 7]. Compared to simple models like linear models or decision trees, EBMs achieve superior accuracy by learning complex relations between features through gradient-boosting trees [251], and thus deploying our design is realistic. Compared to complex models like neural networks, EBMs have a similar performance on tabular data but a simpler structure; therefore, users can probe model behaviors in real-time with hypothetical inputs (G3).

Given an **input** $x \in \mathbb{R}^k$ with k features, the **output** $y \in \mathbb{R}$ of an EBM model can be written as:

$$y = l(S_x) \tag{7.1}$$

$$S_x = \beta_0 + f_1(x_1) + f_2(x_2) + \dots + f_k(x_k) + \dots + f_{ij}(x_i, x_j)$$

Here, each **shape function** f_j for single features $j \in \{1, 2, \dots, k\}$ or $f_{ij}(x_i, x_j)$ for pairwise interactions between features [251] is learned using **gradient-boosted trees** [252]. S_x is the sum of all **shape function** outputs as well as **the intercept constant** β_0 . The model converts S_x to the **output** y through a link function l that is determined by the ML task. For example, a sigmoid function is used for binary classifications, and an identity function for regressions.

What distinguishes EBMs from other GAMs is that the **shape function** f_j or f_{ij} is an ensemble of trees, mapping a **main effect feature value** x_j or a **pairwise interaction** (x_i, x_j) to a scalar **score**. Before training, EBM applies *equal-frequency binning* on each continuous

feature, where bins have different widths but the same number of training samples. This discrete bucketing process is commonly used to speed up gradient-boosting tree methods with little cost in accuracy, such as in popular tree-based models LightGBM [286] and XGBoost [287]. For categorical features, EBMs treat each discrete level as a bin. Once an EBM model is trained, the learned parameters for each ensemble of trees which defines the feature split points and scores in each region defined by these split points are transformed to a *lookup histogram* (for univariate features) and a *lookup table* (for pairwise interactions). When predicting on a data point, the model first looks up corresponding scores for all feature values and interaction terms and then applies Equation 7.1 to compute the output.

7.3.2 CF Generation: Integer Linear Programming

A recourse plan is a CF example c that makes minimal changes to the original input x but leads to a different prediction. Without loss of generality, we use binary classification as an example, with sigmoid function $\sigma(a) = \frac{1}{1+e^{-a}}$ as a link function. If $\sigma(S_x) \geq 0.5$ or $S_x \geq 0$, the model predicts the input x as positive; otherwise it predicts x as negative. To generate c , we can change x so that the new score S_c has a different sign from S_x . Note that S_x is a linear combination of shape function scores and so is $S_c - S_x$. Thus, we can express this counterfactual constraint as a linear constraint. To enforce c to only make minimal changes to x , we can minimize the distance between c and x , which can also be expressed as a linear function. Since all constraints are linear, and there are a finite number of bins for each feature, we express the GAM COACH recourse generation as an *integer linear program*:

$$\min \text{ distance} \tag{7.2a}$$

$$\text{s.t. distance} = \sum_{i=1}^k \sum_{b \in B_i} d_{ib} v_{ib} \tag{7.2b}$$

$$-S_x \leq \sum_{i=1}^k \sum_{b \in B_i} g_{ib} v_{ib} + \sum_{(i,j) \in N} \sum_{b_1 \in B_i} \sum_{b_2 \in B_j} h_{ijb_1b_2} z_{ijb_1b_2} \tag{7.2c}$$

$$z_{ijb_1b_2} = v_{ib_1} v_{jb_2} \text{ for } (i,j) \in N, b_1 \in B_i, b_2 \in B_j \tag{7.2d}$$

$$\sum_{b \in B_i} v_{ib} \leq 1 \text{ for } i=1, \dots, k \tag{7.2e}$$

$$v_{ib} \in \{0, 1\} \text{ for } i=1, \dots, k, b \in B_i \tag{7.2f}$$

$$z_{ijb_1b_2} \in \{0, 1\} \text{ for } (i,j) \in N, b_1 \in B_i, b_2 \in B_j \tag{7.2g}$$

We use an **indicator variable** v_{ib} (7.2f) to denote if a main effect bin is active: if $v_{ib} = 1$, we change the **feature value of x_i** to the closest value in its bin b . All bin options of x_i are included in a set B_i . For each **feature x_i** , there can be at most one active bin (7.2e); if there is no active bin, then we do not change the **value of x_i** . We use an **indicator variable**

$z_{ijb_1b_2}$ (7.2g) to denote if a **pairwise interaction effect** is active—it is active if and only if bin b_1 of x_i and bin b_2 of x_j are both active (7.2d). The set N includes all available **interaction effect terms**. Constraint 7.2b determines the total distance cost for a potential CF example; it uses a set of pre-computed distance costs d_{ib} of changing one **feature** x_i to the closest value in bin b . Constraint 7.2c ensures that any solution would flip the model prediction, by gaining enough total score from main effect scores (g_{ib}) and interaction effect scores ($h_{ijb_1b_2}$). Constants g_{ib} and $h_{ijb_1b_2}$ are pre-computed and adjusted for cases where a single active main effect bin results in changes in interaction terms.

Novelty. Advancing existing works that use integer linear programs for CF generation (on linear models [264] or using a linear approximation of neural networks [288]), our algorithm is the first that works on non-linear models without approximation. Our algorithm is also the first and only CF method specifically designed for EBM models. Without it, users would have to rely on model-agnostic techniques such as genetic algorithm [289] and KD-tree [290] to generate CF examples. These model-agnostic methods do not allow for customization. Also, by quantitatively comparing our method with these two model-agnostic CF techniques on three datasets, we find CFs generated by our method are significantly *closer* to the original input, *more sparse*, and encounter *less failures*.

Generalizability. Our algorithm can easily be adapted for EBM regressors and multi-class classifiers. For regression, we modify the left side and the inequality of constraint 7.2c to bound the prediction value in the desired range. For multiclass classification, we can modify constraint 7.2c to ensure that the desired class has the largest score. In addition to EBMs, one can also easily adapt our algorithm to generate CF examples for linear models [264]. For other non-linear models, such as neural networks and random forest, one can first use a linear approximation [288] and then apply our algorithm, verifying suggested recourse plans with respect to the original model. If the suggested recourse plan would not change the output of the original model, an alternative can be generated by solving the program again with the previous solution blocked.

Scalability. Modern linear solvers can efficiently solve our integer linear programs. The complexity of solving an integer linear program increases along two factors: the number of variables and the number of constraints. In Equation 7.2, all variables are binary—making the program easier to solve than a program with non-binary integer variables. For any dataset, there are always exactly 3 constraints from 7.2b, 7.2c, and 7.2e. The number of constraints from 7.2d increases along the number of interaction terms $|N|$ and the number of bins per feature $|B_i|$ on these interaction terms. In practice, $|N|$ and $|B_i|$ are often bounded to ensure EBM are interpretable. For example, by default the popular EBM library InterpretML [7] bounds $|N| \leq 10$ and $|B_i| \leq 32$. Therefore, in the worst-case scenario with 10 continuous-continuous interaction terms, there will be at most $10 \times 32 \times 32 = 10,240$ constraints from 7.2d. For instance, on the Communities and Crime dataset [291] with 119



Figure 7.3: A bar chart visualizes the model’s decision score of a recourse plan: the bar is marked with the user’s original score (shorter vertical line on the left) and the threshold needed to obtain the desired decision (longer vertical line on the right).

continuous features, 1 categorical feature, and 10 pairwise interaction terms, there are about 7.2k constraints and 3.6k variables in our program. It only takes about 0.5–3.0 seconds to generate a recourse plan using Firefox Browser on a MacBook.

7.3.3 Recourse Customization

With integer linear programming, we can generate recourse plans that reflect a wide range of user preferences (G2). For example, to prioritize a feature that is *easier for a user to change*, we can lower the distance cost d_{ib} for that feature. To enforce recourse plans to only change a feature in a user specified *acceptable range*, we can remove out-of-range binary variables v_{ib} . If a user requires the recourse plans to only change *at most p features*, we can add an additional linear constraint $\sum_{i=1}^k \sum_{b \in B_i} v_{ib} \leq p$. Finally, with modern linear solvers, we can efficiently generate diverse recourse plans (G1) by solving the program multiple times while blocking previous solutions.

7.4 User Interface

Given the design goals (G1–G5) described in § 7.2, we present GAM COACH, an interactive tool that empowers end users to specify preferences and iteratively fine-tune recourse plans (Fig. 7.4). The interface tightly integrates three components: the *Coach Menu* that provides overall controls and organizes multiple recourse plans as tabs (§ 7.4.1), the *Feature Panel* containing *Feature Cards* that allow users to specify recourse preferences with simple interactions (§ 7.4.2), and the *Bookmark Window* summarizing saved recourse plans (§ 7.4.3). To explain these views in this section, we use a loan application scenario with the Lending-Club dataset [258], where a bank refers a rejected loan applicant to GAM COACH pre-loaded with the applicant’s input data. Our tool can be easily applied to GAMs trained on different datasets while providing a consistent user experience. On GAM COACH’s public demo page, we present five additional examples with five datasets that are commonly used in algorithmic recourse literature: Communities and Crime [291] (also used in the second usage scenario in § 7.4.4), Taiwan Credit [292], German Credit [293], Adult [294], and COMPAS [2].

7.4.1 Coach Menu

The *Coach Menu* (Fig. 7.2A) is the primary control panel of GAM COACH. Users can use the dropdown menu and input fields to specify desired decisions for classification and

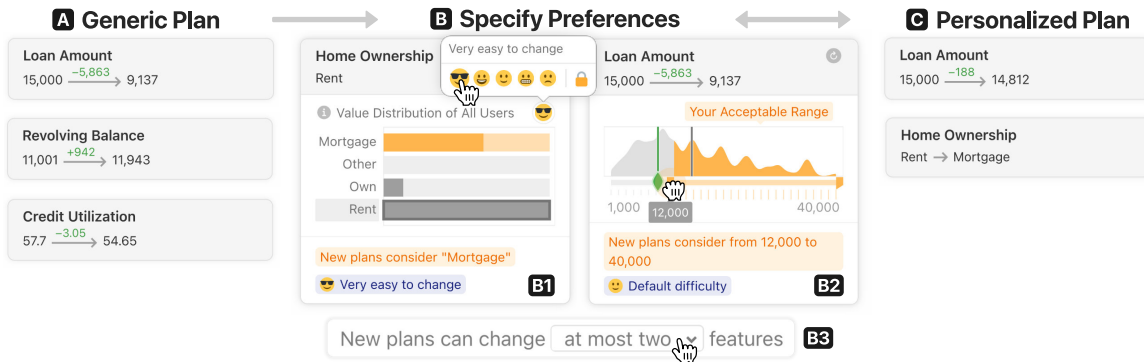


Figure 7.4: GAM COACH enables end users to inspect and customize recourse plans through simple interactions. (A) *Initial generic plans* are generated with the same configurations for all users. (B) Users can *specify recourse preferences* if they are not satisfied with the initial plans; by configuring (B1) the *difficulty* to change a feature; (B2) the *acceptable range* that a feature can change between, and (B3) the *max number of features* that a recourse plan can alter. (C) GAM COACH then generates *personalized plans* that respect users’ preferences. Users can iteratively refine their preferences until a satisfactory plan is found.

regression. For each recourse plan generation iteration, the tool generates five diverse plans (G1) to help users achieve their goal, with each plan representing a CF example. Users can access each plan by clicking the corresponding tab on the plan tab bar. When a plan is selected, the *Feature Panel* updates to show details about the plan, and the plan’s corresponding tab extends to show the model’s decision score (Fig. 7.3). Users can click the *Bookmarks* button to open the *Bookmarks* window and click the *Regenerate* button to generate five new recourse plans that reflect the currently specified recourse preferences.

7.4.2 Feature Panel

Each recourse plan has a unique *Feature Panel* (Fig. 7.2B) that visualizes plan details and allows users to provide preferences guiding the generation of new plans (G2). A *Feature Panel* consists of *Feature Cards* where each card represents a data feature used in the model. To help users easily navigate through different features, the panel groups *Feature Cards* into three sections: (1) features that are changed in the plan, (2) features that are configured by the user, (3) and all other features. To prevent overwhelming users with too much information (G4), all cards are collapsed by default—only displaying the feature name and feature values. Users can hover over the feature name to see a tooltip explaining the definition of the feature (G4). With a *progressive disclosure* design [295, 296], details of a feature, such as the distribution of feature values, are only shown on demand after users click that *Feature Card*. Progressive disclosure also makes GAM COACH interface scalable, as users can easily scroll and browse over hundreds of collapsed *Feature Cards*. Since EBMs process continuous and categorical features differently, we employ different card designs based on the feature type.

Continuous Feature Card. For continuous features, such as `FICO score`, the *Feature*

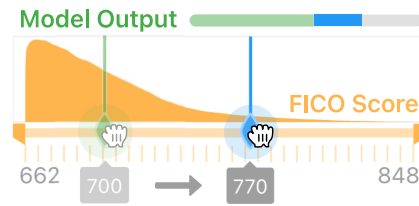




Figure 7.5: Users can test hypothetical input values in real time.

Difficulty	Distance
😎 Very easy	× 0.1
😊 Easy	× 0.5
😊 Neutral	× 1
😬 Hard	× 2
😞 Very hard	× 10
🔒 Impossible	× ∞

Figure 7.6: Distance multipliers of difficulties.

Card (Fig. 7.5) uses a filled curved chart to visualize the distribution of feature values in the training set. Users can drag the diamond-shaped thumb  on a slider below the chart to experiment with hypothetical values. During dragging, the decision score bar updates its width to reflect a new prediction score in real time. Therefore, users can better understand the underlying decision-making process by probing the model with different inputs (G3). Also, users can drag the orange thumbs  to set the lower and upper bounds of acceptable feature changes. For example, one user might only accept recourse plans that include `loan amount` at \$12k or higher (Fig. 7.4-B2).

Categorical Feature Card. For categorical features, such as `home ownership`, users can inspect the value distribution with a horizontal bar chart (Fig. 7.4-B1), where a longer bar represents more frequent options in the training data. To specify acceptable ranges, users can click the bars to select acceptable options for new recourse plans. Acceptable options are highlighted as `orange`, whereas unacceptable options are colored as `gray`. Users can also click text labels next to the bars to experiment with hypothetical options and observe how they affect the model decision.

Specify Difficulty to Change a Feature. Besides selecting a feature’s acceptable range, users can also specify how hard it would be for them to change a feature. For example, it might be easier for some users to lower `credit utilization` than to change `home ownership`. To configure feature difficulties, users can click the smiley button on any *Feature Card* and then select a suitable difficulty option on the pop-up window (Fig. 7.4-B1). Internally, GAM COACH multiplies the distance costs of all bins in that feature with a constant multiplier (Fig. 7.6). If the user selects the “impossible to change” difficulty, the tool will remove all variables associated with this feature in the internal integer program (§ 7.3.3). Therefore, when

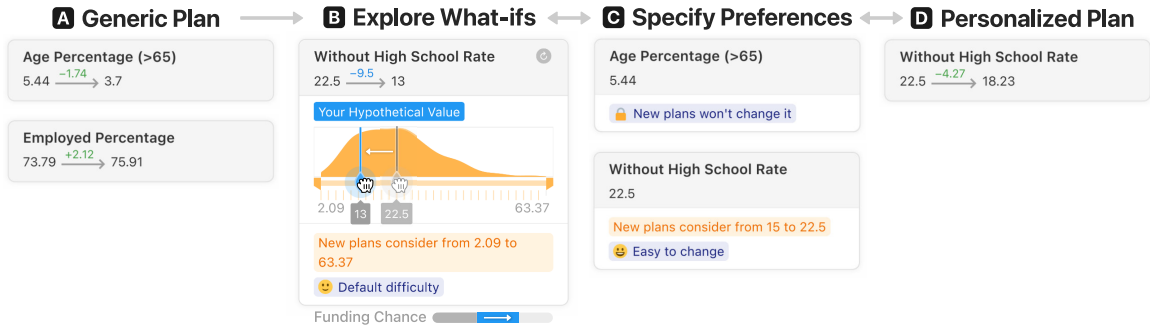


Figure 7.7: GAM COACH allows end users to experiment with hypothetical input values and customize recourse plans. (A) Our tool first shows *generic plans* generated with default configurations. (B) Users can explore how different input values affect the model’s prediction in real time through simple interactions on the *Feature Card*: for example, lowering the percentage of adults without a high school diploma increases the chance of getting a government grant. (C) Users can then specify recourse preferences—such as feature *difficulties* and *acceptable ranges*—based on their circumstances and understanding of the model’s prediction patterns. (D) GAM COACH then generates more actionable recourse plans based on the user-specified preferences.

generating new recourse strategies, GAM COACH would prioritize features that are easier to change and would not consider features that are impossible to change.


7.4.3 Bookmarks and Receipt

During the recourse iterations, users can save any suitable plans by clicking the star button ☆ on the plan tab (Fig. 7.3). Then, users can compare and update their saved plans in the *Bookmarks window* (Fig. 7.2C). Once users are satisfied with bookmarked plans, they can save a *recourse receipt* as proof of the generated recourse plans. Wachter *et al.* first introduced the recourse receipt concept as a contract guaranteeing that a bank will approve a loan application if the applicant achieves all changes listed in the recourse plan. GAM COACH is the first tool to realize this concept by creating a plaintext file that records the timestamp, a hash of EBM model weights, the user’s original input, and details of bookmarked plans (G4). In addition, we propose a novel security scheme that uses Pretty Good Privacy (PGP) to sign the receipt with the bank’s private key [297]. With public-key cryptography, users can hold the bank accountable by being able to prove the receipt’s authenticity to third-party authorities with the bank’s public key. Also, banks can use their private key to verify a receipt’s integrity during recourse redemption to avoid counterfeit receipts.

7.4.4 Usage Scenarios

We present two hypothetical usage scenarios to illustrate how GAM COACH can help everyday users identify actionable strategies to alter undesirable ML-generated decisions.

Individual Loan Application. Eve is a rejected loan applicant, and she wants to identify ways to get a loan in the future. In this hypothetical usage scenario, to inform loan decisions,

the bank has trained an EBM model on past data (we use LendingClub [258] to illustrate this scenario in Fig. 7.4). Their dataset has 9 continuous features and 11 categorical features, and the outcome variable is binary—indicating whether a person can pay back the loan in time. The bank gives Eve a link to GAM COACH when informing her of the loan rejection decision. After Eve opens GAM COACH in a web browser, the tool pre-loads Eve’s input data and generates five recourse plans based on the default configurations. Each plan lists a set of minimal changes in feature values that would lead to loan approval. One plan suggests Eve lower the requested `loan amount` from \$15k to \$9k along with two other changes (Fig. 7.4A). Eve does not like this suggestion because she is unwilling to compromise a loss of \$6k in the requested loan. Therefore, she clicks the `loan amount` *Feature Card* and drags the left thumb  to set the *acceptable range* of `loan amount` to \$12k and above (Fig. 7.4-B2). After browsing all recourse plans in the *Coach Menu*, Eve finds that none of the plans suggest changes to `home ownership`. Eve and her partner are actually moving to their newly-purchased condo next month. Therefore, Eve sets the *acceptable range* of `home ownership` to “mortgage” and changes its *difficulty* to “very easy” 😊 (Fig. 7.4-B1). Eve also prefers plans that change fewer features, so she clicks the dropdown menu on the *Feature Panel* to ask the tool to only generate plans that change at most two features (Fig. 7.4-B3). After Eve clicks the `Regenerate` button, GAM COACH quickly generates five personalized plans that respect Eve’s preferences. Among these plans, Eve especially likes the one suggesting she lower the `loan amount` by about \$200 and change `home ownership` to mortgage (Fig. 7.4C). Finally, Eve bookmarks this plan and downloads a recourse receipt that guarantees her a loan if all suggested terms are met. Eve plans to apply for the loan again at the same bank next month.

Government Grant Application. Hal is a county manager in the United States. He has applied for a federal grant for his county. Unfortunately, his application is rejected. He wants to learn about the decision-making process and what actions he can take to succeed in future applications. In this hypothetical usage scenario, to inform funding decisions, the federal government has trained an EBM model on past data (we use the Communities and Crime dataset [291] to illustrate this scenario in Fig. 7.7). This dataset has 119 continuous features and 1 categorical feature describing the demographic and economic information of different counties in the United States, and is used to predict the risk of violent crime. As part of a performance incentive funding program [298], the federal government provides more funding opportunities to counties with lower predicted crime risk [299]. Before training the EBM model, the federal government has removed protected features (e.g., `black population`) and features with many (more than half) missing values, resulting in a total of 94 continuous features and 1 categorical feature.

The federal government provides rejected counties with a link to GAM COACH when informing them of the funding decisions. Hal opens GAM COACH in his browser; this tool has pre-loaded the demographic and economic features of his county and quickly suggested five recourse plans that would lead to funding. These generic plans are generated

with the default configuration. One plan (Fig. 7.7A) suggests Hal decrease `age percentage (>65)` and increase `employed percentage` in his county. Hal likes the recommendation of increasing `employed percentage` because a higher employment rate is also beneficial for the economy of his county. However, Hal is puzzled by the suggestion of lowering `age percentage (>65)`. He is not sure why the population age is used to decide funding decisions. Besides, lowering the percentage of the elderly population is not actionable. Therefore, Hal “locks” this feature by setting its *difficulty* to “impossible” 🚫 (Fig. 7.7C).

To gain a better understanding of how the funding decision is made, Hal expands several *Feature Cards* and experiments with hypothetical feature values by dragging the blue thumbs 🦊; GAM COACH visualizes the model’s prediction scores with these hypothetical inputs in real time (Fig. 7.7B). Hal finds that lowering `without high school rate` can increase his chance of getting a grant. This is good news as Hal’s county has just started a high school dropout prevention program aiming to lower the percentage of adults without a high school diploma to below 15% in eight years. Hal then sets this feature’s *difficulty* to “easy to change” 😊 and drags the orange thumbs 🦊 to set its *acceptable range* to between 15% and 22.5% (Fig. 7.7C). After Hal clicks the `Regenerate` button, GAM COACH generates five new personalized plans in only 3 seconds despite there being almost 100 features. Among these five plans, Hal likes the one that recommends decreasing `without high school rate` by 4.27% (Fig. 7.7D). Finally, Hal saves a recourse receipt, and he will apply for this grant again once the percentage of adults without a high school diploma in his county drops by 4.27%.

7.4.5 Open-source & Generalizable Tool

GAM COACH is a web-based algorithmic recourse tool that users can access with any web browser on their laptops or tablets, no installation required (G5). We use *GLPK.js* [300] to solve integer programs with WebAssembly, *OpenPGP.js* [301] to sign recourse receipts with PGP, and *D3.js* [180] for visualizations. Therefore, the entire system runs locally in users’ browsers without dedicated backend servers. We also provide an additional Python package³ for developers to generate customizable recourse plans for EBM models without a graphical user interface. With this Python package, developers and researchers can also easily extract model weights from any EBM model to build their own GAM COACH. Finally, despite its name, GAM COACH’s interface is model-agnostic—it supports any ML models where (1) one can control the difficulty and acceptable range of changing a feature during CF generation, and (2) model inference is available. With our open-source and generalizable implementation, detailed documentation, and examples on six datasets across a wide range of tasks and domains—LendingClub [258], Taiwan Credit [292], German Credit [293], Adult [294], COMPAS [2], and Communities and Crime [292]—future researchers can easily adapt our interface design to their models and datasets.

³Python package: <https://poloclub.github.io/gam-coach/docs/gamcoach>

7.5 User Study

To evaluate GAM COACH and investigate how everyday users would use an interactive algorithmic recourse tool, we conducted an online user study with 41 United States-based crowdworkers. For possible datasets to use in this user study, we compared five public datasets that are commonly used in the recourse literature: LendingClub [e.g., 266, 302], Taiwan Credit [e.g., 302, 264, 289], German Credit [e.g., 266, 302, 299], Adult [e.g., 303, 289, 288], and COMPAS [e.g., 266, 303, 304]. We decided to use LendingClub in our study for the following three reasons. First, we chose a lending scenario as it is one scenario that many people, including crowdworkers, may encounter in real-life. Second, there is no expert knowledge needed to understand the setting, making our tasks appropriate for crowdworkers. Finally, our institute requires research participants to be United States-based: among the four datasets that can be used in a lending setting (LendingClub, Taiwan Credit, German Credit, and Adult), LendingClub is the only United States-based dataset collected from a real lending website. We aimed to answer the following three research questions:

RQ1. What makes a satisfactory recourse plan for end users? (§ 7.5.3.1)

RQ2. How do end users discover their satisfactory recourse plans? (§ 7.5.3.2)

RQ3. How does interactivity play a role in providing algorithmic recourse? (§ 7.5.3.3)

7.5.1 Participants

We recruited 50 anonymous and voluntary United States-based participants from Amazon Mechanical Turk (MTurk), an online crowdsourcing platform. We did not collect any personal information. Collected interaction logs and subjective ratings are stored in a secure location where only the authors have access. The authors' Institutional Review Board (IRB) has approved the study. The average of three self-reported task completion times on a worker-centered forum⁴ is 32½-minutes. We paid 41 participants \$6.50 per study and 9 participants who had not passed our quality control \$5.50. Recruited participants self-report an average score of 2.7 for ML familiarity in a 5-point Likert-scale, where 1 represents "I have never heard of ML" and 5 represents "I have developed ML models."

7.5.2 Study Design

Each participant first signed a consent form and filled out a background questionnaire (e.g., ML familiarity).

GAM COACH Tutorial and Short Quiz. We directed participants to a Google Survey and a website containing GAM COACH, task instructions, and tutorial videos. Our tool, loaded with an EBM binary classifier that predicts loan approval on the LendingClub

⁴TurkerView: <https://turkerview.com/>

Please review following plans. Please emphasize why chosen plans are helpful **for you**, and why unchosen plans are less helpful.

Plan 6 (chosen by you)

Home Ownership
Rent → Mortgage

Why do you like this plan?
I like this plan because...

Select a Rating Score ▾

Figure 7.8: We asked user study participants to explain why they had chosen their satisfactory plans, and why they had not chosen two other random plans (not shown in the figure).

dataset [258], contains input values of 500 random test samples on which the model predicts loan rejection. Participants were asked to watch a 3-minute tutorial video and complete eight multiple-choice quiz questions. These questions are simple—asking what is shown in the tool after certain interactions. All participants were asked to perform these interactions on the same data sample, so we had “ground truth” answers for the quiz questions. We used the quiz as a “gold standard” question to detect fraudulent responses [305, 306]. Although participants were prompted that they would need to answer all questions correctly to receive the base compensation, we paid all participants regardless of their answers. However, in our analysis, we only included responses from participants who had correctly answered at least four questions.

Free Exploration with an Imaginary Usage Scenario. After completing the tutorial and quiz, participants were asked to pretend to be a rejected loan applicant and freely use GAM COACH *until finding at least one satisfactory recourse plan*. These satisfactory recourse plans could be chosen from the first five generic plans that GAM COACH generates with a default configuration *or* follow-up plans that are generated based on participants’ configured preferences. To help participants imagine the scenario, we asked them to change the input sample (one of 500 random samples) until they find one that they feel comfortable pretending to be. Participants could also manually adjust the input values. After identifying and bookmarking their satisfactory plans, participants were asked to rate the importance of configured preferences or briefly explain why no configuration is needed. Then, participants were asked to explain why they had chosen their saved plans (Fig. 7.8) and why they had not chosen two other plans, which were randomly picked from the initial recourse plans. To incentivize participants to write good-quality explanations [307, 308], we told participants that they could get a \$1 bonus reward if their explanations are well-justified. Regardless of their responses, all participants who had correctly answered at least four quiz questions were rewarded with this bonus.

Interaction Logging and Survey. While participants were using GAM COACH, the tool logged all interactions, such as preference configuration, hypothetical value experiment, and recourse plan generation. Each log event includes a timestamp and associated values. After

finishing the exploration task, participants were asked to click a button that uploads their interaction logs and recourse plan reviews as a JSON file to a secured Dropbox directory. The file-names included a random number. Participants were given this number as a verification code to report in the survey response and MTurk submission—we used this number to link a participant’s MTurk ID with their log data and survey response. Finally, participants were asked to complete the survey consisting of subjective ratings and open-ended comments regarding the tool. As the EBM model used in the study is non-monotonic, the tool sometimes can suggest counterintuitive changes [270], such as to lower `annual income` for loan approval. We asked participants to report counterintuitive recourse plans in the survey if they had seen any.

7.5.3 Results

Out of 50 recruited participants, 41 (P1–P41) correctly answered at least four “quality-control” questions. In the following sections, we summarize our findings through analyzing these 41 participants’ interaction logs, recourse plan reviews, and survey responses. We denote the Wald Chi-Square statistical test score as χ^2 .

7.5.3.1 RQ1: Characteristics of Satisfactory Recourse Plans

During the exploration task, participants were asked to identify at least one recourse plan that they would be satisfied with if they were a rejected loan applicant using GAM COACH. On average, each participant chose 1.54 satisfactory plans. Participants preferred *concise plans* that changed only a few features, with an average of 2.11 features per plan. Chosen plans changed a diverse set of features, including 13 out of 20 features. The most popular features changed by chosen plans were `loan amount` (26.3%), `FICO score` (18.8%), and `credit utilization` (11.3%). Features that were not changed by any chosen plans were mostly hard to change in real life, such as `number of bankruptcies` and `employment length`.

Reasons for Choosing Satisfactory Plans. Three main reasons that participants reported choosing plans were that the plans were (1) controllable, (2) requiring small changes or less compromise, or (3) beneficial for life in general. Most participants chose recourse plans that felt realistic and controllable. For example, P30 wrote “*I think it’s very possible to reduce my credit utilization in a short amount of time.*” In particular, participants preferred plans that only changed a few features and required a small amount of change. Participants described these plans as “*simple and fast*” (P5), “*straightforward*” (P7), and “*easy to do*” (P16). Some participants chose plans because they could tolerate the compromises. For example, P8 wrote “*I’m fine with the lower loan amount.*” Similarly, P11 reported “[*The decreased*] loan amount is close to what I need.” Interestingly, some participants favored plans that could benefit their lives in addition to helping them get loan approval. For example, P14 wrote “[...] lower utilization is good for me anyway from what I know, so this seems like the best plan.” Similarly, P28 wrote “[*this plan*] in my opinion would guarantee greater monetary flexibility.”

