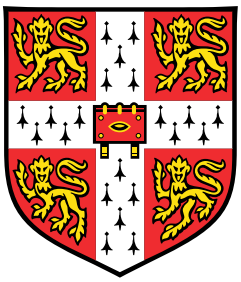
Trustworthy Machine Learning

From Algorithmic Transparency to Decision Support



Umang Sanjiv Bhatt

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

St. Edmund's College

September 2023

To my Dada, Nareshchandra Maneklal Bhatt. I hope you are proud of what our *Math Thinking Process* has become.

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Umang Sanjiv Bhatt
September 2023

Trustworthy Machine Learning
From Algorithmic Transparency to Decision Support

Umang Sanjiv Bhatt

Developing machine learning models worthy of decision-maker trust is crucial to using models in practice. Algorithmic transparency tools, such as explainability and uncertainty estimates, demonstrate the trustworthiness of a model to a decision-maker. In this thesis, we first explore how practitioners use explainability in industry. Through an interview study, we find that, while engineers increasingly use explainability methods to test model behavior during development, there is limited adoption of these methods for the benefit of external stakeholders. To that end, we develop novel algorithmic transparency methods for specific decision-making contexts and test these methods with real decision-makers via human-subject experiments. We first propose DIVINE, an example-based explanation method, which finds training points that are not only influential to the model's parameters but also diversely located in input space. We show how our explanations can improve a decision-maker's ability to simulate a model's decision boundary. We next discuss Counterfactual Latent Uncertainty Explanations (CLUE), a feature importance explanation method that identifies which input features, if perturbed, would reduce the model's uncertainty on a given input. We demonstrate how decision-makers can use our explanations to identify a model's uncertainty on unseen inputs. While each method is successful in its own right, we are interested in understanding, more generally, the settings under which outcomes improve after a decision-maker leverages a form of decision support, be it algorithmic transparency or model predictions. We propose the problem of learning a *decision support policy* that, for a given input, chooses which form of support to provide to decision-makers for whom we initially have no prior information. Using techniques from stochastic contextual bandits, we introduce `THREAD`, an online algorithm to personalize a decision support policy for each decision-maker. We deploy `THREAD` with real users to show how personalized policies can be learned online, and illustrate nuances of learning decision support policies in practice. We conclude this thesis with the promise of personalizing access to decision support, which could include forms of algorithmic transparency, based on decision-maker needs.

Acknowledgements

First and foremost, I want to thank Swaminarayan Bhagwan, God, for granting me the opportunity to a PhD. My gurus, Pramukh Swami Maharaj and Mahant Swami Maharaj, have supported every step of my spiritual and academic journey: for that, I am eternally indebted. Their āgna (wishes) continue to make my decision-making all the more straightforward. *"Prārabdham me tad ichchhaiva"* — Satsang Diksha 45.

I never thought I would do a PhD. At the end of my first year of undergrad at Carnegie Mellon University, I was smitten with the entrepreneurship world, yet my faculty advisor David O'Halloran, with no prompting, handed me a printed copy of Mor Horchol-Balter's infamous guide to *Applying to PhD programs in CS*. He then muttered "go to get a PhD." I had never considered graduate school. I had neither the grades nor the focus at that time. I did not know what he was thinking, so I politely dismissed his idea and trucked on. One year later just as I was dipping my toe into research with Zico Kolter, I stumbled upon Dave's printout in a pile of papers and finally read it. Thanks to a gamut of projects with Fei Fang, Manuela Veloso, Radu Marculescu, and Pradeep Ravikumar during my time in Pittsburgh, I fell in love with the autonomy that research affords and the grit required to validate your ideas (just as with startups). I spent a year and a half working with José Moura, who patiently listened to me muse on explainability and taught me to zoom out to the bigger picture. Carnegie Mellon was the perfect place to immerse myself in research. To CMU, José, Zico, Pradeep, and Dave, thank you for inspiring me to pursue doctoral studies.

Deciding where to go to graduate school felt daunting. Vacillated between satisficing versus maximizing, I met Adrian Weller for a sunset walk along a beach on Oahu in January 2019. Quite the sight for an interview. Over dinner thereafter, we spoke about everything: the challenges of algorithmic transparency, our relationship with faith, our investing philosophies, the role of diversity in higher education, the variance in technical definitions of fairness, and human-machine teams. Later that night I distinctly remember the owner of the tiny sushi shop ushering Adrian and me out, since we accidentally overstayed our welcome, well beyond closing. Truly the start of a human-human team. The next four years were a whirlwind. Adrian is a remarkable and inspiring PhD supervisor. He has taught me so much, including how to think clearly when defining research questions and how to articulate my results succinctly.

Adrian encouraged my interdisciplinary approach to my research and career: he helped me design new human subject experiments, brainstorm algorithms for HMTs, collaborate with policymakers and regulators, get involved with industry applications of our work, and mentor newer students in our group. From gorilla trekking in the Ugandan jungle to enjoying jazz in NYC (after publishing our first paper together) to discussing the definition of an explanation with philosophers to tubing around London for meetings to aligning time-zones for calls during my travels to applying for ethics approval from Cambridge Engineering to having weekly dinners in Darwin to walking on the Heath to everything in between, thank you for a lifetime of memories, Adrian.

I am grateful to the Leverhulme Centre for the Future of Intelligence, which facilitated much of the interdisciplinary work in this thesis, and to Stephen Cave for encouraging me to forge new collaborations. I appreciate CFI for funding my PhD via generous donations from Google DeepMind and the Leverhulme Trust. I thank Peter Eckersley, Alice Xiang, and Terah Lyons for hosting me as one of the first Research Fellows at the Partnership on AI. I am thankful to Mark Surman, Becca Ricks, and Navrina Singh for inviting me to join the Mozilla Foundation as a Fellow. Many thanks to Milind Tambe and Himabindu Lakkaraju for facilitating my time as a Summer Fellow at the Harvard Center for Research on Computation and Society. I thank Ameet Talwalkar for hosting me (back) at CMU to work closely with Valerie Chen, who has become a delightful intellectual sparring partner and research confidant. Manoj Saxena, Ashley Casovan, and Var Shankar were instrumental in my joining the Responsible AI Institute. During my PhD, I had the good fortune of mentoring excellent students and researchers, including Varun Babbar, Matthew Barker, Katherine Collins, Javier Abad Martinez, Daniel Ley, and Vivek Palaniappan. I single out Katie for being an incredibly energetic, patient, and supportive collaborator. I am thankful for my examiners, Neil Lawrence and Eric Horvitz, for taking the time to read this entire thesis and for conducting what was a detailed and exciting viva.

With regard to the papers that will appear in this thesis, I would like to thank my co-authors: Tameem Adel, McKane Andrus, Javier Antorán, Valerie Chen, Isabel Chien, Rumi Chunara, Katherine Collins, Peter Eckersley, Riccardo Fogliato, Joydeep Ghosh, Hoda Heidari, José Miguel Hernández-Lobato, Yunhan Jia, Emma Kallina, Parameswaran Kamalaruban, Ranganath Krishnan, Q. Vera Liao, Gabrielle Melançon, José M.F. Moura, Lama Nachman, Ruchir Puri, Prasanna Sattigeri, Zohreh Shams, Shubham Sharma, Madhulika Srikumar, Jason Stanley, Ameet Talwalkar, Ankur Taly, Omesh Tickoo, Adrian Weller, Alice Xiang, Muhammad Bilal Zafar, John Zerilli, and Yunfeng Zhang.

Cambridge is a wonderful place to pursue graduate studies. The Cambridge Machine Learning Group is full of many talented folks, who have shaped my research taste. Adrian's group provided countless discussions and insightful lunches over the years, thanks to Matthew

Ashman, Yanzhi Chen, Katie Collins, Juyeon Heo, Julius von Kugelgen, Valerii Likhosherstov, Weiyang Liu, Issac Reid, Holli Sargeant, Pingfan Song, Vihari Pitarla, John Zerilli, and Miri Zilka. I would be amiss not to give a shout out to the CBL 2019 Cohort: James Allingham, Javier Antorán, Kris Jensen, Marine Schimel, and Austin Tripp. I thank Rachel Fogg for sorting all of my administrative qualms over the years and Richard Turner for advising me to double down on my interdisciplinary work. At the Alan Turing Institute, I appreciated the support of Ruth Drysdale, Parameswaran Kamalaruban, Chiara Sbruazzo, and Matthew Wicker. The leadership at St Edmunds College—namely Catherine Arnold, Kate Glennie, and Vittorio Montemaggi—were of great help in teaching me the art of philanthropy.

Cambridge, at large, and St Edmund's gave me many memories to cherish: for that, I thank Matthew Ashman, Hamza Chaudhry, Ella Fennelly, Andrew Furlan, Regina Guzman, Hope McGovern, Aneesh Pappu, Sofie Pultz, Tushar Shah, Danielle Sklarew, Zoe Vanhegan, Ralph Wade, and Ahmed Zaidi. I also thank my dear friends in Brian Bakerman, Brian Davis, Olivia Koshy, David Murray, Akshar Patel, Hersh Patel, and Ayushi Sinha for always picking up, even with an ocean between us. I call out Hersh, as someone who was exceptionally supportive, as much of this thesis was written during our travels.

Finally, I would like to thank my family for their unending love and support. My father, Sanjiv, never ceases to be my role model and personal strategist. My mother, Jalaj, reminds me to keep my feet on the ground and pushes me to be the best version of myself. My sister, Naiti, is my number one fan and my first (and best) proofreader. My Nana, Kanakdeo, always stresses that value-driven prioritization and focus will guarantee I fly high. A sunset walk on La Jolla Shores Beach with VJ in the summer of 2021 sparked the idea for Chapter 5. The laughs of M6, inspiration of the Bhatt Family, and support of the Trivedi Family have not gone unnoticed. However, my grandparents are the ones to blame for my obsession with autonomy and decision-making.

Table of Contents

List of Figures xvii

List of Tables xxv

1 Introduction 1

2 Background 5

2.1 On Trust . . . 6

2.2 On Trustworthiness Mechanisms . . . 8

2.2.1 Procedural Transparency . . . 11

2.2.2 Algorithmic Transparency . . . 12

2.3 On Types of Algorithmic Transparency . . . 12

2.3.1 Explainability . . . 12

2.3.2 Uncertainty . . . 22

2.3.3 Case for Opacity . . . 34

2.4 On Human-Machine Teams . . . 35

2.4.1 Incorporating Expert Feedback into Model Updates . . . 36

2.4.2 User Control and Dynamic Allocation . . . 40

3 Algorithmic Transparency in Practice 45

3.1 Overview . . . 45

3.2 Interview Study Methodology . . . 46

3.3 Summary of Findings . . . 48

3.3.1 The Need for Explainability . . . 48

3.3.2 Trends in Using Explainability . . . 49

3.4 Deep Dive: Deploying Local Explainability . . . 52

3.5 Multistakeholder Convening . . . 57

3.6 Convening Takeaways . . . 59

3.6.1 Designing for Explainability . . . 59

3.6.2 Deploying Explainability 62
3.7 Summary 65

4 Algorithmic Transparency in Decision-Making 67
4.1 DIVINE: *DIV*erse *IN*flu*E*ntial Training Points 68
4.1.1 Related Work: Assigning Training Point Importance 69
4.1.2 Method: Selecting Diverse Samples 71
4.1.3 Computational Evaluation 73
4.1.4 User Study: Task Simulatability 78
4.1.5 Limitations 82
4.2 CLUE: *C*ounterfactual *L*atent *U*ncertainty *E*xplanation 83
4.2.1 Related Work: Calculating Uncertainty 84
4.2.2 Method: Generating CLUEs 87
4.2.3 Computational Evaluation 89
4.2.4 User Study: Forward Simulation 92
4.2.5 Limitations 95
4.3 Summary 95

5 Personalizing Access to Decision Support 97
5.1 Introduction 98
5.2 Related Work 100
5.3 Formulating and Learning Decision Support Policies 101
5.3.1 `THREAD`: Learning Personalized Policies 103
5.3.2 Extension to Cost-Aware Decision Support 104
5.3.3 Expertise Profiles 106
5.4 Experimental Setup 107
5.4.1 Decision-making Tasks 107
5.4.2 Human-informed Synthetic Decision-makers 108
5.4.3 Baselines and Other Parameters 108
5.5 Computational Experiments 109
5.6 Human Subject Validation 110
5.7 Summary 113

6 Conclusion 115

References 119

Appendix A Publications 159

Appendix B DIVINE Supplementary Material **165**
B.1 Practitioner Guide . . . 165
B.2 Experimental Setup . . . 173
B.3 DIVINE for Fairness . . . 175
B.4 Additional Experiments . . . 178
B.5 Explanatory Figures for User Studies . . . 182

Appendix C CLUE Supplementary Material **185**
C.1 Experimental Details . . . 185
C.2 Additional Examples . . . 188
C.3 Details on User Study . . . 194

Appendix D Learning Personalized Decision Support Policies **199**
D.1 Comparison Against Prior Work . . . 199
D.2 Additional Details on Problem Formulation . . . 202
D.3 Additional Experimental Details . . . 206
D.4 Additional Computational Experiments . . . 210
D.5 Additional User Study Details and Results . . . 218
D.6 Discussion on Reliance . . . 224

List of Figures

2.1 Scale of stakeholder (e.g. decision-maker) attitudes towards machine learning models in human-machine teams 6

2.2 Depiction of various types of transparency regarding a machine learning model 10

2.3 Top: The same prediction (no cancer) is made in two hypothetical cancer diagnosis cases. However, our model is much more confident in the first. This is reflected in the predictive entropy for each case (the dashed red line denotes the maximum entropy for 3-way classification). Bottom: In a regression task, a predictive distribution contains rich information about our model's predictions (modes, tails, etc). We summarize predictive distributions with means and error bars (here standard deviations). 23

2.4 Uncertainty quantification and evaluation. Left three plots: A 15-element deep ensemble provides an increased model (epistemic) uncertainty in data-sparse regions. A homoscedastic Gaussian noise model provides aleatoric uncertainty matching the noise in the data. Both combine to produce a predictive distribution. Right: calibration plot for a classification task. Each bar corresponds to a bin in which predictions are grouped. Their height corresponds to the proportion of correct predictions. 25

2.5 Examples of uncertainty visualizations. 33

2.6 To incorporate an expert's preferences to improve a model, practitioners must turn non-technical, domain expert preferences into usable model updates. In Chen et al. [2023a], we propose a feedback-update taxonomy that focuses on the ways in which expert feedback can be translated into model updates. This taxonomy has two axes, expert feedback, and model updates, forming six categories of feedback-update interaction. We map existing work onto each category and use our taxonomy to motivate improving the interaction between practitioners and domain experts. 36

2.7 Full summary of the Background. We not only consider the forms of transparency provided to stakeholders, focusing on algorithmic transparency via explainability and uncertainty, but also discuss how stakeholder feedback can be used to update models and learn decision support policies. 44

4.1 In 4.1a, we show that our method selects DIVINE points (yellow circles) that are spread across the feature space. This contrasts IF (red diamonds) and Data Shapley (blue diamonds), which select points located in one region. Note the overlap between IF and DIVINE points in the top right. In 4.1b and 4.1c, we show that DIVINE points (third row) are more diverse than ones selected by IF (first row) or other methods (second row). DIVINE is calculated by trading off IF and $\mathcal{R}_{SR}$. The predicted label is listed under each point. 70

4.2 We characterize the trade-off between influence and diversity in 4.2a by varying γ. We also visualize the top-5 DIVINE points for select values of γ. The red diamond in 4.2a is $\gamma = 0$, which recovers the top points from IF alone plotted in 4.2b. The orange diamond in 4.2a is for the γ we find such that our DIVINE points have 10% less influence than IF points; these are visualized in 4.2c. The yellow diamond in 4.2a is the γ we find such that we maximize the average pairwise distance between our DIVINE points, seen in 4.2d. 74

4.3 We find DIVINE points with different diversity functions: $\mathcal{R}_{\text{FL}}$ and $\mathcal{R}_{\text{MMD}}$. In Figures 4.3a and 4.3c, we select γ by sacrificing 10% influence. In Figures 4.3b and 4.3d, we select γ by maximizing the pairwise distance between selected points. For $\mathcal{R}_{\text{FL}}$, some points are representative (near cluster centers), but others are redundant (top right). For $\mathcal{R}_{\text{MMD}}$, the selected points are representative, though the top cluster is missed. There are no redundant points, in contrast to points selected with $\mathcal{R}_{\text{FL}}$. Compared to $\mathcal{R}_{\text{SR}}$, these functions select points near cluster centers. 74

4.4 For five datasets and four values of $m = \{5, 10, 15, 20\}$, we characterize the influence-diversity trade-off. The red diamond indicates where the top IF points lie. The orange diamond is where 10% of the influence has been foregone for diversity. The yellow diamond is where the average pairwise distance between DIVINE points is maximized. 75

4.5 A Toy Example: valuing data using $\mathcal{L}_{\text{unf}}$ and $\mathcal{L}_{\text{loss}}$ yield different influential points. 77

4.6 Task Simulatability. We show the following sets of $m = 10$ points to a user and ask them to draw a decision boundary for each. Users decide upon a decision boundary by selecting two endpoints, which we then translate into a line. . . . 81

4.7 The resulting user-drawn decision boundaries after observing $m = 10$ points. . 81
4.8 Average User Drawn Decision Boundaries ($m = 10$) 82
4.9 Workflow for automated decision making with transparency. Our probabilistic classifier produces a distribution over outputs. In cases of high uncertainty, CLUE allows us to identify features that are responsible for class ambiguity in the input (denoted by Δ and highlighted in dark blue). Otherwise, we resort to existing feature importance approaches to explain certain decisions. 83
4.10 Left: Training points and predictive distribution for variational Bayesian Logistic Regression on the Moons dataset. Center: Aleatoric entropy H_a matches regions of class non-separability. Right: Epistemic entropy H_e grows away from the data. Both uncertainties are detailed in C.1. 86
4.11 Left: Taking a step in the direction of maximum sensitivity leads to a seemingly noisy input configuration for which H is small. Right: Minimizing CLUE's uncertainty-based objective in terms of a DGM's latent variable z produces a plausible digit with a corrected lower portion. 86
4.12 BNN generates uncertainty estimates for inputs decoded from latent codes; its gradients are backpropagated to latent space. 88
4.13 Example image and tabular CLUEs. 89
4.14 MNIST knee-points for various methods. 91
4.15 Left: CLUEs are similarly *informative* under encoder-based and encoder-free initializations. The color bar indicates the original samples' uncertainty. Its horizontal blue line denotes our rejection threshold. Right: Auxiliary DGMs with more capacity result in more *relevant* CLUEs. 92
4.16 Experimental workflow for our tabular data user study. 93
4.17 Example question shown to main study participants for the COMPAS dataset: *Given the uncertain example on the left and the certain example in the middle, will the model be certain on the test example on the right?* The red text highlights the features that differ between *context points*. 93

5.1 We illustrate the process of learning a decision support policy π_t online to improve a decision-maker Alice's performance. For every decision x_t, our policy selects a form of support $\pi_t(x_t)$ from a set $\mathcal{A} = \{a_1, a_2, a_3\}$. The policy is updated using $\ell(\tilde{y}_t, y_t)$, the loss incurred after observing Alice's decision $\tilde{y}_t$. Without assuming access to prior offline data, our formulation learns a personalized policy *online*. This means Alice's learned policy may differ from that of another decision-maker, Bob, if they have different expertise. We also consider a cost-aware version of this learning problem (e.g. it is more expensive to elicit an LLM summary). 99

5.2 Snapshots of the learned decision support policies computed at the end of 100 interactions with randomly sampled users for CIFAR-2*A*. The forms of support are colored in t-SNE embedding space. In the standard setting, both LinUCB and KNN generally identify the optimal form of support for a given input. However, given the task set-up, the choice of support for images of Birds does not matter (Section 5.4.1). When we introduce cost (i.e. when $\lambda \neq 1$), the learned policies favor the human alone for images of Birds, since it is cheaper than the other two forms of support. All participant plots are included in Figure D.6. 113

5.3 Snapshots of the learned decision support policies computed at the end of 60 interactions for MMLU-3*A*. The forms of support are colored in t-SNE embedding space. All six participants achieve $L_h(\pi) = 0.1875$–comparable to that of H-LLM– but exhibit distinct policies across input space. `Modiste` learns policies that capture each decision-maker's strengths and weaknesses. All other participant plots are in Figure D.7. 114

B.1 Parallel coordinates plot for 10 LSAT data points 173

B.2 DIVINE and dataset redundancy 176

B.3 Training data performance after removal of unfairness-inducing points. Line color denotes method. Blue selects DIVINE points (Equation 4.1) with IF, $\mathcal{R}_{\text{SR}}$, and γ *tuned* via pairwise distance. Orange denotes randomly selected points. Green uses IF to select points ($\gamma = 0$). Red chooses points by maximizing $\mathcal{R}_{\text{SR}}$ alone. Grey indicates original performance. Diamond indicates the point all unfairness-inducing points have been removed, after which we expect unfairness to increase, as we remove low importance (but helpful) points. In IF and DIVINE, valuing data with respect to $\mathcal{L}_{\text{unf}}$ identifies harmful data to remove; after removal, fairness outcomes improve greatly, but accuracy drops slightly. 177

B.4 We show $\frac{I_i+I_j}{I_{\{i,j\}}}$ when $\mathcal{L}$ is equal accuracy. 178
B.5 $\mathcal{R}_{\text{SR}}$ – note this figure is the same as Figure 4.2 from the main text. 179
B.6 $\mathcal{R}_{\text{FL}}$. Some points are representative (near cluster center), but others are redundant (top right). 179
B.7 $\mathcal{R}_{\text{MMD}}$. Our selected points are representative, though the top cluster is missed. There are no redundant points, in contrast to points selected with $\mathcal{R}_{\text{FL}}$. . . . 179
B.8 The DIVINE trade-off as a function of m for various $\mathcal{R}$ with our synthetic data. 180
B.9 Results when using DIVINE atop Data Shapley. 180
B.10 Results when using DIVINE atop leave-one-out. 180
B.11 The DIVINE trade-off as a function of m for various $\mathcal{I}$ with our synthetic data. Note CFP is the method described in [Bhatt et al., 2020c]. 181
B.12 For four datasets and $m = 5$, we characterize the influence-diversity trade-off. In B.12d, we show a trade-off curve for FashionMNIST. Notice that it has a similar shape to 4.2a, even though the model type is a CNN, not LR, and the data type is image not tabular. 181
B.13 Setup of DIVINE user studies. 182
B.14 FashionMNIST User Study with DIVINE. 183
B.15 Simulatability task. Points are shown to users on the coordinate grid for $m = 5$. 183
B.16 User-drawn decision boundaries when $m = 5$. Notice that the user-drawn decision boundaries are most similar to the true decision boundary upon seeing DIVINE points. 184

C.1 The leftmost entry is an uncertain COMPAS test sample. To its right are four candidate CLUEs. The first three successfully reduce uncertainty past our rejection threshold, while the rightmost does not. 188
C.2 We generate 5 possible CLUEs for 11 MNIST digits score above the uncertainty rejection threshold. Below each digit or counterfactual is the predictive entropy it is assigned H and the class of maximum probability c. 189
C.3 High confidence MNIST test examples together with LIME and SHAP explanations for the top 3 predicted classes. The model being investigated is a BNN with the architecture described in Antorán et al. [2021]. The highest probability class is denoted by $\hat{y}$. 191
C.4 Ten MNIST test digits for which our BNN's predictive entropy is above the rejection threshold. A single CLUE is provided for each one. For each digit, the top scoring class is denoted by $\hat{y}$. LIME and SHAP explanations are provided for the three most likely classes. 192

C.5 CLUEs generated for MNIST digits for which our BNN's predictive entropy is above the rejection threshold. The BNNs predictive entropy for both original inputs and CLUEs is shown under the corresponding images. 193
C.6 U-FIDO counterfactuals generated for MNIST digits for which our BNN's predictive entropy is above the rejection threshold. The BNNs predictive entropy for both original inputs and counterfactuals is shown under the corresponding images. 193
C.7 Setup of tabular user studies. 194
C.8 Example Tabular Main Survey questions 195
C.9 Examples of high uncertainty digits containing characteristics that are uncommon in our modified MNIST dataset. Their corresponding CLUEs and ΔCLUEs are displayed beside them. 196
C.10 MNIST User Study Setup 197

D.1 The green curve corresponds to the (unknown) Pareto front of the MOO problem in Equation D.1, e.g. $\mathsf{A} = (1 - L_h(\pi_0^*), -c(\pi_0^*))$, $\mathsf{B} = (1 - L_h(\pi_1^*), -c(\pi_1^*))$, and $\mathsf{D} = (1 - L_h(\pi_\lambda^*), -c(\pi_\lambda^*))$ for some $\lambda \in (0,1)$. The point $\mathsf{E} = (1 - L_h(\pi^\varepsilon), -c(\pi^\varepsilon))$ corresponds to the constrained optimisation problem Equation D.4, which is the point on the curve that we would like our decision support policy to achieve. The dashed line corresponds to the set $\{\pi_\mu := \mu \cdot \pi_0^* + (1-\mu) \cdot \pi_1^* : \mu \in [0,1]\}$, e.g. $\mathsf{C} = (1 - L_h(\pi_\mu), -c(\pi_\mu))$ for some $\mu \in (0,1)$. While the set of Pareto optimal policies Π_h^{opt} may be difficult to compute without further assumptions, we propose a tractable proxy $\{\pi_\mu\}$, that can be recovered if one is able to compute π_0^* and π_1^*, which would already require accurate estimation of $L_h(x,a)$ values for all x and a. 204
D.2 We depict the latent space of the CIFAR subset used. Embeddings without filtering (left) and after filtering out points that violate separability of classes (right). We use the embeddings on the right for our Synthetic-2*A* task and the left for the natural CIFAR tasks. 206
D.3 We plot the expected accuracy and the negative of the expected cost for an individual sampled from the Synthetic-2*A* dataset for $\lambda \in [0,1]$. The ideal policy would lie in the far right corner. The red region denotes policies that fall within ε of the best risk for the sampled individual. We observe that there exists a policy that lies in the red region for both THREAD-KNN and THREAD-LinUCB. We also vary the cost structure (i.e. the cost of $\mathrm{A}_1 = 0$ and the cost of $\mathrm{A}_2 = 0.25, 0.5, 0.75$ in the first, second, and third row respectively). . . . 213

D.4 Top is CIFAR-3*A* and Bottom is MMLU-2*A*. For each of the $N = 10$ simulators, we instantiate using the population data, we compute the difference in expected loss between the optimal policy for that simulator and the learned policy for all values of $\lambda \in [0, 1]$ in increments of 0.05, and compute the difference in expected cost in the same manner. For KNN, we observe overlap between many values of λ that all come within ε of the Best Risk: this implies we can use lower values of λ to potentially achieve the desired level of performance. For LinUCB, we find that large values of λ are required to achieve a suitable level of performance. 214
D.5 Loss plots over time for human simulations (denoted by r_{A_i}) in the CIFAR-3*A* setting. 214
D.6 Snapshots of the recommended forms of support learned via `Modiste` for all participants in the CIFAR task, for different λ. Policies learned using KNN get closer to optimal; see Figure 5.2. 219
D.7 Highlighting individual differences in recommended support policies learned via `Modiste` for participants in the MMLU task. Topics are labeled as in Figure 5.3; clockwise from the top: math (M), biology (B), computer science (CS), foreign policy (FP). Adjacent to each policy visualization, we depict each participants' *unassisted* accuracy per topic; i.e. the accuracy for `DEFER` arms. We sample 7 participants out of the 10 per variant. Just as in our other experiments, we also visualize t-SNE embeddings here [Van der Maaten and Hinton, 2008]. 220
D.8 Example CIFAR interface for the `HUMAN ONLY` form of support. 221
D.9 Example CIFAR interface for a `MODEL` form of support. Note that the image shown in this example is corrupted to create settings where the decision-maker needs to calibrate when to rely on each form of support as described in Section 5.4.1. 221
D.10 Example CIFAR interface for a `CONSENSUS` form of support, using CIFAR-10H labels. Note that the image shown in this example is corrupted to create settings where the decision-maker needs to calibrate when to rely on each form of support as described in Section 5.4.1. 222
D.11 Example CIFAR interface for a `CONSENSUS` form of support when the form of support is intended to be unhelpful. 222
D.12 Example MMLU interface for the `HUMAN ONLY` form of support. 223
D.13 Example MMLU interface for the `LLM` form of support. 223

D.14 Relationship between a participants' sensibility of reliance (measured as the proportion of times they correctly agreed or disagreed with the form of support's prediction) and the loss incurred by the learned policy for the participant. Reliance is computed over all trials (100 for CIFAR, 60 for MMLU), and loss is averaged over the final 10 timesteps. We uncover an inverse relationship between reliance sensibility and the performance of learned policies for MMLU. 225

D.15 There is not a strong relationship between a participants' sensibility of reliance (measured as the proportion of times they correctly agreed or disagreed with the form of support's prediction) and the cost incurred by the learned policy for the participant. 226

List of Tables

2.1 Functional forms for various attribution-based feature importance methods . . 15

2.2 Commonly used metrics for the quantification and communication of uncertainty. 23

2.3 This feedback-update taxonomy illustrates the diverse ways practitioners can convert expert feedback, which generally comes via domain- or observation-level feedback, into model updates, which are either dataset, loss function, or parameter space changes that entail changes to the dataset, loss function, or parameter space respectively. Each cell corresponds to a part of the Feedback-Update Taxonomy. 37

3.1 Summary of select deployed local explainability use cases 50

4.1 Relative performance measure obtained by all methods on all datasets under consideration. Lower is better. The dimensionality of each dataset is beside its name. *e* and *a* indicate results for epistemic and aleatoric uncertainty respectively. 91

4.2 Accuracy (%) of participants on the Tabular main study broken down by dataset and by the certainty of test points. 94

5.1 In the standard setting, we evaluate the average excess loss (i.e. $L_h(\pi) - L_h^{opt}$), where lower is better, and standard deviation across 5 individuals in each expertise profile (Section 5.3.3) for both CIFAR-3*A* (Left) and MMLU-2*A* (Right). $L_h(\pi)$ is computed by averaging across the last 10 steps of 100 total time steps. We find that learning decision support policies are not helpful for decision-makers with "invariant" expertise profiles. However, for individuals who fall under the "strictly better" and "varying" profiles, we **bold** the variant with lowest excess loss, and find at least one `THREAD` variant, particularly KNN, outperforms offline policies. 109