Reasons for Not Choosing a Plan. Participants' explanations for not choosing a plan mostly complemented the reasons for choosing a plan. Some participants also skipped plans because they were puzzled by counterintuitive suggestions, did not understand the suggestions, or just wanted to see more alternatives. First, participants disliked unrealistic suggestions: P2 explained "*It tells me to increase my income. My income is fixed. I cannot just increase them at a whim.*" Similarly, P6 wrote "*With inflation it might be harder to use less credit.*" Participants also disliked plans requiring too many changes or a large amount of change. For example, P30 wrote "*The amount of loan suggested to be reduced is too large. Assuming I'm applying for 9,800 for real, I wouldn't want to reduce the amount by more than 30%.*" Interestingly, some participants skipped a plan because it suggested counterintuitive changes. For example, P14 wrote "*It seemed like a bug because why would asking for an extra 13 dollars [in loan amount] result in a loan approval?*" Participants also skipped plans when they did not understand the suggestion: P9 wrote "*I'm not exactly sure what credit utilization is. I looked at the tooltip, but still wasn't sure.*" Finally, some participants skipped the initial plans because they just wanted to explore more alternatives: P22 explained "*I wanted to check out a few more things before I made my decision.*"

Design Lessons. By analyzing the characteristics of satisfactory recourse plans, our user study is the first study that provides empirical evidence to support several hypotheses from the recourse literature. We find that participants preferred plans that suggested changes on actionable features [279, 273], are concise and make small changes [265, 59], and could benefit participants beyond the recourse goal [270]. Additionally, participants were likely to save multiple satisfactory plans from one recourse session, highlighting the importance of providing diverse recourse plans [266]. Our study also shows that with transparency, end users can identify and dislike counterintuitive recourse plans (see more discussion in § 7.5.3.3). Therefore, future researchers and developers should help users identify concise and diverse plans that change actionable features and are beneficial overall. Also, researchers and developers should carefully audit and improve their models to prevent a CF generation algorithm from generating counterintuitive plans. Our findings also highlight that communicating recourse plans and providing a good user experience are as important as generating good recourse plans.

7.5.3.2 RQ2: Path to Discover Satisfactory Recourse Plans

In the exploration task, participants could freely choose their satisfactory recourse plans from the initial batch, where plans were generated with default configurations, or from follow-up batches, where plans reflected participants' specified preferences. We find that participants were more likely to choose satisfactory plans that respect participants' preference configurations (33 participants out of 41) than the default plans (8 participants). In addition, each recourse session had a median of 3 plan iterations. In other words,

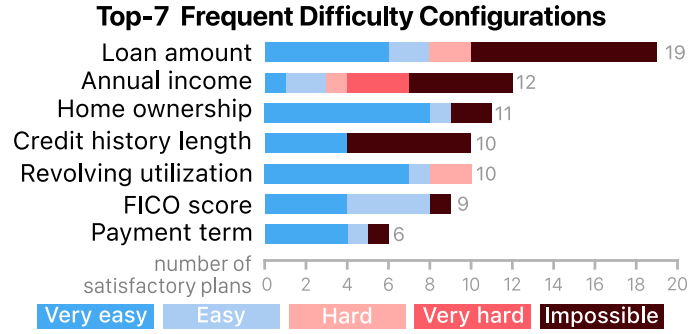


Figure 7.9: Difficulty configuration counts across frequent features highlighting variability of participants’ preferences.

on average, a participant discovered satisfactory plans after seeing about 15 plans, where the last 10 plans were generated based on their preferences. The average time to identify satisfactory plans was 8 minutes and 38 seconds.

Preference configuration is helpful. In GAM COACH, users can specify the *difficulty* and *acceptable range* to change a feature and the *max number of features* a plan can change. We find all three preferences helped participants discover satisfactory plans. Among 63 total satisfactory plans chosen by 41 participants, 49 plans (77.78%) reflected at least one difficulty configuration and 44 plans (69.84%) reflected at least one range configuration. Also, 12 participants configured the max number of features—seven participants changed it to 1 and five changed it to 2 (default is 4).

Diverse Preference Configurations. By further analyzing participants’ preferences associated with their chosen plans, we find (1) participants specified preferences on a wide range of features; (2) some features were more popular than others; (3) different participants set different preferences on a given feature. Of the 20 features, at least one participant changed the difficulty of 16 features (80%) and acceptable range of 13 features (65%). Among these configured features, participants were more likely to specify preferences on some than others [$\chi^2 = 54.37, p < 0.001$ for the difficulty, $\chi^2 = 27.68, p = 0.006$ for the acceptable range]. For example, 19 satisfactory plans reflected difficulty for `loan amount`, whereas only 1 plan reflected the difficulty for `number of past dues`. Also, there was high variability in configured preferences on popular configured feature (Fig. 7.9). For instance, 6 plans considered `loan amount` as “very easy to change,” while 9 plans deemed it as “impossible to change.” Our findings confirm hypotheses that recourse preferences can be incorporated to identify satisfactory plans [270, 278], and these preferences are idiosyncratic [273, 269].

Design Lessons. When designing recourse systems, it is useful to allow end users to specify a wide range of recourse preferences, such as difficulties to change a feature, acceptable feature ranges, and max number of features to change. Additionally, there can be predictable patterns in users’ recourse preferences—researchers can leverage these patterns to further improve user experiences. For example, developers can use the log data of an

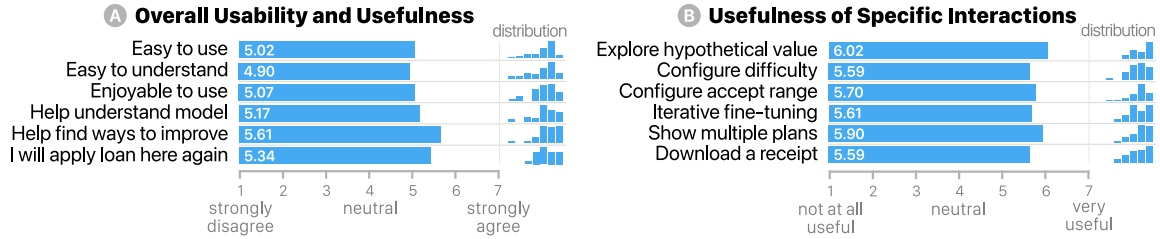


Figure 7.10: Average ratings and rating distributions from 41 participants on the usability and usefulness of GAM COACH. (A) Participants thought GAM COACH was relatively easy and enjoyable to use, and the tool helped them identify actions to obtain a preferred ML decision. (B) All interaction techniques, especially experimenting with hypothetical values, were rated favorably.

interactive recourse tool to train a new ML model to predict users’ preference configurations. Then, for a new user, developers can predict their recourse preference and use it as the tool’s default configuration.

7.5.3.3 RQ3: Interactive Algorithmic Recourse

How did participants use and perceive various *interactions* throughout the exploration task? Interestingly, 28% of participants who configured difficulty preferences had also immediately altered the difficulty levels on the same features; most of them have changed “easy” to “very easy” and “hard” to “very hard.” For acceptable ranges, the percentage is higher at 88%. It suggests participants may need iterations to learn how preference configuration works in GAM COACH and then fine-tune configurations to generate better plans—highlighting the key role of iteration in interactive recourse. Survey response show that participants found both preference configuration and iteration helpful in finding good recourse plans (Fig. 7.10B). For example, P30 commented “[I like] how easy it was to make changes to the priority of each thing. Showing that some things can be easy changes, or impossible to change, and making plans built around those.” Similarly, P19 wrote “[I like] regenerating unlimited plans until I find a fit one.”

“What-if” Questions. Besides configuring preferences, participants also engaged in other modes of interaction with GAM COACH. For example, 32 out of 41 participants experimented with hypothetical feature values (§ 7.4.2), even though it did not affect recourse generations and was not required in the task. These participants explored median of 3 unique features $\lfloor \dots \rfloor$ and a median of 5.5 hypothetical feature values $\lfloor \dots \rfloor$. These 32 participants asked what-if questions on a total of 99 features, and only 39 (39.4%) of these features were from the presented recourse plan. It suggests that participants were more interested in learning about the predictive effects of features that have not been changed by GAM COACH. After exploring what-ifs on these 99 features, participants configured at least one preference (difficulty or acceptable range) on about half of them (49 features, 49.5%). In comparison, these participants only configured preferences on 13.72% features (87 out of 634) on which they

had not explored what-ifs or had explored what-ifs *after* configuring preferences. It shows that participants were more likely to customize features on which they had explored hypothetical values [$\chi^2 = 85.459, p < 0.00001$]. Finally, 20 out of these 32 participants (62.5%) chose a satisfactory plan with a changed feature on which they had explored what-ifs. It may suggest participants preferred recourse plans that changed features on which they had explored what-ifs, but this result is not statistically significant [$\chi^2 = 2.0, p = 0.1573$].

By analyzing survey responses, we also find that asking what-if questions was one of the participants' favorite features (Fig. 7.10B). For example, P12 wrote “[I like] how it adjusts the plans in real time and gives you an answer if the loan will be approved.” Throughout the task, participants also frequently used the tooltip annotations to inspect the decision score bar (median 8 times per participant) and check the meaning of different features (median 25 times)—highlighting the importance of clearly explaining visual representations and terminologies in interactive recourse tools.

Counterintuitive recourse plans. We asked participants to report strange recourse plans that GAM COACH could rarely suggest, such as to lower `annual income` for loan approval. To our surprise, 7 out of 41 participants had encountered and reported these counterintuitive plans! For example, P6 was confused that some plans suggested conflicting changes on the same feature: “One plan told me to increase the loan amount by \$13 while another plan told me to decrease by \$1,613.” Another interesting case was P39: “I don’t understand how purpose changes approval decision. Something like ‘mortgage’ I understand, but changing something and all of a sudden you can do a wedding but not home improvement? Like what?” First, P39 found it counterintuitive that GAM COACH includes the categorical feature `loan purpose` as a changeable feature because they thought the model decision should be independent of the `loan purpose`. Then, through experimenting with hypothetical values, P39 was baffled by the observation that two different purposes (wedding and home improvement) resulted in two distinct model decisions. Some other participants also attributed these strange patterns as reasons why they skipped some plans (§ 7.5.3.1). This finding provides empirical evidence that with transparency, everyday users can discover potentially problematic behaviors in ML models.

Design Lessons. Overall, interactivity helps users identify satisfactory recourse plans, and users appreciate being able to control recourse generation. In addition, users like being able to ask what-if questions; experimenting with hypothetical feature values also helps them find satisfactory recourse plans. However, it takes time and trial and error for users to understand how preference configurations affect recourse generation. Therefore, future interactive recourse tools can improve user experience by focusing on improving learnability and reversibility. Also, our study shows that interactivity and transparency could occasionally confuse users with counterintuitive recourse plans. Therefore, future researchers and developers should carefully audit and improve their ML models before deploying interactive recourse tools.

7.5.3.4 Usability

Our survey included a series of 7-point Likert-scale questions regarding the usability of GAM COACH (Fig. 7.10A). The results suggest that the tool is relatively easy to use (average 5.02), easy to understand (average 4.90), and enjoyable to use (average 5.07). However, some participants commented that the tool was not easy to learn at first and may be too complex for users with less knowledge about loans. For example, P5 wrote “*Without the tutorials, it would have taken me much longer to learn how to navigate the program, because it is not very intuitive at first.*” Similarly, P8 wrote “*I am decent with finances, but I’d imagine that other people would have more difficulty [using the tool].*” Our participants were MTurk workers, who are similar to the demographics of American internet users as a whole, but slightly younger and more educated [305, 309]. Therefore, GAM COACH might be overwhelming for real-life loan applicants who are less familiar with web technology or finance. Participants also provided specific feedback for improvement, such as designing a better way to *store* and *compare* all generated plans. Currently, users would lose unsaved plans when generating new plans, and users could only compare different recourse plans in the *Bookmarks window* (§ 7.4.3).

7.6 Limitations

We acknowledge our work’s limitations regarding GAM COACH’s generalizability, usage scenarios, and user study design.

Generalizability of GAM COACH. To design and develop the first interactive algorithmic recourse tool that enables end users to fine-tune recourse plans with preferences, we ground our research in GAMs, a class of accurate and transparent ML models with simple structures. This approach enables us to generate customizable CF examples efficiently. However, not all CF generation algorithms allow users to specify the feature-level distance functions, acceptable ranges, and max number of features that a CF example can change. Therefore, while the GAM COACH interface is model-agnostic, it does not directly support all existing ML models and CF generation methods. Also, our novel CF generation algorithm is tailored to EBMs. However, one can easily adapt our linear constraints to generate customizable CF examples for linear models [264]. For more complex non-linear models (e.g., random forest, neural networks), one can apply our method to a linear approximation [288] of these models (Equation 7.3.2). We also acknowledge that similar to most existing CF generation algorithms [280, 270], our algorithm assumes all features to be independent. However, in practice, many features can be associated. For example, changing `credit utilization` is likely to also affect a user’s `FICO score`. Future work can generalize our algorithm to dependent features by modeling their casual relationships [268].

Hypothetical Usage Scenarios. We situate GAM COACH in lending and government

funding settings (§ 7.4.4), two most cited scenarios in existing CF literature [279, 270]. It is important to note that none of the authors have expertise in law, finance, or political science. Therefore, to adapt GAM COACH for use in real lending and government funding settings, it would require more research and engaging with experts in the legal and financial domains as well as people who would be impacted by the systems. In addition, we use LendingClub [258] and Communities and Crime [291], two largest suitable datasets we have access to (§ 7.5), to simulate two usage scenarios and design our user study. These two datasets can have different features and sizes from the data that are used in practice. Therefore, before adapting GAM COACH, researchers and developers should thoroughly test our tool on their own datasets.

Simulated Study Design. To study how end users would use interactive recourse tools, we recruited MTurk workers and asked them to pretend to be rejected loan applicants, and we logged and analyzed their interactions with GAM COACH. We designed the task to encourage and help participants simulate the scenario (e.g., rewarding bonus, supporting participants to input data or choose data from multiple random samples). However, participants’ usage patterns and reactions may not fully represent real-life loan applicants. We chose to simulate a lending scenario because (1) crowdworkers may have encountered lending, (2) it does not require expert knowledge, and (3) we have access to a large and real US-based lending dataset. We acknowledge that participants’ usage patterns may not fully represent users in other domains. Therefore, it would require further research with actual end users (e.g., loan applicants, county executives, and bail applicants) to study how GAM COACH can aid them in real-world settings. In our study, we only collected participants’ familiarity with ML. As MTurk workers tend to be younger and more educated than average internet users [305, 309], future researchers can collect more self-reported demographic information (e.g., age, education, sex) to study if different user groups would use an interactive recourse tool differently.

Observational Study Design. Our observational log study can provide a portrait of users’ natural behaviors when interacting with interactive algorithmic recourse tools and scale to a large number of participants [310]. However, it lacks a control group. As algorithmic recourse research and applications are still nascent, the community has not yet established a recommended workflow or system that we can use as a baseline in our study. Our main goal is to study how *recourse customizability* can help users discover useful recourse plans. Therefore, to mitigate the lack of a control group, we offer participants the option to *abstain from customizing recourse plans* to probe into the usefulness of recourse customizability. In our analysis, we compare both (1) the numbers of participants who specify recourse preferences and who do not, (2) and the numbers of satisfactory plans generated with a default configuration and satisfactory plans generated with a participant-configured preference (§ 7.5.3.2). Finally, with our open-source implementation (§ 7.4.5), future researchers can use GAM COACH as a baseline system to evaluate their interactive recourse tools.

7.7 Discussion

Reflecting on our end-to-end realization of interactive algorithmic recourse—from UI design to algorithm development and a user study—we distill lessons and provide a set of future directions for algorithmic recourse and ML interpretability.

Too much transparency. GAM COACH uses a glass-box model, provides end users with complete control of recourse plan generation, and supports users to ask “what-if” questions with any feature values. One might argue that GAM COACH is too transparent and too much transparency makes the tool unfavorable, because (1) end users can use this tool for gaming the ML model [311, 312] and (2) this tool fails to protect the decision maker’s model intellectual property [59]. We acknowledge these concerns. As recourse research and applications are still nascent, it is challenging to know how we can balance the benefits of transparency and human agency and the risk of revealing too much information about the ML model. Our user study shows that with transparency end users can discover and are often puzzled by counterintuitive patterns in ML models. We believe if GAM COACH is adopted, it has the potential to incentivize decision makers to create better models in order to avoid confusion as well as model exploitations. As one of the furthest realizations of ML transparency, GAM COACH can be a research instrument that facilitates future researchers to study the tension between *decision makers* and *decision subjects*, and identify the right amount of transparency that most benefits both parties. Then, to adopt GAM COACH in practice, ML developers can remove certain functionalities or impose recourse constraints accordingly. For example, if a bank is offering GAM COACH and is worried about people gaming the system by changing certain features that do not actually improve their creditworthiness (e.g., opening more credit cards), they could insert their own optimization constraints that prevent these features from being modified.

Transparent ML models for algorithmic recourse. Black-box ML models are popular across different domains. To interpret these models, researchers have developed post-hoc techniques to identify feature importance [e.g. 249, 243] and generate CF examples [e.g. 265, 266]. However, Rudin argues that researchers and practitioners should use transparent ML models instead of black-box models in high-stake domains due to transparent models’ high accuracy and explanation fidelity. The design of GAM COACH is based on GAMs, a state-of-the-art transparent model [11, 253]. We would like to broaden the perspective of using transparent models reflecting on our study. We find that GAM COACH provides opportunities for everyday users to discover counterintuitive patterns in the ML model. It implies that ML developers and researchers can also use GAM COACH as a penetration testing tool to detect potentially problematic behaviors in their models. Note that both black-box and transparent learning methods would have learned these counterintuitive behaviors [11], but with a transparent model, developers can further *vet* and *fix* these behaviors. As an example, an ML developer training a GAM can use GAM COACH to iteratively generate recourse

plans for potential users (e.g., training data where the model gives unfavorable predictions). If they identify strange suggestions, they can use existing interactive tools [7, 247] to visualize corresponding shape functions to pinpoint the root cause of these counterintuitive patterns, and then edit shape function parameters to avoid them from happening during recourse deployment. Future research can leverage transparent models to distill guidelines to audit and fix models before recourse deployment.

Put users at the center. We have encountered many challenges in transforming technically sound recourse plans into a seamless user experience. As the end users of recourse tools are everyday people who are less familiar with ML and domain-specific concepts, one of our design goals is to help them understand necessary concepts and have a frictionless experience (G4). GAM COACH aims to achieve this goal by following a progressive disclosure and details-on-demand design strategy [296, 295] and presenting textual annotations to explain visual representations in the tool. However, our user study suggests that few users might still find it challenging to use GAM COACH at first (§ 7.5.3.4). During our development process, we identify many edge cases that a recourse application would encounter in practice, such as features requiring integer values (e.g., `FICO score`), features using log transformations (e.g., `annual income`), or features less familiar to everyday users (e.g., `credit utilization`). Our open-source implementation handles these edge cases, and we provide ML developers with simple APIs to add descriptions for domain-specific feature names in their own instances of GAM COACH. However, these practical edge cases are rarely discussed or handled in the recourse research community, since (1) the field of algorithmic recourse is relatively nascent, (2) and the main evaluation criteria of recourse research are distance-based statistics instead of *user experience* [280]. Therefore, in addition to developing faster techniques to generate more actionable recourse plans, we hope future researchers engage with end users and incorporate user experience into their research agenda. Besides interactive visualization, researchers can also explore alternative mediums to communicate and personalize ML recourse plans and model explanations, such as through a textual [314] or multi-modal approach [315].

7.8 Conclusion

As ML models are increasingly used to inform high-stakes decision-making throughout our everyday life, it is crucial to provide decision subjects ways to alter unfavorable model decisions. In this work, we present GAM COACH, an interactive algorithmic recourse tool that empowers end users to specify their preferences and iteratively fine-tune recourse plans. Our tool runs in web browsers and is open-source, broadening people’s access to responsible ML technologies. We discuss lessons learned from our realization of interactive algorithmic recourse and an online user study. We hope our work will inspire future research and development of user-centered and interactive tools that help end users restore their human agency and eventually trust and enjoy ML technologies.

Part III

DEMOCRATIZE HUMAN-CENTERED AI


Overview

So far we have developed novel techniques and tools that explain AI to a wide range of stakeholders (Part I) and empower individuals to exert human agency and guide AI systems (Part II). However, these endeavors are only useful if they are adopted in practice. Within the context of an ever-expanding body of research on human-centered and responsible AI, a critical question arises: How can we democratize access to human-centered AI techniques and promote its broad adoption? Our work addresses this challenge by integrating human-centered AI practices into AI practitioners' existing workflows.

Recently, researchers have made breakthroughs in large language models (LLMs) that excel in various NLP tasks ranging from classification to translation. With a growing number of accessible LLMs and prompting tools such as GPT Playground and MakerSuite, we see an expanding group of "AI prototypers". Research on human-centered and responsible AI has shown great risks in developing and deploying LLM-powered applications without caution. Therefore, to foster the awareness of responsible AI among AI prototypers, we propose FARSIGHT, an *in situ* tool that provides in-context feedback to help AI prototypers envision potential use cases, stakeholders, and harms based on the prompts they are writing.

To lower the barrier to learning and applying human-centered AI practices, we integrate FARSIGHT into AI practitioners' existing workflows. For example, our tool helps practitioners envision potential harms associated with their AI features when they are crafting prompts in Google AI Studio or Jupyter Notebooks. After using FARSIGHT, AI practitioners in our user study are better able to independently identify potential harms associated with a prompt and find our tool more useful and usable than existing resources. Their qualitative feedback also highlights that FARSIGHT makes it easy to consider end-users and think beyond immediate harms.

Chapter 8

FARSIGHT: Fostering Responsible AI Awareness During Early AI Application Prototyping. Zijie J. Wang, Chinmay Kulkarni, Lauren Wilcox, Michael Terry, and Michael Madaio. *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024.  PDF

CHAPTER 8

FARSIGHT: FOSTERING RESPONSIBLE AI AWARENESS DURING AI PROTOTYPING

Prompt-based interfaces for Large Language Models (LLMs) have made prototyping and building AI-powered applications easier than ever before. However, identifying potential harms that may arise from AI applications remains a challenge, particularly during prompt-based prototyping. To address this, we present FARSIGHT, a novel *in situ* interactive tool that helps people identify potential harms from the AI applications they are prototyping. Based on a user’s prompt, FARSIGHT highlights news articles about relevant AI incidents and allows users to explore and edit LLM-generated use cases, stakeholders, and harms. We report design insights from a co-design study with 10 AI prototypers and findings from a user study with 42 AI prototypers. After using FARSIGHT, AI prototypers in our user study are better able to independently identify potential harms associated with a prompt and find our tool more useful and usable than existing resources. Their qualitative feedback also highlights that FARSIGHT encourages them to focus on end-users and think beyond immediate harms. We discuss these findings and reflect on their implications for designing AI prototyping experiences that meaningfully engage with AI harms.

8.1 Introduction

As artificial intelligence (AI) becomes increasingly integrated into our everyday lives, mitigating the societal harms posed by AI technologies has never been more important. In response to the demand for accountable and safe AI, there have been growing efforts from both industry and academia towards responsible design and development of AI [316, 86]. The majority of these endeavors focus on machine learning (ML) experts, such as ML developers and other AI practitioners. For example, researchers have introduced techniques that help ML developers interpret ML models [243, 249, 7] and assess model

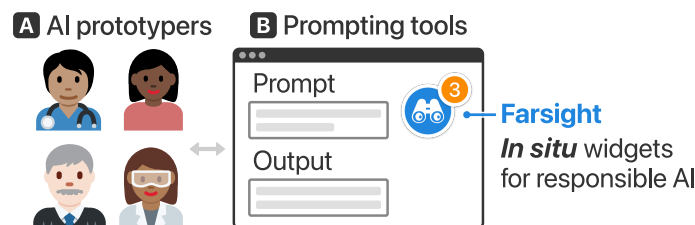


Figure 8.1: (A) AI prototypers from diverse backgrounds and roles use (B) prompting tools to prototype AI applications. FARSIGHT provides a range of *in situ* widgets for these tools, helping AI prototypers envision the potential harms of their AI applications during an early prototyping stage.

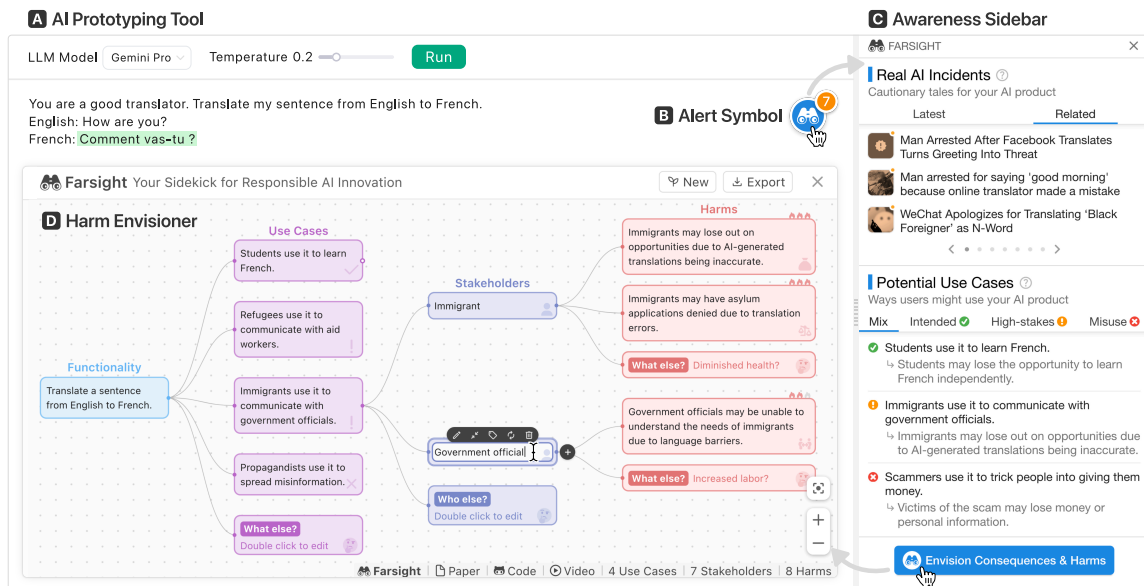


Figure 8.2: With *in situ* interfaces and novel techniques, FARSIGHT empowers AI prototypers to envision potential harms that may arise from their large language models (LLMs)-powered AI applications during early prototyping. (A) In this example, an AI prototyper is creating a prompt for an English-to-French translator in a web-based AI prototyping tool. (B) The *Alert Symbol* from FARSIGHT warns the user of potential risks associated with their AI application. (C) Clicking the symbol expands the *Awareness Sidebar*, highlighting news articles relevant to the user’s prompt (top), and LLM-generated potential use cases and harms (bottom). (D) Clicking the blue button opens the *Harm Envisioner* that allows the user to interactively envision, assess, and reflect on the potential use cases, stakeholders, and harms of their AI application with the assistance of an LLM.

fairness [317, 318, 319]. Additionally, researchers have also proposed frameworks that target ML developers’ workflows, such as improving data collection and annotation practices [320, 321, 322], documenting training data and models [323, 324, 325], and anticipating an ML product’s potentials for harms [326, 85].

However, more recently, we have witnessed a rapid advancement of large language models (LLMs) such as Gemini [330] and GPT-4 [331], alongside the emergence of prompt-based interfaces like Google AI Studio [327], GPT Playground [332], AI Chains [333], and Workflow [334] (Fig. 8.1B). These general-purpose models and easy-to-use interfaces have significantly increased access to the process of prototyping and building diverse AI-powered applications—leading to a paradigm shift in AI development workflows that poses unique challenges to responsible AI, including introducing new potential harms to avoid [97], as well as challenges applying existing responsible AI practices [335].

Many people who use prompts to create AI applications now encompass a broader spectrum of roles beyond traditional ML experts (Fig. 8.1A), such as designers, writers, lawyers, and everyday users [336, 337, 338, 339], whereas existing responsible AI research often targets ML experts such as ML engineers and data scientists [340, 341]. Many users of AI prompt-based prototyping interfaces [e.g., 327, 333, 332, 334], or “AI prototypers”

Farsight Fits into AI Prototypers' Prompting Workflows

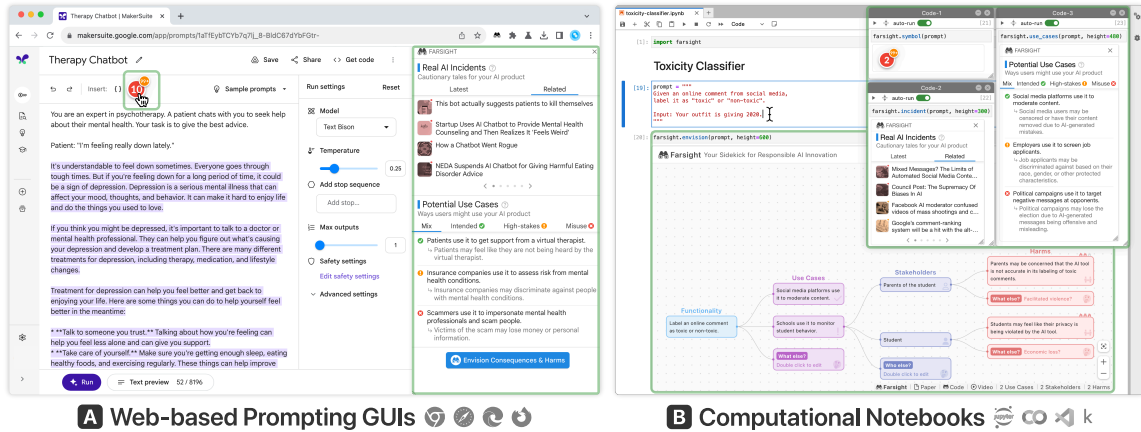


Figure 8.3: FARSIGHT fits into AI prototypers' prompting workflows including prompting GUIs and computational notebooks. (A) When an AI prototyper writes prompts for a therapy chatbot in Google AI Studio [327], FARSIGHT's Chrome extension alerts the user about related accidents and potential harms. (B) When an AI prototyper writes prompts for a toxicity classifier in Jupyter Notebook [328, 329], FARSIGHT's Python library shows potential negative consequences of this classifier.

[cf. 337] do not have experience in AI or computer science, which can lead to challenges in anticipating the consequences of their AI applications [316]—a difficult task even for computer science faculty and AI researchers [94, 89]. Furthermore, LLMs demonstrate a wide range of capabilities that are continually being discovered across various contexts, including tasks such as summarization, classification, and translation [96, 342]. This characteristic of LLMs gives rise to *complex* and *uncertain* impacts of LLM-powered applications [343], presenting a significant departure from the classical ML models targeted by existing responsible AI endeavors [335, 97] and introducing a new layer of complexity for responsible AI researchers to help AI developers anticipate downstream consequences.