5.2 In the cost-aware setting for MMLU-2*A*, we compare the expected loss $L_h(\pi)$ and the expected cost $c(\pi)$—lower is better—averaged across the last 10 steps of 100 total time steps of 5 separate runs. From the data collected in Section 5.4.2, we select the two individuals who have a "varying" profile (corresponding to the decision-makers with the first and second lowest risk without support) and two individuals who have a "strictly better" profile (corresponding the decision-makers with the lowest and highest L_h^{opt}). We follow the workflow described in Section 5.3.2 to find a suitable λ, reported for each person in Table D.2. We **bold** the algorithm that achieves the lowest cost within ε risk of the L_h^{opt}, and find that `THREAD` outperforms offline policies. 110

5.3 We report expected loss $L_h(\pi)$ and expected cost $c(\pi)$ incurred (lower is better) in the last 10 trials by Prolific participants for each Algorithm and **bold** the variant with the lowest $L_h(\pi)$. `Modiste` learns effective, low-cost policies: for CIFAR, `Modiste` outperforms all baselines. For MMLU, we find that `Modiste` learns a policy with roughly the same performance as the best offline policy but at *half* the cost. We also consider different choices of λ, where $\lambda = 1.0$ corresponds to the standard setting and $\lambda \neq 1.0$ corresponds to a cost-aware setting, where the choice of λ was selected according to Section 5.3.2. . . . 112

B.1 Practitioners can leverage DIVINE to value data points based on their contributions to model-specific evaluation metrics and then select a diverse, influential subset of points as a summary. We list other methods based on (1) their dependence on model parameters (e.g. prototypes are model-independent), (2) the diversity in the points to which they assign high importance, and (3) the metric with respect to which the data points are valued. 166

B.2 Various candidate evaluation functions $\mathcal{L}$; note $P_{j,a,j} = P(\hat{y} = j | A = a, y = j)$. 167

B.3 Summary of datasets used in our experiments. (*)We use a 7 feature version of COMPAS; however, other versions exist. 174

B.4 LSAT Unfairness Inducing Points 176

C.1 Summary of datasets used in our experiments. (*) We use a 7-feature version of COMPAS, however, other versions exist. 185

D.1 Unlike Table 5.2 where risks come from pilot studies, we instantiate simulated **synthetic** humans, each specified by the human's loss on each form of support H-ONLY (Human only) and H-MODEL (Human+Model), for MMLU-2*A*. We report the expected loss $r_h(\pi)$ and the expected cost $c(\pi)$—for both metrics, lower is better—averaged across the last 10 steps of 100 total time steps along with their standard deviations (across 5 runs). We bold the algorithm that achieves the lowest cost within ε of the Best Risk for each human simulation and find that our algorithm outperforms baselines on various simulated humans. For each form of support, we specify risk r over each of the 4 topics for MMLU-2*A*. We fix the cost structure: $c(\text{H-ONLY}) = 0.0$ and $c(\text{H-MODEL}) = 0.1$. The selected λ values used for `THREAD` are in Table D.2. For `THREAD`, we select the best policy over a sweep of λ values that minimizes Eq. D.4 under $\varepsilon = 0.05$. We also indicate the values of λ chosen beneath the results table. 215

D.2 The best value of λ that is selected from doing a sweep over $\lambda \in [0,1]$ that corresponds to the human simulators in Table 5.2. The column ordering is the same as the main text. 216

D.3 Vary exploration parameters (α for LinUCB and γ for KNN). We find that LinUCB generally performs better when exploration increases and KNN generally performs better when exploration decreases. 216

D.4 We vary the K and W parameters used to instantiate KNN across three datasets and report the expected risk $L_h(\pi)$ (lower is better) and expected cost $c(\pi)$ (lower is better) across 5 runs. The human simulation used here is the same as the one used in the second setting of Table D.1. 216

D.5 On the CIFAR-2*A* dataset, we explore the effect of varying the embedding size for KNN, with $K = 8$, $W = 25$ (Top), and LinUCB (Bottom). We report the expected risk $L_h(\pi)$ (lower is better) and expected cost $c(\pi)$ (lower is better) averaged over 5 runs. Recall that in our main text experiments we used two-dimensional t-SNE embeddings, not these model embeddings, for ease of visualization. 217

D.6 We evaluate $|\mathcal{A}| > 3$ in the cost-agnostic setting, $L_h(\pi_t)$ for $t = 50, 100, 250$ for the same setup as Table 5.1, but with more arms. We find that `THREAD` works well over time. The success of LinUCB is highly dependent on the embedding space and its geometry. 217

D.7 We report expected loss $L_h(\pi)$ and expected cost $c(\pi)$ incurred (lower is better) across *all* trials by Prolific participants for each Algorithm and **bold** the variant with the lowest $L_h(\pi)$. `Modiste` learns effective, low-cost policies: for CIFAR, `Modiste` outperforms all baselines. For MMLU, we find that `Modiste` learns a policy with roughly the same performance as the best offline policy but at *half* the cost. We also consider different choices of λ, where $\lambda = 1.0$ corresponds to the standard setting and $\lambda \neq 1.0$ corresponds to a cost-aware setting, where the choice of λ was selected according to Section 5.3.2. 220

Chapter 1

Introduction

Decisions are difficult. Our daily lives are peppered with incessant strings of decision-making. Indeed, it is not easy to decide when to wake up, pick a country to visit next, or choose a route for a hiking trip. We are showered with optionality and the freedom to choose based on our individual preferences and personal limitations [Schwartz et al., 2002]. But, we are imperfect. Our decisions may be sub-optimal, suffer from our irrationality, or result in harm [Kahneman, 2011]. To improve our decision outcomes, we may leverage a host of decision aids: conferring with those we trust, studying the lives of those before us, or querying algorithmic decision-making systems. In the past half-century, we have seen algorithmic decision-making blossom. Such algorithmic aids are generally powered by artificial intelligence (AI), which usually attempts to mimic human decision-making with the help of computation. Machine learning (ML) models are a popular type of AI-powered decision aid: they have tremendous predictive power but often come with a burden of opacity. As a society, we now let ML models help us decide where to dine with friends, which videos to show our children, and how to treat the ailing. The sheer scale at which such systems are used is unfathomable to the uninitiated: models have permeated many aspects of life, including high-stakes settings such as healthcare [Rajpurkar et al., 2022], national security [Schmidt et al., 2021], and air traffic control [Debernard et al., 1993]. Much of this has occurred without us demanding transparency from these models, but this attitude is being increasingly challenged. For us to trust a model to serve our best interests, we need to have certain expectations for the model's behavior. We want the model to display trustworthiness, which begs the question: *"How can we trust an AI system to help us make decisions?"*

This thesis concerns trustworthy machine learning. For ML models, trustworthiness can be shown in many ways, which may include transparency into the models' behavior. A transparent AI system may be worthy of a decision-maker's trust, as we discuss in Chapter 2. Transparency encompasses a variety of efforts to provide stakeholders, particularly model

developers and decision-makers, with relevant information about how a model works [Weller, 2019]. Many technology companies and regulatory bodies have released AI principles that include transparency as a core value [IBM, 2019; Microsoft, 2019; OECD, 2019]. As a result, there is a good deal of important work on what we will call procedural transparency – insight into the sociotechnical ecosystem of the model. This has included documentation of the training procedure, dataset, or organizational best practices [Arnold et al., 2019; Gebru et al., 2021; Mitchell et al., 2019]; disclosure of how personal data is stored and used by the system [Council of European Union, 2018]; third-party certification of model behavior [Cihon et al., 2021]; or release of the model's code or weights [Askell et al., 2019]. Here, we focus on algorithmic transparency.

Algorithmic transparency exposes properties of models to stakeholders for purposes that include understanding, improving, and contesting model recommendations. Tools, like explainability and uncertainty estimation, modulate a stakeholder's trust in algorithmic decision-making systems. In this thesis, we will discuss how to create models that explain their predictions to stakeholders and that leverage decision-maker expertise to improve human-machine team performance. To accomplish this, we use many different research tools, which include convening stakeholders to understand existing shortcomings (Chapter 3), devising principled methods to address stakeholder needs (Chapter 4 and 5), and running large-scale user studies to study the efficacy of our methods (Chapter 4 and 5).

A popular form of algorithmic transparency is explainability, which provides stakeholders with insights about a model's behavior on specific data [Gilpin et al., 2018]. For example, governments use models to detect anomalous behavior in chat rooms and flag potential criminal activity; explainability provides the *why* behind a positive flag. With the Partnership on AI, we conducted an influential, multi-stakeholder interview study with over twenty multinational organizations about how they view and use explainability [Bhatt et al., 2020b]. We report our findings in Chapter 3: we found that there was little use of any explainability methods in production. Under little regulatory pressure, industry organizations had no accountability for deploying explainability tools. Based on participatory design sessions that collated the needs of policymakers, data scientists, and executives with respect to explainability, our findings push for systematic evaluation of explanations when deploying ML models [Bhatt et al., 2021c] and for context specificity when providing transparency to regulators and auditors [Bhatt et al., 2020a]. These findings inspired our subsequent work to build new methods for algorithmic transparency. In Chapter 4.1, we demonstrate how decision-makers can leverage example-based explanations to simulate model behavior [Bhatt et al., 2021b]. However, explainability alone might not be enough for decision-makers to gauge whether a model has sufficient knowledge about a task.

Beyond explainability, we argue for a complementary form of algorithmic transparency by estimating and communicating the uncertainty associated with model predictions [Bhatt et al., 2021a]. For example, notions of uncertainty are critical in public health [Spiegelhalter et al., 2011]. A model for easing lockdowns ought to expose its blind spots, allowing officials to override a model's recommendations when appropriate. In Chapter 4.2, we show that uncertainty could be rendered more actionable if it were expressed in terms of model inputs, answering the question: *"Which input patterns lead my prediction to be uncertain?"* Our approach finds explanations of uncertainty in input space, by searching in the latent space of a deep generative model [Antorán et al., 2021]. Understanding which input features are responsible for uncertainty can help stakeholders learn where training data is lacking or erroneous. When using models to analyze radiology scans, a clinician can identify under-represented subgroups in the training data for which the model should not be used [Bhatt and Shams, 2021]. Treating those cases carefully, until more data is collected, leads to more accurate predictions for a range of patients. Decision-makers can thus use models more effectively after they see uncertainty visualized in input space.

Unfortunately, there are severe shortcomings of algorithmic transparency. Notably, algorithmic transparency usually implies a unilateral discourse: decision-makers passively consume information about models but cannot express feedback about models' behavior. Transparency, in some sense, may not be appropriate for all decision-makers [Amershi et al., 2019; Weller, 2019]. We develop methods to learn policies of when experts ought to rely on models and their explanations as decision aids in Chapter 5. In certain cases, models outperform experts; in this regime, we may want to deploy mechanisms that promote expert adherence to model recommendations [Bhatt et al., 2023]. Alternatively, the expert may be competent without a model; here, the model could ignore these cases and focus on cases where the expert performs poorly [Wilder et al., 2020; Zerilli et al., 2022]. Imagine a clinician who uses a model, trained on knee X-rays, to decide what type of knee injury a patient has. If a model has only seen ligament injuries, the clinician may need to intervene and override a model's erroneous prediction when observing an out-of-distribution meniscal tear never-before-seen by the model. A human-machine team reduces potentially flawed decision-making by better allocating cases for human review [Babbar et al., 2022]. Adaptively allocating scans for review can help a clinician maintain agency in decision-making while leveraging the predictive power of models. We can therefore build complementary human-machine teams. This turns algorithmic transparency into a bilateral dialogue between decision-makers and models, thus empowering experts to tailor models to their needs.

This thesis puts humans at the center of algorithmic decision-making. To build a human-machine team, we harness the power of ML models infused with algorithmic transparency and

decision-maker expertise. We first provide a background of trust, transparency, and human-machine teams in Chapter 2. We then discuss our study of how algorithmic transparency is used in practice in Chapter 3. We find that many organizations do not leverage explainability tools for a wide range of stakeholders. In Chapter 4, we study how transparency can be used in decision-making tasks. Specifically, we develop an example-based explanation method to assist decision-makers in simulating model behavior in Chapter 4.1 and a feature-based explanation method to help decision-makers understand where uncertainty lies in input space in Chapter 4.2. In Chapter 5, we study how to personalize access to a model as a decision aid based on where a particular decision-maker's expertise lies.

Publications and Contributions

Chapter 2 synthesizes multiple review papers I wrote during my PhD [Bhatt and Shams, 2021; Bhatt et al., 2021a; Chen et al., 2023a; Zerilli et al., 2022]. I led the writing of [Bhatt et al., 2021a] (which appeared at AIES 2021) and wrote [Bhatt and Shams, 2021] (which appeared as a book chapter) alone as a primer on trustworthy ML for healthcare practitioners. I co-led [Chen et al., 2023a] (which appeared in Patterns 2023) to cover how to update models based on stakeholder feedback. I wrote much of the technical portions of [Zerilli et al., 2022] (which appeared in Patterns 2022) to show the interplay between reliance and trust. In Chapter 3, we discuss how algorithmic transparency is used in practice. This chapter is based on [Bhatt et al., 2020b] (which appeared at ACM FAccT 2020) and on [Bhatt et al., 2020a] (which appeared at ICML WHI 2020). I led both works: I designed and conducted the interview study in the former, and I devised the day-long convening for the latter. I then synthesized notes from the study and the convening into their respective papers. In Chapter 4, we devise two explainability methods and measure their efficacy via user studies. This chapter is based on [Antorán et al., 2021] (which appeared at ICLR 2021) and on [Bhatt et al., 2021b] (which is under review). For [Antorán et al., 2021], my contributions were on the empirical front and included designing and running the human subject experiments. For [Bhatt et al., 2021b], I led the entire work, from inception to computational experiments to user studies. In Chapter 5, I co-led [Bhatt et al., 2023], wherein I devised the algorithm, ran all of the experiments, and co-designed the user studies. I provide an exhaustive list of the publications from my PhD in Appendix A.

Chapter 2

Background

The study of human-machine systems is central to a variety of behavioral and engineering disciplines, including management science [Dietvorst et al., 2015, 2018; Lee, 2018; Lewandowsky et al., 2000; Logg et al., 2019; Simon, 1960], human factors [Dzindolet et al., 2003; Manzey et al., 2012; McGuirl and Sarter, 2006; Parasuraman and Riley, 1997], robotics [Andrist et al., 2016; Bainbridge et al., 2011; Desai et al., 2013; Dragan et al., 2015; Gombolay et al., 2015; Robinette et al., 2015; Salem et al., 2015], and human-computer interaction [Amershi et al., 2019; De-Arteaga et al., 2020; Jacovi et al., 2021; Kaur et al., 2020; Montague and Xu, 2012; Schmidt et al., 2020; Suresh et al., 2020; Weerts et al., 2019; Yang et al., 2020]. Recent advances in artificial intelligence (AI) and machine learning (ML) have brought the study of human-machine teams (HMTs) into sharper focus. An important set of questions for those designing HMT interfaces concerns trust: specifically, human trust in the AI systems (i.e. ML models) with which they form teams. Trust in machines has been defined as "the attitude that a [machine] will help an individual achieve her goals in a situation characterized by uncertainty and vulnerability" [Botsman, 2017; Jacovi et al., 2021; Lee and See, 2004]. More generally, trust is "a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behavior of another" [Rousseau et al., 1998]. Trust is therefore a subjective attitude of the vulnerable party, to be distinguished from *trustworthiness*, which is an *objective* attribute of the trustee [O'Neill, 2018]. Just as human collaboration would be impossible without some degree of trust between team members, some form of trust in algorithmic systems is necessary for HMTs to perform effectively. It follows that if trust is ever violated, its repair will be crucial in any attempt to rehabilitate team performance.

Displays of *trustworthiness* are essential to consider when discussing HMTs; of particular interest is how ML models can show trustworthiness when aiding human decision-makers. We start by discussing how perceiving an ML model making mistakes (i.e. an unreliable ML model) violates trust in HMTs [Zerilli et al., 2022]. We then review how such violations might be



Fig. 2.1 Scale of stakeholder (e.g. decision-maker) attitudes towards machine learning models in human-machine teams

repaired. In doing so, we discuss the role played by algorithmic transparency [Bhatt and Shams, 2021; Bhatt et al., 2021a] in the process of demonstrating trustworthiness to and repairing the trust of human decision-makers. We end with a brief background on how to update ML models based on decision-maker feedback in HMTs [Chen et al., 2023a].

2.1 On Trust

In an ideal world, only AI systems that are trustworthy would be trusted [Botsman, 2017]. If a system performs considerably worse than a human (or human team) acting alone, or if a system is opaque or ethically suspect, then distrust would be justified. But distrust is problematic when the distrusting behavior to which it leads—what has been termed algorithm "aversion"—is an overreaction to having witnessed the system's mistakes [Bansal et al., 2021b; Dietvorst et al., 2015, 2018; Dzindolet et al., 2003]. In the most extreme case, algorithm aversion results in a refusal to engage with a system at all or a blatant disregard of its recommendations—an attitude we term "opposition."

Conversely, there is such a thing as too much trust—algorithm "appreciation" [Logg et al., 2019] or overtrust —where a human is so impressed by a system that they cease actively monitoring its outputs [Dzindolet et al., 2003; Hou and Jung, 2021; Parasuraman and Riley, 1997], and, in the limiting case, follow its every recommendation without question—an attitude we term "loafing." As one might have guessed, appreciation is not a problem for systems that pass a very high threshold of accuracy [Goddard et al., 2014; Zerilli et al., 2019a]. Accordingly, the AI systems of interest to HMT research are generally trustworthy in the sense that they are not so error-prone that algorithm aversion becomes rational, but not so adept that overtrust ceases to be a problem. Both aversion and appreciation are inappropriate attitudes towards systems that are generally trustworthy in this sense [Dietvorst et al., 2015; Parasuraman and Riley, 1997; Sunstein, 2023].

To our knowledge, these various attitudes are rarely cast within a single frame of reference. Papers tend to problematize overtrust or distrust, failing to demonstrate that any one system can

engender both phenomena. Hence we envisage opposition and loafing as lying at opposite ends of a spectrum, with algorithmic "vigilance" representing an ideal mid-point between them, and aversion and appreciation lying mid-way between this ideal and each extreme: see Figure 2.1. Algorithmic vigilance, as we use the term, is an attitude of active user engagement and healthy skepticism. It marks the level of trust that a human (or human team) should display towards an AI system from the point of view of optimal HMT performance. Confusingly, this attitude is sometimes given the name "complementarity," presumably to indicate that some ideal division of labor has been struck between human and machine, such that humans will focus on tasks too difficult for machines and vice versa [Bansal et al., 2021b; Horvitz and Paek, 2007; Steyvers et al., 2022]. But complementarity in this sense may be compatible with human loafing, so we prefer the term vigilance.

In human factors engineering and human-computer interaction, overtrust has been extensively researched for close to four decades [Hoff and Bashir, 2015; Körber, 2018; Lee and See, 2004; Parasuraman and Riley, 1997]. In human factors, the phenomenon goes by the names of "automation complacency" and "automation bias" [Parasuraman and Manzey, 2010]. Though similar, these effects are not the same. Automation complacency describes the state of passivity, diffidence, or deference into which the user of a system falls when uncritically relying on technology they deem more proficient than themselves [Pazouki et al., 2018]. In effect, it is the failure to attend to the possibility that a system may be wrong by not seeking out either confirmatory or disconfirmatory evidence [Manzey et al., 2012]. Automation bias is a more extreme variant of this attitude and manifests when a user actively prefers a system's signals over actual—i.e. overtly—contradictory information, including information from more reliable sources such as the user's senses [Manzey et al., 2012; Pazouki et al., 2018]. Crucially, it is the perception of a system's superior performance that induces these states: they are rarely observed when a system is considered liable to even occasional error [Bagheri and Jamieson, 2004; Banks et al., 2018a,b; Dzindolet et al., 2003; Kleinberg et al., 2018; Manzey et al., 2012].

By contrast, algorithm aversion has not been nearly as well researched or theorized. But some results are notable. Users of AI systems in many lab-based settings have been shown to display unrealistically high levels of trust initially, only for that trust to drop precipitously in response to seeing a system err [Dietvorst et al., 2015, 2018; Dzindolet et al., 2003]. Human decision-makers then typically retreat to their judgment, even when doing so leads demonstrably to even more errors [Dietvorst et al., 2015, 2018; Dzindolet et al., 2003]. For example, during an incentivized task, when given the choice between relying on their judgment exclusively or relying on an algorithm's forecasts exclusively, most participants who had not seen the algorithm perform chose to rely on the algorithm exclusively, while most of those who had seen the algorithm perform (and hence err) chose to rely on human judgment, despite observing the

algorithm's better performance [Dietvorst et al., 2015]. It has been suggested that this effect is greater for obvious errors than for subtle ones because obvious errors can quite drastically upset a user's initially high expectations of a system's competence [Dzindolet et al., 2003]. Moreover, a decision-maker's expertise can affect their perception of machine errors [Logg et al., 2019]. Decision-makers, who are adept or self-confident in tasks that have been delegated to automation, tend to ignore machine advice [Lee and Moray, 1994] and, as a result, make less accurate predictions relative to lay people willing to follow machine advice [Dietvorst et al., 2015, 2018; Dzindolet et al., 2003].

The pattern of trust → error → distrust in which trust becomes difficult to restore despite impressive system performance could be explained by a decision-maker's "diminishing sensitivity to error." Throughout five studies, Dietvorst and Bharti [2020] found that participants displayed error intolerance when confronted with systems that were highly reliable on average but incapable of perfect forecasts, and error tolerance when confronted with systems that were less reliable on average but that had at least a chance of making near-perfect forecasts. If decision-makers have diminishing sensitivity to error, it would plausibly explain why AI systems that make even a single error are penalized so harshly: decision-makers' hopes for near-perfect automated forecasting having thus been dashed, the more volatile and error-prone decision-making option (human judgment) suddenly looks like the most appealing one. Human forecasters can at least stumble on near-perfect forecasts after all. In any event, errors seem to have a stronger impact on trust than correct outputs [Dzindolet et al., 2003; Manzey et al., 2012]. This phenomenon is indeed so pronounced that cumulative feedback about a system's superior performance presented at the end of a task session may not be enough to counteract human decision-makers' misgivings after having had their expectations disappointed throughout a task session [Dzindolet et al., 2003].

Curiously, while higher levels of trust generally lead to greater reliance, trust and reliance are not monotonic. An untrustworthy system may rightly arouse distrust (measured subjectively by self-evaluation and report) and yet continue to be relied upon (judging by actual usage data) [Chavaillaz et al., 2016; Dzindolet et al., 2003; Manzey et al., 2012]. The converse of this situation has also been observed so that even when the subjective feeling of trust eventually recovered after witnessing a system failure, immediate post-failure behavior (e.g. scrupulous cross-checking) did not revert to the pre-failure norm [Manzey et al., 2012].

2.2 On Trustworthiness Mechanisms

We should lodge an important caveat at the outset. Trust is, in the first instance, an inter*perso*nal attitude between humans, not between humans and machines. Interpersonal trust has been the

subject of investigation in organizational and social psychology for several decades [Epley et al., 2007; Evans and Krueger, 2009; Fiske, 2018; Lewicki and Brinsfield, 2017; Mayer et al., 1995; Siegrist et al., 2003, 2005; Thielmann and Hilbig, 2015], and in these fields, trust is understood to be influenced by at least two factors: (i) the competence of the trustee; and (ii) the degree to which the trustee exhibits good faith/benevolent intentions (e.g. in a contractual setting, the desire to support the other party's efforts in performing the contract) but more generally, the absence of ill will or ulterior motives in the trustee [Epley et al., 2007; Evans and Krueger, 2009; Fiske, 2018; Lewicki and Brinsfield, 2017; Thielmann and Hilbig, 2015].[1] Recast into language more appropriate for AI systems, we can take *competence* to denote a system's accuracy, and *good faith* to denote a system's transparency, as judged by a range of criteria including, but not limited to its explainability. Indeed, good faith is not, strictly speaking, the same thing as transparency, and transparency is often a means of verifying good faith (as well as accuracy). However, it is also true that transparency can itself be an expression of good faith on the trustee's part, as when someone who is "open" or "forthright" is understood to harbor no ill will. In other words, while good faith encompasses more than transparency, it often encompasses at least that much. Accuracy and transparency are by no means the only antecedents of trust in AI systems [Glikson and Woolley, 2020; Hoff and Bashir, 2015]. Other important, if less marked, determinants of trust in automation include ergonomic and demographic factors, team size and composition (e.g. in terms of active vs. passive users), and task type and complexity.

In the previous section, we discussed the interplay between trust and a decision-maker's reliance on an AI system. Many of those results could easily lead one to the cynical conclusion that the best way for AI systems to promote the right amount of trust is simply by shielding users from information about the system's decisions—in effect, by being less transparent [Dietvorst et al., 2015; Dzindolet et al., 2003].[2] Yet there is reason to believe that a better calibration of trust to a system's actual level of accuracy can be achieved by providing more of the right kind of transparency: not just cumulative performance feedback (delivered at the end of a task session), but continuous performance feedback that allows the user to maintain a better picture of the system's relative superiority in real time [Dzindolet et al., 2003; Wang et al., 2009]. Some researchers have even noticed a pattern in the way accuracy information interacts with

[1] We wholly ignore a third common factor of trust: the affability or "warmth" of the trustee, asking can we assess the integrity or honesty of the trustee [O'Neill, 2018]. The warmth of AI systems is more relevant for anthropomorphic AI systems (e.g. virtual agents and robots), not ML models.

[2] Dzindolet et al. [2003] report that: "eliminating operators' awareness of an automated decision aid's obvious errors (through blinding the participants to the decisions of the aid) was useful in promoting appropriate automation reliance if participants were continually reminded of their and their aid's performance. Unfortunately, applying these techniques outside the laboratory is problematic. It would not be reasonable to provide someone with an automated decision aid but not allow them to see the decisions the aid has made."

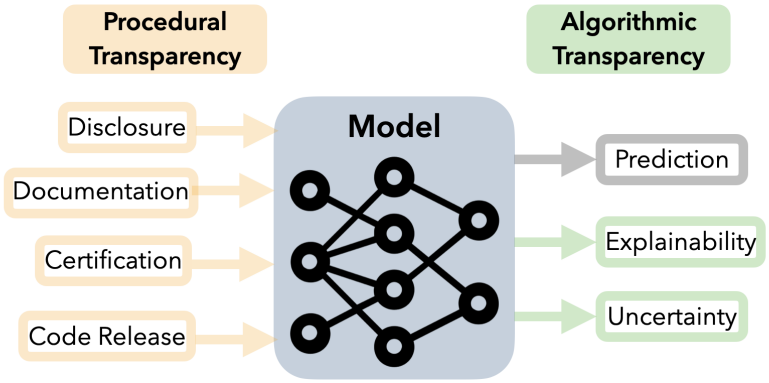


Fig. 2.2 Depiction of various types of transparency regarding a machine learning model

user attitudes. Metainformation about low-reliability automation runs the risk of promoting overtrust (as measured by higher trust ratings), but metainformation about high-reliability automation seems to have the opposite effect. Presumably, this is because, in the first case, users are placed on notice, ready to step in and override the system when it fails, which could, perversely, contribute to a sense that the system is actually more reliable than it is; while in the second case, metainformation may consolidate users' unrealistic expectations, which are inevitably contradicted on witnessing errors, with the attendant fallout [Seong and Bisantz, 2008].

Transparency is one such mechanism by which an AI system can display its trustworthiness (i.e. good faith) to decision-makers. Throughout this thesis, we take transparency to mean any relevant information about an AI system that can be provided to stakeholders (e.g. decision-makers, policymakers, social workers, ML engineers). Transparency allows stakeholders to audit systems to see if the system behaves as desired: we depict various forms of transparency in Figure 2.2. In its simplest form, the prediction (or output) of an ML model would be a relevant piece of information about which the decision-maker can reason. Nonetheless, the prediction alone may not suffice as a form of transparency to understand the trustworthiness of an AI system. Transparency can be broken into two broad categories: procedural and algorithmic [Bhatt and Shams, 2021]. Procedural transparency captures information about the sociotechnical environment in which an AI system was developed and deployed. In some sense, this information does not depend on the output of the system, but rather on the ecosystem around a system. Algorithmic transparency, on the other hand, encompasses what the AI system can communicate about its behavior. Upon probing the system, the latter type of transparency, which includes explanations and uncertainty estimates, can be used to assess if the system is behaving in good faith. While the focus of this thesis is algorithmic transparency, we now describe procedural transparency.

2.2.1 Procedural Transparency

Procedural transparency entails conveying information about how an AI system is trained and how it will be used. AI systems represent a chain of models, data, and human decisions [Lawrence, 2019], or, in other words, a distinctly sociotechnical system: see [Selbst et al., 2019] for a summary of common issues faced with sociotechnical systems. One AI system can consist of a single ML model or many ML models, each requiring a different level and style of transparency for a wide set of stakeholders to operate cohesively. At times, transparency requirements can be just a matter of *disclosure* of the data used to train and maintain the model(s): such disclosure may ostensibly be in line with the recent European General Data Protection Regulation (GDPR) [Council of European Union, 2018]. One form of procedural transparency would be to publish an algorithm's *code*, though this transparency would not provide an intelligible explanation to most non-expert stakeholders and would have little utility without the training data used.

More recently, regulatory communities have suggested *certification* as a mechanism for assessing AI systems [Cihon et al., 2021]. Certification programs are a form of compliance-based regulation (sometimes referred to as soft law or private governance), implying that they are not enforced by law nor bound to a specific jurisdiction. AI certification could provide the necessary translation from high-level principles [Microsoft, 2019] and policy [Veale, 2020] to actionable components measured against an international standard [Squirrell, 2008]. To date, no certification has managed to find the correct balance between general applicability (i.e. relevant for all AI systems) and specificity (i.e. enough detail to be actionable). However, a successful certification program could yield a certification mark for acceptable AI systems and would be a welcome form of procedural transparency.

Another form would be to disclose properties of the training procedure and datasets used: this *documentation* is sometimes referred to as a model card [Mitchell et al., 2019]. Disclosing the procedure for training can include properties of the model (developers, version, licensing), intended use cases for the model (primary use, out-of-scope use cases, intended users), details about the training data (diversity, preprocessing, feature selection), performance metrics (decision thresholds, qualitative results, unitary/intersectional analyses), and ethical considerations [Arnold et al., 2019; Gebru et al., 2021; Mitchell et al., 2019; Raji and Yang, 2019]. Though making this information available could be nontrivial and may expose proprietary information, such documentation can help stakeholders understand the functionalities and limitations of models. While some stakeholders are generally not equipped to understand how raw data and code translate into benefits or harms that might affect them individually, procedural transparency is still paramount to displaying trustworthiness in an HMT and ensuring the safe adoption of ML models in production.

2.2.2 Algorithmic Transparency

In this thesis, algorithmic transparency refers to information an AI system can expose about its behavior: this is in contrast to procedural transparency and the sociotechnical information it captures. Algorithmic transparency encompasses explainability (the ability to explain how the system made a prediction), uncertainty (the ability to convey when the system is unsure), and other technical aspects of the system. Again, algorithmic transparency is just one way for a model to display its trustworthiness via good faith [O'Neill, 2018].

In the context of ML, Lipton [2018] argues that algorithmic transparency enables people to *simulate* model behavior (e.g. reproduce exactly how the model comes to its output), *decompose* the model into smaller intelligible components (e.g. use methods proposed in Lou et al. [2012]), or *understand* what is important to the model (e.g. expose post-hoc model explanations). Zhang et al. [2020b] explore the impact of algorithmic transparency (i.e. providing model explanations) for fostering end user trust. Much of the existing literature limits algorithmic transparency to explainability [Lu et al., 2020]; however, communicating predictive uncertainty can be seen as a form of transparency as well [Jiang et al., 2018; Volk et al., 2014]. Algorithmic transparency beyond explainability can be salient in building effective HMTs, wherein people use models to complete tasks [Bansal et al., 2021b].

Beyond ML researchers, the study of algorithmic transparency has garnered interest from philosophers, psychologists, sociologists, lawyers, visualization experts, and human-computer interaction scientists [Lipton, 2009; Miller, 2019]. The interdisciplinary nature of transparency research makes it an exciting direction. A concerted effort will ensure that algorithmic transparency is built with real-world adoption in mind. In Chapter 3, we take a critical look at the practical utility of algorithmic transparency to date.

2.3 On Types of Algorithmic Transparency

As algorithmic transparency comes in many forms, we give a detailed background on two types, explainability and uncertainty, and then discuss whether transparency is necessary.

2.3.1 Explainability

No doubt one of the most pertinent forms of transparency is an explanation, which can enhance a stakeholder's understanding of how an ML model works, and hence why it might commit the sorts of errors it does [Bansal et al., 2021b; Dzindolet et al., 2003; Lee and Moray, 1992]. Explainability (hereafter used interchangeably with explainable ML) loosely refers to any technique that helps stakeholders of ML models understand why models behave the way they

do, and usually results in an explanation of a model's behavior. Explainability is arguably the most popular form of algorithmic transparency [Bhatt et al., 2020b; Lu et al., 2020]. Explanations have practical utility: from telling patients which symptoms were indicative of a particular diagnosis [Lundberg et al., 2018] to helping factory workers analyze inefficiencies in a production pipeline [Dhurandhar et al., 2018b].

By providing an explanation for how the model made a decision, explainability techniques seek to provide transparency directly targeted at a specific stakeholder, often aiming to increase the model's trustworthiness [O'Neill, 2018]. The importance of explainability as a concept has been reflected in legal and ethical guidelines for data and ML [Selbst and Barocas, 2018]. In cases of automated decision-making, Articles 13-15 of GDPR require that data subjects have access to "meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject" [Council of European Union, 2018].

While there is promise in explainable ML providing explanations of model behavior to non-expert stakeholders, explainability is mostly used by model developers to date: see Chapter 3. Explanations may be provided on a global level, summarizing model behavior for multiple data points or for the entire training dataset, or on a local level, explaining an individual prediction [Adadi and Berrada, 2018; Brundage et al., 2020]. Some global explainability techniques attempt to characterize the concepts learned by the model [Koh et al., 2020], find simpler models learned from the representation of complex models [Dhurandhar et al., 2018b], find prototypical points that summarize a dataset [Kim et al., 2016], or model the topology of the data itself [DuMouchel, 2002]. While global explainability is an important form of algorithmic transparency, we dive deeper into local explainability in this section.

Local explainability provides an explanation for a model's behavior on a single test point. Explanations can be generated post-hoc (i.e. after a model has been trained, a different model can generate an explanation for a specified point) or by-design (i.e. the model itself is transparent and provides explanations at test time) [Arrieta et al., 2020; Carvalho et al., 2019; Rudin, 2019]. Popular local post-hoc explanations come in the form of which features are important to the model when performing prediction (feature importance) [Davis et al., 2020; Lundberg and Lee, 2017; Ribeiro et al., 2016], which training points were most important to a particular prediction (training point importance) [Khanna et al., 2019; Koh and Liang, 2017; Yeh et al., 2018], or what needs to change in an input to change a model's prediction (counterfactual explanation) [Dhurandhar et al., 2018a; Ustun et al., 2019; Wachter et al., 2017]. We now detail methods for each type of local, post-hoc explanation.

Notation

A black-box model $f \in \mathcal{F}$ maps input values $x \in \mathcal{X}$ to output values $y \in \mathcal{Y}$. When we assume f has a parametric form, we write f_θ to denote a model parameterized by $\theta \in \Theta$. We use $\ell(y, f_\theta(x))$ to denote the loss function used to train f on a dataset $\mathcal{D} = \{z^{(1)}, z^{(2)}, \ldots, z^{(n)}\}$ of input-output pairs, $z^{(i)} = (x^{(i)}, y^{(i)}) \in \mathcal{X} \times \mathcal{Y}$. We sometimes abuse notation and write $\ell(z, \theta)$ to indicate the loss for a point $z = (x, y)$ under a model f parameterized by θ. We use g to denote an explanation function whose functional form will be defined inline. Additional notation will be introduced as needed.

Feature Importance

Feature importance (also known as feature-level interpretation, attribution, or saliency map) is by far the most widely used and well-studied explainability technique [Baehrens et al., 2010; Gilpin et al., 2018]. Feature importance methods define an explanation function $g : f \times \mathcal{X} \mapsto \mathbb{R}^d$ that takes in a model f and a point of interest x and returns real-valued importance scores $g(f, x) \in \mathbb{R}^d$ for all d input features. $g(f, x)_i$ is the importance of (or attribution for) feature x_i of x.

Common approaches to feature importance roughly fall into two categories: surrogate-based and attribution-based. A popular surrogate-based feature importance method is LIME [Ribeiro et al., 2016]. LIME learns a linear model locally around each test point and takes the coefficients of the linear model as the feature importance for the test point's prediction [Slack et al., 2020b]. LIME solves the following objective: $\arg\min_{g \in \mathcal{G}} \; L(f, g, \pi_x) + \Omega(g)$, where $L(f, g, \pi_x) = \sum_{z \in \mathcal{Z}} \pi_x(z)[f(z) - g(z)]^2$. With $\mathcal{G}$ restricted to the class of linear models, we want to find a simple model g that locally approximates f near x. LIME learns g using $\mathcal{Z}$, a set of instances generated by perturbing x, and using a similarity kernel $\pi_x(z)$, which indicates the similarity between x and z. While Ribeiro et al. [2016] resort to random sampling to generate $\mathcal{Z}$ (leading to differing explanations over multiple runs), Zafar and Khan [2019] use k-nearest neighbors and hierarchical clustering to generate deterministic LIME explanations. Ω regularizes the complexity of g: Ribeiro et al. [2016] take this to be the number of non-zero coefficients of g. Several practitioners have critiqued LIME for its inefficient sampling, its instability, and its ambiguous definition of neighborhood (i.e. selection of π) [Molnar, 2019; Zhang et al., 2019]. As a result, recent work has attempted to fit other interpretable approximations locally like rules, which might be more suitable for stakeholder consumption [Plumb et al., 2018; Ribeiro et al., 2018].

Explanation Functions	Formulation
Gradient Saliency	$g(f,x) = \frac{\partial f}{\partial x}$
Gradient $\odot$ Input	$g(f,x) = x \odot \frac{\partial f}{\partial x}$
SmoothGrad	$g(f,x,g_o) = \frac{1}{K}\sum_{i=1}^{K} g_o(f, x+\varepsilon_i)$ with each $\varepsilon_i \sim \mathcal{N}(0,1)$
Integrated Gradients	$g(f,x,\bar{x}) = (x-\bar{x}) \times \int_0^1 \frac{\partial f(\bar{x}-\alpha(x-\bar{x}))}{\partial x} d\alpha$
SHAP	$g(f,x)_i = \frac{1}{\lvert T\rvert} \sum_{S \subseteq T \setminus \{i\}} \binom{T-1}{S}^{-1} (f(\bar{x}_{[\bar{x}_{s\cup\{i\}}=x_{s\cup\{i\}}]} - f(\bar{x}_{[\bar{x}_s=x_s]}))$

Table 2.1 Functional forms for various attribution-based feature importance methods

Attribution Techniques. Attribution-based explanation functions roughly fall into two categories: gradient-based techniques [Ancona et al., 2018; Montavon et al., 2017; Shrikumar et al., 2017; Singla et al., 2019; Smilkov et al., 2017; Sundararajan et al., 2017] and perturbation-based techniques [Ancona et al., 2019; Chen et al., 2019b; Fong and Vedaldi, 2017; Lundberg and Lee, 2017; Martens and Provost, 2014; Štrumbelj and Kononenko, 2014]. Note that gradient-based techniques can be seen as a special case of a perturbation-based technique with an infinitesimal perturbation size.

We include the functional form of some explanation functions in Table 2.1. Baehrens et al. [2010] propose Gradient Saliency, which simply amounts to the gradient of the model output with respect to the input. This captures how much the model's predictions change in a small neighborhood around the input. Shrikumar et al. [2016] find that taking the element-wise product of the input and the gradient results in more visually intuitive explanations (dubbed Gradient $\odot$ Input): Ancona et al. [2018] find that Gradient $\odot$ Input is equivalent to DeepLift [Shrikumar et al., 2017] and ε-LRP [Binder et al., 2016] for a multilayer perceptron (MLP) with ReLu activations and no additive biases. Smilkov et al. [2017] propose SmoothGrad, which must be applied to another explanation function, g_o. This technique generates noisy inputs via additive Gaussian noise and averages the explanations of the noisy inputs, effectively smoothing out g_o. Yeh et al. [2019] show that SmoothGrad is the most robust explanation to adversarial attacks on inputs, yet fails the parameter randomization sanity checks proposed by Adebayo et al. [2018].

Sundararajan et al. [2017] propose Integrated Gradients, which take a path integral from a fixed reference baseline $\bar{x}$ to x (i.e. the method accumulates the gradients for each point on the straight line path between $\bar{x}$ and x). This method circumvents gradient saturation [Shrikumar et al., 2017], wherein many unimportant features have high gradients for all inputs. If a feature i has high importance in both $\bar{x}$ and x, Integrated Gradients will ignore feature i; thus, this method

will only accumulate the relevant, unsaturated gradients with respect to x. However, picking a proper reference baseline $\bar{x}$ is non-trivial. Sundararajan et al. [2017] suggest using a baseline where $f(\bar{x}) \approx 0$, while others have proposed taking the baseline to be the mean of the training data. Chang et al. [2019] note that the baseline can be learned using generative modeling. Instead of using a single baseline, Erion et al. [2019] take an expectation of Integrated Gradients over multiple random baselines, and Bhatt et al. [2021c] take an expectation of explanations where x's nearest neighbors are the baselines.

Shapley Values. A prominent class of perturbation-based methods is based on Shapley values from cooperative game theory [Shapley, 1953]. Shapley values are a way to distribute the gains from a cooperative game to its players. Shapley values denote the marginal contributions of a player to the payoff of a coalitional game. Let T be the number of players, and let $v : 2^T \rightarrow \mathbb{R}$ be the characteristic function, where $v(S)$ denotes the worth (contribution) of the players in $S \subseteq T$. The Shapley value of player i's contribution (averaging player i's marginal contributions to all possible subsets S) is given by: $\phi_i(v) = \frac{1}{|T|} \sum_{S \subseteq T \setminus \{i\}} \binom{T-1}{S}^{-1} (v(S \cup \{i\}) - v(S))$. Let $\Phi \in \mathbb{R}^T$ be a Shapley value contribution vector for all players in the game, where $\phi_i(v)$ is the i^{th} element of Φ. The highlight of Shapley values is that they enjoy axiomatic uniqueness guarantees [Shapley, 1953]. In the feature importance literature, we formulate a similar problem to where the game's payoff is the predictor's output $y = f(x)$, the players are the d features of x, and the ϕ_i values represent the contribution of x_i to the game $f(x)$. Note that $g(f,x)_i = \phi_i(v)$. Aas et al. [2021] define a characteristic function v where $v_x(S) = \mathbb{E}\left[f(z)|z = \bar{x}_{[\bar{x}_s = x_s]}\right]$. For a subset of indices $S \subseteq \{1,2,\ldots d\}$, $x_s = \{x_i, i \in S\}$ denotes a sub-vector of input features that partitions the input, $x = x_s \cup x_c$. $\bar{x}_{[\bar{x}_s = x_s]}$ denotes an input where the features in S are set to the observed values while the rest of the features remain the reference baseline: $\bar{x}_{[\bar{x}_s = x_s]} = x_s \cup \bar{x}_c$. When $|S| = d$, $\bar{x}_{[\bar{x}_s = x_s]} = x$. This characteristic function captures the expected model output given a subset of features that take on the value of some reference baseline. This function is used by Lundberg and Lee [2017], who show an equivalence between attribution-based Shapley values and surrogate methods, like LIME. Li et al. [2018] and Chen et al. [2019b] define a different characteristic function in terms of the expected number of bits required to encode the model's output based on the feature subset S.

Unfortunately, calculating the exact Shapley value is exponential in d, input dimensionality; however, the literature has proposed approximate methods using Monte Carlo approximation [Štrumbelj and Kononenko, 2014], weighted linear regression [Lundberg and Lee, 2017], centroid aggregation [Bhatt et al., 2019], and graph-structured factorization [Chen et al., 2019b]. Kumar et al. [2020] detail issues with using Shapley values as feature importance explanations. They note that the conditional characteristic function, like in Aas et al. [2021] and Frye et al.

[2020], requires assumptions on the relationship between features that may not have been captured by the model. For example, in the presence of redundant features, the conditional characteristic function can attribute influence to arbitrarily large subsets of features without discounting for statistically related features in the subset.

Connections to Global Explanations. While we only cover a sliver of the feature importance literature, other methods worth noting include Global Sensitivity Analysis [Cortez and Embrechts, 2013], Grad-CAM [Selvaraju et al., 2017] and MUSE [Lakkaraju et al., 2017]. Some have also connected saliency maps with the notion of attention mechanisms in neural networks [Cornia et al., 2018; Hendricks et al., 2018; Linsley et al., 2018; Wang and Shen, 2017]. Others view feature importance as part of rationale generation in natural language processing (NLP) [Ehsan et al., 2019; Zaidan et al., 2008]. See Guidotti et al. [2018] for a survey of techniques.

Evaluating Feature Importance With many candidate feature importance explanation functions, ML practitioners can find it difficult to pick which explanation function best captures how a model reaches a specific output for a given input. There has been little exploration into formalizing quantitative techniques for evaluating these model explanations. Recent work has created auxiliary tasks to test if attribution is assigned to relevant inputs [Yang and Kim, 2019], and has developed tools to verify if the features important to an explanation function are relevant to the model itself [Camburu et al., 2019]. Bhatt et al. [2021c] define multiple desirable properties of an attribution, including sensitivity [Melis and Jaakkola, 2018], faithfulness [Yeh et al., 2019], and complexity. They then propose algorithms to optimize an attribution for a specific property. In addition to establishing quantitative evaluation criteria, there has been work that qualitatively evaluates the utility of model explanations with human subjects [Lage et al., 2019]. Kulesza et al. [2013] find that sound explanations (i.e. those that truthfully represent the underlying model) help users develop an accurate mental model of a black-box model. Poursabzi-Sangdeh et al. [2021] conclude that larger explanations (i.e. increasing the number of features k for which importance scores are reported to users) make it hard for a user to simulate a black-box model accurately. Context-specific user studies may help uncover a set of design principles for exposing explanations to various stakeholders [Chen et al., 2022; Jeyakumar et al., 2020].

Training Point Importance

Whereas feature importance provides explanations in terms of individual features, training point importance (also called example-based explanations or sample importance) provides

explanations in terms of data points. We discuss two training importance methods: influence functions and prototypes.

Influential Training Samples. This explanation technique asks the question: Which data point in the training dataset $\mathcal{D}$ is most influential to the model's output $f(x)$ for a test point x? One approach would be to remove each training point and retrain, but this would be prohibitively slow. To answer this, statisticians have used measures like Cook's distance, which estimates the effect of deleting a data point on the model output [Cook, 1977]. However, such measures require an exhaustive search and hence do not scale well for larger datasets.

Translating these learnings to ML, Koh and Liang [2017] develop algorithms to approximate the effect of removing a training point on the loss at a test point by using influence functions. Suppose we have a weighted empirical risk minimization objective: $\sum_{i=1}^{n} w^{(i)} \ell(z^{(i)}, \theta)$, where $w^{(i)}$ is the weight given to training point $z^{(i)}$. When we give each training point equal weight (i.e. $w^{(i)} = \frac{1}{N}$), we obtain the parameters $\hat{\theta}$. We can measure the effect of perturbing a training point by re-weighting its contribution. Suppose we modify the weight of $z^{(i)}$ to $w^{(i)} = \frac{1}{n} + \varepsilon_i$. Let $\hat{\theta}_{\varepsilon_i}$ be the parameters obtained upon re-weighting. If we let $\varepsilon_i = -\frac{1}{n}$, this amounts to dropping $z^{(i)}$ from the training data. Influence functions from robust statistics can be repurposed to approximate $\hat{\theta}_{\varepsilon_i}$ after the re-weighting of $z^{(i)}$ [Hampel, 1974]. We require that the loss is convex and twice differentiable with respect to the model parameters.

Koh and Liang [2017] define the most influential training point z to a test point x as that which maximizes the following: $\mathcal{I}_{\text{up,loss}}(z,x) = -\nabla_\theta \ell\left(x, \hat{\theta}\right)^\top H_{\hat{\theta}}^{-1} \nabla_\theta \ell\left(z, \hat{\theta}\right)$. This quantity measures the effect of upweighting z on the loss at x. The goal here is to uncover which training point, when re-weighted, would have the largest effect (positive or negative) on the loss of a test point. Using a Hessian-vector products from Pearlmutter [1994], approximating $\mathcal{I}_{\text{up,loss}}$ is relatively cheap. Barshan et al. [2020] notice that the top-k influential points selected by Koh and Liang [2017]'s technique tend to be outliers or more generally "atypical." Therefore, Barshan et al. [2020] proposes constraining the objective: $\arg\max_{i\in\{1,\ldots,N\}} \max_{\varepsilon_i} |\ell(x, \hat{\theta}_{\varepsilon_i}) - \ell(x, \hat{\theta})|$ s.t. $s.t.\ \mathbb{E}_{z\sim\mathcal{D}}|\ell(z, \hat{\theta}_{\varepsilon_i}) - \ell(z, \hat{\theta})| \leq \delta$.

Essentially, this objective asks: "For some allowable change in expected loss, which training point should be reweighted to maximally change the loss at x?" The influential points found by this constraint are not global to the entire model like in Koh and Liang [2017] but are specific to the training point. Koh et al. [2018] find that this influence function setup can be used to perform data poisoning attacks, and Koh et al. [2019] study how to extend the above framework to dropping multiple points at once, dubbed group influence.

Khanna et al. [2019] use Fisher kernels combined with Sequential Bayesian Quadrature to select influential training points efficiently. They show that their method recovers the approach

from Koh and Liang [2017] if $\ell(\cdot,\cdot)$ is negative log-likelihood. Their method is also flexible enough to identify maximally influential subsets of points (i.e. group influence).

Specific to deep neural networks, Yeh et al. [2018] decompose the (pre-activation) prediction for a test point into a linear combination of activations for training points, using a modified representer theorem [Schölkopf et al., 2001]. The resulting training point weights correspond to helpful (large positive) and harmful (large negative) training points. In practice, influential training samples can be useful for displaying a model's trustworthiness: Zhou et al. [2019] find that, when a model is accurate, users use the model's outputs on influential training points to justify the model's output on a test point. Cai et al. [2019] develop an interface for pathologists to verify influential training points for cancer detection.

Prototypes. Prototypical points are representative points that summarize a dataset [Bien and Tibshirani, 2011]. Prototypes can be used to build more efficient classifiers (e.g. faster nearest neighbors), by selecting points that maximize intra-class coverage but minimize inter-class coverage [Kim et al., 2014]. Just as one may use feature selection to induce sparsity in the number of features a model uses, one can use prototype-finding algorithms to achieve sparsity in the number of samples used to build a model. Kaufmann [1987]'s proposal for k-medoids can be seen as a form of using prototypes for clustering. Prototypes are related to the *coreset* literature, which attempts to approximate the model learned on an entire dataset only using a subset of training points [Feldman and Langberg, 2011; Huggins et al., 2016; Jubran et al., 2019].

While prototypes can be helpful for learning better models, there is little work applying them as a form of explanation beyond the preliminary user study of Kim et al. [2014], in which users use prototypes and their labels to identify the label of a test point (recipe identification task). Gurumoorthy et al. [2019] propose ProtoDash, which speeds up the computation of Kim et al. [2016] and provides non-negative weights for prototype importance. As shown in Gurumoorthy et al. [2019]'s case study with a public health professional, prototypes might be a sensible way to provide summaries of model performance to stakeholders.

Counterfactual Explanations

Counterfactual explanations are points close to an input for which the decision of a classifier changes. Counterfactual explanations answer the question: "What is the minimal change to the input x required to change a model's output $f(x)$?" For example, for a person who was rejected for a loan by a model, a counterfactual explanation would possibly suggest: "Had your income been greater by $5000, the loan would have been granted, ceteris paribus." Counterfactual explanations not only explain individual predictions (and the model's decision boundary) but

also recommend a means for recourse [Ustun et al., 2019]. Note that the term counterfactual has a different meaning in the causality literature [Holland, 1986; Pearl, 2000]. We now discuss how to formulate counterfactual explanations.

As proposed in Wachter et al. [2017], a counterfactual explanation c, given an input x, a model f, and a distance metric d, is found by solving the optimization problem: $\min_{c \in \mathcal{X}} d(x,c)$ s.t. $f(x) \neq f(c)$. Adversarial examples follow a nearly identical formulation whereby they find a nearby point that deceives the model into a false prediction [Szegedy et al., 2013]; however, they contrast the intent of counterfactual explanations to interpret model predictions. To prevent c from simply devolving into an adversarial example, approaches to counterfactual explanation have imposed two types of constraints: plausibility and feasibility [Karimi et al., 2020]. Plausibility captures whether the counterfactual example is in-distribution (i.e. in a high-density region of the input) [Joshi et al., 2019; Karimi et al., 2020] – this constraint ensures the counterfactual is realistic. Feasibility ensures that the features that change can actually be actioned upon [Lash et al., 2017; Ustun et al., 2019] – this constraint ensures the counterfactual does not make impossible recommendations (e.g. change your race or decrease your age).

To solve for counterfactuals, some suggest using integer programming [Russell, 2019; Ustun et al., 2019], and others conduct latent space optimization [Joshi et al., 2019; Pawelczyk et al., 2020]. Feasibility requirements can be easily added as constraints in any optimization setup or can be folded into a cost function: Ustun et al. [2019] use percentile shift to capture the effort of making a particular change. Plausibility requirements are more difficult but better handled by latent space optimization. For example, Joshi et al. [2019] learn a variational autoencoder (VAE) on the training set. They then encode the original point and perform optimization in the latent space of the VAE to find an on-manifold counterfactual.

Counterfactual explanations do not come with an optimality guarantee: there may not be a recourse recommendation that is superior to all others. As a result, others have looked into finding and exposing diverse counterfactuals to stakeholders [Ley et al., 2022; Mothilal et al., 2020; Russell, 2019]. Barocas et al. [2020] explore the practical issues with deploying counterfactual explanations. They conclude that the recommendations made (also called the flipset) may be both feasible and plausible technically, but, in reality, an individual may not be able to make those changes due to social constraints not considered by the model. This idea is also echoed by Venkatasubramanian and Alfano [2020], who call for stakeholder involvement when deciding what is actionable (i.e. feasible). Barocas et al. [2020] also advise the development of counterfactual explanations alongside affected decision subjects to understand if the prescribed recommendation aligns with everyday mitigation strategies.

Other Explanation Forms

Beyond the three aforementioned local, post-hoc explanation types, the ML community has taken other approaches to generate explanations from models. Some have advocated for natural language explanations that provide rationales for model output [Ehsan et al., 2018]. Others have advocated for concept-based explanations: instead of finding semantically meaningless yet important pixels when classifying an image, Kim et al. [2018] and Ghorbani et al. [2019] label higher-level concepts (like stripes or wheels) as important in image classification settings. On another front, some have argued against generating explanations from black-box models. Rudin [2019] argues that we ought to use interpretable models that are explainable by design (e.g. decision trees [Breiman et al., 1984] or generalized additive models [Hastie and Tibshirani, 1990]). One middle ground between interpretable and black-box models is imposing desired explanations while learning a black-box model: Ross et al. [2017] regularize the input gradients for known irrelevant features, thus leading to models that are not only accurate but also have the correct feature importance. Adel et al. [2018] leverage side information to learn better representations in generative models, which also leads to better performance on discriminative tasks. Hancock et al. [2018] learn language models using annotators' natural language explanations to assign noisy labels to unlabelled data, leading to more accurate models. Hind et al. [2019] require an explanation for each training point and develop an algorithm to learn explanations and outputs simultaneously.

While a well-attested and important means of establishing appropriate levels of trust [Lai and Tan, 2019], explanations can easily backfire. Some explanations of AI systems, for example, appear to induce automation complacency [Bansal et al., 2021b; Buçinca et al., 2021; Dzindolet et al., 2003]. Feature importance explanations are particularly prone to misleading users in this regard [Carton et al., 2020; Kenny et al., 2021; Shen and Huang, 2020], though similar training point importance explanations have, admittedly, been shown to be conducive to HMT performance [Jeyakumar et al., 2020; van der Waa et al., 2021]. In the same vein, when explanations are provided before users are in a position to assess a situation for themselves, decision-makers may be led to anchor on the first data they receive, conditioning subsequent deliberation [Bansal et al., 2021b]. More perversely, "too much transparency can cause people to incorrectly follow a model when it makes a mistake, due to information overload" [Kaur et al., 2020]. On other occasions, poor or confusing explanations can lead to algorithm aversion [Kaur et al., 2020; Logg et al., 2019].

2.3.2 Uncertainty

The ML community has mostly considered explainability, which attempts to provide reasoning for a model's behavior to stakeholders, as a proxy for algorithmic transparency [Lucic et al., 2021]. In Bhatt et al. [2021a], we encourage researchers to study uncertainty as an alternative form of algorithmic transparency and practitioners to communicate uncertainty estimates to stakeholders. Uncertainty is crucial yet often overlooked in the context of ML-assisted, or automated, decision-making [Kochenderfer, 2015; Schum et al., 2014]. If well-calibrated and effectively communicated, uncertainty can help stakeholders understand when they should trust model predictions and can even help developers address fairness issues in their models [Tomsett et al., 2020; Zhang et al., 2020b].

Uncertainty refers to our lack of knowledge about some outcome [Spiegelhalter et al., 2011], and will be characterized differently depending on the task. In regression, uncertainty is often expressed in terms of error bars, also known as confidence intervals. For example, when predicting the number of crimes in a given city, we could report that the number of predicted crimes is 943 ± 10, where "± 10" represents a 95% *confidence interval* (capturing two standard deviations on either side of the mean estimate). The smaller the interval, the more certain the model. In classification, probabilities are often used to capture how confident a model is in a prediction. For example, a classification model may decide that a person is at high risk for developing diabetes given a prediction of 85% chance of diabetes. Broadly, uncertainty in algorithmic decision-making systems may stem from different sources and thus communicate different information to stakeholders [Gal, 2016; Hora, 1996].

Measuring Uncertainty

In ML, we use the tools of probability to reason about and quantify uncertainty. The Bayesian school of thought interprets probabilities as subjective degrees of belief in an outcome of interest occurring [MacKay, 2003]. For frequentists, probabilities reflect how often we would observe the outcome if we were to repeat our observation multiple times [Bland and Altman, 1998; Pek and Van Zandt, 2020]. Fortunately for end users, uncertainty from Bayesian and frequentist methods conveys similar information in practice [Xie and Singh, 2013], and can almost always be treated interchangeably in downstream tasks.