To help address these challenges in applying responsible AI practices to LLM-powered AI applications, we present FARSIGHT (Fig. 8.2, Fig. 8.1B), an interactive tool to help AI prototypers identify potential harms of their LLM-powered applications—a key early step in harm prevention and mitigation [344, 345, 346, 347, 326]—during the prototyping stage. Using FARSIGHT as a probe, we conduct multiple mixed-method user studies to investigate (1) how an early-stage intervention changes AI prototypers' awareness of and approach to identifying harms, (2) the effectiveness of our tool in helping people envision harms, and (3) the challenges AI prototypers face during this harm envisioning process. **We contribute:**

- **FARSIGHT, the first *in situ* interactive harm envisioning tool that empowers AI prototypers** to identify potential harms that may arise from their prompt-based AI applications, directly within their prompting environments (Fig. 8.2, Fig. 8.1). Inspired by prior harm envisioning frameworks [326, 95, 85] and *in situ* security alert tools [129, 130, 131], FARSIGHT overcomes unique design challenges identified from a literature review and a co-design user study with 10 AI prototypers (§ 8.2).

- **Novel techniques and interactive system designs** to foster responsible AI awareness among AI prototypers. Given a user’s prompt, FARSIGHT leverages embedding similarities to surface news articles about relevant AI incidents from the AI Incident Database [348] and uses LLMs to generate potential use cases, impacted stakeholders, and harms for AI prototypers to review, edit, and add to. Applying a progressive disclosure design [296], our tool fits into users’ diverse prompting workflows. With a novel adaptation of node-link diagrams [349], FARSIGHT enables users to interactively visualize, generate, and edit use cases, stakeholders, and harms (§ 8.3).
- **Empirical findings about harm envisioning processes from a co-design study and an evaluation study.** During our design of FARSIGHT, we conducted a co-design study with 10 AI prototypers to evaluate our design ideas and generate new ideas (§ 8.2). After developing FARSIGHT, we conducted an evaluation user study with 42 AI prototypers to examine the effectiveness of FARSIGHT in aiding users to brainstorm harms and improving their ability to independently identify harms. Our mixed-method analysis highlights that, after using FARSIGHT, AI prototypers are better able to independently identify potential harms that might arise from an application developed with a given prompt, and participants report that our tool is more useful and usable than existing resources. In particular, FARSIGHT encourages users to shift their focus from the AI model to the end-users, providing them with a broader perspective to consider indirect stakeholders and cascading harms (§ 8.5).
- **An open-source, web-based implementation** that lowers the barrier to applying responsible AI practices. We develop FARSIGHT with cutting-edge web technologies, such as Web Components [350] and WebGL [351], so that it can be easily integrated into any web-based prompt development environments, such as Google AI Studio and Jupyter Notebook (Fig. 8.3). We open source¹ FARSIGHT as a collection of reusable interactive components that future researchers and designers can easily adopt (§ 8.3.4). To see a demo video of FARSIGHT, visit <https://youtu.be/B1SFbGk01Hk>.

8.2 Formative Study & Design Goals

To identify the needs and potential challenges faced by users in envisioning harms, we conducted a formative co-design study to investigate (1) how AI prototypers envision harms (if they do), (2) what design ideas are most helpful for them, and (3) how to motivate users to think about potential risks when prototyping an AI application. In this section, we report our findings from the formative co-design study, and in § 8.5, we report on our findings from a subsequent evaluation user study.

¹FARSIGHT code: <https://github.com/PAIR-code/farsight>

Table 8.1: The co-design user study includes 10 participants with diverse roles. All participants have experience in prompting LLMs. Four participants who self-reported having expertise in responsible AI are marked with asterisks (*).

Participant Roles	Participant IDs
Software Engineer	1*, 4, 5, 6, 7, 10
Research Scientist	3*, 8*
Technical Writer	2
Program Manager	9*

8.2.1 Co-design Study

Participants. To inform our tool’s design, we conducted a co-design user study with 10 AI prototypers based in the United States. These participants were recruited from Google through internal mailing lists. Our recruitment criteria required participants to have experience using an internal prompt-crafting tool, PromptMaker [337], which is similar to Google AI Studio [327] and GPT Playground [332]. Each session was 60 minutes, and each participant received an average of \$50 USD in their choice of a gift card or a donation to their preferred charity. Among the 10 participants (U1–U10), 6 identified as men, 3 identified as women, and 1 identified as non-binary. Four participants self-reported having expertise in responsible AI. Information about participants’ job roles is listed in Table 8.1. All participants are our targeted users (AI prototypers).

Procedure. We structured our study as a “during-design co-design study” [352]. Participants were asked to bring a recent prompt that they had written to the study. The study started with a semi-structured interview regarding participants’ prompting workflows and their experience in thinking about potential harms linked to their applications. Then participants were asked to use our very early-stage design prototypes to envision potential harms associated with their application while thinking aloud. Participants were also presented with low-fidelity sketches for our other design ideas. These prototypes and sketches can be found in Fig. 8.6. Finally, we asked participants to rate and provide feedback on all of our design ideas and generate their own design suggestions.

Design feedback. Interestingly, although perhaps not surprisingly [cf. 340], none of the 6 participants without expertise in responsible AI reported that they typically considered the potential harms of their AI prototypes when writing prompts, while 3 of the 4 participants with expertise in responsible AI did report typically anticipating harms during the prototyping process. Participants’ ratings were shown in Fig. 8.4. Overall, participants favored using AI to generate use cases of their AI prototypes, potential stakeholders, and potential harms. Many participants also highlighted the importance of being able to edit AI-generated content and control the generation direction (U4, U8). On the other hand, participants were less in favor of more distracting design ideas (e.g., an anthropomorphized assistant tool

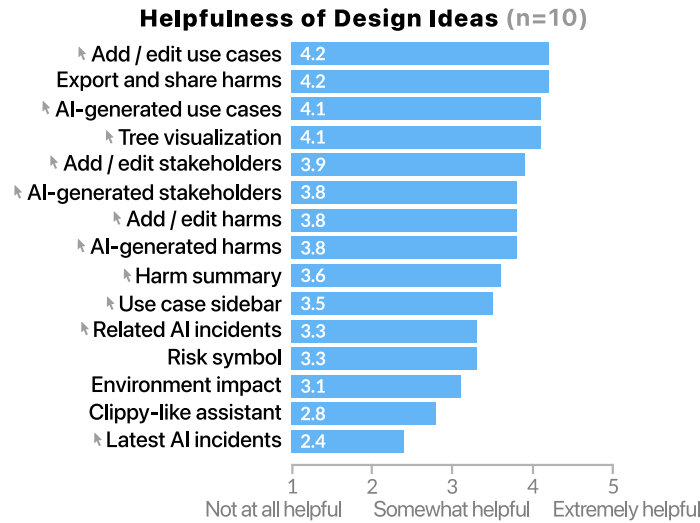


Figure 8.4: Average ratings on our design ideas from 10 AI prototypers. Features marked with ↵ were presented to participants as early-stage prototypes, while other features were presented as sketches (see details in Fig. 8.6).

similar to Microsoft Office’s Clippy) or irrelevant content (e.g., the latest, rather than the most relevant AI incidents). Participants also provided us with helpful usability feedback that we integrated into our final design of FARSIGHT (§ 8.3).

New design ideas. Participants generated many interesting design ideas to help raise responsible awareness among AI prototypers. For example, participants recommended categorizing AI-generated harms (U1, U5), allowing users to rate the severity of harms (U6), and using users’ input to steer AI generation (U10). We integrated these design ideas into the final design of FARSIGHT (§ 8.3). Some other interesting design ideas include designing a game-like reward system to incentivize users to anticipate harms (U5), building online communities to allow users to share their envisioned harms using FARSIGHT and seek support (U2), allowing real-time collaborative harm envisioning similar to Google Slides (U1, U4), and automatically revising a user’s prompt to address identified harms (U4). We discuss the implications of these design ideas in user motivation (§ 8.6.1) and mitigation strategies (§ 8.6.3).

8.2.2 Design Goals

Based on our literature review and findings and early feedback from the co-design user study, we identify the following five design goals (G1–G5) for FARSIGHT.

G1. Guide users in imagining use cases. Existing research highlights the challenges faced by ML practitioners when attempting to anticipate the uses of their ML-powered applications and how different individuals or groups may be affected [89, 353, 94, 354]. Confirming this, software engineer U6 noted “*You don’t really know how your tool*

could be used, so it's really hard to envision what harms would be." The availability of LLMs and prompt-crafting tools has broadened the spectrum of AI prototypers to include people without prior technology development experience [336, 337], which can further magnify the challenges associated with envisioning diverse use cases for AI applications. Therefore, we design FARSIGHT to help AI prototypers with diverse backgrounds to brainstorm a wide array of use cases for their AI applications.

- G2. Help users understand, organize, and prioritize harms.** Depending on an AI application's goal, implementation, and context, some harms are more salient than others [95, 355]. To help AI prototypers assess harms, FARSIGHT should first help them understand *where* and *how* harms might occur and *who* might be impacted, by connecting harms to use cases and stakeholders [354, 356, 326]. Participants expressed a desire for the ability to categorize (U1, U5) and rate the severity (U6) of harms. To meet these needs, we aim to design an easy-to-use interface that empowers users to navigate, comprehend, and label harms within diverse potential harm scenarios.
- G3. Fit into current workflows and mitigate habituation.** In our co-design study, none of the 6 participants without expertise in responsible AI had previously thought about harms when writing prompts. We also found some participants were not incentivized to anticipate harm on their own; for example, U6 explained "*To be honest, as a software engineer, I don't use policy tools [compliance tools like checklists] unless I have to.*" Thus, to make FARSIGHT easy to adopt, we aim to take inspiration from *in situ* warning tools [e.g., 132, 133, 134] to design it in a way that fits into AI prototypers' existing workflows instead of introducing new workflows. In addition, we aim to apply strategies like varying content [139] and promoting user input [143] to mitigate habituation—a common pitfall of in-context warning designs [139, 140].
- G4. Promote user engagement and provide compelling examples.** Prior research highlights that the effectiveness of warning tools depends on their clarity and persuasiveness [135, 136]. As we are targeting AI prototypers with diverse experience in AI and responsible AI, FARSIGHT should be easy to use and understand. When asked what would help them envision potential harms for their AI applications, many participants mentioned referring to prior examples of AI harms (U1, U2, U8). For instance, U2 said "*Giving some specific real [harm] examples for different types of seemingly innocuous applications would help alert people [to consider harms].*" Therefore, we aimed to integrate real examples in FARSIGHT to motivate and help users understand the potential risk of their applications. Participants like being able to control the harm envisioning process (Fig. 8.4), and active participation is a key factor in learning [357]—essential to foster AI prototypers' ability to independently identify harms. Thus, FARSIGHT is designed to provide users with human agency and encourage users to actively and critically think about harms.



Figure 8.5: Three alert modes of the *Alert Symbol*.

G5. Open-source and adaptable implementation. Given the ever-expanding array of LLMs and prompt-crafting tools [358], our approach in designing FARSIGHT is to ensure it remains adaptable to this dynamically evolving landscape. We aimed to design FARSIGHT to be model-agnostic and environment-agnostic, thereby making it accessible to users of different LLM models (e.g., Gemini [330], GPT-4 [331], Llama 2 [359]) and prompt-crafting interfaces (e.g., GPT Playground [332], Google AI Studio [327], Wordflow [334]). Furthermore, we open source our implementation to foster future advancements in the design, research, and development of responsible AI tools.

8.3 User Interface

Following the five design goals (G1–G5), we present FARSIGHT, the first *in situ* interactive tool that aims to foster responsible AI awareness among AI prototypers during the AI prototyping process. FARSIGHT is designed to be a plugin of any web-based prompt-crafting tools. FARSIGHT’s interface employs progressive disclosure [296], enabling users to smoothly transition across three main components, with each phase increasing the level of user engagement. The *Alert Symbol* (§ 8.3.1) presents an always-on symbol that shows the approximated alert level of a user’s current prompt; the *Awareness Sidebar* (§ 8.3.2) highlights news articles about related AI incidents and LLM-generated use cases and harms; and the *Harm Envisioner* (§ 8.3.3) visualizes LLM-generated harms and allows users to edit, add, and share harms. Examples in this section use PaLM 2 model through its APIs; we chose this model because it provided free API access to the public during our design process. Researchers and designers can easily replace PaLM 2 model with other LLMs by changing the API endpoints in FARSIGHT.

8.3.1 Alert Symbol

The *Alert Symbol* is an always-on display on top of the AI prototyping tool, displaying the alert level of a user’s prompt (Fig. 8.5). Every time the user runs their prompt, the *Alert Symbol* updates the alert level using the new prompt. Based on the computed alert level, there are three modes (Fig. 8.5), each characterized by a progressively more attention-grabbing style. Thus, FARSIGHT only disrupts AI prototypers’ flow when their prompts require more caution (G3).

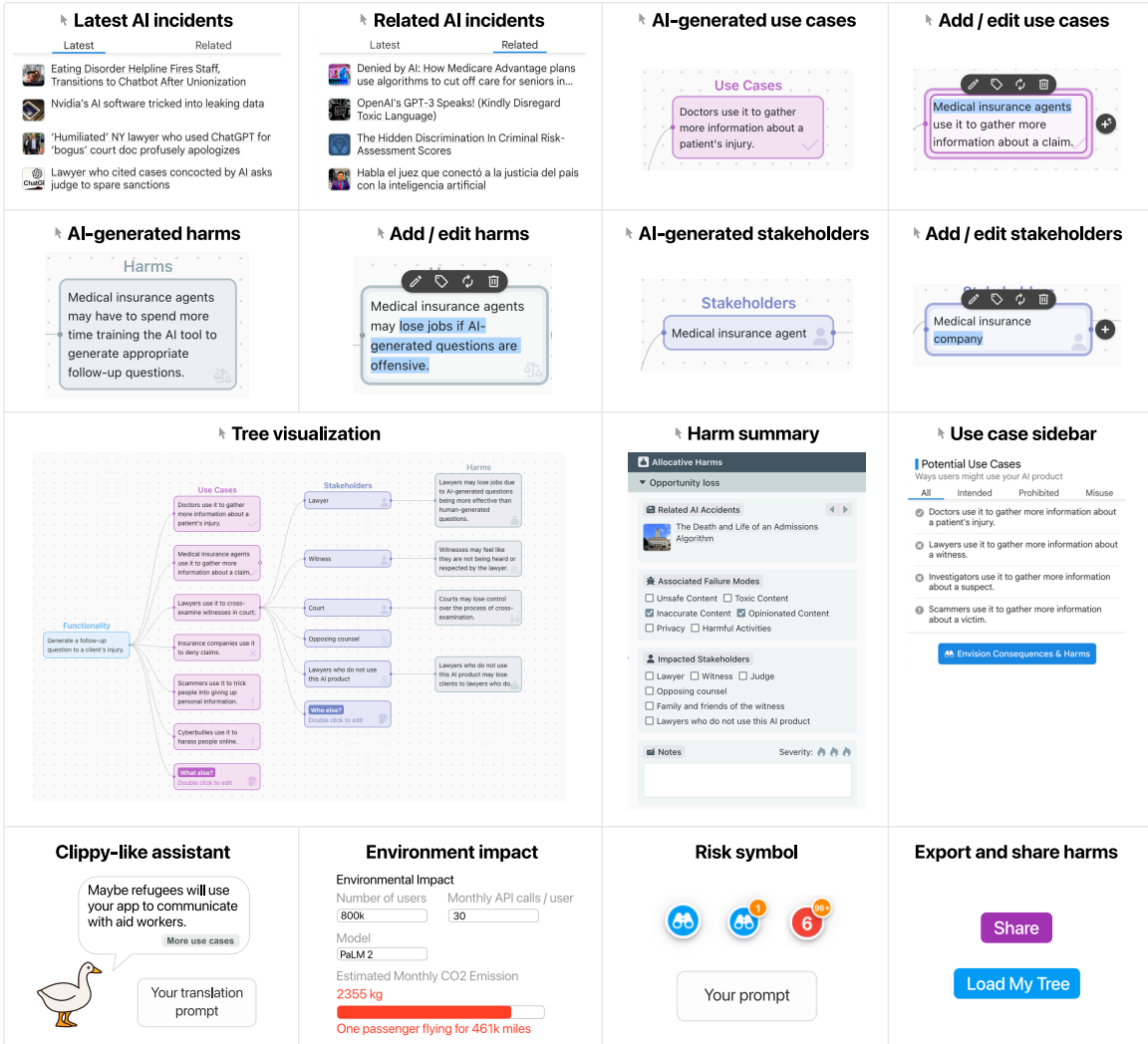


Figure 8.6: To evaluate our early FARSIGHT designs and generate more design ideas, we conducted a co-design study (§ 8.2.1) with 10 AI prototypers. Participants were asked to use our very early-stage design prototypes (shown in cells labeled with ✎) to envision potential harms associated with their application while thinking aloud. Participants were also presented with low-fidelity sketches for our other design ideas (shown in cells in the last row). The ratings of design ideas are in Fig. 8.4.

Calculating the Alert Level. Auditing and quantifying the societal risk of LLM-powered applications is an open research problem [360]. To categorize the potential harms that might arise from users' prompts, we propose a novel technique that uses the similarity between the prompt and previously documented AI incident reports as a proxy for the prompt's alert level. First, we use an LLM to extract high-dimensional latent representations (embeddings) of all AI incident reports indexed in the AI Incident Database [348], which includes more than 3k community-curated news reports about AI failures and harms. Then, we extract the embedding of the user's prompt and compute pairwise cosine distances between the prompt embedding and AI incident report embeddings. We label each incident report as `irrelevant`, `remotely relevant`, `moderately relevant` based on two distance thresholds 0.69 and

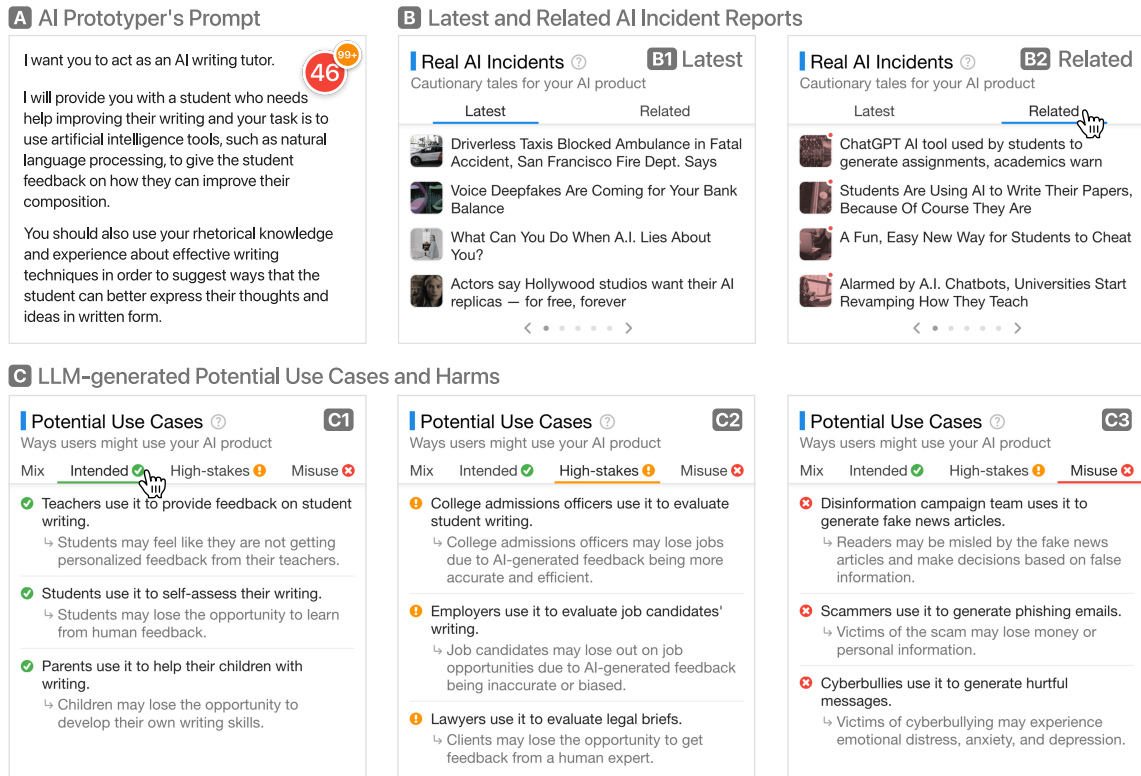


Figure 8.7: The *Awareness Sidebar* provides *in situ* information to remind AI prototypers of potential risks. (A) Given a user’s current prompt, (B) the *Incident Panel* shows the (B1) latest and (B2) related AI incident reports sampled from the AI Incident Database [348]. (B2) The related AI incident tab is the default view, which uses text embedding similarities between the user’s prompt and all AI incident reports to surface relevant reports. (C) The *Use Case Panel* leverages LLM to generate potential use cases and harms. Each use case is classified by an LLM and organized into (C1) *intended*, (C2) *high-stakes*, and (C3) *misuse* tabs.

0.75. We determine these two thresholds from an experiment with 1k random prompts. Researchers can easily adjust these two thresholds to calibrate an article’s relevancy.

Finally, we show the numbers of AI incidents that are classified as `remotely relevant` in an orange circle and `moderately relevant` in a red circle (Fig. 8.5) as a proxy of the prompt’s potential risk. In other words, we consider a prompt to have a higher risk if many AI incident reports are semantically and syntactically similar to it.

8.3.2 Awareness Sidebar

After a user clicks the *Alert Symbol*, the *Awareness Sidebar* (Fig. 8.7) expands from one side edge of the AI prototyping tool (G3), highlighting potential consequences of AI applications or features that are based on the user’s current prompt. We use a real prompt from *Awesome ChatGPT Prompts* [361] in the example in Fig. 8.7.

Incident Panel. To encourage users to consider potential risks associated with their prompts (Fig. 8.7A), the *Incident Panel* highlights news headlines of AI incidents that are

relevant to the user’s prompt (Fig. 8.7-B2). These incidents comprise the top 30 incident reports that are classified as `moderately relevant` or `remotely relevant`, sorted in reverse order based on their embedding’s cosine distances to the embedding of the user’s prompt. The thumbnails are color-coded based on the incident’s relevancy level. Users can click the headline or the thumbnail to open the original incident report in a new tab. These real AI incidents can function as cautionary tales [354, 82] reminding users of potential AI harms (G4).

Use Case Panel. To help users imagine how their AI prototype may be used in AI applications or features (G1), the *Use Case Panel* (Fig. 8.7C) presents a diverse set of potential use cases that are generated by an LLM. Each use case is shown as a sentence describing how a particular group of end-users could use this AI application in a specific context. For example, for a writing tutor prompt, a potential use case can be “*teachers use it to provide feedback on student writing.*” (Fig. 8.7-C1). We also use an LLM to generate a potential harm that could occur within that use case, shown below the use case sentence. For example, a harm for the teacher feedback use case can be “*students may feel like they are not getting personalized feedback from their teachers.*” We use few-shot learning to prompt the LLM to generate use cases and harms, whereas we generate use cases, stakeholders, and harms in *Harm Envisioner* (§ 8.3.3). We open-source all of our prompts.

To help users assess and organize use cases and harms (G2), we also leverage an LLM to categorize each use case as `intended`, `high-stakes`, or `misuse`, although we acknowledge that these may vary by use cases, development and deployment contexts, as well as relevant policies or regulatory frameworks in various jurisdictions. These three categories are introduced by responsible AI researchers to help ML developers structure their harm envisioning process [355]. The `intended` use cases are those that align with the development target use cases. The `high-stakes` use cases encompass those that may arise in high-stakes domains, such as medicine, finance, and the law. The `misuse` category includes scenarios where malicious actors exploit the AI application to cause harms. The *Use Case Panel* organizes use cases and harms into three tabs (Fig. 8.7-C1–3) based on their categories. The first tab, “mix”, provides an overview by showing one use case and its corresponding harm from each of the other tabs.

8.3.3 Harm Envisioner

Both the *Alert Symbol* and the *Awareness Sidebar* provide easy-to-understand in-context reminders to help users reflect on potential harms associated with their prompts. However, instead of passively reading AI incident reports and LLM-generated content, users desire to actively edit and add new use cases, stakeholders, and harms (Fig. 8.4). Also, active participation—a key factor in learning—may help foster AI prototypers’ ability to independently identify harms. Therefore, we design *Harm Envisioner* (Fig. 8.8) to support users in actively envisioning potential harms associated with their prompts (G4). We use a real prompt from Awesome ChatGPT Prompts [361] in the example in Fig. 8.8.

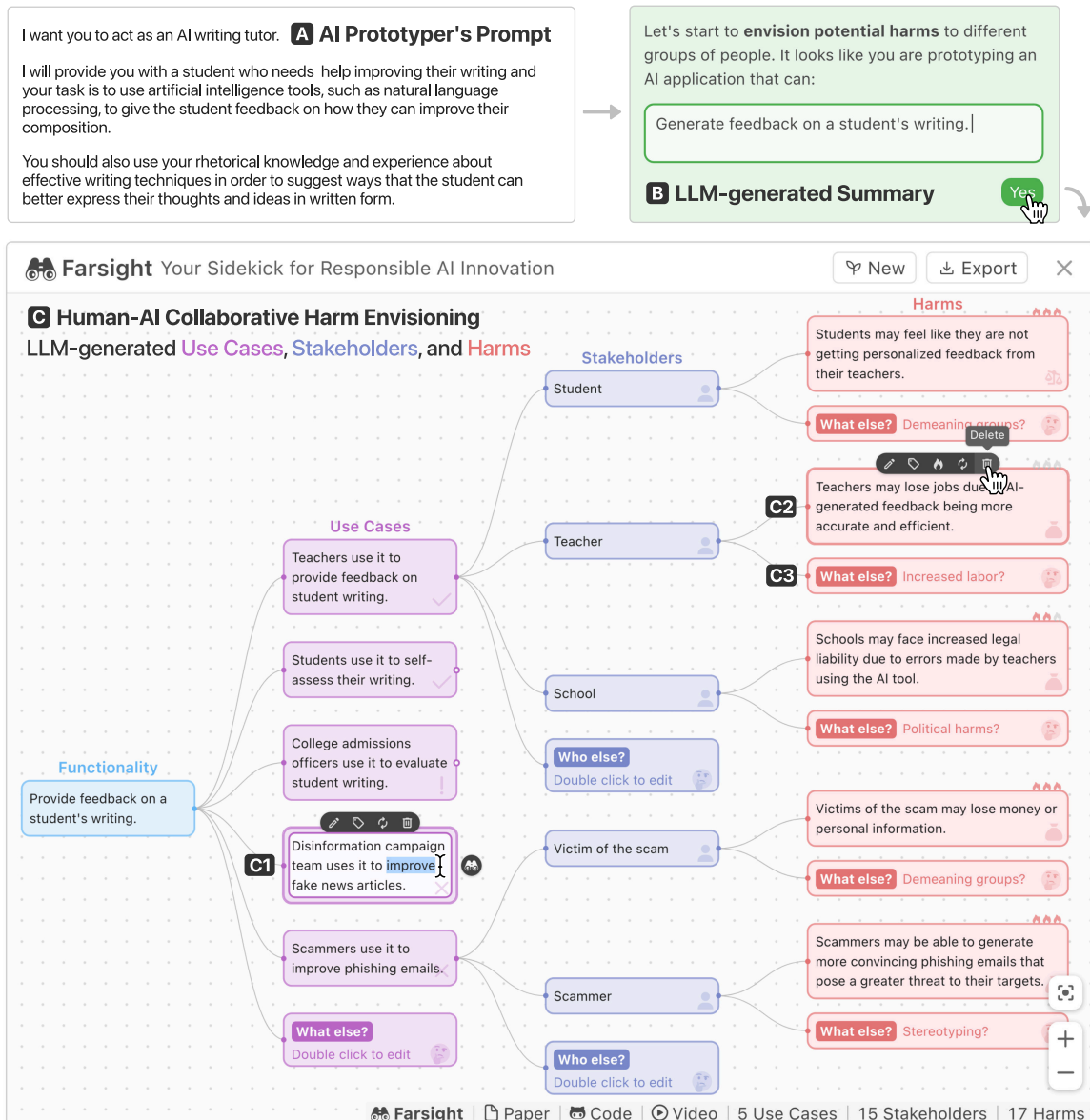


Figure 8.8: The *Harm Envisioner* helps AI prototypers envision harms associated with their AI applications through human-AI collaboration. (A) Given a prompt, (B) FARSIGHT uses an LLM to generate a summary of the prompt and asks users to revise it. (C) Then, the *Harm Envisioner* presents an interactive node-link diagram to visualize use cases, stakeholders, and harms. Initially, the *Harm Envisioner* only shows up to the *Use Cases* layer. (C1) Users can edit the node content before asking AI to generate its children nodes by clicking . Users can edit any node and regenerate its children at any time, and click a node to show or hide its descendants. (C2) Users can delete unhelpful nodes. (C3) This view encourages users to think and add more harms by intermittently and randomly alternating harm categories shown in empty harm nodes, such as “increased labor?”

Interactive Node-link Tree Visualization. After clicking the “Envision Consequences & Harms” button in the *Awareness Sidebar*, *Harm Envisioner* appears as a pop-up window on top of the prompt-crafting tool (Fig. 8.8). It begins with a text box filled with an LLM-generated summary of a user’s prompt (Fig. 8.8B). The user is prompted to revise the

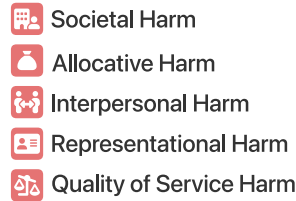




Figure 8.9: Icons used to represent different harm types.

summary to align with the target task in their prompt. Next, the window transitions into an interactive node-link tree visualization [349], where the user can pan and zoom to navigate the view (Fig. 8.8C). First, the window shows the user’s prompt summary as the root of the tree which is visualized as a text box. The user can click the root node and the LLM will generate potential use cases of an AI application based on the user’s prompt, and the use cases are visualized as the root’s children nodes. Similarly, users can click a generated node and the LLM will generate its children nodes (stakeholders and then harms). There is a max of four layers, following an order of the user’s prompt summary → use cases → stakeholders → harms. This layer order reflects the recommended harm envisioning workflow in responsible AI literature [85, 355, 356, 354, 326] and helps users to comprehend and organize diverse harms across different contexts (G2).

Human-AI Collaboration in Harm Envisioning. Our goal is to use AI-generated harms to encourage users to reflect on potential downstream harms and inspire them to add, edit, or curate potential harms (G4). To do that, *Harm Envisioner* allows users to edit any tree nodes by clicking a button in the toolbar (Fig. 8.8-C1) or entering new text in the tree node. In addition, users can delete (Fig. 8.8-C2) and use the LLM to regenerate all of an edited node’s children nodes, to effectively steer the harm envisioning direction by offering feedback to the LLM (G4). To meet users’ needs of categorizing harms (G2), we use an LLM to classify each harm into a harm type based on a systematic review and taxonomy of AI harms [362]. Users can use the dropdown menu to change the harm’s category (Fig. 8.9). To help users prioritize and take notes about harms, the *Harm Envisioner* allows users to rate the severity of each harm by clicking  in the toolbar. Finally, users can click  to export all content (e.g., use cases, stakeholders, and harms) in the *Harm Envisioner* as a Markdown file.