Metrics for Uncertainty. The metrics used to communicate uncertainty vary between research communities and application domains. Predictive distributions, shown in Figure 2.3, tell us about our models' degree of belief in every possible prediction. Despite containing a lot of information (prediction modes, tails, etc.), a full predictive distribution may not always be

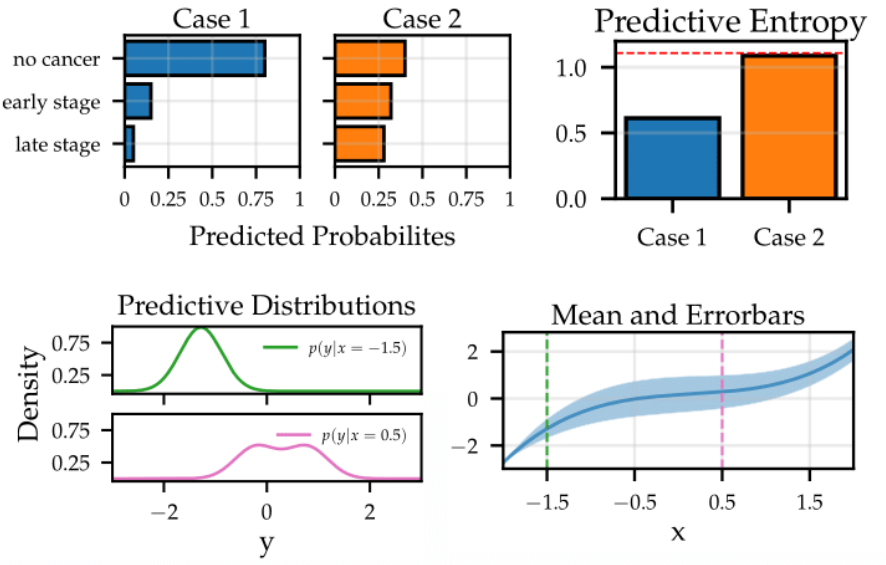


Fig. 2.3 Top: The same prediction (no cancer) is made in two hypothetical cancer diagnosis cases. However, our model is much more confident in the first. This is reflected in the predictive entropy for each case (the dashed red line denotes the maximum entropy for 3-way classification). Bottom: In a regression task, a predictive distribution contains rich information about our model's predictions (modes, tails, etc). We summarize predictive distributions with means and error bars (here standard deviations).

desirable. For this reason, **summary statistics** of the predictive distribution are often used to convey information about uncertainty. For classification, the predictive distribution is composed of class probabilities. These intuitively communicate our degree of belief in an outcome. On the other hand, predictive entropy decouples our predictions from their uncertainty, only telling us about the latter. For regression, the predictive distribution is often summarized by a predictive mean together with error bars (written $\pm\sigma$). These commonly reflect the standard deviation or some percentiles of the predictive distribution. We show common summary statistics for uncertainty in Table 2.2.

Table 2.2 Commonly used metrics for the quantification and communication of uncertainty.

	Full Information	Summary Statistics
Regression	Predictive Density	Predictive Variance, Percentile (or Quantile) Confidence Intervals
Classification	Predictive Probabilities	Predictive Entropy, Expected Entropy, Mutual Information, Variation Ratio

The Different Sources of Uncertainty. While there can be many sources of uncertainty [Van Der Bles et al., 2019], we focus on sources we can quantify in ML models: *aleatoric* (or indirect

uncertainty) and *epistemic* (or direct uncertainty) [Depeweg et al., 2018; Der Kiureghian and Ditlevsen, 2009; Gal, 2016].

Aleatoric uncertainty stems from noise, or class overlap, in our data [Der Kiureghian and Ditlevsen, 2009]. Noise in the data is a consequence of unaccounted-for factors that introduce variability in the inputs or targets. Examples of this could be background noise in a signal detection scenario or the imperfect reliability of a medical test in a cancer diagnosis scenario. Aleatoric uncertainty is also known as irreducible uncertainty: it cannot be decreased by observing more data. If we wish to reduce aleatoric uncertainty, we may need to leverage different sources of data (e.g. switching to a more reliable clinical test). In practice, most ML models account for aleatoric uncertainty through the specification of a noise model or likelihood function.

Epistemic uncertainty stems from a lack of knowledge about which function best explains the data we have observed [Der Kiureghian and Ditlevsen, 2009]. There are two reasons why epistemic uncertainty may arise. Consider a scenario in which we employ a very complex model relative to the amount of training data available. We will be unable to properly constrain our model's parameters. This means that, out of all the possible functions that our model can represent, we are unsure of which ones to choose. We can refer to uncertainty about a model's parameters as *model uncertainty*. We might also be uncertain of whether we picked the correct model class in the first place. Perhaps we are using a linear predictor but the phenomenon we are trying to predict is non-linear. We can refer to this as *model specification uncertainty* or *architecture uncertainty*. Epistemic uncertainty can be reduced by collecting more data in input regions where the training dataset was sparse. It is less common for ML models to capture epistemic uncertainty. Often, those that do are referred to as probabilistic models.

Given a probabilistic predictive model, aleatoric and epistemic uncertainties can be quantified separately. We depict them separately in Figure 2.4. Being aware of which regions of the input space present large aleatoric uncertainty can help practitioners identify issues in their data collection process. On the other hand, epistemic uncertainty tells us about which regions of input space we have yet to learn about. Thus, epistemic uncertainty is used to detect dataset shift [Ovadia et al., 2019] or adversarial inputs [Ye and Zhu, 2018]. It is also used to guide methods that require exploration like active learning [Houlsby et al., 2011], continual learning [Nguyen et al., 2018], Bayesian optimisation [Hernández-Lobato et al., 2014], and reinforcement learning [Janz et al., 2019].

Methods to Quantify Uncertainty. Most ML approaches involve a noise model, thus capturing aleatoric uncertainty. However, few are able to express epistemic uncertainty. When we say a method is able to quantify uncertainty, we are implicitly referring to those that capture both

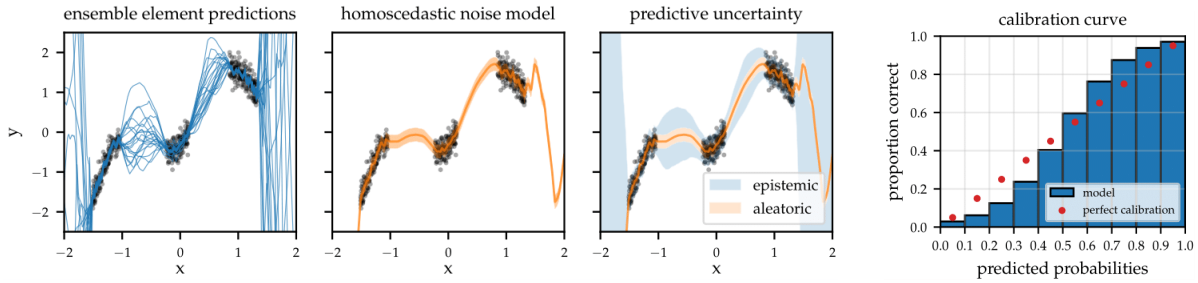


Fig. 2.4 Uncertainty quantification and evaluation. Left three plots: A 15-element deep ensemble provides an increased model (epistemic) uncertainty in data-sparse regions. A homoscedastic Gaussian noise model provides aleatoric uncertainty matching the noise in the data. Both combine to produce a predictive distribution. Right: calibration plot for a classification task. Each bar corresponds to a bin in which predictions are grouped. Their height corresponds to the proportion of correct predictions.

epistemic and aleatoric uncertainty. These methods can be broadly classified into Bayesian approaches [Ghahramani, 2015; Neal, 2012; Rasmussen and Williams, 2005] and Frequentist approaches [Breiman, 1996; Lakshminarayanan et al., 2017; Shalev-Shwartz and Ben-David, 2014]

Bayesian methods explicitly define a hypothesis space of plausible models *a priori* (before observing any data) and use deductive logic to update these priors given the observed data. In parametric models, like Bayesian Neural Networks (BNNs) [MacKay, 1992; Neal, 2012], this is often done by treating model parameters as random variables instead of single values and assigning them a prior distribution $p(\theta)$. Given some observed data $\mathcal{D}$, the conditional likelihood $p(y\,|\,x,\theta)$ tells us how well each parameter setting θ explains our observations. The likelihood is used to update the prior, yielding the posterior distribution over the parameters $p(\theta|\mathcal{D})$:

$$p(\theta|\mathcal{D}) = \frac{p(y\,|\,x,\theta)\,p(\theta)}{\int p(y\,|\,x,\theta)\,p(\theta)\,d\theta} \tag{2.1}$$

Prediction for a test point x^* is made via marginalization: all possible weight configurations are considered with each configuration's prediction being weighed by that parameter's posterior density. The disagreement among predictions from different plausible parameter settings induces model (epistemic) uncertainty. The predictive posterior distribution:

$$p(y|x^*) = \int p(y|x^*,\theta)p(\theta|\mathcal{D})\,d\theta \tag{2.2}$$

captures both epistemic and aleatoric uncertainty.

Frequentist methods do not specify a prior distribution over hypotheses. They exclusively consider how well the distribution over observations implied by each hypothesis matches

the data. Here, uncertainty stems from how we expect our chosen hypothesis to change if we were to repeatedly sample different sets of data. Perhaps the most salient Frequentist technique is ensembling [Dietterich, 2000; Hernández-Lobato et al., 2009]. This consists of training multiple models in different ways to obtain multiple plausible fits. At test time, the disagreement between ensemble elements' predictions yields model uncertainty, as shown in Figure 2.4. Currently, deep ensembles [Lakshminarayanan et al., 2017] are one of the best-performing uncertainty quantification approaches for neural networks [Ashukha et al., 2019] and retain calibration even under dataset shift [Ovadia et al., 2019]. Unfortunately, the computational cost of running multiple models at both train and test times makes ensembles one of the most expensive methods. There is a vast heterogeneous literature on frequentist uncertainty quantification [Alaa and van der Schaar, 2020; Liu et al., 2020; Schulam and Saria, 2019; Xiao et al., 2022].

Uncertainty Evaluation. Calibration is a form of quality assurance for uncertainty estimates. It is not enough to provide larger error bars when our model is more likely to make a mistake. Our predictive distribution must reflect the true distribution of our targets. Consider a cancer diagnosis scenario, where a system declines to make a prediction when uncertainty is above a threshold and a doctor is queried instead. Since a doctor's time is limited, we might design a system such that it only declines to make a prediction if it estimates there is a probability greater than 0.05 of the prediction being wrong. If our system is underconfident instead of well-calibrated, we would over-query the doctor in situations where the systems' predictions are correct. Overconfidence would result in taking action on unreliable predictions: delivering unnecessary treatment or abstaining from providing necessary treatment.

Calibration is orthogonal to accuracy. A model with a predictive distribution that matches the marginal distribution of the targets $p(y|x) = p(y) \forall x$ would be perfectly calibrated but would not provide any useful predictions. Thus, calibration is usually measured in tandem with accuracy through either one general fidelity metric (most often chosen to be a proper scoring rule [Gneiting and Raftery, 2007]) which subsumes both objectives, or two separate metrics. The most common metrics of the former category are *negative log-likelihood* (NLL) and *Brier score* [Brier, 1950]. Of the latter type, *Expected calibration error* (ECE) [Naeini et al., 2015], illustrated in Figure 2.4, is popularly used in classification scenarios.

Using Uncertainty

Uncertainty estimation can be used in many ways. It can be used to detect out-of-distribution (OOD) data [Lee et al., 2018]: Ovadia et al. [2019] study the effect of distribution shift, specifically covariate shift [Sugiyama et al., 2017], on uncertainty estimates. Similar analysis

can be done for uncertainty under label shift [Garg et al., 2020]. Uncertainty estimates can also be used to build selective classifiers (i.e. classification with abstention or rejection-option classification) [Bartlett and Wegkamp, 2008; Hellman, 1970; Nalisnick et al., 2019]. Such a model will abstain from providing a prediction for inputs that yield high uncertainty; in such cases, the model may defer prediction to an expert [Madras et al., 2018; Mozannar and Sontag, 2020; Wiener and El-Yaniv, 2011]. Instead of abstaining from prediction, Shafer and Vovk [2008] suggest predicting a set of class labels (instead of the most likely label) such that, with high probability, the true label is contained within the set: uncertainty can be used to determine how many labels are included in the set. These conformal models can be seen as treating uncertainty as algorithmic transparency: both help stakeholders understand what the model has not learned [Babbar et al., 2022; Vovk et al., 2005]. We now discuss the use of uncertainty for fairness, decision-making, and trust in automation.

Uncertainty and Fairness. An unfair ML model is one that exhibits unwanted bias towards or against one or more target classes [Dwork et al., 2012]. We discuss possible ways in which bias can appear as a consequence of unaccounted-for uncertainty.

Measurement bias, also known as feature noise, is a case of aleatoric uncertainty. It arises when one or more of the features in the data only represent a proxy for the features that would have ideally been measured, and can be mitigated by a properly specified nose model. We first describe the effect of noisy sensitive attributes. In contexts such as medical diagnosis, information on the race and ethnicity of patients may not be collected [Chen et al., 2019a]. The experimental results of Gupta et al. [2018] have shown that enforcing fairness constraints on a noisy sensitive attribute, without assumptions on the structure of the noise, is not guaranteed to lead to any improvement in the fairness properties of a model. The "mutually contaminated learning model" assumes the noise only depends on the true unobserved value of the sensitive attribute [Scott et al., 2013]. Here, measures of demographic parity and equalized odds computed on the observed data are equal to the true metrics up to a scaling factor, which is proportional to the value of the noise [Lamy et al., 2019]; if the noise rates are known, then the true metrics can be directly estimated. When information on the sensitive attribute is unavailable (e.g. information on gender was not collected) but can be predicted from an auxiliary dataset, disparity measures are generally unidentifiable [Chen et al., 2019a; Kallus et al., 2022].

Second, we describe the effect of noise in the targets (i.e. aleatoric uncertainty in the labels). This source of bias has attracted less attention in the fairness community despite being similarly pervasive. Obermeyer et al. [2019] found that when medical expenses were used as a proxy for illness, their algorithm severely underpredicted the prevalence of the illness for the population

of Black patients. Similarly to noise in the sensitive attribute, Blum and Stangl [2019] show that fairness constraints are guaranteed to improve the properties of a predictor only under appropriate assumptions on the label noise. Indeed, Fogliato et al. [2020] show that even a small amount of label noise can greatly impact the assessment of the fairness properties of the model.

Representation bias stems from how we define the population under study and how we sample our observations from said population [Suresh and Guttag, 2021]. Representation bias is epistemic in nature and thus may be reduced by collecting additional, potentially more diverse, data. Models trained in the presence of representation bias could exhibit unwanted bias towards an under-represented group. For example, the differential performance of gender classification systems across racial groups may be due to the under-representation of Black individuals in the sampled population [Buolamwini and Gebru, 2018]. Similarly, historical over-policing of some communities has unavoidable impacts on the sampling of the data used to train models for predictive policing [Ensign et al., 2018; Lum and Isaac, 2016].

To tackle representation bias, we must first detect that such bias exists. This can be done by leveraging uncertainty as a form of transparency. ML practitioners building probabilistic models could check for representation bias by ensuring that their validation datasets closely match the distribution expected at deployment. Their model presenting large epistemic uncertainty on this validation set would indicate the existence of representation bias in the training data. The practitioner could then identify which subgroups are unequally represented between their training and validation sets [Antorán et al., 2021]. Finally, they could leverage this knowledge to improve the data collection procedure. Unfortunately, sample size may still represent an issue. It is not always simple to collect more diverse data. Additionally, in many domains, the existing sample size may also not be large enough to assess the existence of biases [Ethayarajh, 2020]. We refer to Mehrabi et al. [2019] for a more detailed breakdown of the potential sources of representation bias.

Uncertainty and Decision-making. Depending on the context, AI systems can be used to support human decision-making to various degrees. Suppose a human decision-maker is tasked with making a decision while being aided by an ML model. Here, uncertainty plays a key role, as the decision-maker has to weigh how much they should trust the model's output. This question corresponds to prototypical tasks[3] studied in the Judgment and Decision-Making (JDM) literature (i.e. "action threshold decision" and "multi-option choices" [Fischhoff and Davis, 2014]). The ML literature has only just begun to examine how uncertainty estimates affect user interactions with ML models and task performance [Arshad et al., 2015; Zhang et al.,

[3] Social scientists often use the terms "risk" and "uncertainty" differently [Knight, 1921; Rakow, 2010].

2020b]. However, we highlight relevant conclusions from the JDM literature that pertain to using uncertainty estimates in decision-making. Prospect Theory suggests that uncertainty (or risk) is not considered independently but together with the expected outcome [Kahneman and Tversky, 2013; Tversky and Kahneman, 1992]. This relationship is non-linear and asymmetrical. A certain prediction of a large loss is often perceived more negatively than an uncertain prediction of a small loss. When risk is presented positively (e.g. a 95% chance of the model being correct), people tend to be risk-averse; when it is presented negatively, they are risk-seeking. As the stake of the outcome increases, tolerance for uncertainty seems to decrease at a superlinear rate [Van Der Bles et al., 2020]. Of course, differences among individuals' tolerances for uncertainty play an important role [Heath and Tversky, 1991; Miller, 1987; Politi et al., 2007; Sorrentino et al., 1988].

Assessment of risk, however, also depends on how uncertainty is communicated and perceived. Both lay people and experts rely on mental shortcuts, or heuristics, to *interpret* uncertainty [Tversky and Kahneman, 1974]. This could lead to biased appraisals of uncertainty even if model outputs are well-calibrated. Stowers et al. [2017] find that communicating and visualizing uncertainty information to operators of unmanned vehicles helped improve human-AI team performance; however, they note that their findings may not generalize to other tasks and contexts. To our knowledge, the empirical understanding of how decision-makers make use of aleatoric versus epistemic uncertainty is limited [Kruschke, 2014]. Furthermore, the JDM literature has mostly focused on discrete outcomes. There is not a good understanding of how people perceive uncertainty over continuous outcomes (e.g. error bars). We judge these to be important gaps in the human-machine interaction literature.

Uncertainty and Trust Formation. While trust could be implicit in a decision to rely on a model's prediction, the communication of a model's uncertainty can also affect decision-maker's general trust in an ML model. At a high level, communicating uncertainty is a form of algorithmic transparency that can display a model's trustworthiness and repair a decision-maker's trust. The relationship between uncertainty estimates and trust in automation is a relatively unexplored idea. To anticipate how uncertainty estimates, and ways to communicate them, could impact stakeholder trust, we highlight existing "process models" on how people develop trust. Rooted in information-processing and decision-making theories [Chaiken, 1999; Kahneman, 2011; Petty and Cacioppo, 1986], "process models" differentiate between an analytic (or systematic) process of trust formation and an affective (or heuristic) process of trust formation [Lee and See, 2004; Metzger and Flanagin, 2013; Sundar, 2008]. The former involves a rational evaluation of a trustee's characteristics; systematic trust formation in an ML model could be facilitated by providing detailed probabilistic uncertainty estimates. The

latter process relies on feelings or heuristics to form a quick judgment to trust or not; when lacking either the technical ability or motivation to perform an analytic evaluation, people rely more on the affective or heuristic route [Petty and Cacioppo, 1986; Sundar and Kim, 2019]. For some users, the mere presence of uncertainty information could signal that the engineers are transparent and sincere, enhancing trust via good faith [Hovland et al., 1953]. For others, uncertainty could invoke negative heuristics [Van Der Bles et al., 2019].

Prior work suggests that the style in which uncertainty estimates are communicated is highly relevant to how these are perceived [Parasuraman and Miller, 2004]. The goal of presenting uncertainty estimates to stakeholders should support forming *appropriate* trust, rather than blindly enhancing trust. A well-measured and well-communicated uncertainty estimate should not only facilitate the *calibration* of overall trust on a system but also the *resolution* of trust [Cohen et al., 1998; Lee and See, 2004]. The latter refers to how precisely the judgment of trust could differentiate types of model capabilities in decision-making. Thus, leveraging uncertainty as a form of transparency can be helpful for trust formation. However, how uncertainty estimates are processed for trust formation and what affective impact uncertainty invokes remain open questions and merit future research.

Communicating Uncertainty

Many application domains involve communicating uncertainty estimates to the general public to help them make decisions (e.g. weather forecasting, transit information delivery [Kay et al., 2016], medical diagnosis and interventions [Politi et al., 2007]). However, even well-calibrated uncertainty estimates could be perceived inaccurately by people because (a) they have varying levels of understanding about probability and statistics, and (b) human perception of uncertainty quantities is often biased by decision-making heuristics.

Issues in Understanding Uncertainty. One key issue in communicating uncertainty is that the intended audience may not have the numeracy skills required to interpret uncertainty correctly. In a survey conducted in 2010 on statistical numeracy across the US and Germany, Galesic and Garcia-Retamero [2010] found that many people do not understand relatively simple statements that involve statistics concepts. For example, 20% of the German and US participants could not say "Which of the following numbers represents the biggest risk of getting a disease: 1%, 5%, or 10%," and almost 30% could not answer whether 1 in 10, 1 in 100, or 1 in 1000 represents the largest risk. Another study found that people's numeracy skills significantly affect how well they comprehend risks [Zikmund-Fisher et al., 2007]. Besides numeracy skills, research shows that humans in general suffer from a variety of cognitive biases, some of which hinder our understanding of uncertainty [Kahneman, 2011; Reyna and

Brainerd, 2008; Spiegelhalter, 2017]. One is called ratio bias, which refers to the phenomenon where people sometimes believe a ratio with a big numerator is larger than an equivalent ratio with a small numerator. For example, people may see 10/100 as a larger odds of having breast cancer than 1/10. This same phenomenon is sometimes manifested as an underweighting of the denominator, e.g. believing 9/11 is smaller than 10/13. This is also called denominator neglect.

In addition to ratio biases, people's perception of probabilities is distorted in that they tend to underweight high probabilities while overweighting low probabilities. This distortion prevents people from making optimal decisions. Zhang and Maloney [2012] showed that when people are asked to estimate probabilities or frequencies of events based on memory or visual observations, their estimates are distorted in a way that follows a log-odds transformation of the true probabilities. Research also found that this bias occurs when people are asked to make decisions under risk and that their decisions imply such distortions [Tversky and Kahneman, 1992; Zhang et al., 2015]. Therefore, when communicating probabilities, we need to be aware that people's perception of high risks may be lower than the actual risk, while that of low risks may be higher than actual. A different kind of cognitive bias that impacts people's perception of uncertainty is framing [Kahneman, 2011]. Framing has to do with how information is contextualized. Typically, people prefer options with positive framing (e.g. 80% likelihood of surviving breast cancer) than an equivalent option with negative framing (e.g. 20% likelihood of dying from breast cancer). This bias has an effect on how people perceive uncertainty information. A remedy for this bias is to always describe the uncertainty of both positive and negative outcomes, rather than relying on the audience to infer what was left out of the description.

Communication Methods. Choosing the right communication methods can address some of the above issues. Hullman et al. [2018] review methods for evaluating the success of uncertainty visualization. Cosmides and Tooby [1996] find that people may be good at interpreting well-calibrated probabilities in JDM tasks. More recent work suggests that humans are better at interpreting log-odds [Zhang and Maloney, 2012; Zhang et al., 2020a]. Spiegelhalter et al. [2011] suggest a framework for the public communication and visualization of uncertainty; however, Hullman [2019] finds that, while researchers acknowledge that uncertainty is important to quantify and communicate, many do not visualize (or even present) uncertainty when communicating scientific facts or model outputs to end users. Nonetheless, conveying uncertainty for specific predictions has shown initial promise in building accurate HMTs [Babbar et al., 2022; Bansal et al., 2021b].

Van Der Bles et al. [2019] categorize the different ways of expressing uncertainty into nine groups with increasing precision, from explicitly denying that uncertainty exists to displaying

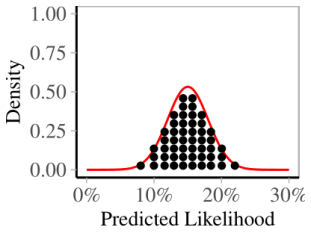
a full probability distribution. While high-precision communication methods help experts understand the full scale of the uncertainty of ML models, low-precision methods can suffice for lay people, who may have potentially low numeracy skills. We focus on the pros and cons of the four methods of communicating uncertainty: 1) describing the degree of uncertainty using a predefined categorization, 2) describing a numerical range, 3) showing a summary of a distribution, and 4) showing a full probability distribution. The first two methods can be communicated verbally, while the last two often require visualizations.

Predefined categorizations of uncertainty reduce the cognitive effort needed to comprehend uncertainty estimates, and therefore are particularly likely to help people with low numeracy skills [Peters et al., 2007]. Uncertainty ratings are also frequently used by financial agencies to communicate the overall risks associated with an investment instrument [Dionisio et al., 2007]. The main drawback of communicating uncertainty via predefined categories is that the audience, especially non-experts, might not be aware of or even misinterpret the threshold criteria of the categories. Many studies have shown that although individuals have internally consistent interpretations of words for describing probabilities (e.g. likely, probably), these interpretations can vary substantially from one person to another [Budescu and Wallsten, 1985; Clark, 1990; Lichtenstein and Newman, 1967]. Recently, Budescu et al. [2012] investigated how the general public interprets the uncertainty information in a climate change report published by the Intergovernmental Panel on Climate Change (IPCC). They find that people generally interpreted the IPCC's categorical description of probabilities as less likely than the IPCC intended: people took the word "very likely" as indicating a probability of around 60%, whereas the IPCC's guideline specifies that it indicates a greater than 90% probability. To avoid such misinterpretation, categorical and numerical forms of uncertainty can be communicated when possible.

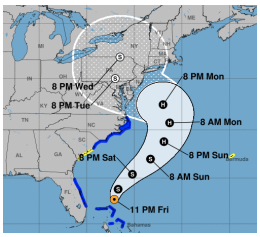
Numerical ranges are more precise than categorical scales in communicating uncertainty; however, they are harder to understand for people with low numeracy and can induce ratio biases. To overcome the adverse effect of denominator neglect, it is important to present ratios with the same denominator so that they can be compared with just the numerator [Spiegelhalter et al., 2011]. Denominators that are powers of 10 are preferred since they are easier to compute. There are no conclusive findings on whether frequency format (*"Out of every 100 patients, 2 are likely to be misdiagnosed"*) is easier to understand than ratios/percentages (*"Out of every 100 customers, 2% are likely to be misdiagnosed"*): people do seem to perceive risk probabilities represented in the frequency format as showing higher risk than those represented in the percentage format [Reyna and Brainerd, 2008]. Therefore, it is helpful to use a consistent format to represent probabilities. If the audience underestimates risk levels, the frequency format may be preferred.



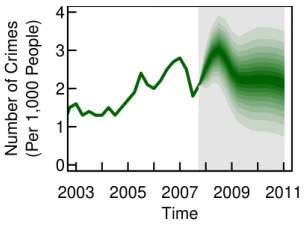
(a) An example icon array chart, used to represent the chance of a patient having breast cancer.



(b) Quantile dot plot, which can be used to show the uncertainty around the predicted likelihood of a patient having breast cancer.



(c) Cone of uncertainty, showing the uncertainty around the predicted path of the center of a hurricane. Taken from [Ruginski et al., 2016].



(d) Fanchart, which can be used to show how the predicted crime rate of a city evolves over time.

Fig. 2.5 Examples of uncertainty visualizations.

Summary statistics can capture the uncertainty of a continuous variable or model prediction that has the form of a distribution. Some commonly reported summary statistics include mean, median, confidence intervals, standard deviation, and quartiles [Alexander McFarlane Mood and Boes, 1974]. These statistics are often depicted graphically as error bars and boxplots for univariate data, and two-dimensional error bars and bagplots for bivariate data [Rousseeuw et al., 1999]. Error bars only have a few graphical elements and are hence relatively easy to interpret. However, since they have represented a range of different statistics in the past, they are ambiguous if presented without explicit labeling [Wilke, 2019]. Uncertainty estimates represented with graphics have several advantages over verbal communication, such as attracting and holding the audience's attention, revealing trends or patterns in the data, and evoking mental mathematical operations [Lipkus and Hollands, 1999]. Commonly used visualizations include pie charts, bar charts, and icon arrays (Figure 2.5a).

Posterior predictive distributions, shown in their entirety, may be helpful when presenting uncertainty about individual model predictions. This can avoid over-emphasis of the within-bar range and allow more granular visual inferences. Popular visualizations of distributions are histograms, density plots, and violin plots,[4] which seem to be hard for an uninitiated audience to grasp. They are often mistaken as bivariate case-value plots in which the lines or bars denote values instead of frequencies [Boels et al., 2019]. Kay et al. [2016] develop quantile dot plots to convey distributions (see Figure 2.5b for an example). These plots use stacked dots, where each dot represents a group of cases, to approximate the data frequency at particular values. This method translates the abstract concept of a probability distribution into a set of discrete outcomes, which are more familiar to people who have not been trained in statistics. Kay et al. [2016]'s study showed that people could more accurately derive probability estimates from

[4] Hintze and Nelson [1998] show multiple density plots side-by-side

quantile dot plots than from density plots. Kruschke [2014] defines a method that attempts to simultaneously convey aleatoric and epistemic uncertainty in a single plot by showing the predictive densities resulting from various samples of the posterior distribution; however, the efficacy of such plots in practice is still unknown.

The above methods are designed to communicate uncertainty around a single quantity, so they need to be extended for visualizing uncertainty around a range of predictions, such as those in time-series forecasting. The simplest form of such visualization is a quantile plot, which uses lines to connect predictions at equal quantiles of the uncertainty distribution across the output range. When used in time-series forecasting, these plots are called cone-of-uncertainty plots (see Figure 2.5c), in which the cone enlarges over time, indicating increasing uncertainty. Gradient plots, or fan charts (see Figure 2.5d) in the context of time series forecasting, can be used to show more granular changes in uncertainty, but they require extra visual encoding that may not be easily understood by the viewer. Hypothetical outcome plots (HOP) [Hofman et al., 2020; Hullman et al., 2015; Kale et al., 2020] can be used to show uncertainty estimates over a range of predictions by showing each model's predictions in an animation frame [Wilke, 2019]. The plethora of methods for communicating uncertainty makes it an appealing form of algorithmic transparency to provide to decision-makers.

2.3.3 Case for Opacity

While transparency may be preferable in many contexts, transparency may not always be desirable: opacity might be a better route. In the presence of perfect assistance, a person does not expect other people to be transparent in their decision-making [Zerilli et al., 2019b]; similarly, if models perform well over time, transparency may not be necessary in the face of high reliability. Opacity may suffice if one fears that added transparency will lead to privacy violations. Transparency can be seen as the antithesis of privacy, which is an intentional lack of information. Milli et al. [2019b] shows that a malicious actor can reconstruct a potentially proprietary, black-box model via gradient-based attributions from the model; surprisingly, they perform this without knowledge of the model class or the data distribution. The desire to provide explanations may be at odds with the intention to keep the model hidden: the cost of transparency here may justify opacity.

Instead of complete opacity, Weller [2019] discusses the upside of *selective transparency*. First, they show that providing full transparency (i.e. faithful information) about a system to every autonomous agent in a multi-agent economy can lead to worse outcomes for all agents. Then, they show that selective transparency (only providing information to select agents) can lead every agent to perform better. This raises questions at the intersection of transparency and fairness: how does one decide which agents will be privy to additional information?