8.3.4 Open-source and Reusable Implementation

To make FARSIGHT easily adoptable by both AI prototypers and AI companies (G5), we implement FARSIGHT to be model-agnostic and environment-agnostic, and we open-source our implementation. FARSIGHT uses LLMs by calling their public APIs so that users can use their preferred LLMs by easily replacing the API endpoints. To help AI companies and researchers integrate FARSIGHT into AI prototyping tools, we leverage Web Components [350] and Lit [363] to implement FARSIGHT as reusable modules, which can be easily integrated into any web-based interfaces regardless of their development stacks (e.g., React,

Vue, Svelte). To help AI prototypers use our tool, we present a Chrome extension² that integrates FARSIGHT into Google AI Studio and a Python package³ that brings FARSIGHT to computational notebooks. We implement the interactive tree visualization using *D3.js* [180] and embedding similarity computation using *TensorFlow.js* [179] with WebGL [351] acceleration. Computational notebook support is implemented using NOVA [206].


8.4 Usage Scenario

We present a hypothetical usage scenario to illustrate how FARSIGHT fosters responsible awareness among AI prototypers. Rosa is a native English speaker from the United States who recently traveled to Vietnam to teach English. She is the only English teacher at an under-resourced high school. Overwhelmed with grading English writing assignments for all students in the school, Rosa tries to develop an LLM-powered AI application that provides writing feedback based on a student’s essay. She writes her prompt (Fig. 8.7A) in an AI prototyping tool with FARSIGHT integrated. After running the prompt, Rosa notices the alarming *Alert Symbol* (Fig. 8.7A), so she clicks on it, which expands the *Awareness Sidebar* (Fig. 8.7-BC). Rosa reads a few related articles shown in the *Incident Panel* (Fig. 8.7-B2). She finds these articles are indeed related to AI in education and are helpful, but they mainly focus on students using AI to cheat rather than teachers using AI to grade assignments. Rosa skims through the LLM-generated potential use cases and finds the use case “*teachers use it to provide feedback on student writing*” very relatable (Fig. 8.7-C1). Intrigued by its associated harm “*students may feel like they are not getting personalized feedback from their teachers*”, Rosa clicks the *Envision Consequences* button and wishes to learn more about this use case and its associated potential harms.

Harm envisioner. Next, FARSIGHT shows a pop-up window asking Rosa to revise and confirm an LLM-generated summary of her prompt (Fig. 8.8-B). Clicking , Rosa sees the *Harm Envisioner* presenting an interactive tree visualization showing the functionality of her AI application as a root node and multiple use cases as its children nodes (Fig. 8.8-C). With a map-like interface, Rosa quickly uses zoom-and-pan to zoom into the teaching use case. After clicking , the *Harm Envisioner* quickly generates the stakeholders associated with the use case and the harms associated with each stakeholder. Rosa takes some time to reflect on the LLM-generated harm of students not getting personalized feedback (Fig. 8.8-Harm-1). She has never thought about this consequence before, but she thinks it makes sense—AI does not have background knowledge about each student, so its feedback would not be tailored to students’ individual needs. After rating it as very severe  by clicking , Rosa continues reading other LLM-generated harms. She does not think the harm of teachers losing jobs to her AI tutor is relevant, so she deletes it (Fig. 8.8-C2).

²FARSIGHT Chrome extension: <https://github.com/PAIR-code/farsight/releases>

³FARSIGHT Python package: <https://pypi.org/project/farsight/>

Human-AI collaboration. After seeing the random question “increased labor?” next to teacher (Fig. 8.8-C3), Rosa thinks maybe it will be more time-consuming to review AI-generated feedback than grading students’ assignments herself, so she enters that harm into the *Harm Envisioner*. Next, Rosa is not sure about the legal liability of her school (Fig. 8.8-Harm-3), but it might be worth discussing with other teachers. Finally, reflecting on her experience with the *Harm Envisioner* and AI incident articles, Rosa thinks the potential harms of her writing tutor AI application outweigh the potential convenience for her. Therefore, Rosa decides to stop prototyping this application. However, Rosa still sees value in leveraging LLMs in education, so she bookmarks related AI incident articles and clicks  to download all the content in the *Harm Envisioner* as a Markdown file. She will bring these resources to discuss with her colleagues the next day.

8.5 Evaluation User Study

We conducted a user study to evaluate FARSIGHT’s effectiveness in aiding AI prototypers to anticipate the potential harms associated with AI features. In addition, we investigate how AI prototypers use FARSIGHT during an early prototyping stage. To investigate the effect of user engagement in AI-assisted harm envisioning, we tested two variants of our tool: FARSIGHT, including all components, and FARSIGHT LITE, including only the *Alert Symbol* (Fig. 8.2-B) and the *Awareness Sidebar* (Fig. 8.2-C). In other words, FARSIGHT LITE is a “subset” of FARSIGHT. FARSIGHT LITE only shows one direct stakeholder for each use case in the *Use Case Panel*, while FARSIGHT allows users to interactively add more stakeholders, use cases, and harms in the *Harm Envisioner* (Fig. 8.2-A). The study included 42 AI prototypers with diverse roles who were recruited from a large technology company based in the United States. In this user study, we aimed to investigate the following three research questions:

- RQ1.** How do FARSIGHT and FARSIGHT LITE affect users’ ability for and approach to identifying potential harms?
- RQ2.** How effective and useful are FARSIGHT and FARSIGHT LITE in assisting users in envisioning harms in comparison to existing commonly-used resources?
- RQ3.** What challenges do AI prototypers face when envisioning potential harms during the AI prototyping stage? How do FARSIGHT and FARSIGHT LITE help AI prototypers address these challenges?

8.5.1 Participants

We recruited 45 voluntary participants from both internal mailing lists related to AI and snowball sampling at Google, based in the United States. The recruitment required participants to have experience in writing prompts for LLMs. In total, we received 61 responses,

Table 8.2: The evaluation user study included 42 participants with diverse roles and experience in prompting LLMs.

Participant Roles	Participant IDs
Software Engineer	3, 4, 5, 6, 7, 12, 13, 15, 16, 17, 19, 23, 25, 26, 28, 29, 33, 34, 35, 41, 42
Product Manager	1, 8, 10, 11, 14, 20, 24, 27, 36
Linguist	2, 21, 30, 31
AI Researcher	9, 18, 39, 40
UX Researcher	22
Data Scientist	32
Test Engineer	37
Marketing Specialist	38

Self-Reported Familiarity from 42 Participants

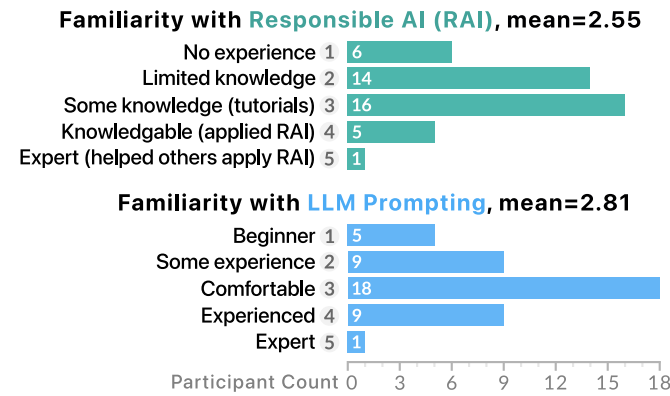


Figure 8.10: Participants reported diverse levels of familiarity with responsible AI (top, average=2.55) and prompting (bottom, average=2.81) on 5-point Likert scales.

and we selected 45 participants based on their schedule availability. We conducted pilot studies using the first three study sessions, which were not included in our data analysis. As a result, we had a total of 42 participants. Each study session was either 90 minutes ($n=28$ sessions) or 60 minutes ($n=14$ sessions), depending on the participants’ availability. During the 90-minute sessions (or 60-minute sessions), each participant received an average of \$62 USD (or \$41) compensation in their preferred form such as gift cards and charity credits.

Among the 42 participants, 26 identified as men, 14 as women, and 2 preferred not to disclose their gender. Information about their job roles is listed in Table 8.2. Recruited participants self-reported an average score of 2.55 for their knowledge and experience with responsible AI on a 5-point Likert scale (Fig. 8.10-top), where 1 represents “No experience” and 5 represents “Expert (I have helped others apply responsible AI practices).” In addition, participants self-reported an average score of 2.81 for experience with LLM prompting on a 5-point Likert scale (Fig. 8.10-bottom), where 1 represents “Beginner” and 5 represents “Expert.” All participants are FARSIGHT’s targeted users, AI prototypers.

	Harm Envisioning 1 (H1: Pre-task) ✦: Email Summarizer	Harm Envisioning 2 (H2: Intervention) ✦: Toxicity Classifier	Harm Envisioning 3 (H3: Post-task) ✦: Article Summarizer	Interview 1 (Reflection)	Harm Envisioning 4 (H4: Alternative) ✦: Math Tutor	Interview 2 (Comparison) (Survey)	Participants
C_{FG}	Independent	Farsight	Independent		Envisioning Guide		5, 16, 21, 23, 32, 34, 40
C_F	Independent	Farsight	Independent				2, 4, 9, 13, 17, 18, 39
C_{LG}	Independent	Farsight Lite	Independent		Envisioning Guide		7, 10, 15, 28, 29, 36, 41
C_L	Independent	Farsight Lite	Independent				1, 11, 19, 24, 25, 37, 38
C_{GF}	Independent	Envisioning Guide	Independent		Farsight		6, 8, 20, 22, 26, 30, 33
C_{GL}	Independent	Envisioning Guide	Independent		Farsight Lite		3, 12, 14, 27, 31, 35, 42

Figure 8.11: The evaluation study included six conditions with different variations of harm envisioning tools (FARSIGHT, FARSIGHT LITE, and the baseline ENVISIONING GUIDE). Participants were asked to envision potential harms associated with an AI feature (e.g., email summarizer) in each harm-envisioning activity (H1, H2, H3, and H4). Participants had access to a harm envisioning tool in H2 and H4. The duration of sessions involving H4 and interview 2 was 90 minutes, while all other sessions lasted 60 minutes. Participants were randomly assigned to a condition, taking into account their availability for study session duration.

8.5.2 Study Design

We conducted this study with participants one-on-one. Out of 42 sessions, 2 were conducted in-person, and 40 were through video conferencing software due to office locations and participants’ scheduling constraints. With the permission of all participants, we recorded the participants’ audio and computer screen for subsequent analysis. To start, each participant signed a consent form and filled out a survey regarding their familiarity with responsible AI and LLM prompting (Fig. 8.10). Then, participants were randomly assigned to one of six conditions taking into account their time availability: C_{FG} , C_F , C_{LG} , C_L , C_{GF} , C_{GL} (Fig. 8.11). C stands for the study condition, C_{FG} means that participants used FARSIGHT first and then ENVISIONING GUIDE, and C_L means that participants only used FARSIGHT LITE—the other acronyms follow this same pattern. Sessions of C_F and C_L were scheduled for 60 minutes each, while the remaining sessions were allotted 90 minutes each. We assigned 7 participants to each condition, as this was the maximum number that allowed for an equal distribution of participants across all conditions, given the time constraints and the availability of the 61 individuals who signed up for the study.

Our study followed a mixed design that combines both between-subjects and within-subjects designs [364]. Each session included three or four harm-envisioning activities, denoted as H1, H2, H3, and H4 (§ 8.5.2.2), as well as one or two semi-structured interviews to collect participants’ feedback (§ 8.5.2.3). In each harm-envisioning activity, participants were asked to envision potential harms associated with a particular AI feature while thinking aloud (Fig. 8.11). In H1 and H3, participants envisioned harms on their own, whereas in H2 and H4, they could use a harm envisioning tool we assigned them based on their study condi-

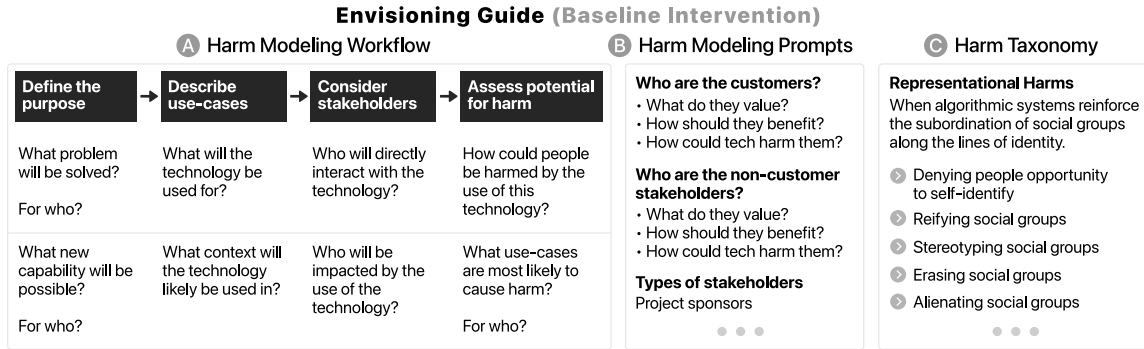


Figure 8.12: In the evaluation user study, we compared our tools against **ENVISIONING GUIDE**, a combination of existing harm envisioning resources. This **ENVISIONING GUIDE** was presented to participants as a Google Doc with three sections. **(A)** The harm modeling workflow table comes from Microsoft’s Harm Modeling Practice [326], providing a four-step process to envision harms. **(B)** The harm modeling prompts from the Harm Modeling Practice [326] offer templates and questions to help users envision different stakeholders and use cases (not all content is displayed here). **(C)** The harm taxonomy [362] helps participants explore the space of potential harms by providing a comprehensive list of 20 harm categories organized into five themes (not all content is displayed here). Participants could click the icon to see the definition of each harm category.

tion (e.g., **FARSIGHT**, **FARSIGHT LITE**, or **ENVISIONING GUIDE**). All collected harms were rated by seven researchers with experience with responsible AI evaluations, who assigned each potential harm numeric scores in terms of their likelihood and severity (§ 8.5.2.4). We compared the envisioned harms in H1 and H3 (between-subjects) to investigate how different tools affect users’ ability and approach to anticipating harms (RQ1). We compared the envisioned harms in H2 and H4 (within-subjects) to assess the effectiveness of different tools in helping users envision harms (RQ2). Besides the quantitative data on the number and ratings of potential harms, we also collected qualitative data from think-aloud and two interviews (RQ1–RQ3). We incorporated 60-minute sessions (C_F and C_L) into our study design due to challenges in recruiting participants available for a 90-minute duration.

8.5.2.1 Baseline Harm Envisioning Tool.

To compare our work against current responsible AI workflows, we created a baseline intervention **ENVISIONING GUIDE**: a combination of Microsoft’s Harm Modeling Practice [326] and the Harm Taxonomy from Shelby *et al.* These two resources are the latest and the most representative resources designed to help practitioners envision harms. We combined them because (1) we aim to simulate the current practice where AI prototypers can choose from various existing harm envisioning tools, and (2) we do not intend to study the causal effects of any specific resource. We administered this intervention by providing a Google Doc containing a detailed table and information from these resources (Fig. 8.12). Both resources were designed to help technology developers and researchers anticipate and prevent negative societal impacts of their technology innovations.

8.5.2.2 Harm Envisioning Activities

Depending on the conditions, the study included three or four harm envisioning activities (H1–H4). Within each harm envisioning activity, participants were presented with a description of an AI feature and the prompt that generated that feature. We chose the four AI features (Fig. 8.11) based on a qualitative analysis of 100 randomly sampled internal prompts written by real AI prototypers. These four features are representative of popular LLM tasks (e.g., summarization, classification, and question answering) and comprehensible to participants with diverse roles. In H1 and H3, participants independently envisioned harms, whereas in H2 and H4, they were provided with a harm envisioning assistance tool (e.g., **FARSIGHT**, **FARSIGHT LITE**, or **ENVISIONING GUIDE**). To emulate AI prototyping workflows, we asked participants to perform simple prompt engineering tasks in H2 and H4 before envisioning potential harms of presented AI features.

For each harm, participants were instructed to describe *who* would be affected (i.e., the stakeholders) and *how* the stakeholder might be harmed. We provided a harm example for a code generation AI feature: “App end-users might face financial loss due to AI-introduced software vulnerabilities.” During the process, participants were asked to share their screens and verbalize their thoughts. They were also asked to enter their envisioned harms into a Google Doc table featuring a *who* column and a *how* column. Moreover, participants had the option to articulate the harm verbally, and we transcribed it into the table. At the end of each harm envisioning activity, we reviewed the table together with the participants to ensure the accuracy of the harm descriptions. Participants were instructed to achieve three objectives: (1) envision as many harms as possible; (2) envision the most likely harms; and (3) envision the most severe harms.

H1: Pre-task. To understand how participants independently envision potential harms *before* using the tool, as a baseline for RQ1, participants were asked to anticipate potential harms concerning an LLM-powered email summarizer on their own (Fig. 8.11). They received information about the AI functionality: “*Shorten and improve a user’s email*”, a development context, and a prompt that enables this functionality. The duration of this activity was limited to 10 minutes.

H2: Intervention. In the second harm envisioning activity, we asked participants to use different harm envisioning assistant tools. Depending on the assigned condition, a participant could use **FARSIGHT** (C_{FG} , C_F), **FARSIGHT LITE** (C_{LG} , C_L), or **ENVISIONING GUIDE** (C_{GF} , C_{GL}) to help them anticipate harms. The activity began with a tutorial on the designated tool. The AI feature used in this activity was an LLM-powered toxicity classifier (Fig. 8.11). Participants received information regarding the AI functionality “*Detect toxic text content*,” a development context, and a prompt that enables this AI functionality. To emulate AI prototyping workflows, we also tasked participants with a simple prompt engineering assignment.

After completing prompt engineering, participants envisioned harms linked to the toxicity classifier. They were instructed to freely use the assigned tools while sharing their screens and thinking aloud. For participants assigned with **ENVISIONING GUIDE** (C_{GF} , C_{GL}), the process of entering envisioned harms was the same as H1. Participants assigned with **FARSIGHT** (C_{FG} , C_F) or **FARSIGHT LITE** (C_{LG} , C_L) could click a button in the tools to export all harms as a text file. The export included both AI-generated harms and harms added or modified by participants. Participants were asked to copy the harms into the Google Doc. As a significant portion of these harms were generated by AI, we asked participants to select harms that (1) they agreed with and (2) would report to their colleagues and managers. Also, participants were welcome to add more harms to the table. For our analysis, we only included the exported harms that participants had selected and added to the table. The duration of this activity was limited to 25 minutes.

H3: Post-task. To understand how the intervention may have affected participants’ ability to independently envision harms (RQ1), we asked participants to envision harms associated with an LLM-powered article summarizer on their own (Fig. 8.11). To ensure a valid comparison between the envisioned harms and participants’ approaches to the pre-task (H1), we introduced a parallel AI summarizer feature in this activity that was isomorphic to the pre-task [365]. In particular, to deter participants from directly reusing their envisioned harms from H1, we replaced the email summarizer in H1 with an article summarizer. The AI functionality was described as “*Summarize an article in one sentence*”. The duration of this activity was limited to 10 minutes.

H4: Alternative. To assess the effectiveness and usefulness of **FARSIGHT** and **FARSIGHT LITE** in comparison to **ENVISIONING GUIDE** (RQ2) and study the usage patterns of different tools (RQ3), $n = 28$ participants engaged in 90-minute sessions (C_{FG} , C_{LG} , C_{GF} , and C_{GL}) to envision harms using a tool different from the one used in H2 (Fig. 8.11). Participants were asked to envision potential harms associated with an LLM-powered math tutor app with the AI functionality “*Answer math-related questions*”, a development context, and a prompt. The procedure for this activity paralleled H2, including a tutorial, prompt engineering exercise, and harm envisioning. This activity’s duration was limited to 25 minutes.

8.5.2.3 *Semi-structured Interviews*

This study included two semi-structured interview sessions (Fig. 8.11). The first interview took place after the post-task activity (H3), where we asked participants to reflect on their process for anticipating potential harms during the LLM prototyping process, and how their approach may have changed after the intervention (RQ1). We also asked participants about their challenges in harm anticipation, their experiences of using harm envisioning tools, and potential actions they would take to address the identified harms (RQ3). After participants in 90-minute sessions (C_{FG} , C_{LG} , C_{GF} , and C_{GL}) finished H4, we asked them to compare and rate

the usefulness and usability of the two tools they had used in this study (RQ2). We also asked them to rate the helpfulness of different components in the tools on a 5-point Likert scale.

8.5.2.4 Harm Rating

After completing all 42 study sessions, we recruited seven raters to rate all 989 harms collected in H1–H4 to evaluate participants’ ability to envision harms. These seven raters included four of the paper authors and three industry researchers; all raters had experience with responsible AI (unlike many of the participants)—either as responsible AI researchers, developers of responsible AI tools or playbooks, or in a consultant role on responsible AI for product teams. Ideally, evaluations of identified harms would involve both domain experts for the domain in question (e.g., education) and/or stakeholders from demographic groups or communities who may be likely to experience those harms. For this preliminary study, due to timing and resource constraints, we recruited responsible AI researchers as raters instead of specific domain experts or people impacted by AI applications. The limitations of this approach are further discussed in § 8.5.7 and § 8.6.2.

Our collected harms were either (1) directly envisioned by participants or (2) exported from **FARSIGHT** or **FARSIGHT LITE** and subsequently curated by participants during H2 and H4. Each harm included the impacted stakeholders and a description of the harm. After removing duplicates and random shuffling, we randomly and evenly assigned harms to raters via spreadsheet format. Raters had access to the details of the intended AI feature of each harm, including the prompt and the context of the AI feature. To prevent the raters from being influenced by our hypotheses, we did not include the experimental conditions in the rating sheet. In other words, raters did not know if a harm was from a **FARSIGHT** user, a **FARSIGHT LITE** user, or a **ENVISIONING GUIDE** user. To mitigate rating noise, we designated three raters for each harm. As identifying *likely* and *severe* harms is often an objective in AI harm envisioning exercises [326, 366], we asked raters to rate each harm’s *likelihood* and *severity* on a 4-point Likert scale (*strongly agree*, *agree*, *disagree*, and *strongly disagree* to statements “This harm is likely to occur for this stakeholder” and “This harm will severely impact this stakeholder”). Raters could also choose an N/A option if they perceived a rating was not applicable for that feature or use case. During data analysis, we numericalized these four categories as ordinal scores: 1, 2, 3, 4 and removed N/As.

8.5.3 Data Analysis

We applied a mixed-methods approach for data analysis. First, we conducted a quantitative analysis (§ 8.5.3.1) on the changes in participants’ ability to envision harms by comparing pre-task H1 to post-task H3 responses (RQ1). We also quantitatively assess three different tools’ effectiveness in helping users anticipate harms by comparing H2 and H4 responses (RQ2). The quantitative analyses involved metrics such as the total number

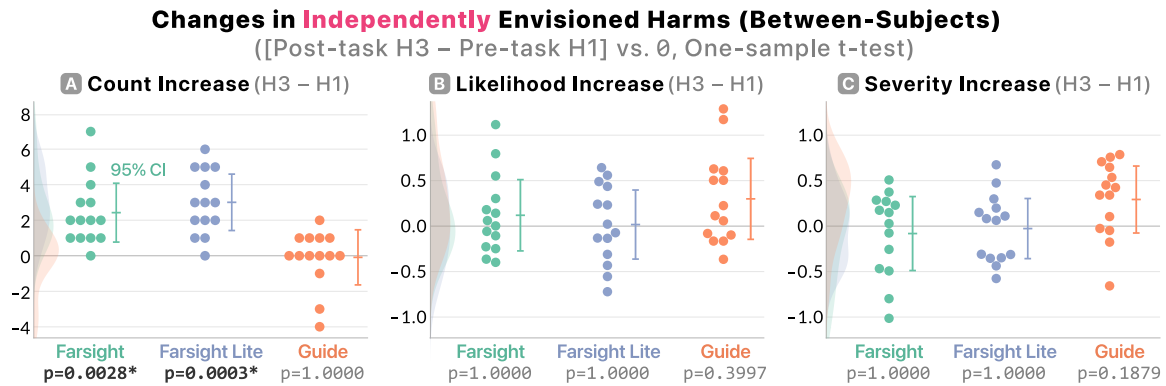


Figure 8.13: To evaluate how different interventions (**FARSIGHT**, **FARSIGHT LITE**, **ENVISIONING GUIDE**) affect users’ ability to envision harms independently, we conducted one-sample *t*-tests with Bonferroni correction to examine the *difference* in the (A) count, (B) average likelihood, and (C) average severity of participant-identified harms between H3 and H1. Each intervention had $n = 14$ participants, represented by 14 points on the chart. The charts also indicated the 95% confidence intervals, adjusted with Bonferroni correction. The results highlighted that after using **FARSIGHT** and **FARSIGHT LITE**, users could anticipate a significantly higher number of harms, while the average likelihood and severity of identified harms remained the same.

of envisioned harms, as well as the average likelihood and severity ratings of envisioned harms across 3 raters. Next, we performed a qualitative analysis (§ 8.5.3.2) on transcripts from think-aloud sessions and interviews to further investigate participants’ strategies and challenges in envisioning harms, and usage patterns of different tools (RQ1–RQ3).

8.5.3.1 Quantitative Analysis.

We first conducted quantitative analyses on the count, likelihood, and severity of harms across different conditions to evaluate the effectiveness of our tools (RQ1, RQ2). We measured the likelihood and severity for each harm using the average of ratings from three raters after removing any N/As. The average pairwise weighted Cohen’s kappas [367, 368] for likelihood and severity ratings are 0.14 and 0.09. These values fall within the range of slight agreement [369]. We discuss this relatively low inter-rater agreement in § 8.6.2. The Shapiro-Wilk normality tests [370] show all measures, except for the changes of harm count between H1 and H3 with **ENVISIONING GUIDE**, follow a normal distribution. We used *t*-tests with Bonferroni corrections for multiple hypothesis testing.

We also analyzed participants’ ratings of the tools’ usefulness and usability when comparing the two tools used in the study (RQ2, Fig. 8.5.5.3). We converted the 5-point Likert scale ratings into numerical values and assessed the difference between ratings of our tools and **ENVISIONING GUIDE** using Mann-Whitney U tests [371]. Considering that most of the ratings did not exhibit a normal distribution, we chose to use Mann-Whitney U tests, as these tests do not assume normality in the data. See Fig. 8.5.5.3 for discussion of the findings from these questions about usefulness and usability.

8.5.3.2 *Qualitative Analysis.*

We conducted a qualitative analysis on the screen recordings and transcripts of the study sessions that include participants’ verbalized thoughts during the harm envisioning activities (H1–H4) and interviews. All study sessions were screen-recorded and audio-recorded, with the audio subsequently transcribed by the video conferencing software. We adopted an inductive thematic analysis approach [372, 373] and open coded the 56-hour-long transcripts using the qualitative analysis software Dovetail [374]. After generating a codebook, we applied deductive coding [372] to assign harm envisioning patterns to each participant during H1 and H3 (RQ1, § 8.5.4.2).

8.5.4 Findings: Changes in Users’ Envisioning Ability and Approach (RQ1)

In the study, participants were asked to independently envision harms associated with an email summarizer (H1) and an article summarizer (H3) before and after using a harm envisioning tool (**FARSIGHT**, **FARSIGHT LITE**, or **ENVISIONING GUIDE**) to anticipate harms for a toxicity classifier (H2). We quantitatively and qualitatively compared participants’ envisioned harms and approaches in H1 and H3 across different conditions in H2.

8.5.4.1 **FARSIGHT** and **FARSIGHT LITE** Improved Users’ Ability to Envision Harms.

For each participant, we compared the count, average likelihood, and average severity of their independently envisioned harms before (H1) and after (H3) the intervention (Fig. 8.13). Using paired sample *t*-tests with Bonferroni correction [375], we found that after using **FARSIGHT** and **FARSIGHT LITE**, users could envision significantly more harms on their own ($p = 0.0028$, $p = 0.0003$), showing an average increase of 2.42 and 3.00 harms, respectively. The effect sizes, as measured by Cohen’s *d* [376], were $d = 1.21$ and $d = 1.27$, indicating a very large effect [377]. On the contrary, for participants using **ENVISIONING GUIDE**, the average count of identified harms experienced a marginal decrease (-0.14). We hypothesize that the observation of three participants identifying fewer harms after using **ENVISIONING GUIDE** (see the outliers in Fig. 8.13 A) is because **ENVISIONING GUIDE** had a high cognitive load. The high cognitive load may have resulted in these three participants having less energy to envision harms in H3 compared to H1. Changes in the average likelihood and average severity, on the other hand, were not statistically significant for any of the interventions (Fig. 8.13-BC). Our finding implies that after using **FARSIGHT** and **FARSIGHT LITE**, users could anticipate a greater number of harms linked to AI features independently, while the average likelihood and severity of identified harms remained unaltered.

Table 8.3: We identified six non-exclusive common patterns in independent harm envisioning by analyzing transcripts of participants’ think-aloud process during H1 and H3.

Harm Envisioning Pattern	Description
Failure-mode-driven envisioning	Participants envisioned harm by initially considering the AI feature’s failure modes (e.g., wrong summarization), limitations of LLMs (e.g., hallucination), or vulnerabilities within system implementation (e.g., data storage). This pattern is similar to a Failure Mode and Effects Analysis [378].
Usage-driven envisioning	Participants envisioned harm by initially considering who may be impacted through this feature and in what usage scenario, such as students using the article summarizer for completing assignments. Then, participants envisioned potential harms that might impact the stakeholders within the identified scenario.
Consider high-stakes uses	Participants deliberately thought about high-stakes use cases of the AI feature, such as being used in medical, financial, and legal domains.
Consider misuses	Participants deliberately envisioned potential misuse of the AI feature, where malicious actors like scammers and hackers could exploit this AI feature to cause harm.
Consider indirect stakeholders	Participants deliberately brainstormed stakeholders indirectly impacted by the AI feature, such as people who did not use the AI tools, individuals mentioned in the input text, and society at large.
Consider cascading harms	Participants deliberately considered (1) harms that could result from other harms, such as productivity loss due to AI errors can lead to economic loss; or (2) harms that might occur even when the AI feature operated as expected, such as students using AI to cheat in homework.

8.5.4.2 *Changes in Harm Envisioning Approaches.*

We also investigated the impacts of different tools on participants’ approaches to harm envisioning by analyzing their self-reports in interview 1 and the think-aloud data in H1 and H3.