Recall our definition of transparency as any information provided about an AI system beyond its outputs. While transparency is often beneficial, we briefly note several potential dangers [Weller, 2019]. Just as model outputs can be wrong, so too can additional transparency information. Since this information might be relied upon in making decisions, incorrect transparency can cause harm. Incorrect transparency might be unintentional [Ehsan and Riedl, 2021], or could be deliberately deceptive [Dimanov et al., 2020; Heo et al., 2019; Slack et al., 2020a]. Even correct information might be misleading. In human communication, we often leave certain points unsaid, assuming our counterpart has background knowledge of the context. This creates the potential for information to be misleading if it is not carefully presented [Gigerenzer et al., 2010]. Hence, algorithmic transparency ideally should satisfy properties similar to those required in a courtroom. However, this is often not easy to measure or achieve in practice, hence is an important area of research.

2.4 On Human-Machine Teams

We widen our discussion to the broader field of human-machine teams. As mentioned before, when a decision-maker leverages an AI system or ML model in their decision-making process, we refer to the tandem of the two as a HMT. HMTs have been shown to outperform either teammate alone in galaxy identification [Kamar et al., 2012], breast cancer detection [Wang et al., 2016], and other applications [Horvitz and Paek, 2007]. We include a brief recap of Section 2.1. Automation bias refers to the phenomena when humans do not consider contradictory information in the presence of output from a model [Cummings, 2004; Mosier and Skitka, 1996; Parasuraman and Riley, 1997]: that is, decision-makers may wrongfully overtrust models. Recall that Logg et al. [2019] call this *algorithm appreciation*, as decision-makers can tend to follow a model's recommendation blindly. Bansal et al. [2021b] find that decision-makers follow a model's recommendations, even when the model is wrong and the decision-maker is shown an explanation for the incorrect recommendation. However, Dietvorst et al. [2015] notice that decision-makers avoid model recommendations after seeing the model err and tend to select another human as a teammate thereafter: they term this phenomenon *algorithm aversion*. Evidently, in complementary settings, decision-makers may not display *algorithmic vigilance* towards the model [Lee and See, 2004]. If designed and deployed thoughtfully, transparency may be helpful in displaying the trustworthiness of AI systems in HMTs [Glikson and Woolley, 2020]. Making transparency, perhaps in the form of explanations or calibrated uncertainty estimates, accessible and available to decision-makers could improve team performance [Amershi et al., 2019].

We provide background on two broad areas of HMTs. First, we describe how stakeholders (including domain experts and decision-makers) can provide feedback to update the AI system, which is assisting the stakeholder in the HMT. Specifically, we review existing methods for incorporating the provided feedback into an actionable update to the AI system itself. Second, we discuss methods for instantiating selective transparency [Weller, 2019], wherein decision-makers are only provided with access to the AI system (and forms of algorithmic transparency) sometimes. In some sense, we first discuss how to provide human control over AI systems in HMTs (e.g. editing AI systems to be better teammates) and then discuss how to lace a layer of AI control underneath human control (e.g. learning when to show AI system output based on the system's uncertainty and decision-maker expertise).

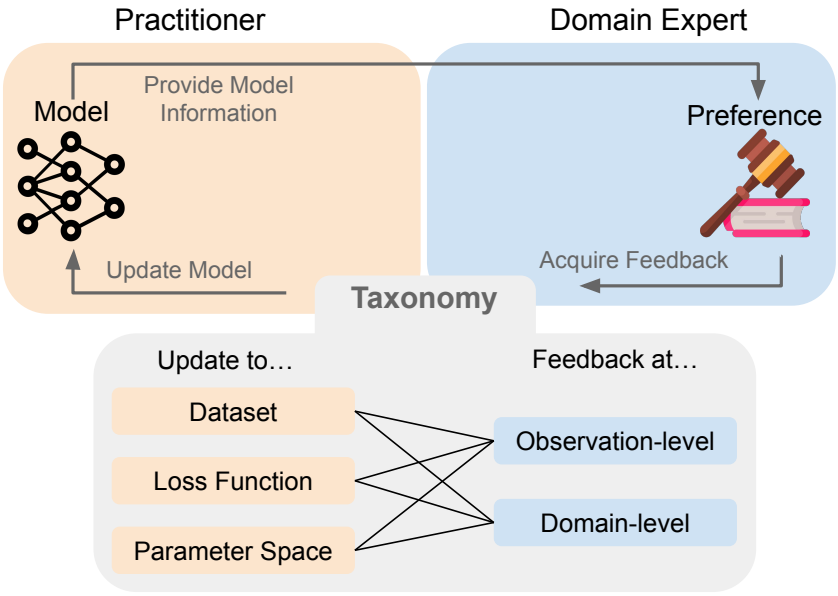


Fig. 2.6 To incorporate an expert's preferences to improve a model, practitioners must turn non-technical, domain expert preferences into usable model updates. In Chen et al. [2023a], we propose a feedback-update taxonomy that focuses on the ways in which expert feedback can be translated into model updates. This taxonomy has two axes, expert feedback, and model updates, forming six categories of feedback-update interaction. We map existing work onto each category and use our taxonomy to motivate improving the interaction between practitioners and domain experts.

2.4.1 Incorporating Expert Feedback into Model Updates

Before deploying an ML model in high-stakes use cases, *practitioners*, who are responsible for developing and maintaining models, may solicit and incorporate feedback from *experts* [Amershi et al., 2014; Cui et al., 2021; Fails and Olsen Jr, 2003; Nushi et al., 2018]. Prior work

has largely focused on incorporating feedback from technical experts (herein ML engineers, data scientists, etc.) into models [Adebayo et al., 2020; Li et al., 2021; Liu et al., 2017; Ross et al., 2017; Song et al., 2013; Wang et al., 2020]. The feedback of technical experts might be immediately actionable, as likely few communication barriers exist between technical experts and practitioners. In contrast, the relationship between a practitioner and non-technical expert (herein doctors, lawyers, elected officials, policymakers, social workers, etc.), as illustrated in Figure 2.6, is more complex [Bhatt et al., 2020b; Chen et al., 2022]. Upon seeing transparency information about the model, the expert provides feedback based on their preference to practitioners, who can then update the model. For example, Wu et al. [2019] create an interactive error analysis tool that allows experts to explore where NLP models err: this tool may be a form of algorithmic transparency, but how can an expert then provide actionable feedback to improve the model based on its errors? There has been insufficient consideration on how to incorporate feedback from non-technical experts [Bhatt et al., 2020a; Kulynych et al., 2020] into models.

Table 2.3 This feedback-update taxonomy illustrates the diverse ways practitioners can convert expert feedback, which generally comes via domain- or observation-level feedback, into model updates, which are either dataset, loss function, or parameter space changes that entail changes to the dataset, loss function, or parameter space respectively. Each cell corresponds to a part of the Feedback-Update Taxonomy.

	Dataset Update	Loss Function Update	Parameter Space Update
Domain Feedback	*Dataset modification* Augmentation [Dao et al., 2019] Preprocessing [Feldman et al., 2015; Iyengar, 2002] *Data generation from constraint* Fairness [Xu et al., 2018] Interpretability [Lee et al., 2020; Plumb et al., 2022] *Weak supervision* Using unlabeled data [Arazo et al., 2020; Carmon et al., 2019; Ratner et al., 2017] Checking synthetic data [Ryazanov et al., 2021]	*Constraint specification* Fairness [Coston et al., 2021; Zafar et al., 2017b] Interpretability [Lakkaraju et al., 2016; Plumb et al., 2020] Resource constraints [Frankle and Carbin, 2018; Lin et al., 2020]	*Model editing* Rules [Yang et al., 2019] Weights [Wang et al., 2022] *Model selection* Prior update [Lage et al., 2018] Complexity [Rudin, 2019]
Observation Feedback	*Active data collection* Adding data [Ghai et al., 2021; Wan et al., 2020] Relabeling data [Kaushik et al., 2019] Reweighting data [Bourtoule et al., 2021; Hiranandani et al., 2021; Zhao et al., 2019] Collecting expert labels [Peterson et al., 2019] *Passive observation* [Irvin et al., 2019; Laidlaw and Russell, 2021; Swartz et al., 2006]	*Constraint elicitation* Metric learning [Cheng et al., 2021; Hiranandani et al., 2019] Human representations [Hilgard et al., 2021; Santurkar et al., 2021] *Collecting contextual information* Generative factors [Adel et al., 2018] Concept representations [Koh et al., 2020] Explanations [Hind et al., 2019] Feature attributions [Tseng et al., 2020; Weinberger et al., 2020]	*Feature modification* Add/remove features [Bakker et al., 2021; Noriega-Campero et al., 2019] Engineering features [Lou et al., 2013]

One now classic example is a classifier that is supposed to differentiate dogs from wolves and that picks up on a spurious pattern in the data: the presence of snow correlates with wolves and the presence of grass with dogs [Kelly et al., 2019; Ribeiro et al., 2016]. Instead of learning a semantically meaningful representation that distinguishes between dogs and wolves, the classifier fits to the images' background. When such a model uses the wrong features, it is possible to solicit human feedback and tweak the model parameters to fit the true signal (here the dog versus the wolf) [Lee et al., 2020]. Interactive ML comes to the rescue to involve healthcare practitioners directly in a model correction phase, thus building a more effective HMT [Amershi et al., 2019; Nushi et al., 2017]. If a deployed AI system errs, there ought to be an opportunity for a decision-maker to intervene and correct the system's behavior. To bridge

this gap, we examine model updates available to the practitioner and the types of feedback that non-technical experts might provide.

We clarify the mechanisms available to turn feedback into updates and then devise a taxonomy along two axes: (a) levels of expert feedback, and (b) types of model updates. Along the first axis, expert feedback may come as *domain-level feedback*, which captures high-level conceptual feedback that the practitioner must translate into updates, or *observation-level feedback*, which captures how the model should behave on a few, specific data points [Armstrong and Mindermann, 2018; Davies, 2005; Hertwig and Erev, 2009; Swartz et al., 2006]. Along the other axis, we consider the updates a practitioner can make to a supervised learning objective, where feedback typically changes the dataset, the loss function, or the parameter space. Supervised learning covers a broad range of model classes commonly deployed in practice, ranging from vision transformers, large language models, and impactful application areas, like medical diagnostics [Irvin et al., 2019] and criminal justice [Pierson et al., 2020]. Other objectives, which may include reinforcement learning [Christiano et al., 2017; Cui et al., 2021] or unsupervised learning [Coden et al., 2017; Guimaraes Goecks, 2020], are out-of-scope for this thesis.

Feedback-Update Taxonomy

One role of practitioners is to convert non-technical expert feedback into a model update. We note the important case that sometimes valuable expert feedback might be received to say that using *any* model is not appropriate for the setting at hand. While such a concern must be taken seriously and considered carefully with relevant stakeholders, we do not discuss this case further here. We describe the diverse ways that expert feedback can lead to model updates through our *feedback-update taxonomy* (Table 2.3). While experts are often involved prior to training an initial model [Irvin et al., 2019], we focus on the iterative feedback process after a model has been trained. One piece of feedback could be used to alter multiple parts of the objective (e.g. change the dataset and loss) but each update should be considered individually. We elaborate on the two axes of our taxonomy and flesh out each category.

Levels of Domain Expert Feedback Once an expert has observed information about the model, practitioners may ask for feedback to improve the model's behavior in two general ways [Hertwig and Erev, 2009; Wulff et al., 2018].

- **Domain feedback.** It may be natural for non-technical experts to provide high-level conceptual feedback [Hertwig and Erev, 2009]. The expert could provide explicit feedback over a set of good models [Fisher et al., 2019; Semenova and Rudin, 2019]

or suggest data pre-processing to reduce discrimination [Calmon et al., 2017; Feldman et al., 2015; Hajian and Domingo-Ferrer, 2012].

- **Observation feedback.** It may also be possible to learn by observing expert behavior [Armstrong and Mindermann, 2018; Davies, 2005; Swartz et al., 2006]. For example, practitioners can use observations to approximate a property of interest (e.g. fairness) [Hiranandani et al., 2020], or can collect *contextual information*, where every data point is accompanied by auxiliary information that can be used during learning (e.g. feature attributions [Weinberger et al., 2020], style factors [Adel et al., 2018], semantically meaningful concepts [Koh et al., 2020]).

These two types of feedback form one axis of our feedback-update taxonomy in Table 2.3. While these two forms of feedback may be non-exhaustive, they capture a wide variety of mechanisms for non-technical experts to influence the development of models [Hertwig and Erev, 2009]. Neither type requires non-technical experts to have knowledge about the model or training process itself. A radiologist could provide domain-level feedback about X-ray scans via high-level information about the region of interest in each X-ray or pre-processing suggestions for every scan. Examples of observation-level feedback that the same radiologist could provide on X-ray scans include bounding boxes of where in each X-ray a specific fracture lies or additional electronic health record data to co-reference a given scan. The role of practitioners may be expanding, as a practitioner may need to decide, as part of model updating, whether to treat the collected feedback as domain- or observation-level feedback.

We consider other forms of feedback to be out-of-scope for this work because they are less intuitive to elicit from a non-technical domain expert [Schoeffer et al., 2021; Wang and Gupta, 2020]. This includes changing the learning algorithm (e.g. in differential privacy communities [Dwork et al., 2014; Song et al., 2013]), selecting hyperparameters (e.g. in AutoML research [Li et al., 2017, 2021]), and specifying the order of data points given to a learning algorithm (e.g. in machine teaching literature [Liu et al., 2017; Simard et al., 2017]).

Types of Model Updates In the supervised learning setting, a practitioner generally minimizes a loss function on a provided dataset to learn the parameters of a model. Once experts have provided feedback, practitioners can leverage expert input to improve the model in multiple ways: updating the dataset, the loss function, or the parameter space. These update types form the other axis of Table 2.3.

- **Dataset updates.** Feedback can be incorporated by changing the dataset $\mathcal{D}$ that the model is trained on.

- **Loss function updates.** Feedback can also be incorporated by adding a constraint to the optimization objective. This manifests as a change in the loss function ℓ.
- **Parameter space updates.** Finally, feedback can also be provided on the parameters of the model itself, which reflect a change in the parameter space $\theta \in \Theta$.

For example, a public official may ask that as the input feature `population` increases, the likelihood of a project proposal getting `funded` should increase; this implies monotonicity between an outcome and input feature. Practitioners can incorporate this feedback in various ways: the practitioner can *update the dataset* by adding or removing appropriate data points, *update the loss function* by adding a regularizer that penalizes the model for not satisfying this condition, or *update the parameter space* by optimizing over a subspace of parameters that satisfy this condition. While these update types may seem straightforward, it is unclear how to identify the new dataset, loss function, or model parameter space: this involves transforming domain expert feedback into one of these three general updates. After the practitioner incorporates expert feedback, the updated model should ideally reflect the expert's preferences better than the original model. In Chen et al. [2023a], we flesh out this *conversion* from feedback to update.

2.4.2 User Control and Dynamic Allocation

Because algorithmic transparency ultimately satisfies a need for human decision-makers to be in control in HMTs, an effective alternative strategy may be to allow decision-makers a degree of latitude over whether to accept an AI system's outputs at face value [Kleinberg et al., 2018]. For instance, provided decision-makers can modify or override outputs, decision-makers are apparently willing to take an ML model seriously even after seeing the system make occasional mistakes. What is more, the precise degree of control seems to be irrelevant: the ability to modify an output even slightly may be sufficient to induce appropriate reliance [Dietvorst et al., 2018]. Control can be exercised in various ways, including through cognitive "forcing" functions that prompt decision-makers to request additional information in the form of algorithmic transparency should they desire them [Buçinca et al., 2021]. In some sense, we want to avoid inundating decision-makers with unnecessary information, especially in time-critical settings [Brown et al., 1998; Horvitz and Barry, 1995].

The static versus dynamic nature of allocation is also important because tasks in which control flexibly shifts between human and machine in accordance with user needs are better at sustaining operator vigilance [Chavaillaz et al., 2016; Fogliato et al., 2022]: this motivates **personalizing** when an AI system's output is shown as a form of decision support to decision-makers in HMTs. HMTs in which such allocation is dynamic can be further divided between

those in which the allocation is *adaptable*, where decision-makers dictate the allocation, and those in which the allocation is *adaptive*, where the allocation is automated [Chavaillaz et al., 2016; Sauer et al., 2012; Shahaf and Horvitz, 2010; Zhang et al., 2012]. Allocation can then proceed along several lines, but perhaps the most intuitive is along lines of difficulty. A human is likely to find some input examples easy that an AI system will find hard and others hard that an AI system will find easy.[5] Generally, human trust in AI is higher when tasks involve objective calculation—to the point of trusting the AI even after seeing it make mistakes [Dijkstra, 1999]—and lower when tasks involve social and emotional intelligence [Lee, 2018]. Both adaptive and adaptable forms of allocation can help achieve an optimal division of labor from the point of view of difficulty. For example, under adaptable allocation, decision-makers can reserve all the examples they consider easy for themselves and delegate the remaining ones to a machine (i.e. request an AI system's output for those examples). Under adaptive allocation, an AI system could vary the difficulty of the examples it reserved for the decision-maker, so that it referred both moderately difficult as well as easy tasks to the decision-maker, in an attempt to keep them vigilant (e.g. via so-called "catch trials"). In one study, adaptable allocation was found to have a marginal advantage over adaptive allocation, and (unsurprisingly) happens to be easier to design [Sauer et al., 2012]. However, adaptive systems may be able to leverage uncertainty information in ways that are more effective than adaptable systems (catch trials for one) [De et al., 2021].

A large body of research in aviation demonstrates the potential advantages of adaptive allocation [Parasuraman et al., 1996]. Air traffic controllers manage aircraft flow and intervene if aircraft separation is too low [Metzger and Parasuraman, 2005]. The controller is provided with an automated decision aid to handle multiple tasks. In these scenarios, an adaptive allocation strategy is usually preferred [Parasuraman et al., 1996]. One advantage of adaptive strategies is that they can accommodate the use of "catch trials." The point of a catch trial is to ensure the controller is alert and situationally-aware [Davies and Parasuraman, 1982; Gugerty and Tirre, 2000]. They may take the form of randomly generated system errors to "catch out" the decision-maker, or (more commonly) abstentions in which the system declines to recommend a course of action in a specific instance, leaving the decision-maker to fall back on their own skills.

When both the decision-maker and AI system find an example easy, it likely does not matter who provides a response (although decision fatigue is an ever-present risk [Chaparro et al., 1999; Warm et al., 2018]). More interesting are cases in which both the AI system and decision-maker struggle with an example. One approach would be to select an agent at random. If the decision-maker is selected, then the decision-maker must make a decision without the

[5] From the AI system's perspective, difficulty can be understood in terms of predictive uncertainty.

AI system's recommendation; if the AI system is selected, then the decision-maker would be shown the AI system's recommendation before making a decision (i.e. the decision-maker would have a choice whether to accept the AI system's recommendation).

Broadly, we can learn an allocation strategy that considers many forms of decision support to provide to decision-makers: such a strategy would be a mechanism for varying the control given to decision-makers. We call such a strategy a *decision support policy* where decision-makers have multiple options for support, and one form of support is selected to be shown to decision-maker based on the instance (i.e. input). In some sense, the most basic action would be to `DEFER` entirely to decision-maker judgment. Another would be to always `SHOW` the prediction for an ML model. We may decide to `DEFER` or `SHOW` selectively based on the decision-maker's expertise and the model's capabilities. Most papers on human-machine collaboration have considered clever ways for AI-based decision support systems to abstain from prediction on specific inputs [Cortes et al., 2016, 2018], learning deferral functions based on multiple experts [Keswani et al., 2021; Vovk, 1998], or teaching decision-makers when to rely [Mozannar et al., 2022b]. There are also a number of papers from the HCI literature (see survey by [Lai et al., 2023]) that evaluate the setting where a decision-maker is shown either nothing or a model prediction a *static* decision support policy (e.g. always showing the ML model prediction or always showing the model for a fixed class). These static allocation policies can be seen as conducting adaptable allocation, where a priori some examples are deferred to decision-makers to complete without any support. We review various forms of support that can be provided to decision-makers: this will be important for the personalized decision support policies we propose in Chapter 5, where we consider learning an adaptive allocation strategy for when to show a form of decision support (e.g. model prediction or model explanation).

- `DEFER`: This form of support is equivalent to **no support**. Decision-makers are asked to make a decision without any assistance. The ML community has studied how to identify when to defer to a subset of examples to humans based on decision-maker strengths [Bansal et al., 2021a; Wilder et al., 2020] and/or model failures [Chow, 1957; Geifman and El-Yaniv, 2017]. The premise of such an action would be to allow decision-makers to be unaided and squarely place decision liability on the individual.

- `SHOW`: In many settings, ML models are trained to do prediction tasks similar to the decision-making task prescribed to the decision-maker, or in the case of foundation models [Bommasani et al., 2021], ML models can be adapted to aid decision-making, even if the task was not specifically prescribed at train-time [Yang et al., 2023b]. In essence, a model prediction, or associated generation (e.g. a code snippet [Mozannar et al., 2022a]) would be shown

to aid an individual decision-maker. This has been shown to help improve decision-maker performance. The following are variations of showing a model prediction to a decision-maker.

  - `CONFORMAL`: For classification tasks, only displaying the most likely label may not lead to good performance due to various reasons, including uncertainty in the modeling procedure [Bondi et al., 2022; Vovk et al., 2005]; however, such uncertainty can be communicated to decision-maker by showing a prediction set to experts [Babbar et al., 2022]. Such a prediction set might be generated using conformal prediction, which guarantees the true label lies in the set with a user-specified error tolerance [Bates et al., 2021; Straitouri et al., 2022].
  - `CONFIDENCE`: Instead of translating the uncertainty into a prediction set (or interval), one could simply show the confidence or uncertainty of the prediction, which may manifest as displaying probabilities, standard errors, or entropies [Spiegelhalter, 2017]. The visualization mechanism used for displaying confidence may alter the decision-maker's performance [Hullman et al., 2018; Zhang et al., 2020b].
  - `EXPLAIN`: In addition to providing a model prediction, many have considered showing an explanation of model behavior, examples of which include feature attribution [Buçinca et al., 2020; Ribeiro et al., 2016], sample importance [Jeyakumar et al., 2020; Kim et al., 2014], counterfactual explanations [Antorán et al., 2021; Ustun et al., 2019], and natural language rationales [Camburu et al., 2018; Ehsan et al., 2018]. Displaying such explanations to end users has had mixed results on how decision-making performance is affected [Chen et al., 2022; Lai et al., 2022]. Worryingly, in many settings, showing some types of explanations may lead to over-reliance on models by giving the perception of competence [Buçinca et al., 2020; Chen et al., 2023b; Zerilli et al., 2022].

- `CONSENSUS`: One can depict forms of support that are independent of any model, for instance, presenting the belief of one or more decision-makers. Belief distributions can be constructed by pooling over many different humans' "votes" for what a label ought to be [Beyer et al., 2020; Gordon et al., 2021, 2022; Peterson et al., 2019; Uma et al., 2020, 2022] or by eliciting distributions over the likely label directly from each individual decision-maker [Collins et al., 2022, 2023]. These consensus distributions permit the expression of uncertainty *without any model*. However, the elicitation of this form of support may be costly and humans may be fallible in the information they provide (e.g. due to direct labeling errors [Augustin et al., 2017; Dawid and Skene, 1979; Wei et al., 2022; Whitehill et al., 2009] or miscalibrated confidence [Collins et al., 2023; Lichtenstein et al., 1977; O'Hagan et al., 2006; Tversky and Kahneman, 1996]).

- `ADDITIONAL`: While much of this thesis focused on support that provides decision-makers with label information (e.g. recommendation for the task at hand), decision support may also entail acquiring or displaying additional contextual information (e.g. new features [Bakker et al., 2021]) or requesting previously unseen features, for instance, through additional medical diagnostics [Harrell et al., 1982; Mylonakis et al., 2000]. This flavor of support can be varied structurally, ranging from the results of a search query [Nakano et al., 2021] to hierarchical information like exposing the subsidiary ownership structure for multinational corporations [Erramilli, 1996]. In terms of the cost, some pieces of additional information may require additional cost or certification if pertaining to sensitive attributes.

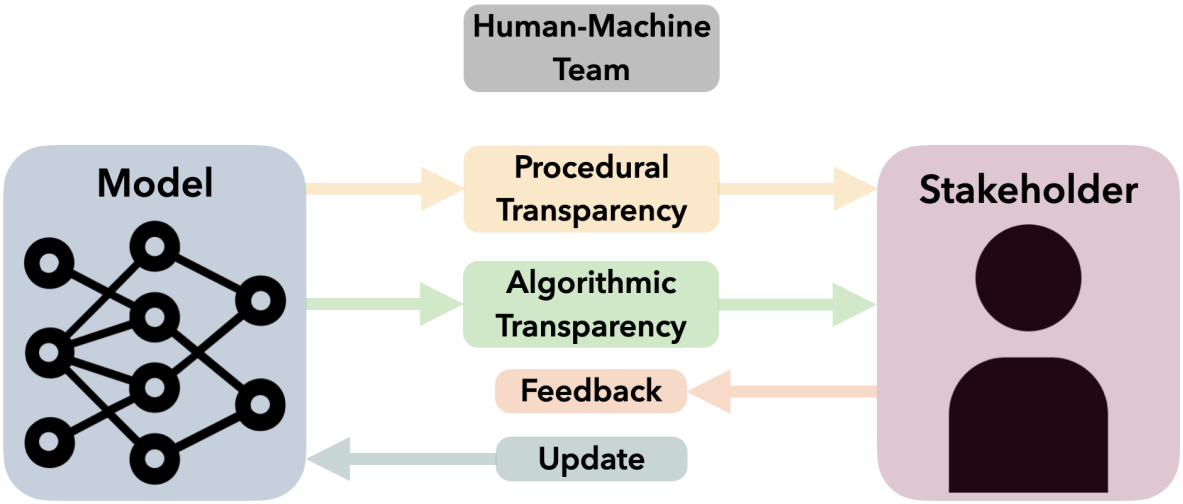


Fig. 2.7 Full summary of the Background. We not only consider the forms of transparency provided to stakeholders, focusing on algorithmic transparency via explainability and uncertainty, but also discuss how stakeholder feedback can be used to update models and learn decision support policies.

Chapter 3

Algorithmic Transparency in Practice

Explainability, a popular form of algorithmic transparency, offers the potential to provide stakeholders with helpful insights into model behavior, yet there is little understanding of how organizations use these methods in practice. In this chapter, we first discuss an interview study we conducted in 2019 exploring how organizations view and use explainability [Bhatt et al., 2020b]. We find that the majority of deployments are not for end users but rather for data scientists and machine learning engineers, who use explainability to debug the model. There is thus a gap between explainability in practice and the goal of external transparency since current approaches to model explanation primarily cater to internal stakeholders. Providing algorithmic transparency requires careful consideration of the needs of stakeholders, including end users, regulators, and domain experts. To help address this need, we convened a closed-door, day-long workshop [Bhatt et al., 2020a] between academics, industry experts, legal scholars, and policymakers. Takeaways from this workshop include developing a shared language around explainability and understanding the shortcomings of using explainability in the service of external transparency goals.

3.1 Overview

With growing interest in "peeking under the hood" of machine learning (ML) models and in providing explanations to human users, explainability (or explainable ML/AI) has become an important subfield of ML. Despite burgeoning literature, there has been little work characterizing how explainability has been deployed by organizations in the real world. With help from the Partnership on AI, we explore how organizations have deployed local explainability techniques so that we can observe which techniques work best in practice, report on the shortcomings of existing techniques, and recommend paths for future research. We focus specifically on local explainability techniques since they explain individual predictions and have been particularly

studied in the ML literature [Gilpin et al., 2018] as we discussed in Chapter 2.3.1. Our interview studies synthesize interviews with roughly fifty people from approximately thirty organizations. At the time of our study, we, to the best of our knowledge, were the first to study how explainability techniques are used by organizations that deploy ML models in their workflows. A contemporary study was done by Hong et al. [2020] on twenty-two individuals, and others have looked into the effect of explanation on data scientists debugging ML models [Kaur et al., 2020; Poursabzi-Sangdeh et al., 2021]. After our interview study, we hosted a multistakeholder convening to explore how to bridge the explainability research and practice gap. This chapter summarizes the insights gained from the interview study [Bhatt et al., 2020b] and from the summary report of the workshop of experts [Bhatt et al., 2020a]. The chapter is organized as follows:

1. We discuss the methodology of our interview study in Section 3.2.
2. We summarize the findings of our study in Section 3.3.
3. We detail use cases for local explainability in Section 3.4.
4. We describe our multi-stakeholder convening in Section 3.5.
5. We discuss findings from our convening in Section 3.6.

3.2 Interview Study Methodology

In the spirit of Holstein et al. [2019], we study how industry practitioners view and use explainable ML. Specifically, we study how particular organizations deploy explainability tools, including who consumes the explanation and how it is evaluated by the intended stakeholder. We conduct two sets of interviews: Group 1 consisted of twenty data scientists who are not currently using explainable ML and hope to leverage it. Group 2, the crux of our study, consisted of people who were actually using explainability in practice. This group spanned roughly thirty people across approximately twenty different organizations, both for-profit and non-profit. Most of these organizations were members of the Partnership on AI, which is a global multi-stakeholder non-profit established to study and formulate best practices for AI to benefit society.

With each individual, we held a thirty-minute to two-hour semi-structured interview to understand the state of explainability in their organization, their motivation for using explanations, and the benefits and shortcomings of the methods used. Some organizations asked to stay anonymous, not to be referred to explicitly in the prose, or not to be included in the

acknowledgments. Of the people we spoke with in Group 2, around one-third represented non-profit organizations (academics, civil societies, and think tanks), while the rest worked for for-profit organizations (corporations, industrial research labs, and start-ups). Around one-third of the interviewees were executives, around half were research scientists or engineers, and the remainder were professors at academic institutions, who commented on the consulting they had done with industry leaders to commercialize their research.

For Group 1, we, with the help of Fiddler Labs, led a set of around twenty interviews to assess explainability needs across various organizations in the technology and financial services sectors. We specifically focused on teams that do not currently employ explainability tools. These semi-structured, hour-long interviews included, but were not limited to, the following questions:

- What are your ML use cases?
- What is your current model development workflow?
- What are your pain points in deploying ML models?
- Would explainability help address those pain points?