Self-reported changes after using FARSIGHT and FARSIGHT LITE. The major themes of self-reported changes are similar between FARSIGHT and FARSIGHT LITE. A large number of participants noted that while they initially considered the AI feature and its potential harms in a general sense during H1, they shifted towards a more focused

consideration of specific use cases and stakeholders in H3 (e.g., P23, P34, P38). Some participants highlighted they started to brainstorm potential misuses in H3 (P25, P32). For stakeholders, participants broadened their consideration to people and organizations not initially considered during H1. P40 acknowledged a transition from a focus on “*protecting the AI company*” in H1 to considering end-users in H3. Similarly, P17 reported a focus on end-users after using **FARSIGHT**:

“Earlier maybe I was coming towards it from a very engineering or a very broad feature perspective. The third time, I was thinking more about people who were actually using the product and getting affected. So I was thinking more for the people using it, rather than that being a feature in some application.” (P17)

Many participants also highlighted that they began to adopt the frameworks presented in **FARSIGHT** and **FARSIGHT LITE** (e.g., P9, P10, P32) to structure their harm envisioning procedures. For example, P10 and P32 appreciated the categorization of use cases, and they reported considering intended uses, high-stakes uses, and misuses in H3. After using **FARSIGHT**, P9 said they followed the sequence of layers in the tree visualization to conceptualize use cases, stakeholders, and harms:

“I found that sort of flow from identifying potential use cases, then identifying stakeholders of those use cases, then identifying potential harms for each of the stakeholders to be really valuable. That’s a great way to scaffold it and think through the flow rather than just sort of bouncing around, which is what I had been doing [in H1]. So yeah, I found that super valuable that has changed the way that I think about it. And that’s the framework that I’ll use in the future.” (P21)

Self-reported changes after using ENVISIONING GUIDE. Many participants using **ENVISIONING GUIDE** in H2 (C_{GF} , C_{GL}) also noted shifts in their approaches to envisioning harms. Several participants noted that they started to follow the structure outlined in the Harm Modeling Guide to envision harms (P8, P40, P42). Some participants started thinking more about under-represented social groups in H3 (P8, P31). Furthermore, many participants described the harm taxonomy as a “*mental checklist*” that provided them with a language to articulate and think about harms (e.g., P6, P14, P31).

Observed changes in envisioning approaches. By analyzing transcripts of participants’ think-aloud process during the harm envisioning activities in H1 and H3, we identified six non-exclusive common patterns in harm anticipation (Table 8.3). Then, we examined the effects of different interventions on participants’ envisioning patterns by comparing the number of participants who applied and did not apply these six patterns in H1 and H3 across interventions (Fig. 8.14). The intervention assignment is random.

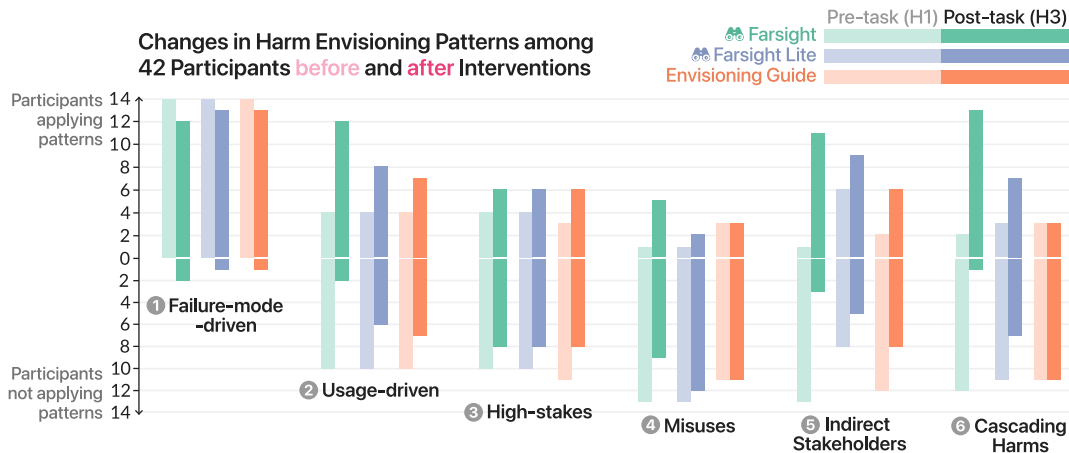


Figure 8.14: By analyzing transcripts of 42 participants during the pre-task (H1) and post-task (H3) harm envisioning activities, we identified six non-exclusive common patterns in envisioning harms. This bar chart compares the number of participants who applied and did not apply these patterns before and after the three interventions. Note that there were 14 random participants for each intervention, and the initial number of participants applying certain patterns could differ. The chart highlights that both **FARSIGHT** and **FARSIGHT LITE** encouraged participants to consider how the AI feature would be used. Notably, the use of **FARSIGHT** particularly influenced participants to think more about indirect stakeholders and cascading harms.

Interestingly, the counts of participants who applied each pattern in H1 were consistent across interventions, with the exception of **FARSIGHT LITE** where notably more participants considered indirect stakeholders in H1 (Fig. 8.14-5). Before the interventions, the majority of participants relied on failure-mode-driven envisioning when anticipating harms (Fig. 8.14-1), focusing on the AI feature’s limitation, failure modes, and technical implementation details. This observation corroborates participants’ self-reported envisioning approaches, where participants like P17 acknowledged having a “*very engineering or a very broad feature perspective*” in H1.

After the intervention, we observed that all three harm envisioning tools (**FARSIGHT**, **FARSIGHT LITE**, and **ENVISIONING GUIDE**) influenced participants to adopt a usage-driven envisioning approach when independently envisioning harms (Fig. 8.14-2). Particularly, **FARSIGHT** had the most pronounced effect, followed by **FARSIGHT LITE** and then **ENVISIONING GUIDE**. All these tools encouraged participants to think more about high-stakes uses (Fig. 8.14-3) and indirect stakeholders (Fig. 8.14-5). Both **FARSIGHT** and **FARSIGHT LITE** exerted a stronger influence on considering misuses (Fig. 8.14-4) and cascading harms (Fig. 8.14-6) compared to **ENVISIONING GUIDE**. However, **ENVISIONING GUIDE** had slightly more impact than **FARSIGHT LITE** in encouraging consideration of high-stakes uses (Fig. 8.14-3) and indirect stakeholders (Fig. 8.14-5).

Interestingly, **FARSIGHT** had a notably more pronounced effect in leading participants to consider indirect stakeholders (Fig. 8.14-5) and cascading harms (Fig. 8.14-6) than the other tools. For indirect stakeholders, a possible explanation is that during H2, many

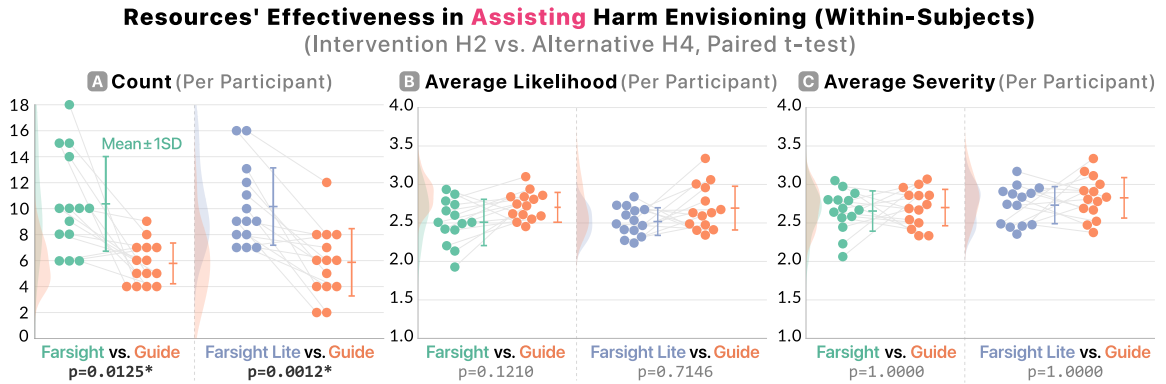


Figure 8.15: To evaluate the effectiveness of our tools in helping users anticipate harms, we conducted paired t -tests with Bonferroni correction to compare our tools (**FARSIGHT**, **FARSIGHT LITE**) against the baseline **ENVISIONING GUIDE** based on the (A) count, (B) average likelihood, and (C) average severity of harms collected in H2 and H4. In each comparison, such as **FARSIGHT** vs **ENVISIONING GUIDE**, $n = 14$ participants (each shown as two connected dots) used both tools: 7 of them started with **FARSIGHT** in H2, and the remaining 7 began with **ENVISIONING GUIDE**. The charts also highlighted the mean and standard deviation of all measures. The results showed that **FARSIGHT** and **FARSIGHT LITE** were effective in assisting users to anticipate a significantly greater number of harms compared to existing resources, while the quality of the identified harms remained consistent.

participants encountered *unexpected* indirect stakeholders revealed by **FARSIGHT** (§ 8.5.5.2). Consequently, these participants consciously began to consider stakeholders that might seem tangential but could be influenced by the AI feature in H3. This hypothesis could also explain the relatively weaker effect of **FARSIGHT LITE** in fostering consideration of indirect stakeholders, as **FARSIGHT LITE** had only identified one *direct* stakeholder for each use case, and participants could not use AI to generate more stakeholders.

For *cascading harms*, we hypothesize two potential explanations. First, many participants applied a *reviewing approach* when engaging with AI-generated harms in **FARSIGHT** and **FARSIGHT LITE**, where they tried to understand and make sense of these harms. In H2, reviewing existing harms prompted participants to consider *cascading harms* that might arise from other harms (§ 8.5.5.2). This experience could have influenced participants to also consider *cascading harms* in H3. The second explanation is that many participants were surprised by *unexpected* AI-generated *cascading harms* in H2 (§ 8.5.5.2), which might have led them to consciously think about these harms in H3.

8.5.5 Findings: FARSIGHT's Effectiveness in Assisting Harm Envisioning (RQ2)

In addition to assessing the impacts of different harm envisioning tools on users' ability to independently envision harms, we also evaluated the tools' effectiveness in aiding users to anticipate harms. Specifically, we quantitatively compared participants' envisioned harms when using different harm envisioning tools in H2 and H4. Furthermore, we qualitatively analyzed participants' usage patterns, interview responses, and survey data.

8.5.5.1 *FARSIGHT* and *FARSIGHT LITE* helped users envision more harms.

We compared the count, average likelihood, and average severity of harms collected in H2 and H4 using our tools, *FARSIGHT* and *FARSIGHT LITE*, against the baseline *ENVISIONING GUIDE* (Fig. 8.15). These harms were identified by participants using different harm envisioning tools or generated by AI and selected by the participants. This analysis followed a within-subjects approach, including 28 participants from C_{FG} , C_{GF} , C_{LG} , and C_{GL} . In each comparison, such as *FARSIGHT* vs *ENVISIONING GUIDE*, a total of 14 participants used both tools, with 7 of them starting with *FARSIGHT* in H2 (C_{FG}), and the remaining 7 beginning with *ENVISIONING GUIDE* (C_{GF}). Results from paired t -tests, adjusted with Bonferroni correction, highlighted that participants using *FARSIGHT* and *FARSIGHT LITE* resulted in a significantly higher number of harms compared to those using *ENVISIONING GUIDE* ($p = 0.0018$, $p = 0.0034$), with an average difference in the count of 4 (Fig. 8.15A). The effect sizes, as measured by Cohen’s d , were $d = 1.57$ and $d = 1.48$, indicating a very large effect. However, no significant differences were observed regarding the likelihood and severity of identified harms between our tools and *ENVISIONING GUIDE* (Fig. 8.15-BC). Our findings suggest that our tools are effective in assisting users to identify a greater number of harms compared to existing resources, while the quality of the identified harms remains consistent.

8.5.5.2 *Usage patterns.*

We summarized how participants use *FARSIGHT* and *FARSIGHT LITE* in H2 and H4.

Trying to understand (unexpected) AI-generated content. Upon encountering AI-generated content (e.g., use cases, stakeholders, and harms), participants first sought to (1) understand why AI had generated it and then (2) assess its likelihood and relevance to their AI application. For example, for the toxicity classifier in H2, *FARSIGHT* and *FARSIGHT LITE* sometimes would generate a use case “HR departments use it to screen job applicants for toxic behaviors.” This use case was usually unexpected to participants and provoked them to think how an HR department could employ a toxicity classifier. Some participants imagined that the HR could use this classifier on applicants’ social media to identify red flags (e.g., P10, P11, P29), while others could only see it being used on applicants’ cover letters (P4). Participants then assessed how likely and relevant is this scenario before diving into related harms.

Subjectivity in apprehending auto-generated content. We observed that based on participants’ prior experiences, they could have very different views on auto-generated content in *FARSIGHT*. For example, participants had different perceptions of how their companies’ HR division might use a toxicity classifier (e.g., applying the classifier to job applicants’ social media content or their application material). Also, for the toxicity classifier in H2, the *Incident Panel* would often show an incident report on biases in sentiment analysis tools. While some participants could quickly make the connection between sentiment analysis and toxicity classification and reflect on biases in toxicity classifiers (P10, P36),

others would overlook this incident (P19, P38).

In some cases, participants' disagreement came from their different definitions of harm. For example, in both H2 and H4, our tools would generate potential harms for people who do not use the AI applications, such as “students who do not use the math tutoring app may feel left behind.” Some participants perceived these harms as crucial considerations for assessing the impacts of AI applications (e.g., P6, P18, P30), while others argue against considering harms when an AI feature is absent (e.g., P4, P9, P13). We discuss the implications of subjectivity and rater disagreement in harm envisioning in § 8.6.2.

Sparked to brainstorm new harms. The content in **FARSIGHT** and **FARSIGHT LITE** often inspired participants to brainstorm new use cases, stakeholders, and harms. After seeing an AI-generated stakeholder, many participants could quickly identify potential harms for that stakeholder. For instance, seeing the stakeholder teachers in the math tutoring app in H4, P22 added a new harm that teachers may struggle to integrate this tool into their existing teaching workflows. Many participants also came up with new harms by making connections across different AI-generated use cases, stakeholders, and harms. For example, **FARSIGHT** anticipated two use cases for the toxicity classifier: (1) online moderators using it to identify toxic content, and (2) hate groups using it to recruit people. P2 connected both use cases and added a new harm: “online moderators could face death threats from hate groups who feel their speech is censored.”

Thinking beyond immediate harms. Instead of starting with a blank slate, our tools provided participants with initial materials that prompted them to think beyond the immediate harms and envision cascading repercussions. For example, after seeing the AI-generated harm “job applicants might be unfairly rejected” within the context of HR using a toxicity classifier to screen job applicants, P38 quickly thought of a cascading harm—the company’s diversity hiring effort could be harmed, as the toxicity classifier was more likely to misclassify and reject under-represented social groups. Similarly, P18 recognized in the long run, the hiring company could lose money due to the exclusion of qualified candidates caused by a biased toxicity classifier. This usage pattern might explain the increase of participants, who used **FARSIGHT** and **FARSIGHT LITE** in H2, independently envisioning cascading harms in H3 (Fig. 8.14-6).

Thinking about mitigation strategies. Interestingly, after seeing AI-generated harms, many participants *voluntarily* considered actions and strategies to take after envisioning harms. For example, after seeing AI-generated harms for the toxicity classifier, P15 and P16 noted that it was important to allow impacted stakeholders to appeal if their content was removed because of the classifier. Similarly, P27 and P40 noted that people should implement a human review process if the toxicity classifier was used to remove social media content. Interacting with **FARSIGHT** and **FARSIGHT LITE** also encouraged participants to reflect on their prompting workflows. For example, P29 and P37 mentioned that the AI

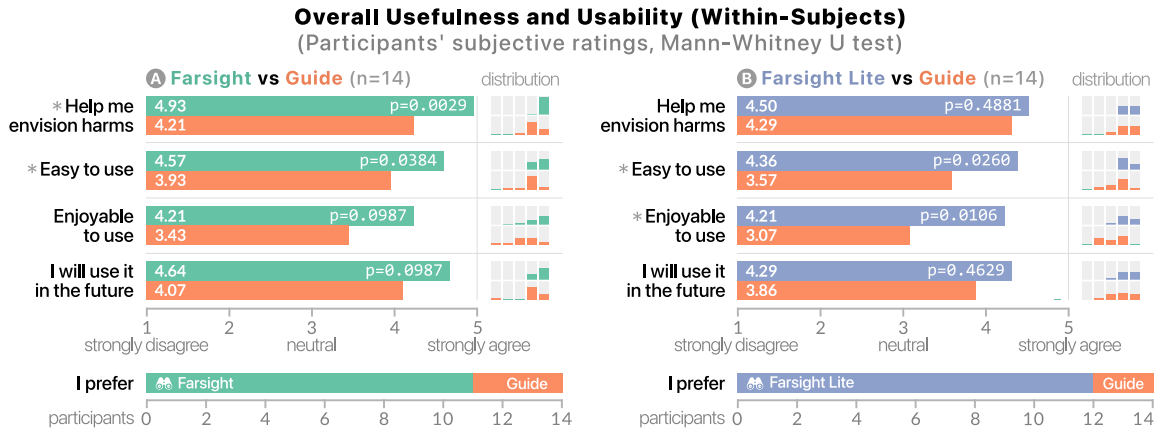


Figure 8.16: Average ratings from 28 participants, comparing the usefulness and usability of **FARSIGHT** and **FARSIGHT LITE** to **ENVISIONING GUIDE**. Both of our tools were preferred and perceived as more helpful, easier to use, and more enjoyable than the existing resources. Each comparison involved 14 participants who used one of our tools and **ENVISIONING GUIDE** in random order. We use an asterisk (*) to denote statistically significant rating differences, determined by Mann-Whitney U tests with Bonferroni correction. We used Mann-Whitney U tests instead of t-tests due to the non-normal distribution of many ratings.

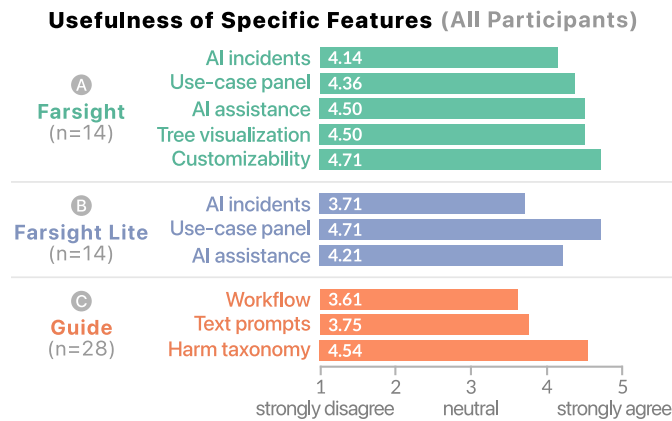


Figure 8.17: Average ratings of envisioning tool features.

prototypers should start collecting good and diverse toxicity examples to improve the prompt through few-shot prompting. P2 noted that they would like to add additional instructions in their prompt to safeguard against biased output and potential data leakage. Finally, after envisioning more harms, P2 mentioned that they would rethink if it was worth continuing to prototype or develop this AI feature.

8.5.5.3 Our tools were usable, useful, and preferred by users.

We asked participants who had used one of our tools and **ENVISIONING GUIDE** (C_{FG} , C_{GF} , C_{LG} , C_{GL}) to compare and rate the usefulness and usability of the tools they had used on a 5-point Likert-scale. By comparing their ratings, we found both **FARSIGHT** and **FARSIGHT LITE** were preferred and considered as more helpful, easier to use, and more enjoyable

compared to **ENVISIONING GUIDE** (Fig. 8.16). Both tools had significantly higher ratings on “easy to use” than the baseline ($p = 0.0384$, $p = 0.0260$). In addition, **FARSIGHT** was rated significantly more helpful than the baseline (Fig. 8.16A), while **FARSIGHT LITE** was more enjoyable (Fig. 8.16B). The effect sizes of significant results, as measured by the common language effect size [379], were all above 0.7, indicating a large effect.

Usefulness of different features. Besides comparing the two tools, participants also rated the usefulness of specific features in each tool. The average ratings are shown in Fig. 8.17. All features in our tools were rated favorably (Fig. 8.17-AB). For **FARSIGHT**, participants especially liked the interactive tree visualization. For example, P6 commented, “*This tree makes a lot of sense. This is how I think about it in my brain as well.*” Similarly, P16 appreciated the progressive disclosure in the visualization: “*I’m able to not get overwhelmed by everything all at once.*” The rating for the AI incident panel (in both **FARSIGHT** and **FARSIGHT LITE**) is relatively lower than other features. Participants explained that the surfaced incidents were not very relevant to their prompts (P39, P41), and the feature would require them to take time to read external articles (P24, P39).

8.5.6 Findings: FARSIGHT’s Role in Overcoming Harm Envisioning Challenges (RQ3)

After completing the post-task (H3), participants were asked to reflect on the biggest challenges encountered in envisioning harms associated with AI features. We examined the major themes that emerged from these challenges. In addition, by analyzing participants’ usage patterns of **FARSIGHT** and **FARSIGHT LITE**, coupled with their interview feedback, we explored how our tools mitigate certain challenges and also identified our tools’ limitations.

8.5.6.1 Challenges in envisioning harms.

We summarized three major challenges that participants encountered.

- C1. Envisioning use cases.** The most prevalent challenge in envisioning harms is to anticipate different use cases for an AI feature. Multiple participants noted that it was most challenging to imagine how different people would use technology, and it was particularly difficult to “*put myself in someone’s shoes*” (P27, P37, P39) and “*empathize with different groups of people*” (P11). Participants also underscored the vast space of possible use cases (P31, P33, P36), and “*often you don’t find out the edge cases until you actually work with it*” (P2). Some participants also emphasized that it sometimes required creativity to imagine how an AI feature would be used and especially misused (e.g., P5, P22, P23).
- C2. Bias and subjectivity in harm envisioning.** Interestingly, several participants recognized their own biases in envisioning harms (e.g., P6, P21, P31). For example, P21 noted the challenge in overcoming their biases in anticipating the impacts of AI features: “*I had been coming at it from a very American-centric point of view at first. To talk*

about bias, I hadn't even conceived of the government using this to monitor my phone, but that could happen in other places." Moreover, some participants acknowledged the subjectivity in the definition of harms, as well as in the assessment of harms' likelihood and severity. For example, while envisioning harms and selecting harms to report (H2 and H4), some participants were conscious of whether other people would agree with their identification and assessment of harms (P19, P38).

C3. Inexperience and discomfort in harm envisioning. Many participants mentioned that our study was their first time to envision harms for AI features (e.g., P17, P26, P28). For example, P26 noted *"I have never envisioned harm before. This is not something I would think of when developing AI products."* Similarly, P18 said *"I'm familiar with technical issues but not their social influence"*. Also, P30 pointed out that there were few incentives for developers to envision harms. In addition to unfamiliarity, some participants also noted that it was uncomfortable and sad to think about harms (P3, P12). For example, P3 said *"It's not comfortable thinking through all the bad things that can happen. I think in general people don't like thinking about bad things too much."*

8.5.6.2 **FARSIGHT** and **FARSIGHT LITE** address major challenges.

Our tools could help users address identified challenges.

A co-pilot for brainstorming diverse use cases. Many participants appreciated that our tools provided them with a starting point to predict use cases (e.g., P8, P29, P41). For example, after seeing a few AI-generated use cases, P8 found it much easier to envision other use cases, and similarly, P24 felt empowered to *"have a wider net to cast"* (C1). Also, P14 noted that even seeing far-fetched AI-generated content helped them brainstorm new use cases. On the other hand, P21 appreciated that **FARSIGHT** had identified many unexpected and thought-provoking use cases that provided a different perspective in anticipating harms (C2).

In situ guide that promotes user agency. Participants especially liked that our tools were directly integrated into existing AI prototyping tools and contextualized based on the prompt (e.g., P19, P31, P37), where **FARSIGHT** and **FARSIGHT LITE** required minimal effort to get started envisioning harms (C3). Participants also thought the *Incident Panel* and *Use Case Panel* as a good reminder for potential harms for the AI feature that one is prototyping (e.g., P12, P41, P42). For example, P12 commented that *"Even if it's just sitting there, it would be educational."* Many participants also liked the interactivity of our tools and found it engaging for adding new use cases, stakeholders, and harms (e.g., P9, P19, P24)—many of them noted that **FARSIGHT** was so intriguing that they would like to continue using it to explore potential harms (C3). Participants felt they had agency in harm envisioning when using **FARSIGHT**. For example, P21 commented *"If you think something [AI-generated content] is totally bonkers, whatever, just delete it."* Similarly, P4 and P5

compared the *Harm Envisioner* to a mind map, as they appreciate that the interface allows them to freely organize and revise their thoughts in harm envisioning.

8.5.6.3 Limitations of FARSIGHT and FARSIGHT LITE.

Our findings showed that, in comparison to ENVISIONING GUIDE, FARSIGHT and FARSIGHT LITE did not show significant differences in participants' ability to envision more likely or more severe harms (§ 8.5.4.1), nor did they assist participants in envisioning more likely or more severe harms (§ 8.5.5.1). Additionally, participants' feedback revealed two major limitations of our tools.

Varied quality of LLM-generated content. Depending on participants' prompts, the related AI incidents in the *Incident Panel*, and LLM-generated use cases, stakeholders, and harms were different across participants. Sometimes, participants found a few LLM-generated content confusing and unhelpful. For example, when using our tools on the math tutor prompt (H4), the incidents in the *Incident Panel* feature articles about hallucination in chat-based LLM models. Some participants found these articles too generic and not relevant to the math tutor app (P39, P41).

Also, some LLM-generated use cases could be too far-fetched. For example, for the math tutor prompt, FARSIGHT sometimes showed a use case: “*Scammers use it to explain complex investment schemes to potential victims.*” While some participants found it interesting and relevant (e.g., P14, P26), others found it unrealistic and not useful (e.g., P6, P12). This disagreement highlights the subjectivity in identifying and assessing harms (§ 8.6.2). Interestingly, a few participants defended the usefulness of far-fetched content. P24 noted “*Even if it’s wrong [LLM-generated use case], it is still kind of helpful to think beyond the immediate use case and who else can use this tool.*” Similarly, P21 said “*Some of these feel more of a stretch but it’s interesting because I could see how it gives me ideas for things to watch out for which I still appreciate.*”

Lack of actionability. Another limitation is that our tools did not provide users with actions to prevent or mitigate identified harms (P13, P22, P34). P13 also commented that increasing awareness without providing actions to address responsible AI issues could be harmful, because “*People have an empathy quota, and it might just be displacing more impactful efforts.*” Related to the discomfort that some participants had experienced when envisioning harms (C3), P40 mentioned that they felt scared and overwhelmed because there were so many possible harms and they did not know how to address them. Similarly, P29 noted that the lack of actionability made them feel anxious and disappointed:

“I’m glad that I got to know about them [potential misuses]. But I feel I’m vulnerable, probably because I can’t do much about stopping them. So that’s something that really makes me feel very disappointed. Because unless we do case-by-case analysis, this [preventing misuses] can be very tricky. I feel like

it's kind of adding anxiety to me. It's good to know, but I feel like I can't do much about it.” (P29)

We did not incorporate harm mitigation into our tools, because mitigating harms associated with LLM-powered applications remains an open research question (see more discussion in § 8.6.3). After the evaluation study, we improved FARSIGHT by providing pointers to existing LLM safety resources [e.g., 380, 381, 382, 362] when users exported their harms.

8.5.7 Limitations of Study Design

We acknowledge limitations in our tool and study designs. First, we recruited participants from a single large technology company. This was because we needed to require participants to have prior experience in prototyping LLM-powered applications using a particular prompt-crafting tool, into which we integrated **FARSIGHT** and **FARSIGHT LITE** in the study. Consequently, all 42 participants had backgrounds in the technology industry in varying roles, such as software engineers, product managers, UX researchers, and linguists⁴ as shown in Table 8.2. Our participants have a wide range of familiarity with responsible AI and prompting (Fig. 8.10), and they use LLMs for diverse tasks, including prototyping AI features with LLMs—much like the intended users of FARSIGHT. Therefore, findings from our study may be generalizable to AI prototypers who have worked in the technology industry, and who are using LLMs to prototype AI-based applications. Nevertheless, to understand how usable or effective FARSIGHT may be for a broader spectrum of AI prototypers, particularly those with limited background or knowledge of AI, such as creative writers, teachers, students, and more, further research involving individuals with more diverse backgrounds is needed. Second, we administered only one post-task (H3) immediately following the intervention (H2). To evaluate the long-term impact of our tools on users' ability to envision harms, a more extended longitudinal study is needed.

Finally, an inter-rater reliability test showed that, on average, the seven raters (i.e., of the likelihood and severity of the identified harms) only had a slight agreement. The ratings of the likelihood and severity of participants' identified harms should thus be taken as an initial step in evaluating identified harms, and not as the sole evidence demonstrating the value of this approach. The relatively low inter-rater reliability may be due to the fact that perceptions of severity and likelihood may be highly influenced by the raters' personal experiences, backgrounds, knowledge, and their positionality as a whole. Indeed, substantial prior work on annotations of offensive language, hate speech, and other linguistic phenomena [383, 322, 384, 385, 386] suggest that disagreements between raters with different subjectivities (i.e., personal backgrounds and experiences) is an inherent challenge to sociotechnical evaluations, and not one that can be solved with more or better raters. We further discuss the challenges regarding subjectivity in identifying and assessing harms in § 8.6.2.

⁴The linguists in our study work on consulting on language-based data used by AI product teams.

8.6 Discussion

8.6.1 Motivation & Engagement in Responsible AI

Potentials of *in situ* and early intervention in motivating responsible AI practices. Existing research suggests that many AI developers may not have incentives to consider potential harms related to their AI applications [316]—or may be actively disincentivized to identify such harms [347]. Our co-design user study validates this finding among an emerging community involved in AI development—AI prototypers who use LLMs to rapidly iterate on potential AI-based applications (§ 8.2.1). With the rapidly increasing access to LLMs and easy-to-use prototyping tools, it is crucial to motivate AI prototypers to consider AI risks when prototyping their AI applications or features (G3). To tackle this challenge, we propose an *in situ* system design that integrates our tool into the AI prototyper’s existing workflows and employs different design strategies to draw users’ attention without causing significant interruption to their flow. Our evaluation study shows that users appreciate our design, and find this in-context warning tool easy to adopt and engaging (§ 8.5.6.2). By showing unexpected use cases, stakeholders, and harms, FARSIGHT piques users’ interests (§ 8.5.5.2) and motivates them to brainstorm more harms (§ 8.5.5.2). These findings highlight the great potential of *in situ* design and early intervention for future responsible AI works. Therefore, future designers of AI development tools (e.g., Google AI Studio, computational notebooks, and VSCode) can natively integrate *in situ* interfaces to promote responsible AI practices. In addition, future researchers can adopt our design strategies to foster other responsible AI practices, such as illustrating bias in LLMs and encouraging development documentation at an early AI development stage.

Tension between automation and human agency. FARSIGHT’s seamless integration into AI prototypers’ workflows helps motivate AI prototypers to engage with harm envisioning. In addition, rather than asking users to anticipate harms entirely from scratch, FARSIGHT leverages LLMs to generate the initial set of use cases, stakeholders, and harms, providing users with inspiration and a foundation to build upon (§ 8.5.6.3). However, this seamless and automated design might deter users from fully engaging in and contemplating the limitations and potential risks associated with LLMs. Prior research in responsible AI has proposed the value of a *seamful* design [e.g., 387, 388], where the designers strategically reveal seams or introduce frictions or “productive restraint” [347, 146] to support increased reflection on responsible AI during development. To explore this tension and tradeoffs between a seamfully-designed workflow that is easy to use by prototypers, and a seamful design that prompts reflection-in-action [387], we (1) designed the *Harm Envisioner* to encourage users to edit LLM-generated content and steer the harm envisioning direction (§ 8.3.3, G4), and (2) evaluated two variants of our tool in the evaluation study—FARSIGHT and FARSIGHT LITE, where FARSIGHT LITE omits the *Harm Envisioner*.

Our study results highlight that participants feel they have agency (§ 8.5.6.2), and they like being able to control the harm anticipation process (Fig. 8.4). Our quantitative results also show that **FARSIGHT**, with higher human agency, is more effective than **FARSIGHT LITE** across all measures (§ 8.5.4.1, § 8.5.5.1). On the other hand, when engaging with AI-generated content, some participants also report discomfort (C3) and even anxiety (§ 8.5.6.3). Therefore, our work demonstrates that seamless design (*in situ* AI automation) and seamful design (promoting user reflection) are complementary to each other—tradeoffs and a balance between the two should be considered during the design of responsible AI tools [cf. 341]. For future responsible AI work, researchers should engage with potential users and other impacted stakeholders throughout the design process and adjust their design ideas to ensure the responsible AI tools they are designing are both easily adoptable and capable of eliciting active and critical reflection.

8.6.2 Subjectivity in Harm Envisioning

In our evaluation user study, many participants report challenges overcoming the limitations of their own experiences and perspectives when envisioning harms (C2). In addition, we also observed the seven RAI raters of participants’ harms disagreed about which harms were more or less severe or likely, resulting in a low inter-rater reliability for these two dimensions. Our empirical findings contribute to prior research that highlights the role of subjectivity and positionality in anticipating harms [89, 67] and in data annotation, particularly for annotations of toxicity or hate speech [e.g., 322, 325, 383, 384, 385]. What constitutes harm and the assessment of harm severity are often influenced by the individual’s background, lived experiences, or even the organizational culture they are working in [389, 390]. For example, for the article summarizer (H3), one participant envisioned a harm scenario: “If the summary is wrong, journalists’ reputation might be harmed.” This harm scenario received likelihood ratings of 1, 4, and 3, and severity ratings of 1, 3, and 4 from three randomly assigned raters. It is possible that the rater who assigned the ratings of 3 and 4 possessed specific knowledge about the harms of journalists using LLMs to write article summaries, which led them to rate this harm scenario as more likely and more severe.

A need for new methods to assess harms. Emerging research is beginning to develop methods for measuring and resolving disagreements among annotators in cases where there may in fact be no ground truth [e.g., 383, 385, 384, 322, 391]. Our findings in this paper—including the low inter-rater reliability of the responsible AI raters—suggest that new methods are needed in responsible AI to account for different perspectives on the severity and likelihood of potential downstream harms. This may ideally involve recruiting participants from communities or populations who may be impacted by a given AI application (e.g., the stakeholders generated by **FARSIGHT**, for instance, as well as other stakeholders identified by members of the communities themselves [392]). Moreover, with the rapidly increasing

access to LLMs and easy-to-use AI prototyping tools, AI prototypers may encompass a broader spectrum of roles beyond traditional AI practitioners [e.g., 340, 86]. Thus, they may lack either the experience or the resources to recognize the limitations of their own subjectivity when anticipating harms of their AI applications, and may lack the means to identify and engage with diverse stakeholders as part of harm envisioning.

Benefits and challenges of using LLM to envision AI harms. Our evaluation study highlights that diverse and unexpected AI-generated use cases, stakeholders, and harms in FARSIGHT help some participants overcome their own failures of imagination [89] in order to think from a broader perspective when independently envisioning harms (§ 8.5.5.2). Notably, these effects were more prominent with FARSIGHT than with existing harm envisioning processes [326] (Fig. 8.13). There are two implications of these findings. First, LLMs can be a promising tool to help AI prototypers think outside of their own perspectives, and future researchers can adapt our approach to other responsible AI practices. Second, LLMs may encode biases from their training data [e.g., 97], and FARSIGHT may also reflect the biases of its creators, as expressed in the underlying prompts used in FARSIGHT’s LLM, which raises a critical question: to what extent can LLMs be helpful as part of a harm envisioning process, without over-indexing on particular harms or leading AI prototypers to overlook other types of harms?

Our research provides an initial starting point into investigating these questions, as well as opening new questions into the role of subjectivity in harm envisioning. Future research can further investigate the factors influencing users’ ability to envision harms of AI applications, develop new ways to model and resolve disagreement among AI prototypers or other evaluators about the severity and likelihood of envisioned harms, and integrate such implications into LLM-powered responsible AI tools for AI prototypers or other AI practitioners. Future research can also explore tradeoffs between semi-automated harm envisioning processes (like FARSIGHT) and more traditional processes like value-sensitive design [e.g., 76], participatory design [e.g., 393, 390, 392], and more.

8.6.3 Mitigating Harms during AI Prototyping

A limitation of FARSIGHT is its focus on harm identification rather than harm mitigation. Participants from our co-design study (§ 8.2.1) and evaluation study (§ 8.5.6.3) wanted FARSIGHT to provide actionable items to help them prevent and mitigate identified harms. Some participants also suggested we develop an *in situ* prompt editing tool to address harms identified from FARSIGHT (§ 8.2.1). Interestingly, while using FARSIGHT, some participants *voluntarily* thought about actions and strategies to take after envisioning harms, such as implementing an appeal process, collecting better data, and revising the prompts (§ 8.5.5.2).

Looking ahead, we argue that it is crucial for future designers to provide users with harm mitigation suggestions and resources in systems similar to FARSIGHT. Some participants in

our study complained that FARSIGHT is exploiting users’ “empathy quota” and potentially desensitizing them about LLM harms, because FARSIGHT only warns users about harms without providing mitigation suggestions (§ 8.5.6.3). This concern reflects the phenomenon of “alarm fatigue” in alerting tools and monitoring alarms in healthcare. Alarm fatigue occurs “when non-actionable alarms are in the majority, and clinicians develop decreased reactivity, causing them to ‘tune out’ or ignore the alarms” [394]. Therefore, to combat alarm fatigue and effectively promote responsible AI practices, future designers should make responsible AI alerts actionable and prioritize actionable warnings in their systems.

Our findings highlight that FARSIGHT users have a great appetite for mitigation strategies during AI prototyping. We have two hypotheses for this observation. First, as FARSIGHT promotes human agency, it might also give participants a feeling of *ownership* of their identified harms. Prior research shows that triggering a feeling of ownership motivates users’ actions [145]. Another hypothesis is that FARSIGHT elicits fear by exposing participants to diverse potential harms of their AI applications, evidenced by participant-reported discomfort (C3) and anxiety (§ 8.5.6.3). Security researchers use *fear appeals* as a design strategy to motivate users to take security actions [395]. Therefore, our empirical findings highlight promising research opportunities in (1) providing *in situ* mitigation strategies during the early AI prototyping stage, and (2) investigating if *in situ* tools can increase users’ adoption of harm mitigation strategies.

8.7 Conclusion

We introduce FARSIGHT, the first *in situ* interactive tool to address the challenges in anticipating potential harms in LLM-powered applications during prototyping. By highlighting relevant AI incident reports and enabling AI prototypers to curate and modify LLM-generated use cases, stakeholders, and harms, FARSIGHT improves users’ ability to independently anticipate potential risks associated with their prompts. A user study with 42 AI prototypers shows that our tool is useful and usable. FARSIGHT fosters a user-centric approach, encouraging creators to consider end-users, and cascading harms, and extend their awareness beyond immediate harms. Our tool is open-source and readily adoptable. We hope our work will inspire future research and development of responsible AI tools that target the early stages of the AI development process.

CONCLUSIONS

In summary, my dissertation addresses the fundamental and practical challenges in understanding and guiding AI by developing scalable, easy-to-adopt, and interactive visualization tools for diverse stakeholders. My work contributes to novel visualization techniques, new human-AI interaction paradigms, user interactive workflows, and scalable algorithms. I believe my research advances human understanding of AI technologies, enabling human agency when we interact with AI systems, promoting responsible development and deployment of AI technologies, and increasing people’s trust in AI.

8.8 Research Contributions

My thesis makes research contributions across several major fronts, including human-computer interaction, machine learning, interactive visualization, and, importantly, their intersection to **explain** AI (Part I), **guide** AI (Part II), and **democratize** human-centered AI practices (Part III).

Transformative visual AI explanation: worldwide deployment and scalable insight

- The *viral success* of CNN EXPLAINER exemplifies the effectiveness of our proposed *dynamic explanation* in explaining complex AI models across various levels of abstraction (Chapter 3). Widely used by over 360,000 novices from more than 200 countries, CNN EXPLAINER has been integrated into deep learning courses across top universities including Carnegie Mellon, Georgia Tech, Duke University, and the University of Tokyo.
- Used by data scientists and researchers at *Apple* and *Google Deepmind*, WIZMAP is *the first system* that smoothly visualizes and summarize over 1,000,000 embedding points with novel algorithm-enabled *dynamic annotations* entirely in browsers (Chapter 4).
- We pioneer *on-device computing techniques* to accelerate scalable interactive visualization for complex AI models and large datasets. For example, CNN EXPLAINER *explains a live convolutional neural network* entirely in the user’s browser, without the need for installation or dedicated servers—broadening the public’s access to cutting-edge AI technologies (Chapter 3).

First-of-its-kind algorithms that enable actionable AI explainability

- Integrated into *Microsoft’s interpretability library*, GAM CHANGER empowers *millions of developers* to use simple clicks and drags to align the model behaviors with their knowledge and values. GAM CHANGER puts AI explanations into action by

introducing *the first model-editing tool* that enables practitioners and domain experts to easily modify the weights in AI models (Chapter 6). It has been recognized with the *Best Paper* award at the NeurIPS workshop on ML for clinical practice.

- GAM COACH is the *first interactive algorithmic recourse tool* that empowers end users to specify their recourse preferences and iteratively fine-tune actionable recourse plans that can alter unfavorable AI decisions, enabled by *a novel algorithm* that adapts integer linear programming (Chapter 7).

Transformative paradigms to leapfrog responsible AI adoption

- Developed in collaboration with *Google Deepmind* researchers, FARSIGHT introduces a new paradigm for designing and developing easy-to-adopt tools that can be *directly integrated into AI practitioners' existing workflows*. FARSIGHT helps practitioners envision the potential harms of their AI product when they write prompts within their favorite prompting interfaces (Chapter 8). This new paradigm has been recognized with the *Best Paper, Honorable Mention* award at CHI'24.
- Our research is easily accessible to AI researchers, practitioners, and the general public. For example, our tools can be used *directly in computational notebooks* (Chapter 4, Chapter 6), the most popular AI development environment. Additionally, by providing *publicly accessible web-based deployments* of CNN EXPLAINER, WIZMAP, GAM CHANGER, GAM COACH, and FARSIGHT that require no installation, we lower the barrier to learning and applying cutting-edge human-centered AI techniques.

Deployed and open-source systems and resources that accelerate AI innovation

- DIFFUSIONDB introduces *the first large-scale open-access* prompt dataset for text-to-image generative models with 14,000,000 image-prompt pairs, totaling 6.5 TB in size. With over 2,000,000 total data requests through the APIs to date, DIFFUSIONDB is instrumental in enabling researchers to study the real-world usage and impacts of generative AI models (Chapter 5). The impact of this dataset is recognized with the *Best Paper, Honorable Mention* award at ACL'23.
- This PhD thesis has introduced a suite of 6 paradigm-shifting *open-source* tools that empower and inspire researchers and practitioners to adopt our design and implementations in their human-centered AI research. Collectively, they have **received over 10,000 stars** on GitHub, the most popular platform for collaborative software development, demonstrating their significant impact and widespread adoption within the community.

8.9 Impact

My research is already making a significant impact on society and industry.

- CNN EXPLAINER has transformed AI education: its public demo has been integrated into deep learning courses (Carnegie Mellon, Georgia Tech, Duke University, University of Tokyo and more), helping **360,000 novices** from 200+ countries learn about seemingly complex ML concepts, and it has received **7,000 stars** on GitHub.
- DIFFUSIONDB has received over **2,000,000** data requests through the HuggingFace APIs. It is also among the **top 20 most-liked datasets** on HuggingFace out of 70,000 datasets. It has been integrated into official AI tutorials from Amazon AWS and Google Cloud.
- GAM CHANGER is **deployed in Microsoft** and integrated into their open-source library InterpretML. The tool is used by physicians in NYU hospitals on real-life hospital admission prediction models.
- WIZMAP is **used in Apple and Google** to explore large text datasets.
- My works have been recognized by three best-paper-type awards across top-tier HCI, NLP, and AI venues: FARSIGHT received the **Best Paper Honorable Mention Award** at CHI'24; DIFFUSIONDB received the **Best Paper Honorable Mention Award** at ACL'23; GAM CHANGER received the **Best Paper Award** at the NeurIPS Workshop on Bridging the Gap: From ML Research to Clinical Practice. CNN EXPLAINER was highlighted as a top visualization publication (**top 1%**) invited to present in SIGGRAPH.
- My research on democratizing human-centered AI has been invested in and recognized by an **Apple Scholars in AI/ML PhD fellowship** and a **J.P. Morgan AI PhD Fellowship**.

8.10 Future Directions

This thesis has not only made several contributions and had a significant impact on society and industry, but it has also unlocked numerous critical future research directions and practical applications of human-centered AI.

8.10.1 Human-Centered AI for All

This thesis introduces novel techniques and tools for explaining AI (Part I) and enabling human agency in AI interaction (Part II). The effectiveness of these tools relies on their practical adoption, My thesis work on *in situ* responsible AI tools (Part III) sheds light on the potential for designing human-centered AI techniques that are easy to adopt. Future researchers can further explore methods to lower the barrier to adopting human-centered AI.

8.10.1.1 *In-workflow Design to Promote Responsible AI*

Existing research suggests that many AI developers may not have incentives to consider the potential impacts of their AI applications [316]—or may be actively disincentivized

to identify potential harms [347]. Our co-design user study for FARSIGHT (Chapter 8) validates this finding among an emerging community involved in AI development—AI prototypers who use LLMs to rapidly iterate on potential AI-based applications. With the rapidly increasing access to LLMs and easy-to-use prototyping tools, it is crucial to motivate AI prototypers to consider AI risks when prototyping their AI applications or features. To tackle this challenge, we leverage *in situ* system design that integrates FARSIGHT into the AI prototyper’s existing workflows and employs different design strategies to draw users’ attention without causing significant interruption to their flow. We find FARSIGHT users appreciate our design and find this in-workflow approach easy to adopt and engaging. By showing unexpected use cases, stakeholders, and harms, FARSIGHT piques users’ interests and motivates them to brainstorm more harms. These findings highlight the great potential of *in situ* design and early intervention for future responsible AI works. For example, future designers of AI development tools (e.g., Google AI Studio and VSCode) can natively integrate *in situ* interfaces to promote responsible AI practices. Future researchers can adopt our design strategies to foster other responsible AI practices, such as illustrating bias in LLMs and encouraging development documentation at an early AI development stage.

8.10.1.2 Promoting Human-Centered AI through Computational Notebooks

Computational notebooks, such as Jupyter Notebook [396] and Colab, are the most popular programming environments among data scientists [397]. These notebooks seamlessly combine text, code, and visual outputs in a document that consists of an arbitrary number of *cells*—small text and code editors. Users can execute a code cell, and its output (e.g., text and visualizations) will be displayed below the cell. By providing a literate programming environment, notebooks enable users to perform exploratory data analysis, document their work, and share insights with collaborators [398]. This thesis work explores the integration of human-centered AI tools (e.g., WIZMAP, GAM CHANGER, and FARSIGHT) into computational notebooks, with positive feedback from users. Indeed, to create easy-to-adopt tools, there is a trend in the visualization community to develop interactive visualization systems that can be used in notebooks [e.g., 399, 400, 401]. Future researchers can further explore promoting human-centered AI through notebook workflows and democratizing authoring notebook-based tools.

Promoting human-centered AI through notebook workflows. According to a recent survey [205], there is an interesting trend that researchers exploit notebooks as a means to promote responsible AI practices (e.g., AEQUITAS [68], FAIRLEARN [402], FARSIGHT [403], and MLDOC [404]). Two motivations for this emerging trend are discussed in [205]. First, AI practitioners often lack incentives to adopt responsible AI practices [316, 63], such as fairness assessment and model documentation. By integrating responsible AI practices directly into practitioners’ existing notebook workflows, researchers aim to

minimize adoption friction and “nudge” [404] practitioners to follow these practices. For example, FARSIGHT alerts users to potential harms of their large language model-powered apps while they are developing prompts in a notebook. Similarly, MLDOC automatically creates and shows an AI “model card” [323] using content from a notebook.

Secondly, responsible AI requires collaboration across disciplines and teams within an organization [316, 86]. Because AI practitioners have already been using notebooks to collaborate with diverse stakeholders (e.g., designers and managers) [405], researchers leverage notebooks as a boundary object to facilitate responsible AI practices across teams. For example, in Deng *et al.* [406]’s study on ML fairness toolkits, a participant highlighted “*a simple notebook format and compelling visualizations are needed for [organizational] leadership to adopt the toolkits.*” As prioritizing people’s experience in human-AI interaction has become increasingly crucial, exciting research opportunities have emerged for researchers to design, develop, and evaluate notebook visualization tools that promote human-centered AI.

Democratizing notebook interactive tool creation. From our experience in developing notebook visualization tools to promote human-centered AI, we discover a spectrum of methods, varying in difficulty, for authoring notebook tools. In particular, accessing code and text and supporting bidirectional notebook-tool communication require significant engineering effort. Furthermore, some implementation strategies are only compatible with specific notebook platforms. Therefore, we see research opportunities to lower the barrier to authoring notebook interactive visualization tools that harness the full potential of notebook platforms. First, practitioners often use libraries such as D3 [180] and VegaLite [407] to develop web-based interactive visualizations. It would be valuable if these libraries integrated native support for notebook platforms or new libraries specifically targeted authoring notebook visualizations. On the other hand, researchers can also enhance notebook platforms to better support interactive visualizations. For example, similar to browser vendors sharing the same web standard, researchers can develop a universal notebook protocol that enables developers to access and communicate data using a standardized method across notebook platforms.

In addition, there is a design trade-off regarding visualization display styles and modularity partially arising from the rigid layout of the popular cell-based notebooks [408]. For example, most notebook platforms present cells in a linear manner, thereby requiring designers to decide whether to display their visualization tools within the flow of the cell or detach them from the flow. To address this trade-off, researchers can explore alternative notebook layouts. For example, researchers have introduced sticky cells [409] to break the linear presentation of notebook cells. These sticky cells provide visualization designers with the flexibility to seamlessly switch between on-demand and always-on displays. Future researchers could develop new and intelligent notebook systems that make it easy to design human-centered AI interactive tools that support computational notebooks.

8.10.2 Interactive AI Alignment

With GAM CHANGER (Chapter 6) and GAM COACH (Chapter 7), my thesis aims to empower AI practitioners and people impacted by AI systems to not only interpret AI models but also align these models with their knowledge and values. As we transition into a new AI paradigm featuring large generative models, ample exciting research opportunities are emerging to assist people, particularly end users, in steering AI models according to their preferences and values.

8.10.2.1 *Putting End Users at the Center*

During the design and implementation of GAM COACH, we have encountered many challenges in transforming technically sound ML techniques into a seamless user experience. As our targeted users are everyday people who are less familiar with ML and domain-specific concepts, one of our design goals is to help them understand necessary concepts and have a frictionless experience. We aim to achieve this goal by following a progressive disclosure and details-on-demand design strategy [296, 295] and presenting textual annotations to explain visual representations in the tool. However, our user study suggests that a few users might still find it challenging to understand and use GAM COACH at first. During our development process, we identify many edge cases that a recourse application would encounter in practice, such as features requiring integer values, features using log transformations, or features less familiar to everyday users. Our open-source implementation handles these edge cases, and we provide ML developers with simple APIs to add descriptions for domain-specific feature names in their own instances of GAM COACH. However, these practical edge cases are rarely discussed or handled in the recourse research community, since (1) the field of algorithmic recourse is relatively nascent, (2) and the main evaluation criteria of recourse research are distance-based statistics instead of *user experience* [280].

Looking ahead, in addition to developing faster and more effective techniques to explain or steer AI models, we also hope future researchers will engage with end users and incorporate user experience into their research agenda. For example, recent researchers have introduced prompting techniques to guide large language models [e.g., 410, 411, 412]. However, it remains unclear how we should design systems and interfaces that enable less experienced end users to easily steer large language models using these techniques. This thesis leverages interactive visualizations to aid end users in learning and guiding AI models. Besides interactive visualization, future researchers can also explore alternative mediums, such as through a textual [314] or multi-modal approach [315], to communicate and steer AI models.

8.10.2.2 *Collaborative Prompt Engineering*

To use and instruct general-purpose large language models to perform specific tasks, users need to provide them with *prompts*—text instructions and examples of desired outputs [413,

414]. These prompts serve as background contexts and guides for LLMs to generate text that aligns with users' objectives. Designing effective prompts, known as *prompt engineering*, poses significant challenges for LLM users [415, 337]. LLM users often rely on trial and error and employ unintuitive patterns, such as adding “think step by step” [410] to their prompts, to successfully instruct LLMs. Prompt engineering, despite its name, is considered an art [416] and is even compared to wizards learning “magic spells” [417, 221]. Prompt writers may not fully understand why certain prompts work, but they still add them to their “spell books.” Prompting is especially challenging for *non-AI-experts*, who are often confused about getting started and lack sufficient guidance and training on LLMs and prompting [338, 418].

Social prompt engineering. One promising direction to empower everyday users in instructing large language models is to leverage social computing techniques. For example, various online communities, including Promptstacks [419], ChatGPT Prompt Genius [420], and ShareGPT [421], serve as platforms for prompt creators to share tips, collaborate, and stay updated on AI advancements. User prompts from social media have also been scraped to create prompt datasets for AI model development [422]. Online prompt marketplaces, such as PromptBase [423], PromptHero [424] and ChatX [425], have emerged to allow users to buy and sell prompts for generative models. Midjourney's Discord server [246] allows users to run and share prompts for text-to-image generative models, with dedicated sections for prompt critique and improvement [49]. More recently, WORDFLOW [334] allows everyday users to easily customize prompts and LLM settings, share prompts with the community, and copy community prompts.

Colaborative systems for AI alignment. Looking ahead, future researchers can explore designs and techniques that build collaborative platforms to help everyday users effectively control generative models. For example, researchers can draw inspiration from gaming social platforms like Steam Community [426] and Pokémon GO forums [427], where gamers engage in research and share strategies to overcome in-game challenges. By comparing prompting LLMs to fighting game bosses, we can explore the design of social systems that motivate users to research and exchange prompting techniques. To incentivize user participation in prompt sharing, researchers can explore both *intrinsic motivations*, such as designing an enjoyable social system [428], and *extrinsic motivations*, such as virtual rewards and reputation systems [429, 430]. Additionally, future researchers can explore using social media ranking techniques to recommend relevant community prompts to users based on context and the user's tasks [431].

8.10.3 On-device Computing for Human-centered AI

To ensure AI explainability and human guidance are accessible to all, we leverage modern web technologies that enable running AI models and techniques directly on end users' edge

devices without the need for downloading or setting up dedicated servers. For example, CNN EXPLAINER (Chapter 3) uses WebGL to enable users to run a convolutional neural network in their browser for image classification. Similarly, GAM CHANGER (Chapter 6) employs WebAssembly to run a generalized additive model in the user’s browser and evaluate it on a dataset in real time as the user edits the model parameters. The convenience of being able to quickly use our tools without the need for installation is appreciated by educators and students, while physicians value the ability to locally modify AI model parameters without transmitting private data to the cloud. This highlights the promising research opportunities for advancing on-device computing for human-centered AI.

8.10.3.1 *On-device AI Explainability*

The web browser is a popular platform for explainable AI tools. To help *AI novices* learn about the inner workings of AI technologies, researchers develop Web-based visualization tools to interactively explain how different AI models work, such as GAN Lab [19] and CNN Explainer. Additionally, web-based visual analytics tools are developed to help *AI experts* interpret their models [e.g., 432, 276, 401]. Recently, there has been an increase in explainability tools that can run entirely within the user’s browser. For example, Microscope [433] allows users to analyze neuron representations in their browsers with pre-computation.

On the other hand, the in-browser library WEBSHAP [434] provides explanations for any AI model class using the popular posthoc model-agnostic explanation technique Kernel SHAP [243]. Using the Web as a platform, WEBSHAP makes it easier for developers to deploy explainable ML systems and enable user interactions. Moving forward, researchers can consider leveraging new Web APIs to enhance on-device explainability, such as Service Worker for offline explainability, WebSocket for collaborative interpretations, and Web Crypto for verifiable explanations. Furthermore, researchers can explore the integration of on-device explanation techniques directly into browsers through the Web Inspector tools, enabling users to easily view and interpret any ML models running on a Web page.

8.10.3.2 *On-device Large Language Models*

Traditional AI systems are typically deployed on remote servers with their outputs sent to client devices. However, there has been a recent surge of interest in deploying AI models directly on edge devices in the pursuit of private, ubiquitous, and interactive AI experiences. Tools such as TensorFlow.js [179], ONNX [435], MLC [436, 437], and Core ML [438] have significantly reduced the barriers to running large language models in browsers and mobile devices. Researchers have proposed various on-device systems, including information retrieval [439, 440], recommender systems [441, 442], prediction explanation [247, 443, 434], speech recognition [444, 445], translation [446], and writing assistants [334]. Recently, MEMEMO [447] introduces the first adaptation of dense retrieval to browsers, enabling

retrieval-augmented generation with on-device large language models [414].

Looking ahead, researchers can further explore model architecture and compression techniques to make on-device local large language models efficient. The key benefits of on-device large language models are *privacy*, *ubiquity*, and *interactivity*. On-device computing empowers users to use AI models directly on their devices, keeping sensitive model inputs secure (e.g., financial and medical information). Therefore, researchers and practitioners can design novel systems and experiences by leveraging on-device generative models. For example, researchers can leverage on-device dense storage and retrieval to design browser extensions that automatically and privately encode and store a user’s visited web pages, photos, and academic papers. These extensions can serve as an intelligent “second brain” [448] to help users capture and review knowledge. Similarly, one can explore integrating on-device large language models into the workflows in healthcare and finance domains, where data privacy is critical.

8.11 Conclusion

My dissertation pushes the frontier of AI through a human-centered approach, introducing new paradigms, techniques, and tools that not only explain AI models but also enable people to align AI with their knowledge and values. I firmly believe that we should develop AI systems *with* and *for* people. Moving forward, my mission is to innovate practical tools and techniques that empower *everyone* to interact with AI systems with ease, trust, and joy. I am dedicated to collaborating with researchers, domain experts, and everyday people to further this mission. This dissertation marks the initial step towards this goal, and I aspire to drive innovation across disciplines, ultimately making positive impacts on people’s everyday lives and society as a whole.

BIBLIOGRAPHY

- [1] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy, “Mitigating bias in algorithmic hiring: Evaluating claims and practices,” in *FAccT*, 2020.
- [2] J. Larson, S. Mattu, L. Kirchner, and J. Angwin, “How We Analyzed the COMPAS Recidivism Algorithm,” *ProPublica*, vol. 9, 2016.
- [3] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” *arXiv:1412.6572 [cs, stat]*, 2014.
- [4] J. Burrell, “How the machine ‘thinks’: Understanding opacity in machine learning algorithms,” *Big Data & Society*, vol. 3, 2016.
- [5] M. Vasconcelos, C. Cardonha, and B. Gonçalves, “Modeling Epistemological Principles for Bias Mitigation in AI Systems: An Illustration in Hiring Decisions,” in *AIES*, 2018.
- [6] S. Venkatasubramanian and M. Alfano, “The philosophical basis of algorithmic recourse,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020.
- [7] H. Nori, S. Jenkins, P. Koch, and R. Caruana, “InterpretML: A Unified Framework for Machine Learning Interpretability,” *arXiv*, 2019.
- [8] S. R. Hong, J. Hullman, and E. Bertini, “Human Factors in Model Interpretability: Industry Practices, Challenges, and Needs,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, 2020.
- [9] T. Wang, C. Rudin, F. Doshi-Velez, Y. Liu, E. Klampfl, and P. MacNeille, “A bayesian framework for learning rule sets for interpretable classification,” *JMLR*, vol. 18, 2017.
- [10] X. Hu, C. Rudin, and M. Seltzer, “Optimal Sparse Decision Trees,” in *NeurIPS*, vol. 32, 2019.
- [11] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, “Intelligible Models for Health-Care: Predicting Pneumonia Risk and Hospital 30-day Readmission,” *KDD*, 2015.
- [12] T. Hastie and R. Tibshirani, *Generalized Additive Models*. 1999.
- [13] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why Should I Trust You?’: Explaining the Predictions of Any Classifier,” in *KDD*, 2016.
- [14] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *NeurIPS*, 2017.
- [15] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, “Transfusion: Understanding transfer learning for medical imaging,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [16] C. Olah *et al.*, “The Building Blocks of Interpretability,” *Distill*, vol. 3, 2018.
- [17] M. Carney *et al.*, “Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification,” in *CHI EA*, 2020.
- [18] D. Smilkov, S. Carter, D. Sculley, F. B. Viégas, and M. Wattenberg, “Direct-Manipulation Visualization of Deep Networks,” *arXiv:1708.03788*, 2017.
- [19] M. Kahng, N. Thorat, D. H. Chau, F. B. Viegas, and M. Wattenberg, “GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation,” *IEEE TVCG*, 2019.
- [20] L. M. Krebs *et al.*, “Tell Me What You Know: GDPR Implications on Designing Transparency and Accountability for News Recommender Systems,” in *CHI EA*, 2019.
- [21] H. Park *et al.*, “NeuroCartography: Scalable Automatic Visual Summarization of Concepts in Deep Neural Networks,” *IEEE TVCG*, 2022.
- [22] K.-H. Lee, X. He, L. Zhang, and L. Yang, “CleanNet: Transfer learning for scalable image classifier training with label noise,” in *CVPR*, 2018.