The questions we asked Group 2 included, but were not limited to, the following:

- Does your organization use ML model explanations?
- What type of explanations have you used (e.g. feature-based, sample-based, counterfactual, natural language)?
- Who is the audience for the model explanation (e.g. research scientists, product managers, domain experts, end users)?
- In what context have you deployed the explanations (e.g. informing the development process, informing human decision-makers about the model, informing the end user on actions taken based on the model's output)?
- How does your organization decide when and where to use model explanations?

For both groups, we conducted an inductive thematic analysis to (i) identify emergent needs for explainability with Group 1 and (ii) understand the challenges, if any, with using explainability in Group 2. During this data-driven analysis, we coded notes from each meeting in which explainability approaches were discussed, which use cases for explanations were mentioned, and what "pain point" was explained. These codes were then synthesized across

all interviews. Our summarized challenges, or "pain points," from Group 1 are presented in Section 3.3.1. In Section 3.3.2, we report mini-case studies of how explanations are used by various organizations.

3.3 Summary of Findings

3.3.1 The Need for Explainability

In Group 1, we spoke with data scientists from organizations that do not currently use explainability techniques. We now describe the "pain points" in building and using ML models that these data scientists discussed. We summarize the issues they raised and speculate how explainability would address their "pain points." Our discussions with these data scientists can be summarized into the following:

- **Model debugging**: Most data scientists struggle with debugging poor model performance. They wish to identify why the model performs poorly on certain inputs, and also to identify regions of the input space with below-average performance: this goes beyond typical local explanations à la Chapter 2.3.1. They seek guidance on how to engineer new features, drop redundant features, and gather more data to improve model performance. For instance, one data scientist said: "If I have 60 features, maybe it's equally effective if I just have 5 features." Dealing with feature interactions was also a concern, as the data scientist continued, "Feature A will impact feature B, [since] feature A might negatively affect feature B—how do I attribute [importance in the presence of] correlations?" Others mentioned explainability as a debugging solution, helping to "narrow down where things are broken."

- **Model monitoring**: Several individuals worry about drift in the feature and prediction distributions after deployment. Ideally, they would like to be alerted when there is a significant test-time drift relative to the training distribution [Amodei et al., 2016; Pinto et al., 2019]. One organization would like explanations for how drift in input distributions impacts model outcomes and feature importance explanations: "We can compute how much each feature is drifting, but we want to cross-reference [this] with which features are impacting the model a lot."

- **Model management**: Organizations that refresh their models and training data frequently worry about model and data version control: models can get outdated due to large changes in input distribution. One person mentioned: "We do not have a good record of what

has changed over time. That introduces the danger of model degradation that we do not catch."

- **Model transparency**: Organizations that deploy models to make decisions that directly affect end users seek explanations for model predictions. The explanations are meant to increase transparency and help compliance with forthcoming regulations. Data scientists believe that explanations can help communicate predictions to a broader external audience of business teams and customers. One company stressed the need to "show your work" to provide reasons for underwriting decisions to customers, and another company needed explanations to respond to customer complaints.

- **Model audit**: In financial organizations, due to regulatory requirements, all deployed ML models must go through an internal audit. Data scientists building these models need to have them reviewed by internal risk and legal teams. One of the goals of this audit is to conduct various kinds of tests provided by regulations like SR 11-7, which requires ongoing model validation [Board of Governors of the Federal Reserve System, 2011]. An effective model validation framework should include (i) evaluation of the conceptual soundness of the model, (ii) ongoing monitoring including benchmarking [Raji and Yang, 2019], and (iii) outcomes analysis including backward compatibility [Bansal et al., 2019b]. Explainability is viewed as a tool for evaluating the soundness of the model on select data points. Financial institutions would like to conduct sensitivity analyses, checking the impact of small changes to inputs on model outputs. Unexpectedly large changes in outputs can indicate an unstable model.

Hong et al. [2020] find a similar breakdown for when stakeholders want explainability: they split up explainability needs into model conception, building, and deployment.

3.3.2 Trends in Using Explainability

Here we synthesize the results from Group 2. In Table 3.1, we aggregate some of the explainability use cases that we received from different organizations in Group 2. For each use case, we define the domain of use (i.e. the industry in which the model is deployed), the purpose of the model, the explainability technique used, the stakeholder consuming the explanation, and how the explanation is evaluated. Evaluation criteria denote how the organization compares the success of various explanation functions for the chosen technique (e.g. after selecting feature importance as the technique, an organization can compare LIME [Ribeiro et al., 2016] and SHAP [Lundberg and Lee, 2017] explanations via the faithfulness criterion [Yeh et al., 2019]). *In our study, feature importance was the most common explainability technique, and*

Domain	Model Purpose	Explainability Technique	Stakeholders	Evaluation Criteria
Finance	Loan Repayment	Feature Importance	Loan Officers	Completeness [Lundberg and Lee, 2017]
Insurance	Risk Assessment	Feature Importance	Risk Analysts	Completeness [Lundberg and Lee, 2017]
Content Moderation	Malicious Reviews	Feature Importance	Content Moderators	Completeness [Lundberg and Lee, 2017]
Finance	Cash Distribution	Feature Importance	ML Engineers	Sensitivity [Yeh et al., 2019]
Facial Recognition	Smile Detection	Feature Importance	ML Engineers	Faithfulness [Ancona et al., 2018]
Content Moderation	Sentiment Analysis	Feature Importance	QA ML Engineers	ℓ_2 norm
Healthcare	Medicare access	Counterfactual Explanations	ML Engineers	Normalized ℓ_1 norm
Content Moderation	Object Detection	Adversarial Perturbation	QA ML Engineers	ℓ_2 norm

Table 3.1 Summary of select deployed local explainability use cases

Shapley values were the most common type of feature importance explanation. Note that this reflects practice at the time of the interviews; subsequently, a broader set of explainability approaches, including wider use of actionable counterfactuals [Karimi et al., 2020] as well as more user-friendly interfaces [Buçinca et al., 2021], have emerged. The most common stakeholders were ML engineers (or research scientists), followed by domain experts (e.g. loan officers and content moderators). While most stakeholders said they would use an explanation if it were tailored to them, we find that executives are invested, not in the explanation itself, but in ensuring explanations are provided to the right stakeholders within their organization. Section 3.4 provides details on how these techniques were used at organizations in Group 2.

Stakeholders

Most organizations in Group 2 deploy explainability atop their existing ML workflow for one of the following stakeholders:

1. **Executives**: These individuals deem explainability necessary to realize an organization's AI principles. One research scientist felt that "explainability was strongly advised and marketed by higher-ups," and sometimes explainability simply became a checkbox. Executives rarely provide guidance on how to operationalize on explainability or on why the organization needed such algorithmic transparency.

2. **ML Engineers**: These individuals (including data scientists and researchers) train ML models at their organization and use explainability techniques to understand how the trained model works: do the most important features, most similar samples, and nearest training point(s) in the opposite class make sense? Using explainability to debug what the model has learned, this group of individuals was the most common explanation consumers in our study.

3. **End Users**: This is, arguably, the most intuitive consumer of an explanation. The end user is the person consuming the model output and likely making a decision based on

it. Explainability shows the end user why the model behaved the way it did, which is important for showing a model's trustworthiness by providing greater transparency.

4. **Other Stakeholders**: There are many other stakeholders for explainability. One such group is regulators, who may mandate that algorithmic decision-making systems provide explanations to affected populations or to the regulators themselves. It is important that this group understands how explanations are deployed based on existing research, which techniques are feasible, and how techniques can be manipulated to obtain the desired explanation from any model. Another group is domain experts, who are often tasked with auditing model behavior and ensuring it aligns with expert intuition. For many organizations, minimizing the divergence between an expert's intuition and the model's explanation is key to successfully implementing explainability.

Overwhelmingly, we found that local explanations are consumed by ML engineers and data scientists to audit models before deployment rather than to provide explanations to end users. Our interviews reveal factors that prevent organizations from showing explanations to those affected by decisions made based on model outputs.

Key Takeaways

We summarize some key takeaways from Group 2 that shed light on the reasons for the limited use of explainability. Organizations generally consider the judgments of domain experts to be the implicit ground truth for explanations. Since explanations produced by current techniques often deviate from the understanding of domain experts, some organizations still use human experts to evaluate the explanation before it is presented to stakeholders. Part of this deviation stems from the potential for explanations to reflect spurious correlations [Plumb et al., 2022]. As a result, organizations find explainability techniques useful for ML engineers to identify and reconcile inconsistencies between the model's explanations and domain experts' intuition. Recent work has called this into question as explanations might be misleading to end users [Dimanov et al., 2020; Heo et al., 2019; Slack et al., 2020a].

There are technical limitations that make it difficult for organizations to show end users explanations in real-time. The non-convexity of some models makes some explanations (e.g. providing the most influential data points) hard to compute quickly. Some work has attempted to speed up such computations [Guo et al., 2021] but it has yet to be seen if this will affect practical implementations. Moreover, finding actionable counterfactual data points that are feasible in the real world and plausible with respect to the input data manifold is nontrivial [Poyiadzi et al., 2020]. Providing explanations can also raise privacy concerns due to the risk of model

inversion [Milli et al., 2019b] or the leak of private information in the training data [Shokri et al., 2021].

More broadly, organizations lack frameworks for capturing the objectives of an explanation and understanding what type of explanations are reasonable to expect, especially from very complex models. For example, large gradients, representing the direction of maximal variation of output, do not necessarily "explain" anything to stakeholders. Gradient-based explanations provide an interpretation of how the model behaves upon an infinitesimal perturbation not necessarily a feasible one [Hooker and Mentch, 2019], but do not "explain" if the model captures the underlying causal mechanism.

3.4 Deep Dive: Deploying Local Explainability

In this section, we present mini-case studies of how local explainability techniques are used at various organizations (Group 2). Each of these organizations has deployed an ML model. They hope to explain a data point and its corresponding model prediction using an explanation function. Again, we deliberately decide to focus on the more popularly deployed local explainability instead of global explainability. None of our interviewees reported using global explainability, though some studied it in research settings. For each local explainability technique, we discuss organizations' use cases and then report takeaways.

Shapley Values in Practice

Feature importance was by far the most popular technique in our study. It is used across many domains, including finance, healthcare, facial recognition, and content moderation. Organization A works with financial institutions and helps explain models for credit risk analysis. To integrate into the existing ML workflow of these institutions, Organization A proceeds as follows. They let data scientists train a model to the desired accuracy. Organization A focuses mostly on models trained on tabular data, though they are beginning to venture into unstructured data (i.e. language and images). During model validation, risk analysts conduct stress tests before deploying the model to loan officers and other decision-makers. After decision-makers vet the model outputs as a quick check and decide whether or not to override the model output, Organization A generates Shapley value explanations. Before launching the model, risk analysts are asked to review the Shapley value explanations to ensure that the model exhibits expected behavior (i.e. the model uses the same features that a human would for the same task). Notably, the customer support team at these institutions can also use these explanations to provide individuals with information about what went into the decision-making

process for their loan approval or cash distribution decision. They are shown the percentage contribution to the model output (the positive ℓ_1 norm of the Shapley value explanation along with the sign of contribution). An explanation then reads as "55% of the decision was decided by age, which positively correlated with the prediction."

When comparing Shapley value explanations to other feature importance techniques, Organization A found that in practice LIME explanations give unexpected explanations that do not align with human intuition [Ribeiro et al., 2016]. Recent work shows that the fragility of LIME explanations can be traced to the sampling variance when explaining a singular data point and to the explanation sensitivity to sample size and sampling proximity [Zhang et al., 2019]. When asked why they chose Shapley value explanations over other feature importance explanations, Organization A said the axiomatic nature of Shapley values made them appealing and "rigorous;" the well-maintained code for SHAP also made approximate Shapley values easy to use. While the exact computation of Shapley values satisfies specific axioms, it is not clear if these axiomatic guarantees hold when running approximations as noted in [Aas et al., 2021; Chen et al., 2019b; Lundberg and Lee, 2017]. For language models, Organization A uses Integrated Gradients (related to Shapley Values by Sundararajan et al. [2017]) to flag malicious reviews and moderate content at the aforementioned institutions. This information can be displayed to the decision-maker (here, the hired content moderator) since they can now see which word was most important to flag the content.

Though decision-makers have access to the feature importance explanations, end users are still not shown these explanations as reasoning for model output. Organization A aspires to eventually provide Shapley explanations to end users. However, Organization A is proceeding with caution, since multiple works have shown that models can be manipulated to generate misleading explanations [Dimanov et al., 2020; Heo et al., 2019; Slack et al., 2020a]. Going forward, Organization A intends to use a global variant of the Shapley value explanations by exposing how Shapley value explanations work on average for data points of a particular predicted class (e.g. on average someone who was denied a loan had their age matter most for the prediction). This global explanation would help risk analysts get a birds-eye view of how a model behaves and whether it aligns with their expectations.

Takeaway: Shapley values are rigorously motivated, and approximations are simple to deploy. Feature importance is not shown to end users but is exposed to internal stakeholders. Looping in those who make decisions based on model outputs into the development process can ensure models provide the required level of transparency.

Counterfactual Explanations in Healthcare

Organization B uses a faster version of the formulation in [Sharma et al., 2020] to find counterfactual explanations for healthcare applications. For example, when people apply for Medicare, Organization B aims to flag errors in a person's application and to provide explanations on how to correct such errors via counterfactual explanations. The original formulation makes use of a slower genetic algorithm, so they optimized the counterfactual explanation generation process. Their application takes in any black-box model and an associated evaluation dataset and returns a robustness score, fairness measures, and counterfactual explanations, all from a single algorithm.

This approach has several advantages: it applies to black-box models, works for any input data type, and generates multiple explanations in a single run of the algorithm. However, there are shortcomings that Organization B is addressing. One challenge of counterfactual explanations is that the counterfactual might not be feasible. Organization B addresses this by using the training data to guide the counterfactual generation process and by providing a user interface that allows domain experts to specify constraints to guide the generation of actionable counterfactuals. The flexibility of the counterfactual explanations comes with a common drawback: there is no guarantee of the uniqueness of the explanation since there may be multiple routes for recourse [Russell, 2019]. Through the creation of a deployed solution based on this method, the organization realized that some clients also desire an overall assessment of the explainability of the model. This led to the development of numeric scores for model explainability, fairness, and robustness, all of which can be used to compare models alongside standard performance measures. Often a model may have a slightly lower accuracy but is more robust, fair, or explainable; hence, the less accurate model may be preferred as a more reliable candidate for deployment.

Takeaway: Organizations are interested in counterfactual explanation solutions since the underlying method is flexible and such explanations are easy for end users to understand. It is not clear exactly what should be optimized for when generating a counterfactual or how to do it efficiently. Still, approximate solutions may suffice in practice.

Influence Functions in Insurance

For over half of the organizations, influence functions have been the tool of choice for explaining which training points are influential to the model's output for a specific input [Koh and Liang, 2017], though only one organization actually deployed the technique. Organization C uses influence functions to explain risk models in the insurance industry. They hope to identify which customers might see an increase in their premiums based on their driving history in the

past. The organization hopes to divulge to the end user how the premiums for drivers similar to them are priced. In other words, they hope to identify the influential training data points to understand which past drivers had the greatest influence on the prediction for the observed driver. Unfortunately, Organization C has struggled to provide this information to end users since the Hessian computation of Koh and Liang [2017] has made doing so impractical since the latency is high. More pressingly, even when Organization C lets the influence function procedure run, they find that many influential data points are simply outliers that are important for all drivers since those anomalous drivers are far out of distribution. As a result, instead of identifying which drivers are most similar to a given driver, the influential sample identifies drivers that are very different from any driver (i.e. outliers [Barshan et al., 2020; Bhatt et al., 2021b]). While this could be useful for outlier detection, it prevents explanations from being used in practice.

Takeaway: Influence functions can be intractable for large datasets. A significant effort is needed to improve these methods to make them easy to deploy in practice. Influence functions can be sensitive to outliers in the data, such that they might be more useful for outlier detection than for providing end users with explanations.

Heatmaps in Transportation

Organization D looks to detect facial expressions from video feeds of users driving. They hope to use explainability to identify the actions a user performs while driving. Organization D has tried visualizing attributions to obtain heatmaps by backpropagating gradients to regions of interest Adel Bargal et al. [2018]; Zhang et al. [2018]. Specifically, they use these probabilistic Winner-Take-All techniques, variants of gradient-based feature importance techniques [Sundararajan et al., 2017], to localize the region of importance in input space for a particular classification task. For example, when detecting a smile, they expect the mouth of the driver to be important. Though none of these techniques have been deployed to drivers, ML engineers at Organization D found these techniques useful for qualitative review. On small datasets, engineers can figure out which scenarios have false positives (videos falsely detected to contain smiles) and why. However, while trying to understand why the model erred by analyzing similarities in false positives, they have struggled to aggregate local explainability techniques across multiple videos. They are able to qualitatively evaluate a sequence of heatmaps for one video, but doing so across 100M frames simultaneously is far more difficult. Paraphrasing the VP of AI at Organization D, aggregating saliency maps across videos is moot and contains little information. Unlike aggregating Shapley values for tabular data as done at Organization A, taking an expectation over heatmaps does not work, since aggregating pixel attributions is meaningless. One option Organization D discussed would be to cluster low-dimensional

representations of the heatmaps and then tag each cluster based on what the model is focusing on; unfortunately, someone would still have to manually label the clusters of important regions.

Related to model monitoring for feature drift detection, Organization D has encountered issues with spurious correlations in its smile detection models. Their VP of AI noted that "[ML engineers] must know to what extent you want ML to leverage highly correlated data to make classifications." Explainability can help identify models that focus on that correlation and can find ways to have models ignore it. For example, there may be a side effect of a correlated facial expression or co-occurrence: cheek raising, for example, co-occurs with smiling. In a cheek-raise detector trained on the same dataset as a smile detector but with different labels, the model still focused on the mouth instead of the cheeks. Attending to the mouth was undesirable in the cheek-raise detector but allowed in the smile detector. One way Organization D combats this is by using simpler models on top of complex feature engineering. For example, they use black-box deep learning models to build good descriptors robust across camera viewpoints and to detect features that experts deem important for drowsiness. There is one model per important descriptor (i.e. one model for eyes closed, one for yawns). Then, they fit a simple model on the extracted descriptors such that the important descriptors are obvious for the final prediction of drowsiness. Ideally, if Organization D had guarantees about the disentanglement of data generating factors Adel et al. [2018], they would be able to understand which factors (descriptors) play a role in downstream classification.

Takeaway: Heatmaps (and feature importance scores) are hard to aggregate, which makes it hard to do false positive detection at scale. Spurious correlations can be detected via feature importance, even using simple gradient-based techniques.

Adversarial Training in Content Moderation

To ensure the deployed model is robust to variation in inputs, many organizations we interviewed use adversarial training to improve performance and explainability. Organization E moderates user-generated content (UGC) on several public platforms. Specifically, the R&D team at Organization E developed several models to detect adult and violent content from users' uploaded images. Their quality assurance (QA) team measures model robustness to improve content detection accuracy under the threat of adversarial examples. The robustness of a content moderation model is measured by the minimum perturbation required for an image to evade detection [Szegedy et al., 2013]. As is common in the adversarial literature, Organization E applies Projected Gradient Descent (PGD) to search for the minimum perturbation from the set of allowable perturbations. ML engineers on the QA team are shown a ℓ_2-norm perturbation distance averaged over multiple randomly sampled test images. The larger the average perturbation, the more robust the model is, as it takes greater effort for an attacker to

evade detection. The average perturbation required is widely used as a metric when comparing different candidate models and different versions of a given model. Organization E finds that more robust models have more convincing gradient-based explanations (i.e. the gradient of the output with respect to the input shows that the model is focusing on relevant portions of the images), confirming similar research [Etmann et al., 2019; Ilyas et al., 2019; Tsipras et al., 2019]. Researchers at Organization E hope that convincing explanations from robust models mean that the organization will not only attempt adversarial training for many of their models but also consider exposing these explanations to end users some day.

Takeaway: There is a relationship between model robustness and explainability. Model robustness seems to improve the quality of feature importances (specifically saliency maps), confirming research findings [Etmann et al., 2019]. Feature importance helps find minimal adversarial perturbations [Chapman-Rounds et al., 2021] for language models in practice.

3.5 Multistakeholder Convening

Our interview study above suggested that explainability was **not** typically being used in the service of external transparency goals. Much of the ML research claiming to explain how ML models work had not been used in deployed systems to provide explanations to end users, regulators, or other external stakeholders. To ensure explainability reaches beyond internal stakeholders (e.g. ML engineers) in practice, the ML community should account for how and when external stakeholders want transparency. This motivated us to bring together academic researchers, policymakers, and industry experts for a day-long workshop in New York City on February 9th, 2020 to discuss challenges and potential solutions for using explainability at scale for external stakeholders.

Demographics and Methods

33 attendees from five countries, along with seven trained facilitators to moderate the discussion, attended this workshop. Of the 33 attendees, 15 had ML development roles, 3 were designers, 6 were legal experts, and 9 were policymakers. 15 attendees came from for-profit corporations, 12 came from non-profits, and 6 came from academia. First, attendees were clustered into 5- or 6-person groups, with representation from different expertise in each group, wherein they discussed their respective disciplines' notions of explainability and attempted to align on common definitions. Second, attendees were separated into domain-specific groups, each with a combination of domain experts and generalists, to discuss (i) use cases for, (ii) stakeholders of, (iii) challenges with, and (iv) solutions regarding explainability and, more generally, algorithmic

transparency. The domains discussed were finance (e.g. employee monitoring for fraud prevention, mortgage lending), healthcare (e.g. diagnostics, mortality prediction), media (e.g. misinformation detection, targeted advertising), and social services (e.g. housing approval, government resource allocation). In the next section, we discuss emergent themes of the domain-specific portion of the workshop.

Definitions

As "explainability" is usually ill-defined [Lipton, 2018], we asked the interdisciplinary groups in the first part of the workshop to come to a consensus definition of explainability. Below are some definitions provided by attendees.

- Explainability gives stakeholders a summarized sense of how a model works to verify if the model satisfies its intended purpose.
- Explainability is for a particular stakeholder in a specific context with a chosen goal and aims to get a stakeholder's mental model closer to a model's behavior while fulfilling a stakeholder's explanatory needs.
- Explainability lets humans interact with ML models to make better decisions than either could alone.

All definitions of explainability included notions of context (the scenario in which the model is deployed), stakeholders (those affected by the model and those with a vested interest in the model's explanatory nature), interaction (the goal of the model and its explanation serve), and summary (the notion that "an explanation should compress the model into digestible chunks"). In general, attendees concluded that explainability loosely refers to tools that empower a stakeholder to understand and, when necessary, contest the model's reasoning.

One policymaker suggested that technical definitions of explainability can be unsettling since such definitions solely focus on exposing model innards to stakeholders without a clear objective. Explainability tends to not consider the broader context in which the model is deployed. For a given context, the ML community's treatment of explainability fails to capture *what is being explained, to whom, and for what reason*? One academic commented how intelligibility may capture more than explainability; since the former term encapsulates explainability, interpretability, and understandability, capturing all that people can know or infer about ML models [Zhou and Danks, 2020]. Next, we discuss emergent themes of the domain-specific portion of the workshop.

3.6 Convening Takeaways

We first discuss the need for broader community engagement in explainability development and then outline elements of deploying transparency at scale.

3.6.1 Designing for Explainability

Throughout the workshop, attendees spoke about the lack of community engagement in the explainability process. Breaking this down further, the successful design of explainability for external stakeholders will likely require a better understanding of the context of explainability, evaluation of explainability techniques, involvement of affected groups in development, and education of various stakeholders regarding explainability use and misuse.

Context of Explanations

Given the context of a deployed model, an explanation should help stakeholders assess model outcomes based on the additional information provided (e.g. understanding how the model behaves or validating the predictability of the model's output) [Ruben, 2015]. Each stakeholder may require a different type of transparency into the model [Hall et al., 2019]. Expanding the ML community's understanding of stakeholder requirements will allow for model explanations to be personalized. The notion of a good explanation varies by stakeholder and their needs [Miller, 2019]. To further probe these contexts and understand what stakeholders actually need from explanations, many attendees pointed to the need for explainability to incorporate expertise from other disciplines. Introducing researchers from human-computer interaction and user experience research as well as bringing in community experts were seen as ways to enable participatory development [Cornwall and Jewkes, 1995] of explainability. Another dimension of context that attendees noted is that ML systems represent a distinctly sociotechnical system [Selbst et al., 2019], so each model in production will require a different level and style of transparency for each stakeholder. At times, transparency requirements can be procedural; though, making that information available could be nontrivial [Gebru et al., 2021; Mitchell et al., 2019; Raji and Yang, 2019].
Takeaway: Explainability tools should not be developed without regard to the context in which they will be deployed.

Evaluation of Explanations

As part of using explainability techniques in different contexts, practitioners described a need for clarity on how to evaluate explainability's effectiveness. Given the range of potential

uses for explainability, it is not clear how stakeholders should agree upon or test for the desirable properties of an explanation [Doshi-Velez and Kim, 2017]. Quantitative evaluation of explanations, like in [Bhatt et al., 2021c; Hase and Bansal, 2020], are a starting point. Qualitative studies of how to combine models and explanations with stakeholders in a decision-making process [Bansal et al., 2019a] are a critical next step. Even amongst researchers focused on explainability, there is no consensus on how to evaluate an explanation, let alone an understanding of which explanation techniques are good at helping stakeholders achieve their goals in specific contexts [Hoffman et al., 2018]. When people have an understanding of an explanation technique, they underestimate the importance of understanding the data being modeled, which can expose patterns to expect in a model's explanations [Matejka and Fitzmaurice, 2017]. Attendees discussing the role of explainability in journalism pointed to the difficulty of understanding how users understand explanations they are given about mis-/dis-information. Cognitive biases such as the back-fire effect [Peter and Koch, 2016], where users double down on prior beliefs when confronted with contradictory evidence, can completely invert the intended effect of explaining why an article is deemed inaccurate. Attempts at explanation evaluation, especially automated evaluation, can very easily miss these more contextual elements [Doshi-Velez and Kim, 2017]. To effectively evaluate explanations, attendees wanted a rigorous human evaluation of explainability; to date, there are only a few examples [Poursabzi-Sangdeh et al., 2021]. Attendees called for interdisciplinary collaboration by bringing in experts from human-computer interaction, user experience research, and social sciences to help establish explanation evaluation in specific contexts.
Takeaway: Interdisciplinary collaboration is essential to have clarity on what is required and how organizations ought to evaluate explainability methods.

Appropriate Design for Affected Groups

As discussed above, a key component of explainability is answering the question of *what* is being explained *to whom* [Ehsan and Riedl, 2021]. Attendees pointed to scenarios where communities might have disparate capacities to engage with explanations. One scenario posed was the case of an apartment rental application tool, which ought to explain to applicants why they may be denied. Attendees thought it was likely that brokers and applicants with institutional knowledge would be able to modify future applications to improve their chances of success, whereas already disenfranchised applicants would be stuck in cycles of rejection. Understanding these differential responses in non-theoretical cases will likely require designing and evaluating systems directly alongside impacted communities.

In healthcare, protections for how input data is used have already been codified into law. HIPAA [US Department of HHS, 2013] in the US and GDPR [Council of European Union,

2018] in Europe require confidential and transparent management of medical data. As a result of these patient protections, attendees noted that any type of explanation using the training data is unlikely to be deployed without explicit user consent. For other application areas, however, it is less likely that such stringent data protections will apply, leaving it to organizations to decide how protected and transparent individual data use should be. One potential benefit of explainability is that issues of data misuse can be more directly addressed. Explanation recipients, whether they are credentialed experts (e.g. doctors) or the actual subjects of decisions (e.g. rental applicants), likely have a prior understanding of which attributes should be relevant to the decision being made [Miller, 2019]. By having explanations explicitly mention the attributes used in decision-making, these stakeholders may be empowered to contest privacy encroachments and challenge questionable decisions.

Takeaway: Including multiple stakeholders in the development of explanations and striving to better understand stakeholder needs can prevent preferential treatment and data misuse.

Stakeholder Education

Understanding how to listen to stakeholders and inform them about explainability methods is key to widespread adoption. Some attendees noted that data scientists are aware of explainability, but have not been provided with a best practices framework for choosing which explanation technique to use in various contexts [Kaur et al., 2020]. In certain domains, explainability requirements are *top-down* (executives are mandating a specific form of explanation from models); however, widespread adoption of explainability will likely require *grassroots* education of data scientists, who are aware of the context in which the model is deployed [Ehsan et al., 2021]. One issue in explainability stakeholder education is ensuring stakeholders are aware of the limitations of post-hoc explanations. Many attendees agreed that post-hoc explanations are limiting and may not be useful justifications for model behavior in practice. Weller [2019] notes that transparency of ML models can allow malicious attackers to provide deceptive information as an explanation; recent work has concluded that feature importance techniques can be manipulated to fool end users [Slack et al., 2020a]; concurrently, we showed that feature importance explanations can be used to conceal model unfairness [Dimanov et al., 2020]. Informing stakeholders of explainability's potential to mislead unintentionally or to deceive purposefully is critical.

Another attendee from the healthcare domain noted that clinicians have background knowledge and training in making diagnoses, but for the clinician to feel comfortable vetoing a diagnostic model, the clinician may want to be aware of the model's failure modes and understand how the model works. Sometimes there is no time for clinicians to get the training required to do this translation (or no space in the medical school curriculum). There may be an

emerging career where one has specialties in clinical training and ML, who are able to translate model behavior to clinicians and understand the nuances of the model's specification. Beyond data scientist and domain expert education, public education around ML and explainability is crucial. One policymaker noted that people seldom know they are interacting with ML models. Public education would require a common vocabulary that is simple for non-experts to understand and avoids obscure jargon.

Takeaway: Listening to and informing stakeholders will encourage thoughtful adoption of explainability while accounting for differences in expertise and bandwidth.

3.6.2 Deploying Explainability

In addition to engaging with the community while developing explainability tools, attendees also discussed the many nuances of deploying such tools in context.