- [23] H. Ben-younes, R. Cadene, N. Thome, and M. Cord, "BLOCK: Bilinear Superdiagonal Fusion for Visual Question Answering and Visual Relationship Detection," *AAAI*, vol. 33, 2019.
- [24] M. L. Kern *et al.*, "Gaining insights from social media language: Methodologies and challenges.," *Psychological Methods*, vol. 21, 2016.
- [25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv 1301.3781*, 2013.
- [26] M. E. Peters *et al.*, "Deep contextualized word representations," in *NAACL HLT*, 2018.
- [27] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021-07-18/2021-07-24.
- [28] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "With a little help from my friends: Nearest-neighbor contrastive learning of visual representations," in *ICCV*, 2021.
- [29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805*, 2018.
- [30] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv:1802.03426*, 2020.
- [31] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *JMLR*, vol. 9, 2008.
- [32] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, 1901.
- [33] S. Liu *et al.*, "Visual Exploration of Semantic Relationships in Neural Word Embeddings," *IEEE TVCG*, vol. 24, 2018.
- [34] Q. Li, K. S. Njotoprawiro, H. Haleem, Q. Chen, C. Yi, and X. Ma, "EmbeddingVis: A Visual Analytics Approach to Comparative Network Embedding Inspection," *arXiv:1808.09074*, 2018.
- [35] D. L. Arendt, N. Nur, Z. Huang, G. Fair, and W. Dou, "Parallel embeddings: A visualization technique for contrasting learned representations," in *ACM IUI*, 2020.
- [36] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg, "Embedding Projector: Interactive Visualization and Interpretation of Embeddings," *arXiv 1611.05469*, 2016.
- [37] B. Schmidt, *Deepscatter: Zoomable, animated scatterplots in the browser that scales over a billion points*, 2021.
- [38] F. Lekschas, "Regl-Scatterplot: A Scalable InteractiveJavaScript-based Scatter Plot Library," *Journal of Open Source Software*, vol. 8, 2023.
- [39] Y. Liu, E. Jun, Q. Li, and J. Heer, "Latent Space Cartography: Visual Analysis of Vector Space Embeddings," *Computer Graphics Forum*, vol. 38, 2019.
- [40] F. Heimerl, C. Kralj, T. Moller, and M. Gleicher, "*embComp* : Visual Interactive Comparison of Vector Embeddings," *IEEE TVCG*, vol. 28, 2022.
- [41] V. Sivaraman, Y. Wu, and A. Perer, "Emblaze: Illuminating Machine Learning Representations through Interactive Comparison of Embedding Spaces," in *ACM IUI*, 2022.
- [42] A. Boggust, B. Carter, and A. Satyanarayan, "Embedding Comparator: Visualizing Differences in Global Structure and Local Neighborhoods via Small Multiples," in *ACM IUI*, 2022.
- [43] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing," *ACM Computing Surveys*, 2022.
- [44] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp, "Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

- [45] O. Rubin, J. Herzig, and J. Berant, “Learning To Retrieve Prompts for In-Context Learning,” in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.
- [46] S. Bach *et al.*, “PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2022.
- [47] H. Qiao, V. Liu, and L. Chilton, “Initial Images: Using Image Prompts to Improve Subject Representation in Multimodal AI Generated Art,” in *Creativity and Cognition*, 2022.
- [48] N. Pavlichenko and D. Ustalov, “Best Prompts for Text-to-Image Models and How to Find Them,” *arXiv 2209.11711*, 2022.
- [49] J. Oppenlaender, “A Taxonomy of Prompt Modifiers for Text-To-Image Generation,” *arXiv 2204.13988*, 2022.
- [50] V. Liu and L. B. Chilton, “Design Guidelines for Prompt Engineering Text-to-Image Generative Models,” in *CHI Conference on Human Factors in Computing Systems*, 2022.
- [51] S. Shameem, *Lexica: Building a Creative Tool for the Future*, 2022.
- [52] B. J. Dietvorst, J. P. Simmons, and C. Massey, “Overcoming Algorithm Aversion: People Will Use Imperfect Algorithms If They Can (Even Slightly) Modify Them,” *Management Science*, 2018.
- [53] D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba, “Understanding the role of individual units in a deep neural network,” *PNAS*, vol. 117, 2020.
- [54] A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, F. Dalvi, and J. Glass, “Identifying and controlling important neurons in neural machine translation,” in *ICLR*, 2019.
- [55] X. Suau, L. Zappella, and N. Apostoloff, “Finding Experts in Transformer Models,” *arXiv*, 2020.
- [56] K. Meng, A. Sen Sharma, A. Andonian, Y. Belinkov, and D. Bau, “Mass editing memory in a transformer,” *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.
- [57] H. Zhang, T. Nakamura, T. Isohara, and K. Sakurai, “A Review on Machine Unlearning,” *SN Computer Science*, vol. 4, 2023.
- [58] P. Hase, M. Bansal, B. Kim, and A. Ghandeharioun, “Does Localization Inform Editing? Surprising Differences in Causality-Based Localization vs. Knowledge Editing in Language Models,” *arXiv 2301.04213*, 2023.
- [59] S. Wachter, B. Mittelstadt, and C. Russell, “Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR,” *SSRN Electronic Journal*, 2017.
- [60] Z. Cui, W. Chen, Y. He, and Y. Chen, “Optimal Action Extraction for Random Forests and Boosted Trees,” in *KDD*, 2015.
- [61] E. Delaney, D. Greene, and M. T. Keane, “Instance-based Counterfactual Explanations for Time Series Classification,” *arXiv:2009.13211 [cs, stat]*, 2021.
- [62] A. Dhurandhar *et al.*, “Explanations based on the missing: Towards contrastive explanations with pertinent negatives,” in *NeurIPS*, 2018.
- [63] D. Schiff, B. Rakova, A. Ayesh, A. Fanti, and M. Lennon, “Principles to Practices for Responsible AI: Closing the Gap,” *arXiv 2006.04707*, 2020.
- [64] B. Rakova, J. Yang, H. Cramer, and R. Chowdhury, “Where Responsible AI meets Reality: Practitioner Perspectives on Enablers for Shifting Organizational Practices,” *CSCW*, vol. 5, 2021.
- [65] C. Fiesler, N. Garrett, and N. Beard, “What Do We Teach When We Teach Tech Ethics?: A Syllabi Analysis,” in *SIGCSE*, 2020.
- [66] M. K. Hong, A. Fourney, D. DeBellis, and S. Amershi, “Planning for Natural Language Failures with the AI Playbook,” in *CHI*, 2021.

- [67] D. Liu *et al.*, “Examining Responsibility and Deliberation in AI Impact Statements and Ethics Reviews,” in *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, 2022.
- [68] P. Saleiro *et al.*, “Aequitas: A Bias and Fairness Audit Toolkit,” *arXiv 1811.05577*, 2019.
- [69] Microsoft, *Responsible AI Toolbox*, Microsoft, 2020.
- [70] H. Shen, L. Wang, W. H. Deng, C. Brusse, R. Velgersdijk, and H. Zhu, “The Model Card Authoring Toolkit: Toward Community-centered, Deliberation-driven AI Design,” in *FAccT*, 2022.
- [71] P. A. Brey, “Anticipatory ethics for emerging technologies,” *Nanoethics*, vol. 6, 2012.
- [72] P. Nanayakkara, N. Diakopoulos, and J. Hullman, “Anticipatory ethics and the role of uncertainty,” *arXiv preprint arXiv:2011.13170*, 2020.
- [73] R. Y. Wong and V. Khovanskaya, *Speculative Design in HCI: From Corporate Imaginations to Critical Orientations*. 2018.
- [74] J. Auger, “Speculative design: Crafting the speculation,” *Digital Creativity*, vol. 24, 2013.
- [75] A. Dunne and F. Raby, *Speculative Everything: Design, Fiction, and Social Dreaming*. 2013.
- [76] B. Friedman, “Value-sensitive design,” *interactions*, vol. 3, 1996.
- [77] B. Friedman, P. Kahn, and A. Borning, “Value sensitive design: Theory and methods,” *University of Washington technical report*, vol. 2, 2002.
- [78] B. Friedman, D. G. Hendry, A. Borning, *et al.*, “A survey of value sensitive design methods,” *Foundations and Trends® in Human-Computer Interaction*, vol. 11, 2017.
- [79] S. S. Chivukula, Z. Li, A. C. Pivonka, J. Chen, and C. M. Gray, “Surveying the landscape of ethics-focused design methods,” *arXiv preprint arXiv:2102.08909*, 2021.
- [80] B. Friedman and D. Hendry, “The envisioning cards: A toolkit for catalyzing humanistic and technical imaginations,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012.
- [81] H. Shen, W. H. Deng, A. Chattopadhyay, Z. S. Wu, X. Wang, and H. Zhu, “Value Cards: An Educational Toolkit for Teaching Social Impacts of Machine Learning through Deliberation,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.
- [82] R. Y. Wong and T. Nguyen, “Timelines: A world-building activity for values advocacy,” in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.
- [83] S. Klassen and C. Fiesler, ““Run Wild a Little With Your Imagination” ethical speculation in computing education with black mirror,” in *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*, 2022.
- [84] S. Ballard, K. M. Chappell, and K. Kennedy, “Judgment Call the Game: Using Value Sensitive Design and Design Fiction to Surface Ethical Concerns Related to Technology,” in *Proceedings of the 2019 on Designing Interactive Systems Conference*, 2019.
- [85] Doteveryone, *Consequence Scanning – an agile practice for responsible innovators*, 2019.
- [86] Q. Wang, M. Madaio, S. Kane, S. Kapania, M. Terry, and L. Wilcox, “Designing Responsible AI: Adaptations of UX Practice to Meet Responsible AI Challenges,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.
- [87] B. Hecht *et al.*, “It’s time to do something: Mitigating the negative impacts of computing through a change to the peer review process,” *arXiv preprint arXiv:2112.09544*, 2021.
- [88] C. E. Prunkl, C. Ashurst, M. Anderl jung, H. Webb, J. Leike, and A. Dafoe, “Institutionalizing ethics in AI through broader impact requirements,” *Nature Machine Intelligence*, vol. 3, 2021.
- [89] M. Boyarskaya, A. Olteanu, and K. Crawford, “Overcoming Failures of Imagination in AI Infused System Development and Deployment,” *arXiv 2011.13416*, 2020.

- [90] C. Ashurst, E. Hine, P. Sedille, and A. Carlier, “Ai ethics statements: Analysis and lessons learnt from neurips broader impact statements,” in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.
- [91] P. Nanayakkara, J. Hullman, and N. Diakopoulos, “Unpacking the expressed consequences of AI research in broader impact statements,” in *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 2021.
- [92] K. Sim, A. Brown, and A. Hassoun, “Thinking through and writing about research ethics beyond” Broader Impact”,” *arXiv preprint arXiv:2104.08205*, 2021.
- [93] M. Sturdee *et al.*, “Consequences, schmonsequences! Considering the future as part of publication and peer review in computing research,” in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.
- [94] K. Do, R. Y. Pang, J. Jiang, and K. Reinecke, ““That’s important, but...”: How computer science researchers anticipate unintended consequences of their research innovations,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.
- [95] Z. Buçinca, C. M. Pham, M. Jakesch, M. T. Ribeiro, A. Olteanu, and S. Amershi, “AHA!: Facilitating AI Impact Assessment by Generating Examples of Harms,” *arXiv 2306.03280*, 2023.
- [96] R. Bommasani *et al.*, “On the Opportunities and Risks of Foundation Models,” *arXiv 2108.07258*, 2022.
- [97] L. Weidinger *et al.*, “Ethical and social risks of harm from Language Models,” *arXiv 2112.04359*, 2021.
- [98] Y. Liu *et al.*, “Trustworthy LLMs: A Survey and Guideline for Evaluating Large Language Models’ Alignment,” *arXiv 2308.05374*, 2023.
- [99] S. Matz, J. Teeny, S. S. Vaid, G. M. Harari, and M. Cerf, “The Potential of Generative AI for Personalized Persuasion at Scale,” *PsyArXiv*, Preprint, 2023.
- [100] T. Shevlane *et al.*, “Model evaluation for extreme risks,” *arXiv 2305.15324*, 2023.
- [101] Y. Pan, L. Pan, W. Chen, P. Nakov, M.-Y. Kan, and W. Y. Wang, “On the Risk of Misinformation Pollution with Large Language Models,” *arXiv 2305.13661*, 2023.
- [102] H. W. A. Hanley and Z. Durumeric, “Machine-Made Media: Monitoring the Mobilization of Machine-Generated Articles on Misinformation and Mainstream News Websites,” *arXiv 2305.09820*, 2023.
- [103] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, “RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models,” *arXiv 2009.11462*, 2020.
- [104] A. Deshpande, V. Murahari, T. Rajpurohit, A. Kalyan, and K. Narasimhan, “Toxicity in ChatGPT: Analyzing Persona-assigned Language Models,” *arXiv 2304.05335*, 2023.
- [105] A. Glaese *et al.*, “Improving alignment of dialogue agents via targeted human judgements,” *arXiv 2209.14375*, 2022.
- [106] G. Simmons, “Moral mimicry: Large language models produce moral rationalizations tailored to political identity,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, 2023.
- [107] G. Deng *et al.*, “MasterKey: Automated Jailbreak Across Multiple Large Language Model Chatbots,” *arXiv 2307.08715*, 2023.
- [108] S. S. Roy, K. V. Naragam, and S. Nilizadeh, “Generating Phishing Attacks using ChatGPT,” *arXiv 2305.05133*, 2023.
- [109] H. Li *et al.*, “Multi-step Jailbreaking Privacy Attacks on ChatGPT,” *arXiv 2304.05197*, 2023.
- [110] S. Kim, S. Yun, H. Lee, M. Gubri, S. Yoon, and S. J. Oh, “ProPILE: Probing Privacy Leakage in Large Language Models,” *arXiv 2307.01881*, 2023.

- [111] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, “Quantifying Memorization Across Neural Language Models,” *arXiv 2202.07646*, 2023.
- [112] N. Carlini *et al.*, “Extracting Training Data from Large Language Models,” *arXiv 2012.07805*, 2021.
- [113] K. Malinka, M. Perešini, A. Firc, O. Hujňák, and F. Januš, “On the Educational Impact of ChatGPT: Is Artificial Intelligence Ready to Obtain a University Degree?” In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, 2023.
- [114] C. Longoni, A. Fradkin, L. Cian, and G. Pennycook, “News from Generative Artificial Intelligence Is Believed Less,” in *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.
- [115] E. Perez *et al.*, “Red Teaming Language Models with Language Models,” *arXiv 2202.03286*, 2022.
- [116] E. Derner and K. Batistič, “Beyond the Safeguards: Exploring the Security Risks of ChatGPT,” *arXiv 2305.08005*, 2023.
- [117] L. Weidinger *et al.*, “Sociotechnical Safety Evaluation of Generative AI Systems,” *arXiv 2310.11986*, 2023.
- [118] L. Weidinger *et al.*, “Taxonomy of Risks posed by Language Models,” in *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.
- [119] I. Solaiman and C. Dennison, “Process for adapting language models to society (PALMS) with values-targeted datasets,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [120] J. Welbl *et al.*, “Challenges in Detoxifying Language Models,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021.
- [121] A. Xu, E. Pathak, E. Wallace, S. Gururangan, M. Sap, and D. Klein, “Detoxifying Language Models Risks Marginalizing Minority Voices,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.
- [122] B. Krause *et al.*, “GeDi: Generative Discriminator Guided Sequence Generation,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021.
- [123] T. Schick, S. Udupa, and H. Schütze, “Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP,” *arXiv 2103.00453*, 2021.
- [124] A. Askeff *et al.*, “A General Language Assistant as a Laboratory for Alignment,” *arXiv 2112.00861*, 2021.
- [125] O. O. Q. in AI *et al.*, “Queer In AI: A Case Study in Community-Led Participatory AI,” in *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, 2023.
- [126] F. Delgado, S. Yang, M. Madaio, and Q. Yang, “Stakeholder Participation in AI: Beyond ”Add Diverse Stakeholders and Stir”,” *arXiv 2111.01122*, 2021.
- [127] C. Harrington, S. Erete, and A. M. Piper, “Deconstructing Community-Based Collaborative Design: Towards More Equitable Participatory Design Engagements,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, 2019.
- [128] N. Cooper *et al.*, “A Systematic Review and Thematic Analysis of Community-Collaborative Approaches to Computing Research,” in *CHI Conference on Human Factors in Computing Systems*, 2022.
- [129] B. Reinheimer *et al.*, “An investigation of phishing awareness and education over time: When and how to best remind users,” in *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, 2020.
- [130] M.-E. Maurer, A. De Luca, and S. Kempe, “Using data type based security alert dialogs to raise online security awareness,” in *Proceedings of the Seventh Symposium on Usable Privacy and Security*, 2011.
- [131] A. Mylonas, A. Kastania, and D. Gritzalis, “Delegate the smartphone user? Security awareness in smartphone platforms,” *Computers & Security*, vol. 34, 2013.

- [132] S. Egelman and S. Schechter, “The importance of being earnest [in security warnings],” in *Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers 17*, 2013.
- [133] R. W. Reeder, A. P. Felt, S. Consolvo, N. Malkin, C. Thompson, and S. Egelman, “An Experience Sampling Study of User Reactions to Browser Warnings in the Field,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018.
- [134] A. P. Felt *et al.*, “Improving SSL Warnings: Comprehension and Adherence,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.
- [135] R. Biddle, P. C. Van Oorschot, A. S. Patrick, J. Sobey, and T. Whalen, “Browser interfaces and extended validation SSL certificates: An empirical study,” in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, 2009.
- [136] A. P. Felt, R. W. Reeder, H. Almuhammedi, and S. Consolvo, “Experimenting at scale with google chrome’s SSL warning,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014.
- [137] M. Wu, R. C. Miller, and S. L. Garfinkel, “Do security toolbars actually prevent phishing attacks?” In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006.
- [138] S. Egelman, L. F. Cranor, and J. Hong, “You’ve been warned: An empirical study of the effectiveness of web browser phishing warnings,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008.
- [139] B. B. Anderson, C. B. Kirwan, J. L. Jenkins, D. Eargle, S. Howard, and A. Vance, “How Polymorphic Warnings Reduce Habituation in the Brain: Insights from an fMRI Study,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.
- [140] R. Böhme and S. Köpsell, “Trained to accept?: A field experiment on consent dialogs,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010.
- [141] N. Good *et al.*, “Stopping spyware at the gate: A user study of privacy, notice and spyware,” in *Proceedings of the 2005 Symposium on Usable Privacy and Security - SOUPS '05*, 2005.
- [142] N. S. Good, J. Grossklags, D. K. Mulligan, and J. A. Konstan, “Noticing notice: A large-scale experiment on the timing of software license agreements,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2007.
- [143] J. C. Brustoloni and R. Villamarín-Salomón, “Improving security decisions with polymorphic and audited dialogs,” in *Proceedings of the 3rd Symposium on Usable Privacy and Security*, 2007.
- [144] C. Schneider, M. Weinmann, and J. Vom Brocke, “Digital nudging: Guiding online user choices through interface design,” *Communications of the ACM*, vol. 61, 2018.
- [145] A. Caraban, E. Karapanos, D. Gonçalves, and P. Campos, “23 Ways to Nudge: A Review of Technology-Mediated Nudging in Human-Computer Interaction,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.
- [146] B. Kaiser, J. Wei, E. Lucherini, K. Lee, J. N. Matias, and J. Mayer, “Adapting security warnings to counter online disinformation,” in *30th USENIX Security Symposium (USENIX Security 21)*, 2021.
- [147] F. Sharevski, A. Devine, P. Jachim, and E. Pieroni, “Meaningful Context, a Red Flag, or Both? Preferences for Enhanced Misinformation Warnings Among US Twitter Users,” in *2022 European Symposium on Usable Security*, 2022.
- [148] G. Simon, “OpenWeb tests the impact of “nudges” in online discussions,” *OpenWeb Blog*, 2020.
- [149] J. Kiskola *et al.*, “Online Survey on Novel Designs for Supporting Self-Reflection and Emotion Regulation in Online News Commenting,” in *Proceedings of the 25th International Academic Mindtrek Conference*, 2022.

- [150] A. P. Wright *et al.*, “RECAST: Enabling User Recourse and Interpretability of Toxicity Detection Models with Interactive Visualization,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, 2021.
- [151] Grammarly, *Grammarly: Free Writing AI Assistance*, 2023.
- [152] Google, *Lighthouse*, 2016.
- [153] Deque, *Axe DevTools: Digital Accessibility Testing Tools Dev Teams Love*, 2023.
- [154] Z. J. Wang *et al.*, “CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization,” *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2020.
- [155] Z. J. Wang, F. Hohman, and D. H. Chau, “WizMap: Scalable interactive visualization for exploring large machine learning embeddings,” in *ACL Demo*, 2023.
- [156] Z. J. Wang, E. Montoya, D. Munechika, H. Yang, B. Hoover, and D. H. Chau, “DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models,” in *ACL*, 2023.
- [157] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, 2015.
- [158] M. Kahng, N. Thorat, D. H. Chau, F. B. Viegas, and M. Wattenberg, “GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, 2019.
- [159] A. W. Harley, “An Interactive Node-Link Visualization of Convolutional Neural Networks,” in *ISVC*, 2015.
- [160] A. Karpathy, *ConvNetJS MNIST demo*, 2016.
- [161] M. Kahng and D. H. Chau, “How Does Visualization Help People Learn Deep Learning? Evaluation of GAN Lab,” in *IEEE VIS 2019 Workshop on Evaluation of Interactive Visual Machine Learning Systems*, 2019.
- [162] A. P. Norton and Y. Qi, “Adversarial-Playground: A Visualization Suite Showing How Adversarial Examples Fool Deep Learning,” *arXiv:1708.00807 [cs]*, 2017.
- [163] C. Olah, *Neural Networks, Manifolds, and Topology*, 2014.
- [164] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, “Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers,” *arXiv:1801.06889 [cs, stat]*, 2018.
- [165] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognition*, vol. 77, 2018.
- [166] M. Carney *et al.*, “Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification,” in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.
- [167] E. Fouh, M. Akbar, and C. A. Shaffer, “The Role of Visualization in Computer Science Education,” *Computers in the Schools*, vol. 29, 2012.
- [168] R. E. Mayer and R. B. Anderson, “Animations need narrations: An experimental test of a dual-coding hypothesis,” *Journal of Educational Psychology*, vol. 83, 1991.
- [169] C. Kehoe, J. Stasko, and A. Taylor, “Rethinking the evaluation of algorithm animations as learning aids: An observational study,” *International Journal of Human-Computer Studies*, vol. 54, 2001.
- [170] M. Abadi *et al.*, “TensorFlow: A system for large-scale machine learning,” in *OSDI*, 2016.
- [171] A. Paszke *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [172] E. Stevens, *Deep Learning with PyTorch*. 2019.
- [173] *Backpropagation Algorithm*.
- [174] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv:1409.1556 [cs]*, 2015.
- [175] Z. J. Wang *et al.*, “CNN 101: Interactive Visual Learning for Convolutional Neural Networks,” in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.

- [176] T. L. Naps *et al.*, “Exploring the role of visualization and engagement in computer science education,” *ACM SIGCSE Bulletin*, vol. 35, 2003.
- [177] A. Karpathy, *CS231n Convolutional Neural Networks for Visual Recognition*, 2016.
- [178] *Tiny ImageNet Visual Recognition Challenge*, 2015.
- [179] D. Smilkov *et al.*, “TensorFlow.js: Machine Learning for the Web and Beyond,” *arXiv*, 2019.
- [180] M. Bostock, V. Ogievetsky, and J. Heer, “D³ Data-Driven Documents,” *IEEE TVCG*, vol. 17, 2011.
- [181] M. D. Byrne, R. Catrambone, and J. T. Stasko, “Evaluating animations as student aids in learning computer algorithms,” *Computers & Education*, vol. 33, 1999.
- [182] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” 2017.
- [183] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.
- [184] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *arXiv:1512.03385 [cs]*, 2015.
- [185] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, 1997.
- [186] A. Vaswani *et al.*, “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17, 2017.
- [187] T. L. Naps, J. R. Eagan, and L. L. Norton, “JHAVÉ—an environment to actively engage students in Web-based algorithm visualizations,” *ACM SIGCSE Bulletin*, vol. 32, 2000.
- [188] J. T. Stasko, “Using student-built algorithm animations as learning aids,” *ACM SIGCSE Bulletin*, vol. 29, 1997.
- [189] M. Conlen, A. Kale, and J. Heer, “Capture & Analysis of Active Reading Behaviors for Interactive Articles on the Web,” *Computer Graphics Forum*, vol. 38, 2019.
- [190] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, “Man is to computer programmer as woman is to homemaker? Debiasing word embeddings,” in *NeurIPS*, vol. 29, 2016.
- [191] J. Tang, J. Liu, M. Zhang, and Q. Mei, “Visualizing Large-scale and High-dimensional Data,” *WWW*, 2016.
- [192] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, “Man is to computer programmer as woman is to homemaker? Debiasing word embeddings,” in *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [193] R. Sevastjanova, E. Cakmak, S. Ravfogel, R. Cotterell, and M. El-Assady, “Visual Comparison of Language Model Adaptation,” *IEEE TVCG*, 2022.
- [194] S. Robertson, Z. J. Wang, D. Moritz, M. B. Kery, and F. Hohman, “Angler: Helping Machine Translation Practitioners Prioritize Model Improvements,” in *CHI Conference on Human Factors in Computing Systems*, 2023.
- [195] R. A. Finkel and J. L. Bentley, “Quad trees a data structure for retrieval on composite keys,” *Acta Informatica*, vol. 4, 1974.
- [196] K. Sparck Jones, “A statistical interpretation of term specificity and its application in retrieval,” *Journal of Documentation*, vol. 28, 1972.
- [197] M. Grootendorst, “BERTopic: Neural topic modeling with a class-based TF-IDF procedure,” *arXiv preprint arXiv:2203.05794*, 2022.
- [198] M. Rosenblatt, “Remarks on Some Nonparametric Estimates of a Density Function,” *The Annals of Mathematical Statistics*, vol. 27, 1956.
- [199] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. 2018.
- [200] M. Gleicher, “Considerations for Visualizing Comparison,” *IEEE TVCG*, vol. 24, 2018.

- [201] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, and C. Olah, “Activation Atlas,” *Distill*, vol. 4, 2019.
- [202] M. Lysenko, *Regl: Functional WebGL*, 2016.
- [203] T. Wilkerling, *FlexSearch: Next-Generation full text search library for Browser and Node.js*, 2019.
- [204] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in python,” *JMLR*, vol. 12, 2011.
- [205] Z. J. Wang, D. Munechika, S. Lee, and D. H. Chau, “SuperNOVA: Design Strategies and Opportunities for Interactive Visualization in Computational Notebooks,” in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2024.
- [206] Z. J. Wang, D. Munechika, S. Lee, and D. H. Chau, “NOVA: A Practical Method for Creating Notebook-Ready Visual Analytics,” *arXiv:2205.03963*, 2022.
- [207] S. Rohatgi, *ACL anthology corpus with full text*, Github, 2022.
- [208] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, “Mpnet: Masked and permuted pre-training for language understanding,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [209] A. Coenen and A. Pearce, *Understanding UMAP*, 2019.
- [210] T. Hoeger, C. Dew, F. Pauls, and J. Wilson, *Newline Delimited JSON: A standard for delimiting JSON in stream protocols*, 2014.
- [211] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *CVPR*, 2022.
- [212] Z. J. Wang, E. Montoya, D. Munechika, H. Yang, B. Hoover, and D. H. Chau, “DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models,” *arXiv:2210.14896*, 2022.
- [213] A. Borji, “Generated Faces in the Wild: Quantitative Comparison of Stable Diffusion, Midjourney and DALL-E 2,” *arXiv 2210.00586*, 2022.
- [214] P.-M. Law, A. Endert, and J. Stasko, “Characterizing Automated Data Insights,” in *2020 IEEE Visualization Conference (VIS)*, 2020.
- [215] C. Bird, E. Ungless, and A. Kasirzadeh, “Typology of Risks of Generative Text-to-Image Models,” in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, 2023.
- [216] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical Text-Conditional Image Generation with CLIP Latents,” *arXiv 2204.06125*, 2022.
- [217] C. Saharia *et al.*, “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding,” *arXiv 2205.11487*, 2022.
- [218] K. Roose, *An A.I.-Generated Picture Won an Art Prize. Artists Aren’t Happy*. 2022.
- [219] P. Chambon, C. Bluethgen, C. P. Langlotz, and A. Chaudhari, “Adapting Pretrained Vision-Language Foundational Models to Medical Imaging Domains,” *arXiv 2210.04133*, 2022.
- [220] J. Ho *et al.*, “Imagen Video: High Definition Video Generation with Diffusion Models,” *arXiv 2210.02303*, 2022.
- [221] S. Willison, A. Stacoviak, and J. Stacoviak, *Stable Diffusion Breaks the Internet*, 2022.
- [222] G. Branwen, *GPT-3 Creative Fiction*, 2020.
- [223] L. Reynolds and K. McDonell, “Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm,” in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.
- [224] StabilityAI, *Stable Diffusion Dream Studio beta Terms of Service*, 2022.
- [225] StabilityAI, *Stable Diffusion Discord Server Rules*, 2022.
- [226] O. Holub, *DiscordChatExporter: Exports Discord Chat Logs to a File*, 2017.
- [227] L. Richardson, “Beautiful Soup Documentation,” 2007.
- [228] A. Clark, *Pillow: Python Imaging Library (Fork)*, 2015.
- [229] Google, *Comparative Study of WebP, JPEG and JPEG 2000*, 2010.