Uncertainty alongside Explanations

Some attendees noted that predictive uncertainty can be complementary to an explanation. One attendee from a healthcare organization noted that some diseases are more well-understood than others. When using diagnostic decision support tools for predicting which disease a patient has, clinicians need to understand how confident the model is for the suggested prediction. Ideally, the clinicians should decide the threshold at which the model can safely make a prediction of a rare disease. Uncertainty within the model ought to be higher for rare diseases than for common ones, but in practice, it is difficult to quantify and communicate predictive uncertainty. Rigorously measuring and exposing uncertainty alongside an explanation is likely to be useful to clinicians who can leverage their expertise to make informed decisions [Tonekaboni et al., 2019]. It was this discussion with attendees that inspired us to write [Bhatt et al., 2021a], which we included in Chapter 2.3.2.

More generally, attendees agreed that uncertainty can be used as a form of algorithmic transparency, alongside or instead of explainability. Though, we should recognize that it is inherently challenging for people to understand probabilities [Lichtenstein and Newman, 1967]. In specific situations, it may be sensible to expose this uncertainty to a human decision-maker [Zhang et al., 2020b]; for example, showing a mortgage approver for which applicants a model is uncertain could help the approver know when to intervene in an algorithmic decision-making process: one attendee's proposal to view uncertainty in input space encouraged us to develop CLUE, the counterfactual explanation method [Antorán et al., 2021; Ley et al., 2022] which we devise in Chapter 4.2.

Takeaway: By quantifying and communicating uncertainty to stakeholders effectively, the ML community can treat uncertainty as a helpful form of transparency.

Interacting with Explanations

Most existing post-hoc explanation techniques convey information about the model to stakeholders; however, few techniques have been developed to update a model based on the stakeholder's view of the explanation [Bansal et al., 2019b; Lee et al., 2020] or to provide stakeholders with the ability to toggle the information in an explanation. Explanations from ML models effectively provide evidence, and stakeholders then examine that evidence, noting if it aligns with their intuition [Bansal et al., 2019a; Buçinca et al., 2020]. However, stakeholders should be able to interact with the explanation to control how much information is conveyed: if a stakeholder wants less information, the explanation technique used should convey a more summarized explanation without changing the underlying model. One attendee from a civil society organization noted that interactive explanations, which allow stakeholders to peek inside a model's behavior, are important when governments deploy ML models for resource allocation and when developers provide natural language explanations alongside predictions. However, if a front-line practitioner (i.e. a government official checking for a farmer's compliance with local regulation) cannot override the model's prediction, then other stakeholders grow skeptical of the model's utility. Front-line practitioners want to ask questions to the model about its learned reasoning and want to provide feedback to the model in real time. Flexible, interactive models that allow practitioners to alter trained models online to reflect practitioners' mental models are crucial [Fails and Olsen Jr, 2003].

Another attendee noted that, in their organization, language models conflated Paris Hilton with Hilton Hotels and the city of Paris; their organization lacked procedures for a data scientist to expose and alter these correlations to reflect reality. How to mathematically formalize the feedback received from the stakeholder regarding the explanation and how to update the model prior, in some sense, based on expert feedback are open questions, which we went to enumerate in [Chen et al., 2023a] as described in Chapter 2.4.1. Cai et al. [2019] let pathologists use an example-based explanation interface to improve model outcomes. Tools that enable interactions with models, documentation that enumerates implicit assumptions in model training, and interfaces that allow stakeholders to interrogate models are essential for adopting explainability [Amershi et al., 2014]. Model interactivity may require interpretability by design, wherein the model itself is explainable, due to the chosen model class, instead of deriving post-hoc, approximate explanations [Rudin, 2019].

Some elements of explainability are indirect. Another attendee noted that a clinician might want an explanation from a diagnostic model. Developers can create an explanation that the

clinician can deliver verbally to the patient, and statistical rigor (e.g. false positive rate or feature importance) can be reserved for patients who ask for specificity. Explainability in human-machine teams may be necessary up until a certain point. Stakeholders may need interactivity to ensure the model aligns with their own mental model [Roads and Love, 2021]; thereafter, reliability may matter more than transparency [Zerilli et al., 2022]. Interacting with a model, based on its predictions or based on its explanations, is one way to facilitate a synergistic dialogue between humans and machines [Amershi et al., 2019].
Takeaway: Creating interactive explanation techniques and building models that update based on expert feedback will enable the effective use of explainability at scale.

Behavior Changes from Explanations

In many domains, attendees noted that a key component of explanations is how actionable they are for different stakeholders. Whether this was in the case of hospitals improving their health outcomes or journalists removing references to mis-/dis-information, there are specific actions motivated by the explanation a stakeholder receives. Therefore, issues can arise when explanations do not account for how stakeholders might respond to them. Going back to the explainable rental application system, a key motivation for designing such a system should be to inform applicants on how to become better applicants in the future. Counterfactual explanations may be the most useful for such purposes. Such explanations should ideally guide negatively impacted applicants on how to most easily improve their records so as to achieve a more positive outcome. Thus such explanations would reduce what has elsewhere been referred to as the *social cost* of the model [Milli et al., 2019a; Von Kügelgen et al., 2022]. However, in some situations, explanations may hone in on easily modifiable characteristics (i.e. the number of friends one has on a social media account) or seemingly irrelevant characteristics (e.g. current postcode) inducing stakeholders to have less faith in the model or in the decision-making process as a whole.

As models are made more transparent and stakeholder behavior adapts, it is likely that model performance will similarly start to shift. As stated by Goodhart's Law and as re-phrased by Marilyn Strathern, "When a measure becomes a target, it ceases to be a good measure" [Strathern, 1997]. Once a metric is used for informing decisions, practitioners have incentives to optimize that metric to achieve the decision they want. One example from an attendee was the case of health-outcome predictions. Suppose a patient is predicted as high risk and the doctor is given an explanation that attributes much of this risk to the patient's weight. Given this information, the patient might lose weight to improve their prognosis. However, the culprit for high risk could be a factor correlated with weight, such as hypertension, but not reduced concomitantly with weight. If the model does not explicitly include hypertension, it

is likely to underestimate the risk of a patient who has lost weight but still has hypertension. When only a few individuals make this change, overall accuracy might not drop; if encouraging weight loss becomes a standard treatment, we can expect the model's performance to fall if the model is not updated to reflect the new patient archetype.

Beyond distribution shift, attendees also discussed how professionals working closely with a model might adapt to it over time. Drawing from the healthcare conversation once more, one attendee pointed to the trust dynamics between nurses/doctors and explainable ML. At first, there is likely to be a lack of trust, but trust can grow if the tool proves accurate and useful (i.e. displays its trustworthiness via competence). Reaching a more trusting and comfortable state, however, often means not just blindly following the tool's recommendations, but incorporating them into daily judgments. As another attendee mentioned, even if a model update improves accuracy or explanation quality, the change can cause a mismatch between model behavior and user expectations: this would, in turn, worsen team overall performance. There has been some work on the dynamics of updating ML models in human-machine teams [Bansal et al., 2019b]. We learn when stakeholders should receive access to a model as decision support in [Bhatt et al., 2023], upon which we elaborate in Chapter 5.

Takeaway: When designing explainability tools, practitioners need to include how the explanations might be acted upon as a central design question. If the explanations motivate the average user to game or distrust the system, perhaps it indicates that the model is making predictions on unfair/unimportant attributes.

3.7 Summary

This chapter explores how explainability, a popular form of algorithmic transparency, has been used in practice. First, via interview studies, we find that while ML engineers are increasingly using explainability techniques to test model behavior during the development process, there are still significant limitations to current techniques that limit their ability to effectively inform end users. Unfortunately, this suggests that explainability and, more broadly, algorithmic transparency are **not** widely used in practice. Motivated by these findings, we hosted an interdisciplinary convening of stakeholders of explainability. From this workshop, it was apparent that to serve transparency goals, future efforts around explainability would clearly benefit from community engagement in explainability development and from the thoughtful deployment of explainability. Inputs from the full spectrum of stakeholders will increase the adoption of explainability beyond just the ML community. We urge researchers to engage in interdisciplinary conversations with experts and external stakeholders to improve the effectiveness of explainability and, more generally, algorithmic transparency.

Chapter 4

Algorithmic Transparency in Decision-Making

We continue our study of algorithmic transparency by designing explanations to meet stakeholder needs in specific decision-making contexts. In Chapter 3, we discuss how explainability was not used widely across organizations. Through our convening, we identified gaps and opportunities in the explainability landscape, such as building mechanisms to evaluate explainability tools (Chapter 3.6.1) and developing explanations of uncertainty (Chapter 3.6.2). While the previous chapter was written independent of specific explanation methods, we now devise new algorithmic transparency methods and show how they meet the needs of decision-makers, which we then assess via user studies.

In Chapter 4.1, we propose a method for reducing redundancy in training point importance explanations [Bhatt et al., 2021b]. Our method can identify training data that are not only influential to the model but also diversely placed in input space. We show how our explanations can be used to evaluate explanations via task simulatability, wherein decision-makers simulate the entirety of the model's decision boundary. In Chapter 4.2, we suggest a new method to explain outcome uncertainty in terms of input features [Antorán et al., 2021]. These uncertainty explanations can be seen as analogs to feature importance explanations. We demonstrate how decision-makers can use these explanations to simulate a model's uncertainty on an unseen test point. For each proposed method, we provide *food for thought* on how one could deploy the method in practice.

4.1 DIVINE: *DIV*erse *IN*flu*E*ntial Training Points

Training point importance is a useful form of explainability for practitioners when reasoning about a machine learning (ML) model's behavior [Jeyakumar et al., 2020]. This form of explanation identifies which training points are most important to an ML model [Ghorbani and Zou, 2019; Koh and Liang, 2017; Koh et al., 2019; Kwon et al., 2021; Yeh et al., 2018]. Evaluating such explanations can involve estimating the impact on the model of dropping important point(s) or asking the user to provide predictions for unseen points given important point(s). However, the top-m most important points returned by popular methods are often redundant, in the sense that several may be very similar, limiting the extent of explanation provided [Barshan et al., 2020; Bhatt et al., 2020b]. Redundant explanations are known to be less useful to practitioners [Wachter et al., 2017] and have been studied for counterfactual explanations [Russell, 2019]. To address redundancy in training point importance, we devise an approach for selecting a set of DIVINE (DIVerse INfluEntial) training points.

Figure 4.1a shows that the top-5 influential points with respect to the approximate leave-one-out estimate of Koh and Liang [2017] and Data Shapley [Ghorbani and Zou, 2019] are all located in a small vicinity (red and blue diamonds respectively). Due to this lack of diversity, practitioners may miss key insights about model behavior on underrepresented data points. Prototypes [Kim et al., 2016] only depend on the dataset, not on the model, and are located near the large cluster centers. All previous methods ignore the cluster of points in the top left corner. Our DIVINE points not only lie across feature space but also in regions of high influence. Our method provides the flexibility, under a common assumption, to operate on top of training point importance scores from a wide range of methods, including Data Shapley (DS) [Ghorbani and Zou, 2019] and influence functions (IF) [Koh and Liang, 2017]. Beyond the synthetic setting of Figure 4.1a, consider a misclassified test point, as in Figure 4.1c. The influential training points according to IF and RelatIF [Barshan et al., 2020] are very similar to the test point. The resulting explanation contains redundant information. An explanation containing diverse points may be more useful: the coat that appears in the DIVINE points but not in the others.

In our user studies, we study task simulatability as a mechanism for evaluating training point importance methods. To the best of our knowledge, we are the first to study how well practitioners can simulate the *entire* model given an explanation; this starkly differs from previous work that focuses on forward simulation of specific unseen points Jeyakumar et al. [2020]. Such an understanding is essential to getting a holistic sense of the decision boundary. We find that our DIVINE points enable practitioners to simulate model behavior more accurately compared to existing approaches.

Contributions We devise a method to find a diverse set of training points that are influential to a model. Our top m DIVINE points, when trading off influence with diversity objectives, can

Food for Thought: Deploying DIVINE

Our method, DIVINE, provides explanations of model predictions in the form of important training data points. For instance, suppose we want to build a diagnostic model for clinical use. Given the medical history and symptoms of a new patient, our model will suggest potential diagnoses. DIVINE explanations of model predictions will indicate which m previous patients (i.e. training data points) were important to the model-generated diagnoses for a new patient. DIVINE guarantees the m patients are not identical, since diversity in the pool of previous, relevant patients can help clinicians understand how the model works and catch any mistakes the model may have made. This diversity could manifest as previous patients with similar symptoms but different medical history. All of these previous patients could be influential for the model's prediction for the new patient. DIVINE explanations will be crucial in settings where stakeholders benefit from getting a diverse sense of how models behave.

provide a more comprehensive overview of model behavior (Section 4.1.2). Our computational experiments on synthetic and real-world datasets demonstrate that DIVINE can help explore the diverse, influential regions of feature space and can be used to identify and remove unfairness-inducing points (Section 4.1.3). Our extensive user studies reveal that DIVINE leads to better task simulatability (Section 4.1.4).

4.1.1 Related Work: Assigning Training Point Importance

We first detail methods for obtaining training point importance scores, expanding upon Chapter 2.3.1. Consider an ML model f parameterized by $\theta \in \Theta$. Given training data $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$ and a loss function $\ell(y, f_\theta(x))$ like 0-1 loss, weighted empirical risk minimization estimates $\hat{\theta} = \arg\min_{\theta\in\Theta} \sum_{i=1}^n w^{(i)} \ell(y^{(i)}, f_\theta(x^{(i)}))$, where $w^{(i)}$ is the weight given to training point i. Usually, each training point has equal weight (e.g. $w^{(i)} = \frac{1}{n}$). The parameters of the leave-one-out (LOO) model obtained by dropping i-th training point (i.e. zero weight $w^{(i)} = 0$ on point $x^{(i)}$) are denoted by $\hat{\theta}_i^{\text{LOO}}$. Dropping a set of training points $\mathcal{U}$ can be done by setting all $w^{(i)}$s in the set to zero. The resulting model is denoted by $\hat{\theta}_{\mathcal{U}}^{\text{LOO}}$. The importance of $x^{(i)}$ can be written as $I_i^{\text{LOO}} = \mathcal{L}(\hat{\theta}_i^{\text{LOO}}) - \mathcal{L}(\hat{\theta})$ where $\mathcal{L} : \Theta \to \mathbb{R}$ measures a quantity of interest (e.g. loss). In this chapter, we call $\mathcal{L}$ the evaluation function and focus on cases where $\mathcal{L}$ is the loss $\mathcal{L}_{\text{loss}}(\theta) = \sum_{i=1}^n \ell(y^{(i)}, f_\theta(x^{(i)}))$ on n data points or a group fairness metric, like equal accuracy (Section 4.1.3). Unless otherwise specified, a lower $\mathcal{L}(\theta)$ is desirable. A positive $I_i = \mathcal{L}(\hat{\theta}_i^{\text{LOO}}) - \mathcal{L}(\hat{\theta})$ implies that including $x^{(i)}$ is *helpful* for lowering $\mathcal{L}$ when learning θ: upon removing $x^{(i)}$, the value of $\mathcal{L}$ at θ_i^{LOO} increased, which is undesirable. A negative I_i



(a) Synthetic Data (b) Global: FashionMNIST (c) Local: FashionMNIST

Fig. 4.1 In 4.1a, we show that our method selects DIVINE points (yellow circles) that are spread across the feature space. This contrasts IF (red diamonds) and Data Shapley (blue diamonds), which select points located in one region. Note the overlap between IF and DIVINE points in the top right. In 4.1b and 4.1c, we show that DIVINE points (third row) are more diverse than ones selected by IF (first row) or other methods (second row). DIVINE is calculated by trading off IF and $\mathcal{R}_{SR}$. The predicted label is listed under each point.

implies that including $x^{(i)}$ is *harmful* for lowering $\mathcal{L}$. A large absolute magnitude of I_i implies that $x^{(i)}$ is influential.

Influence Functions. Retraining for different weight configurations can be computationally expensive. Koh and Liang [2017] develop algorithms to approximate the effect of removing a training point on the loss at a test point by re-weighting its contribution. Suppose we modify the weight of $x^{(i)}$ from $w^{(i)} = \frac{1}{n}$ to $w^{(i)} = \frac{1}{n} + \varepsilon_i$. Let $\hat{\theta}_{\varepsilon_i}$ be the parameters obtained upon re-weighting. If we let $\varepsilon_i = -\frac{1}{n}$, this amounts to dropping $x^{(i)}$ from the training data. Influence functions (IF) can be used to approximate $\hat{\theta}_{\varepsilon_i}$ [Cook and Weisberg, 1980]. Assuming the loss ℓ is twice differentiable and convex in θ, we can linearly approximate the parameters upon dropping $x^{(i)}$ as $\hat{\theta}_i^{\text{IF}} \approx \hat{\theta}_{\varepsilon_i} \approx \hat{\theta} - H_{\hat{\theta}}^{-1} \nabla_\theta \, \ell(y^{(i)}, f_{\hat{\theta}}(x^{(i)}))\varepsilon_i$, where $H_\theta = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 \ell(y^{(i)}, f_\theta(x^{(i)}))$ is the Hessian of ℓ. Per Koh et al. [2019], we estimate the influence of dropping $x^{(i)}$ on any $\mathcal{L}$ as $I_i^{\text{IF}} \approx I_i := \nabla_\theta \, \mathcal{L}(\hat{\theta})^\intercal H_{\hat{\theta}}^{-1} \nabla_\theta \, \ell(y^{(i)}, f_{\hat{\theta}}(x^{(i)}))$. When $\mathcal{L}$ is *loss*, Koh et al. [2019] note that influence is additive, implying importance scores are additive. Thus, the importance of training points in $\mathcal{U}$ is given by: $I_{\mathcal{U}}^{\text{IF}} = \sum_{i \in \mathcal{U}} I_i^{\text{IF}}$.

Data Shapley. Instead of computing parameters to obtain an importance score with respect to $\mathcal{L}$, techniques like Data Shapley (DS) aim to directly compute importance scores [Ghorbani and Zou, 2019]. Shapley values are a game-theoretic way to attribute value to *players* in a game. Ghorbani and Zou [2019] apply Shapley values to training point importance. They propose to compute the importance of $x^{(i)}$ as $I_i^{\text{DS}} = C^{-1} \sum_{\mathcal{S} \subseteq \mathcal{D} \setminus \{i\}} \mathcal{L}(\hat{\theta}_{\mathcal{U}}^{\text{LOO}}) - \mathcal{L}(\hat{\theta}_{\mathcal{U} \cup \{i\}}^{\text{LOO}})$, where $C = \binom{n-1}{|\mathcal{S}|}$, $\mathcal{S}$ is a subset of the training data and $\mathcal{U} = \mathcal{D} \setminus \mathcal{S}$. We can efficiently approximate DS using Monte Carlo Sampling [Ghorbani and Zou, 2019; Tang et al., 2021].

Other Methods. Khanna et al. [2019] use Fisher kernels to select influential training points efficiently. Others have searched for prototypes, which are representative points that summarize a dataset *independent* of a model [Bien and Tibshirani, 2011; Gurumoorthy et al., 2019]. Kim

et al. [2016] use maximum mean discrepancy (MMD) to find prototypes but do not assign importance scores to the selected points. Prototypes themselves do not explain a model, since changing an underlying model does not change the prototypes selected.

Connection to Other Explanation Types. Most existing diversity-inducing methods have been built for counterfactual explanations [Ley et al., 2022]. This line of work looks to find multiple recourse suggestions for users Ustun et al. [2019]. Russell [2019] searches for a diverse set of counterfactual explanations using mixed integer programming with a novel mixed polytope constraint. Mothilal et al. [2020] obtain diverse counterfactuals based on determinantal point processes. Both authors find multiple, non-redundant counterfactual explanations that provide diverse recourse options to users. More generally, Tan [2022] points out a need for diversity in machine-generated explanations to reflect the diversity in human-provided explanations. To the best of our knowledge, we are the first to consider inducing diversity in example-based explanations. Diversity in training point importance enables users to get a sense of which non-redundant points affect the model most.

4.1.2 Method: Selecting Diverse Samples

In Figure 4.1a, the top-m influential points based on existing importance scores result in a set of points that are similar to each other. We desire m points that are simultaneously influential (high importance) and diverse across the feature space to serve as an explanation of model behavior. To achieve these desiderata, we propose the following objective:

$$\max_{\mathcal{S}\in\mathcal{D},|\mathcal{S}|=m} \mathcal{I}(\mathcal{S}) + \gamma\mathcal{R}(\mathcal{S}), \tag{4.1}$$

where $\mathcal{S}$ is a subset of m important points from the dataset $\mathcal{D}$, $\mathcal{I}(\mathcal{S})$ is a normalized function ($\mathcal{I}(\emptyset) = 0$) that captures the importance of the points in $\mathcal{S}$, $\mathcal{R}(\mathcal{S})$ is a function that captures the diversity of the points in $\mathcal{S}$, and γ controls the trade-off between the two terms. Solving the optimization problem in Equation 4.1 yields a set $\mathcal{S}$ of m DIVerse and INfluEntial points, which we call DIVINE points. Setting $\gamma = 0$ recovers the traditional setup of selecting m points with the highest importance. Our setup is reminiscent of combining loss functions (e.g. one to penalize training error and one to regularize for sparsity or smoothness): we effectively regularize for diversity in the m influential points we select. Our formulation in Equation 4.1 is similar to that of Lin and Bilmes [2011], who select relevant yet diverse sentences to summarize a document, and that of Prasad et al. [2014], who scale diverse set selection to large datasets using submodularity.

We take $\mathcal{I}(\mathcal{S}) = \sum_{i\in\mathcal{S}} I_i$ to be the sum of the importance scores of points in $\mathcal{S}$. We propose three submodular $\mathcal{R}(\mathcal{S})$:

1. Sum-Redundancy [Libbrecht et al., 2018]: $\mathcal{R}_{\text{SR}}(\mathcal{S}) = \kappa - \sum_{u,v\in\mathcal{S}} \phi(u,v)$;
2. Facility-Location [Krause and Golovin, 2014]: $\mathcal{R}_{\text{FL}}(\mathcal{S}) = \sum_{u\in\mathcal{D}} \max_{v\in\mathcal{S}} \phi(u,v)$; and
3. MMD [Kim et al., 2016]: $\mathcal{R}_{\text{MMD}}(\mathcal{S}) = c_1 \sum_{u\in\mathcal{D},v\in\mathcal{S}} \phi(u,v) - c_2 \sum_{u,v\in\mathcal{S}} \phi(u,v)$,

where $\phi(u,v)$ is the similarity between two points, $\kappa = \sum_{u,v\in\mathcal{D}} \phi(u,v)$, $c_1 = \frac{2}{n|\mathcal{S}|}$, and $c_2 = \frac{1}{|\mathcal{S}|^2}$. We let ϕ be the radial basis function kernel. While $\mathcal{R}_{\text{SR}}$ encourages us to find m influential points that are diverse from each other, both $\mathcal{R}_{\text{MMD}}$ and $\mathcal{R}_{\text{FL}}$ encourage our influential points to be representative of the training data. $\mathcal{R}_{\text{SR}}$ is known as penalty-based diversity and penalizes similarity between points in $\mathcal{S}$ [Lin and Bilmes, 2011]. $\mathcal{R}_{\text{FL}}(\mathcal{S})$ maximizes the average similarity between a training point and its most similar point in $\mathcal{S}$. $\mathcal{R}_{\text{MMD}}$ ensures the m selected influential points are similar to the training data while being different from each other. Practitioners can select the $\mathcal{R}$ most appropriate for their use case.

Algorithm 1 Greedy DIVINE selection

Input: Dataset $\mathcal{D}$, Trade-off parameter γ, number of diverse influential points m
for all $x^{(i)} \in \mathcal{D}$ **do**
  $I_i \leftarrow \text{influence}(x^{(i)})$
end for
$\mathcal{S} \leftarrow \emptyset$; $I_{\mathcal{S}} \leftarrow 0$
while $|\mathcal{S}| < m$ **do**
  $\mathcal{S} \leftarrow \mathcal{S} \cup \arg\max_{x^{(i)}\in\mathcal{D}\setminus\mathcal{S}} [I_{\mathcal{S}} + I_i + \gamma\mathcal{R}(\mathcal{S})]$
  $I_{\mathcal{S}} \leftarrow \sum_{i\in\mathcal{S}} I_i$
end while
Output: Set of m DIVINE points $\mathcal{S}$

It is well-known that a modular function $\mathcal{I}(\mathcal{S})$ plus a submodular function $\mathcal{R}(\mathcal{S})$ is submodular, rendering the overall objective submodular [Bach et al., 2013]. Therefore, we take a greedy approach to performing the optimization in Equation 4.1, as outlined in Algorithm 1. Greedy selection returns a set that typically performs very well, and has a guarantee of at worst $(1-\frac{1}{e})$ performance of the optimal set $\mathcal{S}^*$ [Nemhauser et al., 1978]. When the computational cost of full greedy alone can be high (i.e. for large datasets), we can also take a stochastic greedy approach per Mirzasoleiman et al. [2015]. Moreover, some may find our additivity assumption, which lets $\mathcal{I}$ be modular, too restrictive. However, note that, by construction, DS satisfies linearity, which implies modularity. For IF, we show in Appendix B.2 that modularity holds for various $\mathcal{L}$ as long as m is not too large. Furthermore, instead of calculating I_i for the entire dataset and then performing greedy selection, we could select the first point greedily,

recalculate I_i for the remaining points, greedily select the next point, and repeat until we have m points. In [Bhatt et al., 2021b], we show that this works similarly to approaches without recalculation.

In addition to explaining model behavior with a globally diverse set of influential training points, our framework is amenable to local explanations: a diverse set of points that explains the prediction for a specific point, $x^{(i)}$. To accomplish this, we let $\mathcal{L}$ be the loss at $x^{(i)}$: $\mathcal{L}_i(\theta) = l\left(y^{(i)}, f_\theta(x^{(i)})\right)$. Barshan et al. [2020] notice that the top-m influential points selected by Koh and Liang [2017] tend to be outliers, and add locality constraints to the IF objective. However, they solve a different problem than we do: their method, RelatIF, is concerned with data points being atypical, whereas DIVINE focuses on providing a diverse explanation by ensuring that the same region does not get marked as influential repeatedly. Our method would select a diverse set of outliers (if those are indeed influential) whereas the constraints of Barshan et al. [2020] would not permit it.

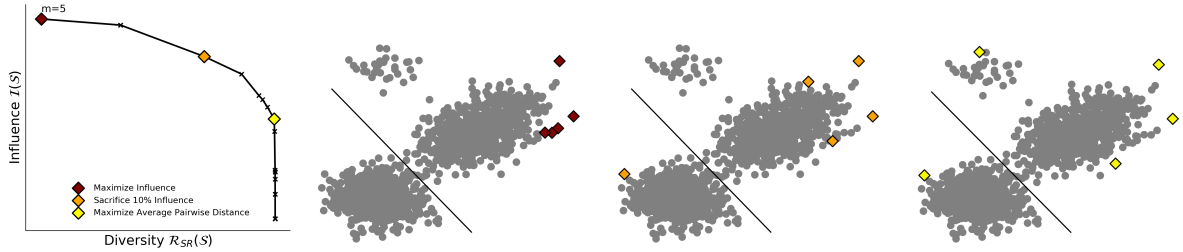
4.1.3 Computational Evaluation

We evaluate our approach on multiple datasets and identify DIVINE points with respect to different evaluation functions. We first visualize the set of DIVINE points found via Algorithm 1 and analyze the trade-off between influence and diversity. We then identify *unfairness-inducing points* and quantify the effect of removing them on model fairness.

We run experiments on the following datasets: synthetic data, LSAT [Kusner et al., 2017], Bank Marketing [Dua and Graff, 2017], COMPAS [Angwin et al., 2016], Adult [Dua and Graff, 2017], a two-class variant of MNIST [LeCun, 1998], and FashionMNIST [Xiao et al., 2017]. We primarily consider logistic loss, $\ell(y, f_\theta(x)) = \log(1+\exp(y\ \theta^\intercal x))$, such that the logistic likelihood is given by $p(y|x) = \sigma(y\ \theta^\intercal x)$ where $\sigma(a) = \frac{1}{1+\exp(a)}$. For all tabular datasets, we use logistic regression and append an intercept to the input features before learning our parameters, $\theta \in \mathbb{R}^d$: this is customary in such settings (i.e. Zafar et al. [2017b] has a similar setup). We learn the classifier's parameters using *scipy.optimize*, using the SLSQP (Sequential Least SQuares Programming) solver and cross-entropy loss. For image datasets, we use *tensorflow* to learn a three-layered multilayer perceptron (for MNIST) and a three-layered convolutional neural network (for FashionMNIST). We then leverage our DIVINE codebase (Appendix B.1) to calculate the DIVINE points for each model.

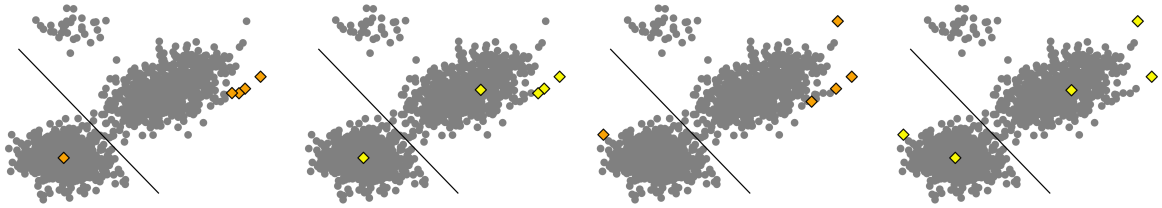
We follow Zafar et al. [2017b] to create a synthetic dataset. We generate $2,000$ binary labels uniformly at random. We then assign a feature vector to each label by sampling from two Gaussian distributions: $p(x|y=1) = \mathcal{N}(x; \mu = [3;3], \Sigma = [2,1;1,2])$ and $p(x|y=-1) = \mathcal{N}(x; \mu = [-2;-2], \Sigma = [1,0;0,1])$. We sample 50 points with label $y=1$ from $p(x|y=$

$1) = \mathcal{N}(x; \mu = [-2; 8], \Sigma = [1, 0; 0, 1])$, and draw a sensitive attribute for each sample from a Bernoulli: $p(A = a) = \frac{p(x'|y=1)}{p(x'|y=1)+p(x'|y=-1)}$, where x' is a rotated version of x.