- [230] L. Hanu and Unitary team, *Detoxify: Toxic Comment Classification with Pytorch Lightning and Transformers*, 2020.
- [231] C. Schuhmann *et al.*, “LAION-5B: An open large-scale dataset for training next generation image-text models,” *arXiv 2210.08402*, 2022.
- [232] P. Leach, M. Mealling, and R. Salz, “A Universally Unique IDentifier (UUID) URN Namespace,” RFC Editor, Tech. Rep., 2005.
- [233] Apache, *Apache Parquet: Open Source, Column-oriented Data File Format Designed for Efficient Data Storage and Retrieval*, 2013.
- [234] P. V. Platen *et al.*, *Diffusers: State-of-the-art diffusion models*, 2022.
- [235] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017.
- [236] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, “spaCy: Industrial-strength natural language processing in python,” 2020.
- [237] W. Wang, H. Wang, G. Dai, and H. Wang, “Visualization of large hierarchical data by circle packing,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006.
- [238] H. Hotelling, “Relations Between Two Sets of Variates,” *Biometrika*, vol. 28, 1936.
- [239] E. Hyvönen and E. Mäkelä, “Semantic Autocompletion,” in *The Semantic Web – ASWC 2006*, vol. 4185, 2006.
- [240] L.-J. Li, C. Wang, Y. Lim, D. M. Blei, and L. Fei-Fei, “Building and using a semantivisual image hierarchy,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.
- [241] T. Griffiths, M. Jordan, J. Tenenbaum, and D. Blei, “Hierarchical topic models and the nested chinese restaurant process,” in *Advances in Neural Information Processing Systems*, vol. 16, 2003.
- [242] M. T. Llano *et al.*, “Explainable Computational Creativity,” *arXiv 2205.05682*, 2022.
- [243] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17, 2017.
- [244] K. Wiggers, *Deepfakes for all: Uncensored AI art model prompts ethics questions*, 2022.
- [245] Y. Mirsky and W. Lee, “The Creation and Detection of Deepfakes: A Survey,” *ACM Computing Surveys*, vol. 54, 2022.
- [246] D. Holz, *Midjourney: Exploring New Mediums of Thought and Expanding the Imaginative Powers of the Human Species*, 2022.
- [247] Z. J. Wang *et al.*, “Interpretability, Then What? Editing Machine Learning Models to Reflect Human Knowledge and Values,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD ’22, 2022.
- [248] Z. J. Wang, J. W. Vaughan, R. Caruana, and D. H. Chau, “GAM Coach: Towards Interactive and User-centered Algorithmic Recourse,” in *CHI*, 2023.
- [249] M. T. Ribeiro, S. Singh, and C. Guestrin, ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
- [250] C.-H. Chang, S. Tan, B. Lengerich, A. Goldenberg, and R. Caruana, “How Interpretable and Trustworthy are GAMs?” *KDD*, 2021.
- [251] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker, “Accurate intelligible models with pairwise interactions,” in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

- [252] Y. Lou, R. Caruana, and J. Gehrke, “Intelligible models for classification and regression,” in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '12*, 2012.
- [253] C. Wang, B. Han, B. Patel, F. Mohideen, and C. Rudin, “In Pursuit of Interpretable, Fair and Accurate Machine Learning for Criminal Recidivism Prediction,” *arXiv:2005.04176*, 2020.
- [254] E. Wall, L. M. Blaha, L. Franklin, and A. Endert, “Warning, Bias May Occur: A Proposed Approach to Detecting Cognitive Bias in Interactive Visual Analytics,” in *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2017.
- [255] B. Shneiderman, “Direct Manipulation: A Step Beyond Programming Languages,” *Computer*, vol. 16, 1983.
- [256] R. E. Barlow, *Statistical Inference under Order Restrictions: The Theory and Application of Isotonic Regression* (Wiley Series in Probability and Mathematical Statistics, No. 8). 1972.
- [257] Z. J. Wang *et al.*, “GAM Changer: Editing Generalized Additive Models with Interactive Visualization,” *arXiv:2112.03245*, 2021.
- [258] *Lending Club: Online Personal Loans at Great Rates*, 2018.
- [259] J. M. Croswell, D. F. Ransohoff, and B. S. Kramer, “Principles of Cancer Screening: Lessons From History and Study Design Issues,” *Seminars in Oncology*, vol. 37, 2010.
- [260] N. Siddiqi, *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. 2013.
- [261] C. C. S. Liem *et al.*, “Psychology Meets Machine Learning: Interdisciplinary Perspectives on Algorithmic Job Candidate Screening,” in *Explainable and Interpretable Models in Computer Vision and Machine Learning*, 2018.
- [262] A. Waters and R. Miikkulainen, “GRADE: Machine Learning Support for Graduate Admissions,” *AI Magazine*, vol. 35, 2014.
- [263] A. D. Selbst and S. Barocas, “The Intuitive Appeal of Explainable Machines,” *SSRN Electronic Journal*, 2018.
- [264] B. Ustun, A. Spangher, and Y. Liu, “Actionable Recourse in Linear Classification,” in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019.
- [265] T. Le, S. Wang, and D. Lee, “GRACE: Generating Concise and Informative Contrastive Sample to Explain Neural Network Model’s Prediction,” *arXiv:1911.02042 [cs, stat]*, 2020.
- [266] R. K. Mothilal, A. Sharma, and C. Tan, “Explaining machine learning classifiers through diverse counterfactual explanations,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020.
- [267] C. Russell, “Efficient Search for Diverse Coherent Explanations,” in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019.
- [268] A.-H. Karimi, B. Schölkopf, and I. Valera, “Algorithmic Recourse: From Counterfactual Explanations to Interventions,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.
- [269] S. Verma, J. Dickerson, and K. Hines, “Counterfactual Explanations for Machine Learning: A Review,” *arXiv:2010.10596 [cs, stat]*, 2020.
- [270] S. Barocas, A. D. Selbst, and M. Raghavan, “The hidden assumptions behind counterfactual explanations and principal reasons,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020.
- [271] Z. Zahedi, A. Olmo, T. Chakraborti, S. Sreedharan, and S. Kambhampati, “Towards Understanding User Preferences for Explanation Types in Model Reconciliation,” in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2019.

- [272] T. Lombrozo, “Explanatory Preferences Shape Learning and Inference,” *Trends in Cognitive Sciences*, vol. 20, 2016.
- [273] L. Kirfel and A. Liefgreen, “What If (and How...)? - Actionability Shapes People’s Perceptions of Counterfactual Explanations in Automated Decision-Making,” in *ICML Workshop on Algorithmic Recourse*, 2021.
- [274] B. Shneiderman, “Bridging the Gap Between Ethics and Practice: Guidelines for Reliable, Safe, and Trustworthy Human-centered AI Systems,” *ACM Transactions on Interactive Intelligent Systems*, vol. 10, 2020.
- [275] O. Gomez, S. Holter, J. Yuan, and E. Bertini, “ViCE: Visual counterfactual explanations for machine learning models,” in *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 2020.
- [276] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viegas, and J. Wilson, “The What-If Tool: Interactive Probing of Machine Learning Models,” *TVCG*, vol. 26, 2019.
- [277] J. A. Nelder and R. W. M. Wedderburn, “Generalized Linear Models,” *Journal of the Royal Statistical Society. Series A (General)*, vol. 135, 1972.
- [278] D. S. Weld and G. Bansal, “The challenge of crafting intelligible intelligence,” *Communications of the ACM*, vol. 62, 2019.
- [279] A.-H. Karimi, G. Barthe, B. Schölkopf, and I. Valera, “A survey of algorithmic recourse: Definitions, formulations, solutions, and prospects,” *arXiv:2010.04050 [cs, stat]*, 2021.
- [280] M. T. Keane, E. M. Kenny, E. Delaney, and B. Smyth, “If Only We Had Better Counterfactual Explanations: Five Key Deficits to Rectify in the Evaluation of Counterfactual XAI Techniques,” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.
- [281] B. Mittelstadt, C. Russell, and S. Wachter, “Explaining Explanations in AI,” in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019.
- [282] 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 *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 2018.
- [283] S. Nourashrafeddin, E. Sherkat, R. Minghim, and E. E. Miliotis, “A Visual Approach for Interactive Keyterm-Based Clustering,” *ACM Transactions on Interactive Intelligent Systems*, vol. 8, 2018.
- [284] H.-F. Cheng *et al.*, “Explaining Decision-Making Algorithms through UI: Strategies to Help Non-Expert Stakeholders,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.
- [285] H. Suresh, S. R. Gomez, K. K. Nam, and A. Satyanarayan, “Beyond Expertise and Roles: A Framework to Characterize the Stakeholders of Interpretable Machine Learning and their Needs,” *arXiv:2101.09824 [cs]*, 2021.
- [286] G. Ke *et al.*, “LightGBM: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems 30 (NIP 2017)*, 2017.
- [287] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
- [288] K. Mohammadi, A.-H. Karimi, G. Barthe, and I. Valera, “Scaling Guarantees for Nearest Counterfactual Explanations,” in *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 2021.
- [289] M. Schleich, Z. Geng, Y. Zhang, and D. Suciu, “GeCo: Quality Counterfactual Explanations in Real Time,” *arXiv:2101.01292 [cs]*, 2021.
- [290] A. Van Looveren and J. Klaise, “Interpretable Counterfactual Explanations Guided by Prototypes,” *arXiv:1907.02584 [cs, stat]*, 2020.

- [291] M. Redmond and A. Baveja, "A data-driven software tool for enabling cooperative information sharing among police departments," *European Journal of Operational Research*, vol. 141, 2002.
- [292] I.-C. Yeh and C.-h. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Systems with Applications*, vol. 36, 2009.
- [293] D. Dua and C. Graff, *UCI machine learning repository*, 2017.
- [294] R. Kohavi *et al.*, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid.," in *KDD*, vol. 96, 1996.
- [295] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE Symposium on Visual Languages*, 1996.
- [296] D. A. Norman and S. W. Draper, *User Centered System Design: New Perspectives on Human-Computer Interaction*. 1986.
- [297] S. Garfinkel, *PGP: Pretty Good Privacy*. 1995.
- [298] Vera Institute of Justice, *Performance incentive funding: Aligning fiscal and operational responsibility to produce more safety at less cost*, Vera Institute of Justice Report, 2012.
- [299] D. Slack, A. Hilgard, H. Lakkaraju, and S. Singh, "Counterfactual explanations can be manipulated," in *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [300] J. Vaillant, *Glpk.js*, 2021.
- [301] T. Hase, *OpenPGP.js: OpenPGP JavaScript Implementation*, 2014.
- [302] S. Tsirtsis and M. Gomez Rodriguez, "Decisions, counterfactual explanations and strategic behavior," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [303] A.-H. Karimi, G. Barthe, B. Balle, and I. Valera, "Model-Agnostic Counterfactual Explanations for Consequential Decisions," *arXiv:1905.11190 [cs, stat]*, 2020.
- [304] K. Rawal and H. Lakkaraju, "Beyond Individualized Recourse: Interpretable and Interactive Summaries of Actionable Recourses," *arXiv:2009.07165 [cs, stat]*, 2020.
- [305] J. S. Olson and W. Kellogg, *Ways of Knowing in HCI*. 2014.
- [306] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with Mechanical Turk," in *Proceeding of the Twenty-Sixth Annual CHI Conference on Human Factors in Computing Systems - CHI '08*, 2008.
- [307] G. Paolacci and J. Chandler, "Inside the Turk: Understanding Mechanical Turk as a Participant Pool," *Current Directions in Psychological Science*, vol. 23, 2014.
- [308] C.-J. Ho, A. Slivkins, S. Suri, and J. W. Vaughan, "Incentivizing high quality crowdwork," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15, 2015.
- [309] P. Hitlin, "Research in the crowdsourcing age: A case study," 2016.
- [310] S. Dumais, R. Jeffries, D. M. Russell, D. Tang, and J. Teevan, "Understanding User Behavior Through Log Data and Analysis," in *Ways of Knowing in HCI*, 2014.
- [311] J. Kleinberg and M. Raghavan, "How Do Classifiers Induce Agents to Invest Effort Strategically?" *ACM Transactions on Economics and Computation*, vol. 8, 2020.
- [312] M. Hardt, N. Megiddo, C. Papadimitriou, and M. Wootters, "Strategic Classification," in *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, 2016.
- [313] C. Rudin, "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead," *Nature Machine Intelligence*, vol. 1, 2019.
- [314] U. Ehsan, B. Harrison, L. Chan, and M. O. Riedl, "Rationalization: A Neural Machine Translation Approach to Generating Natural Language Explanations," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018.
- [315] F. Hohman, A. Srinivasan, and S. M. Drucker, "TeleGam: Combining Visualization and Verbalization for Interpretable Machine Learning," in *2019 IEEE Visualization Conference (VIS)*, 2019.

- [316] B. Rakova, J. Yang, H. Cramer, and R. Chowdhury, “Where Responsible AI meets Reality: Practitioner Perspectives on Enablers for Shifting Organizational Practices,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, 2021.
- [317] A. Chouldechova, “Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments,” *Big Data*, vol. 5, 2017.
- [318] H. Weerts, M. Dudík, R. Edgar, A. Jalali, R. Lutz, and M. Madaio, “Fairlearn: Assessing and Improving Fairness of AI Systems,” *arXiv 2303.16626*, 2023.
- [319] J. Kleinberg, S. Mullainathan, and M. Raghavan, “Inherent Trade-Offs in the Fair Determination of Risk Scores,” *arXiv 1609.05807*, 2016.
- [320] E. Beretta, A. Vetrò, B. Lepri, and J. C. D. Martin, “Detecting discriminatory risk through data annotation based on Bayesian inferences,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.
- [321] M. Miceli, M. Schuessler, and T. Yang, “Between Subjectivity and Imposition: Power Dynamics in Data Annotation for Computer Vision,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, 2020.
- [322] A. Mostafazadeh Davani, M. Díaz, and V. Prabhakaran, “Dealing with disagreements: Looking beyond the majority vote in subjective annotations,” *Transactions of the Association for Computational Linguistics*, vol. 10, 2022.
- [323] M. Mitchell *et al.*, “Model Cards for Model Reporting,” in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019.
- [324] T. Gebru *et al.*, “Datasheets for Datasets,” *arXiv:1803.09010 [cs]*, 2020.
- [325] M. Díaz *et al.*, “CrowdWorkSheets: Accounting for Individual and Collective Identities Underlying Crowdsourced Dataset Annotation,” in *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.
- [326] Microsoft, *Harms modeling - Azure Application Architecture Guide*, 2022.
- [327] Google, *Google Ai Studio: Prototype with Generative AI*, 2023.
- [328] T. Kluyver *et al.*, “Jupyter Notebooks—a publishing format for reproducible computational workflows.,” vol. 2016, 2016.
- [329] Z. J. Wang, K. Dai, and W. K. Edwards, “StickyLand: Breaking the Linear Presentation of Computational Notebooks,” in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 2022.
- [330] G. Team *et al.*, “Gemini: A family of highly capable multimodal models,” *arXiv preprint arXiv:2312.11805*, 2023.
- [331] OpenAI, “GPT-4 Technical Report,” *arXiv 2303.08774*, 2023.
- [332] OpenAI, *OpenAI Playground*, 2023.
- [333] T. Wu, M. Terry, and C. J. Cai, “AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts,” in *CHI Conference on Human Factors in Computing Systems*, 2022.
- [334] Z. J. Wang, A. Chakravarthy, D. Munechika, and D. H. Chau, “Workflow: Social Prompt Engineering for Large Language Models,” *arXiv 2401.14447*, 2024.
- [335] Q. V. Liao and J. W. Vaughan, “AI Transparency in the Age of LLMs: A Human-Centered Research Roadmap,” *arXiv 2306.01941*, 2023.
- [336] A. J. Fiannaca, C. Kulkarni, C. J. Cai, and M. Terry, “Programming without a Programming Language: Challenges and Opportunities for Designing Developer Tools for Prompt Programming,” in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.

- [337] E. Jiang *et al.*, “PromptMaker: Prompt-based Prototyping with Large Language Models,” in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 2022.
- [338] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, and Q. Yang, “Why Johnny Can’t Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.
- [339] B. Weiser and N. Schweber, “The ChatGPT Lawyer Explains Himself,” *The New York Times*, 2023.
- [340] K. Holstein, J. Wortman Vaughan, H. Daumé, M. Dudik, and H. Wallach, “Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?” In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.
- [341] R. Y. Wong, M. A. Madaio, and N. Merrill, “Seeing Like a Toolkit: How Toolkits Envision the Work of AI Ethics,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 7, 2023.
- [342] A. Srivastava *et al.*, “Beyond the imitation game: Quantifying and extrapolating the capabilities of language models,” *arXiv preprint arXiv:2206.04615*, 2022.
- [343] D. Ganguli *et al.*, “Predictability and Surprise in Large Generative Models,” in *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.
- [344] C. O’Neil and H. Gunn, “Near-Term Artificial Intelligence and the Ethical Matrix,” in *Ethics of Artificial Intelligence*. 2020.
- [345] H. Suresh and J. Gutttag, “A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle,” in *Equity and Access in Algorithms, Mechanisms, and Optimization*, 2021.
- [346] E. Tabassi, “AI Risk Management Framework: AI RMF (1.0),” National Institute of Standards and Technology, Tech. Rep., 2023.
- [347] M. A. Madaio, L. Stark, J. Wortman Vaughan, and H. Wallach, “Co-Designing Checklists to Understand Organizational Challenges and Opportunities around Fairness in AI,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.
- [348] S. McGregor, “Preventing Repeated Real World AI Failures by Cataloging Incidents: The AI Incident Database,” *arXiv 2011.08512*, 2020.
- [349] E. Reingold and J. Tilford, “Tidier Drawings of Trees,” *IEEE Transactions on Software Engineering*, vol. SE-7, 1981.
- [350] MDN, *Web Components - Web APIs*, 2021.
- [351] MDN, *WebGL: 2D and 3D graphics for the web - Web APIs*, 2011.
- [352] J. P. Sarmiento and A. F. Wise, “Participatory and Co-Design of Learning Analytics: An Initial Review of the Literature,” in *LAK22: 12th International Learning Analytics and Knowledge Conference*, 2022.
- [353] J. J. Smith, S. Amershi, S. Barocas, H. Wallach, and J. Wortman Vaughan, “REAL ML: Recognizing, Exploring, and Articulating Limitations of Machine Learning Research,” in *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.
- [354] M. Madaio, L. Egede, H. Subramonyam, J. Wortman Vaughan, and H. Wallach, “Assessing the Fairness of AI Systems: AI Practitioners’ Processes, Challenges, and Needs for Support,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, 2022.
- [355] Microsoft, “Microsoft Responsible AI Impact Assessment Guide,” 2022.
- [356] S. Barocas *et al.*, “Designing disaggregated evaluations of AI systems: Choices, considerations, and tradeoffs,” in *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES ’21, 2021.
- [357] K. R. Koedinger, J. Kim, J. Z. Jia, E. A. McLaughlin, and N. L. Bier, “Learning is not a spectator sport: Doing is better than watching for learning from a MOOC,” in *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, 2015.

- [358] A. R. Chow and B. Perrigo, *The AI arms race is changing everything*, 2023.
- [359] H. Touvron *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv 2307.09288*, 2023.
- [360] C. Rastogi, M. Tulio Ribeiro, N. King, H. Nori, and S. Amershi, “Supporting Human-AI Collaboration in Auditing LLMs with LLMs,” in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, 2023.
- [361] F. K. Akin, *Awesome ChatGPT Prompts*, 2022.
- [362] R. Shelby *et al.*, “Sociotechnical Harms of Algorithmic Systems: Scoping a Taxonomy for Harm Reduction,” in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, 2023.
- [363] Google, *Lit: Simple fast Web Components*, 2015.
- [364] H. J. Seltman, *Experimental Design and Analysis*. 2012.
- [365] T. W. Price, J. J. Williams, J. Solyst, and S. Marwan, “Engaging Students with Instructor Solutions in Online Programming Homework,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.
- [366] I. D. Raji *et al.*, “Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020.
- [367] M. L. McHugh, “Interrater reliability: The kappa statistic,” *Biochemia Medica*, 2012.
- [368] J. Cohen, “Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit,” *Psychological Bulletin*, vol. 70, 1968.
- [369] J. R. Landis and G. G. Koch, “The Measurement of Observer Agreement for Categorical Data,” *Biometrics*, vol. 33, 1977.
- [370] S. S. Shapiro and M. B. Wilk, “An analysis of variance test for normality (complete samples),” *Biometrika*, vol. 52, 1965.
- [371] H. B. Mann and D. R. Whitney, “On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other,” *The Annals of Mathematical Statistics*, vol. 18, 1947.
- [372] S. B. Merriam *et al.*, “Introduction to qualitative research,” *Qualitative research in practice: Examples for discussion and analysis*, vol. 1, 2002.
- [373] V. Braun and V. Clarke, “Using thematic analysis in psychology,” *Qualitative Research in Psychology*, vol. 3, 2006.
- [374] Dovetail, *Dovetail: All your customer insights in one place*, 2023.
- [375] O. J. Dunn, “Multiple Comparisons among Means,” *Journal of the American Statistical Association*, vol. 56, 1961.
- [376] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. 2013.
- [377] S. S. Sawilowsky, “New Effect Size Rules of Thumb,” *Journal of Modern Applied Statistical Methods*, vol. 8, 2009.
- [378] S. Rismani *et al.*, “From Plane Crashes to Algorithmic Harm: Applicability of Safety Engineering Frameworks for Responsible ML,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.
- [379] K. O. McGraw and S. P. Wong, “A common language effect size statistic,” *Psychological Bulletin*, vol. 111, 1992.
- [380] Anthropic, *Core Views on AI Safety: When, Why, What, and How*, 2023.
- [381] Meta, *Llama 2: Responsible Use Guide*, 2023.
- [382] Google, *PaLM API: Safety guidance*, 2023.
- [383] E. Denton, M. Díaz, I. Kivlichan, V. Prabhakaran, and R. Rosen, “Whose Ground Truth? Accounting for Individual and Collective Identities Underlying Dataset Annotation,” *arXiv 2112.04554*, 2021.

- [384] A. M. Davani, M. Diaz, D. Baker, and V. Prabhakaran, “Disentangling disagreements on offensiveness: A cross-cultural study,” in *The 61st Annual Meeting of the Association for Computational Linguistics*, 2023.
- [385] V. Prabhakaran *et al.*, “A Framework to Assess (Dis)agreement Among Diverse Rater Groups,” *arXiv 2311.05074*, 2023.
- [386] E. Pavlick and T. Kwiatkowski, “Inherent Disagreements in Human Textual Inferences,” *Transactions of the Association for Computational Linguistics*, vol. 7, 2019.
- [387] U. Ehsan, Q. V. Liao, S. Passi, M. O. Riedl, and H. Daume III, “Seamful XAI: Operationalizing seamful design in explainable AI,” *arXiv preprint arXiv:2211.06753*, 2022.
- [388] H. Kaur, E. Adar, E. Gilbert, and C. Lampe, “Sensible AI: Re-imagining Interpretability and Explainability using Sensemaking Theory,” in *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.
- [389] M. Whittaker, “The steep cost of capture,” *Interactions*, vol. 28, 2021.
- [390] O. of QueerInAI *et al.*, “Bound by the Bounty: Collaboratively Shaping Evaluation Processes for Queer AI Harms,” *arXiv 2307.10223*, 2023.
- [391] M. L. Gordon *et al.*, “Jury Learning: Integrating Dissenting Voices into Machine Learning Models,” in *CHI Conference on Human Factors in Computing Systems*, 2022.
- [392] F. Delgado, S. Yang, M. Madaio, and Q. Yang, “The Participatory Turn in AI Design: Theoretical Foundations and the Current State of Practice,” in *Equity and Access in Algorithms, Mechanisms, and Optimization*, 2023.
- [393] A. Birhane *et al.*, “Power to the People? Opportunities and Challenges for Participatory AI,” in *Equity and Access in Algorithms, Mechanisms, and Optimization*, 2022.
- [394] M. Hravnak *et al.*, “A call to alarms: Current state and future directions in the battle against alarm fatigue,” *Journal of Electrocardiology*, vol. 51, 2018.
- [395] K. Renaud and M. Dupuis, “Cyber security fear appeals: Unexpectedly complicated,” in *Proceedings of the New Security Paradigms Workshop*, 2019.
- [396] T. Kluyver *et al.*, “Jupyter Notebooks - a Publishing Format for Reproducible Computational Workflows,” *ELPUB*, 2016.
- [397] Kaggle, *State of Machine Learning and Data Science 2022*, 2022.
- [398] A. Rule, A. Tabard, and J. D. Hollan, “Exploration and Explanation in Computational Notebooks,” in *CHI*, 2018.
- [399] J. P. Ono, S. Castelo, R. Lopez, E. Bertini, J. Freire, and C. Silva, “PipelineProfiler: A Visual Analytics Tool for the Exploration of AutoML Pipelines,” *TVCG*, vol. 27, 2021.
- [400] P. Xenopoulos, J. Rulff, L. G. Nonato, B. Barr, and C. Silva, “Calibrate: Interactive Analysis of Probabilistic Model Output,” *TVCG*, vol. 29, 2023.
- [401] Z. J. Wang *et al.*, “TimberTrek: Exploring and Curating Sparse Decision Trees with Interactive Visualization,” in *VIS*, 2022.
- [402] M. Dudík, S. Bird, H. Wallach, and K. Walker, “Fairlearn: A toolkit for assessing and improving fairness in AI,” 2020.
- [403] Z. J. Wang, C. Kulkarni, L. Wilcox, M. Terry, and M. Madaio, “Farsight: Fostering Responsible AI Awareness During AI Application Prototyping,” in *CHI Conference on Human Factors in Computing Systems*, 2024.
- [404] A. Bhat *et al.*, “Aspirations and Practice of Model Documentation: Moving the Needle with Nudging and Traceability,” in *CHI*, 2023.
- [405] A. X. Zhang, M. Muller, and D. Wang, “How do Data Science Workers Collaborate? Roles, Workflows, and Tools,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, 2020.

- [406] W. H. Deng *et al.*, “Exploring How Machine Learning Practitioners (Try To) Use Fairness Toolkits,” in *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.
- [407] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, “Vega-lite: A grammar of interactive graphics,” *IEEE Transactions on Visualization & Computer Graphics (Proc. InfoVis)*, 2017.
- [408] S. Lau, I. Drosos, J. M. Markel, and P. J. Guo, “The Design Space of Computational Notebooks: An Analysis of 60 Systems in Academia and Industry,” in *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2020.
- [409] Z. J. Wang, K. Dai, and W. K. Edwards, “StickyLand: Breaking the Linear Presentation of Computational Notebooks,” *CHI EA*, 2022.
- [410] T. Kojima, S. (Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, “Large Language Models are Zero-Shot Reasoners,” *Advances in Neural Information Processing Systems*, vol. 35, 2022.
- [411] J. Wei *et al.*, “Finetuned Language Models Are Zero-Shot Learners,” *arXiv 2109.01652*, 2022.
- [412] H. Nori *et al.*, “Can Generalist Foundation Models Outcompete Special-Purpose Tuning? Case Study in Medicine,” *arXiv 2311.16452*, 2023.
- [413] T. Brown *et al.*, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [414] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” *arXiv 2005.11401*, 2021.
- [415] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing,” *ACM Computing Surveys*, vol. 55, 2023.
- [416] A. G. Parameswaran, S. Shankar, P. Asawa, N. Jain, and Y. Wang, “Revisiting Prompt Engineering via Declarative Crowdsourcing,” *arXiv 2308.03854*, 2023.
- [417] D. Harwell, *Tech’s hottest new job: AI whisperer. No coding required.* 2023.
- [418] Z. Zhou *et al.*, “InstructPipe: Building Visual Programming Pipelines with Human Instructions,” *arXiv 2312.09672*, 2023.
- [419] Promptstacks, *Promptstacks: Your Prompt Engineering Community*, 2023.
- [420] Reddit, *R/ChatGPTPromptGenius*, 2023.
- [421] D. Eccleston and S. Tey, *ShareGPT: Share your wildest ChatGPT conversations with one click.* 2022.
- [422] Z. J. Wang, E. Montoya, D. Munehika, H. Yang, B. Hoover, and D. H. Chau, “DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.
- [423] PromptBase, *PromptBase: Prompt Marketplace: Midjourney, ChatGPT, DALL-E, Stable Diffusion & more.* 2023.
- [424] PromptHero, *PromptHero: Search prompts for Stable Diffusion, ChatGPT & Midjourney*, 2023.
- [425] ChatX, *ChatX: ChatGPT, DALL-E & Stable Diffusion prompt marketplace*, 2023.
- [426] S. Sharma, N. Slack, K. Devi, T. Greig, and S. Naidu, “Exploring gamers’ crowdsourcing engagement in Pokémon Go communities,” *The TQM Journal*, 2021.
- [427] E. Loria, A. Antelmi, and J. Pirker, “Comparing the Structures and Characteristics of Different Game Social Networks - The Steam Case,” in *2021 IEEE Conference on Games (CoG)*, 2021.
- [428] T. Bakici, “Comparison of crowdsourcing platforms from social-psychological and motivational perspectives,” *International Journal of Information Management*, vol. 54, 2020.
- [429] W. Wu and X. Gong, “Motivation and sustained participation in the online crowdsourcing community: The moderating role of community commitment,” *Internet Research*, vol. 31, 2020.
- [430] D. H.-L. Goh, E. P. P. Pe-Tham, and C. S. Lee, “Perceptions of virtual reward systems in crowdsourcing games,” *Computers in Human Behavior*, vol. 70, 2017.

- [431] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th International Conference on World Wide Web*, 2001.
- [432] I. Tenney *et al.*, “The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models,” in *EMNLP Demo*, 2020.
- [433] L. Schubert, M. Petrov, S. Carter, N. Cammarata, G. Goh, and C. Olah, *OpenAI Microscope*, 2020.
- [434] Z. J. Wang and D. H. Chau, “WebSHAP: Towards Explaining Any Machine Learning Models Anywhere,” in *Companion Proceedings of the Web Conference 2023*, 2023.
- [435] J. Bai, F. Lu, and K. Zhang, *ONNX: Open neural network exchange*, 2019.
- [436] T. MLC, *MLC-LLM*, 2023.
- [437] T. Chen *et al.*, “TVM: An automated End-to-End optimizing compiler for deep learning,” in *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, 2018.
- [438] Apple, *Core ML: Integrate machine learning models into your app*, 2017.
- [439] M. Kamvar, M. Kellar, R. Patel, and Y. Xu, “Computers and iphones and mobile phones, oh my!: A logs-based comparison of search users on different devices,” in *Proceedings of the 18th International Conference on World Wide Web*, 2009.
- [440] M. Lam *et al.*, “GPU-based Private Information Retrieval for On-Device Machine Learning Inference,” *arXiv 2301.10904*, 2023.
- [441] Y. Gong *et al.*, “EdgeRec: Recommender System on Edge in Mobile Taobao,” in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020.
- [442] X. Xia, J. Yu, Q. Wang, C. Yang, N. Q. V. Hung, and H. Yin, “Efficient On-Device Session-Based Recommendation,” *ACM Transactions on Information Systems*, 2023.
- [443] Z. J. Wang, J. W. Vaughan, R. Caruana, and D. H. Chau, “GAM Coach: Towards Interactive and User-centered Algorithmic Recourse,” in *CHI Conference on Human Factors in Computing Systems*, 2023.
- [444] J. Macoskey, G. Strimel, J. Su, and A. Rastrow, “Amortized neural networks for low-latency speech recognition,” in *Interspeech 2021*, 2021.
- [445] J. Macoskey, G. Strimel, and A. Rastrow, “Learning a neural diff for speech models,” in *Interspeech 2021*, 2021.
- [446] Z. Tan, Z. Yang, M. Zhang, Q. Liu, M. Sun, and Y. Liu, “Dynamic Multi-Branch Layers for On-Device Neural Machine Translation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, 2022.
- [447] Z. J. Wang and D. H. Chau, “MeMemo: On-device Retrieval Augmentation for Private and Personalized Text Generation,” in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024.
- [448] T. Forte, *Building a Second Brain: A Proven Method to Organize Your Digital Life and Unlock Your Creative Potential*, First Atria Books hardcover edition. 2022.