(a) $\mathcal{I}$-$\mathcal{R}$ Tradeoff (b) DIVINE $\gamma = 0$ (c) DIVINE $\gamma = 161$ (d) DIVINE $\gamma = 434$

Fig. 4.2 We characterize the trade-off between influence and diversity in 4.2a by varying γ. We also visualize the top-5 DIVINE points for select values of γ. The red diamond in 4.2a is $\gamma = 0$, which recovers the top points from IF alone plotted in 4.2b. The orange diamond in 4.2a is for the γ we find such that our DIVINE points have 10% less influence than IF points; these are visualized in 4.2c. The yellow diamond in 4.2a is the γ we find such that we maximize the average pairwise distance between our DIVINE points, seen in 4.2d.

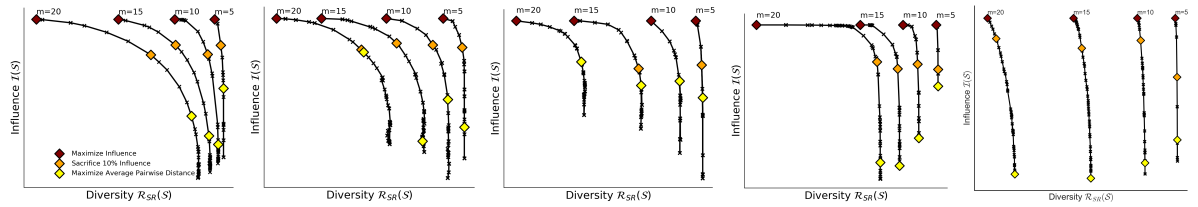


(a) $\mathcal{R}_{\text{FL}}$: $\gamma = 0.526$ (b) $\mathcal{R}_{\text{FL}}$: $\gamma = 0.628$ (c) $\mathcal{R}_{\text{MMD}}$: $\gamma = 201$ (d) $\mathcal{R}_{\text{MMD}}$: $\gamma = 614$

Fig. 4.3 We find DIVINE points with different diversity functions: $\mathcal{R}_{\text{FL}}$ and $\mathcal{R}_{\text{MMD}}$. In Figures 4.3a and 4.3c, we select γ by sacrificing 10% influence. In Figures 4.3b and 4.3d, we select γ by maximizing the pairwise distance between selected points. For $\mathcal{R}_{\text{FL}}$, some points are representative (near cluster centers), but others are redundant (top right). For $\mathcal{R}_{\text{MMD}}$, the selected points are representative, though the top cluster is missed. There are no redundant points, in contrast to points selected with $\mathcal{R}_{\text{FL}}$. Compared to $\mathcal{R}_{\text{SR}}$, these functions select points near cluster centers.

Selecting Diverse Influential Points

First, we validate our approach to greedily select DIVINE points as a global explanation for a logistic regression model trained on our synthetic data. In Figure 4.2, we show how DIVINE values data points using $\mathcal{R}_{\text{SR}}$ and IF with $\mathcal{L}_{\text{loss}}$ on the training data. In Figure 4.2a, we characterize the trade-off between influence and diversity. We obtain the black line by



(a) Synthetic (b) LSAT (c) COMPAS (d) Adult (e) FashionMNIST

Fig. 4.4 For five datasets and four values of $m = \{5, 10, 15, 20\}$, we characterize the influence-diversity trade-off. The red diamond indicates where the top IF points lie. The orange diamond is where 10% of the influence has been foregone for diversity. The yellow diamond is where the average pairwise distance between DIVINE points is maximized.

varying γ. We normalize influence such that we consider how much less influence DIVINE points contain than the top IF points on the y-axis. The red diamond represents $\gamma = 0$, which maximizes influence (i.e. top IF points). We suggest two ways to select γ. One option is to specify a specific amount of influence to sacrifice. In Figure 4.2c, we find γ by specifying that our top-m DIVINE points have 10% less influence than the top-5 IF points; the corresponding point is shown in orange on the trade-off curve. Another option is to find the γ maximizes the average pairwise distance between points in $\mathcal{S}$: $\sum_{u,v \in \mathcal{S}} d(u,v)$, depicted by the yellow diamonds in Figure 4.2d. Both selection mechanisms run a log sweep over $\gamma \in [1e{-}4, 1e5]$ and ensure our set of points has high diversity at the expense of little influence.

We replicate Figure 4.2 with various diversity functions on our synthetic data. Here $m = 5$ and $\mathcal{I}$ is taken to be IF with respect to $\mathcal{L}_{\text{loss}}$. We notice similar trends in Figure 4.3 for $\mathcal{R}_{\text{MMD}}$ and $\mathcal{R}_{\text{FL}}$ as we did for $\mathcal{R}_{\text{SR}}$. We visualize the top-5 DIVINE points for select values of γ. When $\gamma = 0$, $\mathcal{R}$ does not matter. The orange diamonds show the γ where DIVINE points have 10% less influence than IF points. The yellow diamonds represent the γ that maximizes the average pairwise distance between DIVINE points. Notice how both $\mathcal{R}_{\text{MMD}}$ and $\mathcal{R}_{\text{FL}}$ encourage representativeness by selecting a DIVINE point near each Gaussian center. Since $\mathcal{R}_{\text{FL}}$ does not penalize redundancy between points, it selects three points near the top right. As γ approaches ∞, $\mathcal{R}_{\text{MMD}}$ will recover the prototypes of Kim et al. [2016].

We report our trade-off curves for various m from various datasets: Synthetic, LSAT, COMPAS, Adult, and FashionMNIST. We use IF as our influence measure, $\mathcal{R}_{\text{SR}}$ as our diversity function, and $\mathcal{L}_{\text{loss}}$ as our evaluation function. In Figure 4.4, we illustrate the flexibility of DIVINE under changes in model type, input dimensions, and explanation size m. In Figure 4.4a, our trade-off curves vary as we add more DIVINE points ($m = \{5, 10, 15, 20\}$). Each trade-off curve has the same shape as Figure 4.2a, but due to scaling might appear linear when compared to curves for larger m; for example, the rightmost curve in Figure 4.4a is the same as the curve shown in Figure 4.2a. As we increase m, the diversity ($\mathcal{R}_{\text{SR}}$) of the IF points (red) decreases,

implying redundancy in the selected points. This confirms the findings of Barshan et al. [2020]. Practitioners can select anywhere on the black curve to identify positions that trade-off influence and diversity. Our suggested γ selection strategies are shown as yellow and orange diamonds. Over multiple datasets, we notice the same trade-off curve shape.

With COMPAS, the orange and yellow diamonds coincide for multiple values of m. Even with FashionMNIST where the model type is a convolutional neural network, not logistic regression, and the data type is image, not tabular, and where average pairwise distance in input space might not be meaningful, we find that our curves hold (i.e. similar shape to Figure 4.2a). These trade-off curves confirm that DIVINE identifies useful points, as there exists a nearly equal influence explanation that is indeed more diverse in many settings. In Appendix B.4, we show similar curves to Figure 4.4a when valuing points with other methods like DS and when using other diversity measures like $\mathcal{R}_{\text{FL}}$ and $\mathcal{R}_{\text{MMD}}$.

Takeaway: We find that, when optimizing Equation 4.1 and varying γ, we maintain high influence while achieving the desired diversity in our training point importance explanations. We observe this behavior over multiple datasets, multiple explanation sizes (m), and multiple diversity functions ($\mathcal{R}$).

Selecting Unfairness-Inducing Points

In the previous experiments, we value data with respect to loss. We now show how to value data with respect to unfairness, thus helping practitioners analyze where unfairness lies in their models. While Wang et al. [2019] find a counterfactual data distribution that leads to fairer model outcomes, we will now explicitly find points that contribute to a model's unfairness. Here, we define unfairness as the difference in accuracy between subgroups: this is sometimes referred to as (un)equal accuracy. Extension to other group fairness notions is straightforward. Berk et al. [2017] says θ is fair (with respect to equal accuracy) if the following is close to 0: $\mathcal{L}_{\text{unf}}(\theta) = \sum_{j\in\{-1,1\}} |P_{j,a,j} - P_{j,b,j}|$, where $\hat{y} \in \{-1,1\}$ is a binary predicted outcome for some x, y is the actual outcome for x, $A = \{a,b\}$ is a binary sensitive attribute that is contained explicitly in or encoded implicitly in x, $P_{j,a,j} = P(\hat{y} = j | A = a, y = j)$ and $P_{+1,a,+1} = P(\hat{y} = +1 | A = a, y = +1)$ is the true positive rate for Group 1 under θ. When we refer to subgroups, we mean partitions of $\mathcal{D}$ based on A. We take the sum of the absolute difference in true positive rates between subgroups and the absolute difference in true negative rates between subgroups as a measure of unfairness; the larger its magnitude, the more unfair. Practitioners can calculate $\mathcal{L}_{\text{unf}}$ on training, validation, or test data. They may leverage importance scores with respect to $\mathcal{L}_{\text{unf}}$ to identify points hurting their model's fairness. We refer to the points harmful to $\mathcal{L}_{\text{unf}}$ as *unfairness-inducing points*. By removing such points, practitioners can potentially improve model fairness and accuracy.

To check if our method can detect unfairness-inducing points, we construct a toy example. In Figure 4.5, a set of four 2-dimensional data points are drawn on the four corners of a square. The top two points have label -1 ($\times$), and the bottom two points have label 1 ($\bullet$). We assign sensitive attributes by saying the left side is from Group 1 (orange) and the right side is from Group 2 (blue). A fifth point is added farther away in the top left (orange $\times$). A logistic regression model is fit to these five points (dashed line), obtaining perfect accuracy (100%) and ideal unfairness of 0. We inject a poisonous point into the dataset (top right orange $\bullet$). With respect to the original model, this point is incorrectly classified and has an inconsistent sensitive attribute. A logistic regression model is fit to all six points (solid line). This poisoned model gets $\frac{4}{6}$ points correct but has an unfairness of 1.5. We find importance scores for all six points with respect to $\mathcal{L}_{\text{loss}}$ and $\mathcal{L}_{\text{unf}}$. The most influential point with respect to $\mathcal{L}_{\text{loss}}$ is the correctly classified outlier (yellow diamond); however, the most influential point with respect to $\mathcal{L}_{\text{unf}}$ is the poisonous point (red diamond). This demonstrates that $\mathcal{L}_{\text{unf}}$ can detect unfairness-inducing points and does not simply find outliers.

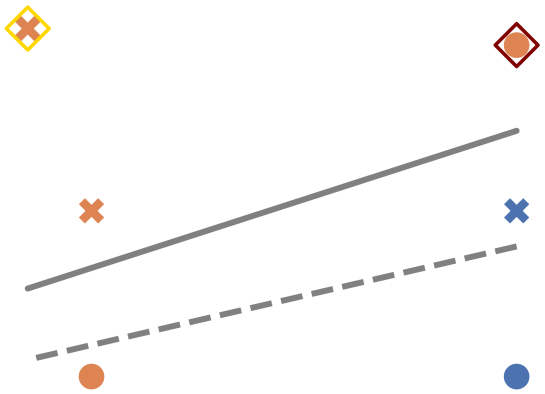


Fig. 4.5 A Toy Example: valuing data using $\mathcal{L}_{\text{unf}}$ and $\mathcal{L}_{\text{loss}}$ yield different influential points.

Once we detect unfairness-inducing points, we may hope to improve our model's fairness outcomes. We consider removing unfairness-inducing points. We first calculate importance scores I_i with respect to $\mathcal{L}_{\text{unf}}$ for each training point. To use Algorithm 1 for removal, we negate each importance score (harmful points now have positive importance), allowing us to perform submodular *maximization* via Equation 4.1. We iteratively select sets of m unfairness-inducing points to remove based on Equation 4.1. We let m be equal to 1% of the training data size. After removing the selected points, we retrain. While more results can be found in [Bhatt et al., 2021b], we plot accuracy and unfairness ($\mathcal{L}_{\text{unf}}$) after removing up to $10\% - 15\%$ of the training data in Figure B.3. For both tabular datasets, $\mathcal{L}_{\text{unf}}$ remains stable or decreases until a large fraction of the dataset has been removed. The corresponding drop in accuracy is minor.

Consider IF on COMPAS: dropping the first 10% of the training points reduces unfairness by almost 70% while only incurring a 2% drop in accuracy. In many cases, a *significant* drop in unfairness can be achieved by dropping a small fraction of the training data points.
Takeaway: DIVINE can be used to identify points that contribute to a model's unfairness. We find that valuing data with respect to unfairness and removing harmful points can improve fairness outcomes.

4.1.4 User Study: Task Simulatability

We conduct user studies to validate how useful DIVINE points are for explanation and task simulatability. For all experiments in this section, we take DIVINE to be IF with $\mathcal{R}_{\text{SR}}$ and find γ by maximizing average pairwise distance.

Experiment Details

We conduct all our user studies on the Prolific platform. We use Google Forms to record our answers. Each study variant had participants, who had a high approval rating on the platform, had computer science experience, and completed a Bachelor's degree. We ran 2 different user studies. Our first study asked about the DIVINE points for FashionMNIST and our second study contained the simulatability experiment. In Appendix B.5, we include the consent form used in our user studies. This user study was performed with the approval of the University of Cambridge's Department of Engineering Research Ethics Committee.

Only two participants who were asked to take the survey did not provide consent and thus exited the form. We still ensured at least 10 participants took each survey variant. The maximum allocated time for either study was 44 minutes. To rule out bogus answers, users were instructed that their responses would be discarded if they finished the task before 5 minutes. The average time spent by a participant was 9 minutes and 53 seconds. In all, we had 40 participants in total, over all studies and variants. Each participant was paid approximately £11.93 per hour. We spent approximately £86 in total. We included an example question, called an "attention check," in each variant: an example is shown in Figure B.13b. The answer to this example question is provided in line. Later in the study, we ask participants this same question. If participants get the attention check wrong, we void their results. We only had to void three results. This did not affect our criteria of ten completed surveys per variant. While the consent form was the same for all study variants, the attention check only appeared in the simulatability user study.

Manually Examining DIVINE Explanations

We consider if DIVINE points are perceived as useful example-based explanations. We conduct a user study to validate the utility of displaying DIVINE points as an explanation for practitioners. We asked 20 participants with computer science experience to do the following.

1. Choose which of top-10 FashionMNIST points from Influence Functions, Prototypes, or DIVINE are more diverse. The question is shown in Appendix Figure B.14a.

2. Rank the trustworthiness of the top-10 FashionMNIST points from different methods from 1 to 5. Answer why they deemed one set of points more trustworthy than the other (min. 10 words). The question is shown in Appendix Figure B.14b.

3. Select which set of top-3 FashionMNIST points from different explanation methods were useful for understanding a misclassified test point. Answer why they deemed one set of points more useful than the other (minimum 10 words).

We prime participants with a randomly selected example from each of the 10 FashionMNIST classes: this ensures they understand the model's intended behavior. When shown the top-5 FashionMNIST points from IF and DIVINE, 100% of participants said DIVINE was more diverse than IF. When shown the top-10 FashionMNIST points from IF and DIVINE, 80% of participants said DIVINE was more diverse. One participant noted that "[DIVINE] seemed to be more distinct and varied," while another said that "with [DIVINE] a more representative selection was used." The same comparison between top-5 (top-10) DIVINE points and prototypes shows that 80% (60%) participants preferred DIVINE.

Next, we displayed 10 FashionMNIST points and asked participants to rate the trustworthiness of the resultant model. We show one set of points from DIVINE and another from IF. Participants provide a trustworthy rating from 1 (Least) to 5 (Most) for each set of points. Asking for the subjective trustworthiness on a Likert scale is common practice [Jacovi et al., 2021; Jeyakumar et al., 2020]. DIVINE, 4.1 ± 0.88, was deemed more trustworthy than IF, 2.6 ± 1.26, ($p = 0.003$, t-test). One participant said "[DIVINE] has more variety so it is more trustworthy." Participants then decided whether IF, RelatIF, or DIVINE provided a useful local explanation for a misclassification on FashionMNIST. Upon seeing the misclassification and the top-3 points from each of the three methods, 50% of participants preferred DIVINE, 30% preferred RelatIF, and 20% preferred IF. One participant said that "the mistake is made obvious since the shape between shirt and coat are shown in [DIVINE]," suggesting our points can be more useful in practice.

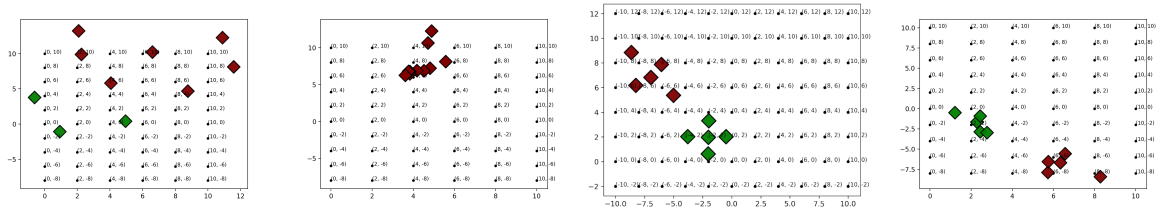
Takeaway: This confirms that not only are DIVINE explanations quantitatively more diverse, but are also qualitatively perceived to be more diverse, trustworthy, and useful.

DIVINE for Task Simulatability

Many user studies for model explainability consider how users can perform forward simulation (i.e. using an explanation to predict the model's behavior on an unseen test point [Doshi-Velez and Kim, 2017; Hase and Bansal, 2020]). While this is a sensible and practical way to validate explanations, they do not provide practitioners with a general sense of how the model behaves [Jeyakumar et al., 2020]. Instead, it is important to consider task **simulatability** [Hoffman et al., 2018; Lipton, 2018], which measures how well a user can reason about an **entire** model given an explanation. To the best of our knowledge, evaluating the importance of training points through task simulatability has not been done before. As it is difficult to measure a practitioner's ability to reason about the entire model, we limit ourselves to the 2D linear setting in this initial inquiry. We posit that diversity in influential samples will help practitioners with simulatability.

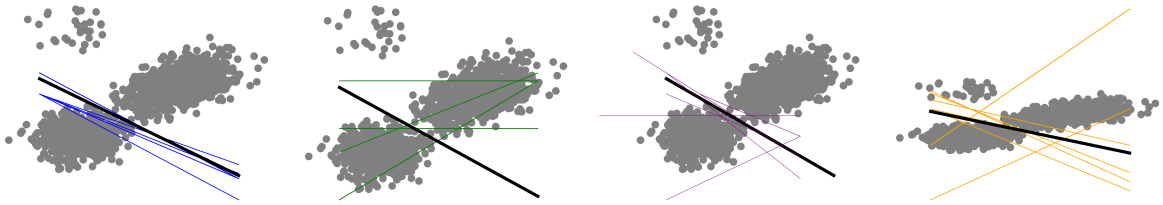
To test the simulatability of DIVINE points, we ask practitioners to reconstruct a model decision boundary given a set of points. Our goal is to measure the similarity between the user-drawn decision boundary and the true decision boundary for a model trained on our synthetic data. To a new set of 20 participants, we display m points on a grid: each point is colored by its predicted class. 10 participants see $m = 5$ points. The rest see $m = 10$. Participants are shown the top-m IF points, top-m DIVINE points, top-m prototypes, or m random points. They are asked to draw a decision boundary that separates the two classes. In Figure 4.6, we show the task setup where the users were asked to guess the classifier boundary given 10 points. Similar task details are shown in Appendix Figure B.15 when $m = 5$. The users were shown the coordinate labels and were asked to enter the coordinates of the start and end points of the inferred line in their response. We then calculate the parameters for the drawn line and compare it to the true decision boundary by calculating the cosine similarity between the drawn and actual boundaries. Note we randomly translate or reflect the points before line drawing to ensure the boundary is not in the same place for all variants. We revert the line drawn between the selected points before calculating the cosine similarity.

In Figure 4.7, we show the user-drawn decision boundaries when shown 10 points from each method. Notice how the boundaries drawn when shown DIVINE points are more similar to the true decision boundary. We represent each user-drawn decision boundary with a slope and a bias term. We then calculate the cosine similarity between the user-drawn decision boundary parameters and the true decision boundary parameters. When shown 5 points, the cosine similarity between the user-drawn boundary and the true boundary is 0.91 ± 0.25 with DIVINE, 0.59 ± 0.78 with IF, 0.37 ± 0.81 with Prototypes, and 0.48 ± 0.81 with Random. When shown 10 points, the cosine similarity between the user-drawn boundary and the true boundary is 0.99 ± 0.00 with DIVINE, is 0.39 ± 0.88 with IF, is 0.60 ± 0.79 with Prototypes,



(a) Top DIVINE Points (b) Top IF Points (c) Top Prototypes (d) Random Points

Fig. 4.6 Task Simulatability. We show the following sets of $m = 10$ points to a user and ask them to draw a decision boundary for each. Users decide upon a decision boundary by selecting two endpoints, which we then translate into a line.



(a) DIVINE (b) IF (c) Prototypes (d) Random

Fig. 4.7 The resulting user-drawn decision boundaries after observing $m = 10$ points.

and 0.33 ± 0.72 with Random. We show the average user-drawn decision boundary (after observing 10 points) in Figure 4.8. Notice that the average decision boundary drawn after observing DIVINE points is closer to the true decision boundary. We find that DIVINE points were considerably more helpful than IF points to participants (p= 0.011, t-test).

We report how the cosine similarity of the true decision boundary and the user-drawn decision boundary after seeing DIVINE points exceeds that of the user-drawn decision boundaries drawn after seeing random points, top IF points, or prototypes. Similar results hold in Figure B.16 for the 10 participants who only saw 5 points from each method. However, note that the cosine similarity was a bit lower implying that it might become easier to simulate the model after seeing more points. Future work can explore how large m needs to be for participants to recover the decision boundary well. While DIVINE points are not optimized for decision boundary reconstruction, it is reassuring to know that they provide sufficiently diverse explanations such that users can reconstruct the decision boundary. Although our study was conducted with synthetic two-dimensional data and linear decision boundaries, our findings take an important step towards showcasing the power of task simulatability within the explainability evaluation community.

Takeaway: We conclude that practitioners find DIVINE points helpful for task simulatability, which enables decision boundary reconstruction directly from an explanation.

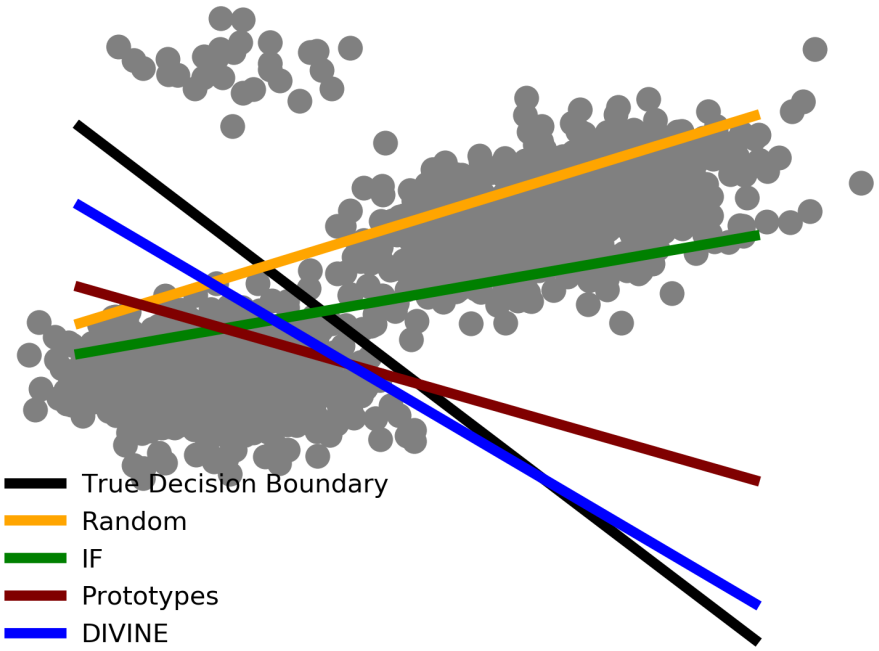


Fig. 4.8 Average User Drawn Decision Boundaries ($m = 10$)

4.1.5 Limitations

In this chapter, we propose an approach for finding DIVerse INfluEntial (DIVINE) training points. Existing training point importance methods tend to assign high importance to similar points; our method selects a diverse and influential subset of training data using submodular optimization. Our approach enables practitioners to obtain more diverse explanations of model behavior. While previous work has mainly investigated influential points with respect to a model's loss, we go further by valuing data points with respect to model fairness. Our experiments on synthetic and real-world datasets demonstrate that by using DIVINE, practitioners can visualize a diverse summary of influential training points and thus understand the possible modes of data that contribute to their model's behavior. In our user studies, we find that practitioners perceive DIVINE points to be more diverse, more useful, and more trustworthy. Practitioners impressively find DIVINE helpful for task simulatability.

We acknowledge that if practitioners do not desire a complete picture of model behavior, they may not want diversity in their training point importance explanations. Referring to local explanations, one participant (Chapter 4.1.4) noted that DIVINE points "may include conflicting and uncomparable [sic] items." Practitioners may want to clarify the goals for using example-based explanations and leverage diversity accordingly. In Appendix B.1, we provide a detailed guide for how practitioners can leverage DIVINE and our codebase. Nonetheless,

practitioners can leverage DIVINE to value training points based on their effect on model-specific evaluation metrics and to summarize model behavior either locally or globally. We hope DIVINE is a helpful tool for practitioners to generate data visualizations and refine their models.

4.2 CLUE: *C*ounterfactual *L*atent *U*ncertainty *E*xplanation

Probabilistic machine learning models aim to provide reliable estimates of uncertainty about their predictions [MacKay, 2003]. These estimates are helpful forms of algorithmic transparency in high-stakes applications, such as recidivism [Zhang and Weiss, 2021] or autonomous vehicles [McAllister et al., 2017]. As discussed in Chapter 2.3.2, well-calibrated uncertainty can be as important as making accurate predictions, leading to increased robustness of automated decision-making systems and helping prevent systems from behaving erratically for out-of-distribution (OOD) test points [Guo et al., 2017]. In practice, predictive uncertainty conveys skepticism about a model's output. However, its utility need not stop there: we posit predictive uncertainty could be rendered more useful and actionable if it were expressed as an explanation in terms of model inputs, answering the question: *"Which input patterns lead my prediction to be uncertain?"* This formulation is reminiscent of feature importance explanations [Lundberg and Lee, 2017; Sundararajan et al., 2017], which asks "Which input features contribute most to model's prediction?" However, feature importance assumes high-accuracy models with predictive certainty.

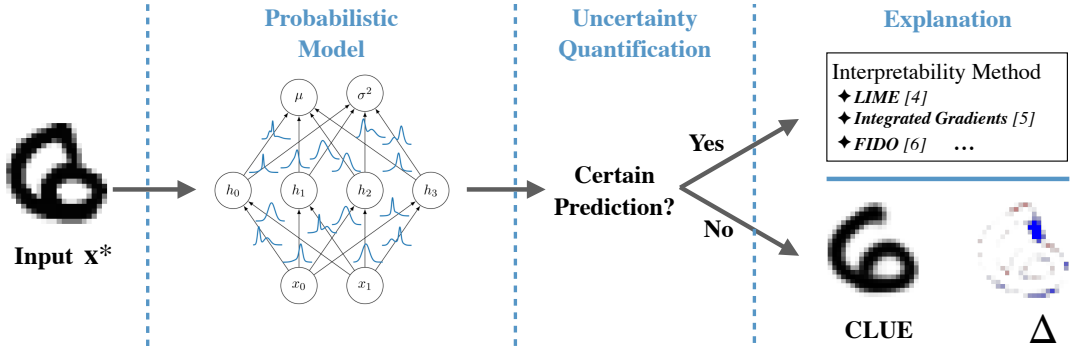


Fig. 4.9 Workflow for automated decision making with transparency. Our probabilistic classifier produces a distribution over outputs. In cases of high uncertainty, CLUE allows us to identify features that are responsible for class ambiguity in the input (denoted by Δ and highlighted in dark blue). Otherwise, we resort to existing feature importance approaches to explain certain decisions.

While explaining predictions from deep models has become a burgeoning field [Bhatt et al., 2020b; Montavon et al., 2018], there has been relatively little research on explaining what leads to neural networks' predictive uncertainty. We introduce Counterfactual Latent Uncertainty Explanations (CLUE), the first approach, to our knowledge, to shed light on the subset of input features that are responsible for uncertainty in probabilistic models. Specifically, we focus on explaining Bayesian Neural Networks (BNNs). We refer to the explanations given by our method as CLUEs. More precisely, CLUEs answer the question: *"What is the smallest change that could be made to an input, while keeping it in distribution, so that our model becomes more certain in its decision for said input?"* CLUEs can be generated for tabular and image data on both classification and regression tasks.

Understanding the input features that are responsible for predictive uncertainty can help practitioners learn in which regions training data is sparse. For example, when training a loan default predictor, a data scientist can identify sub-groups (e.g. by age, gender, race) under-represented in the training data. Collecting more data from these groups, and thus further constraining their model's parameters, could lead to accurate predictions for a broader range of clients. In a clinical scenario, a doctor may use an automated decision-making system to assess whether a patient should receive treatment. In the case of high uncertainty, the system would suggest that the doctor should not rely on its output [Babbar et al., 2022; Madras et al., 2018]. If uncertainty were explained in terms of which features the model finds anomalous, the doctor could appropriately direct their attention.

To maximize transparency of a deployed BNN, we envision CLUE complementing existing approaches to explainability [Chang et al., 2019; Ribeiro et al., 2016; Sundararajan et al., 2017], as shown in Figure 4.9. When the BNN is confident in its prediction, practitioners can generate a feature importance explanation. When the BNN has high uncertainty, CLUE can be used. We proceed to introduce CLUE, an approach that finds counterfactual explanations of uncertainty in input space, by searching in the latent space of a deep generative model (DGM). We put forth an algorithm for generating CLUEs and show how CLUEs are best displayed. We evaluate CLUE quantitatively through comparison to baseline approaches. We then perform a user study, showing that CLUEs allow practitioners to predict for which new inputs a BNN will be uncertain.

4.2.1 Related Work: Calculating Uncertainty

Given a dataset $\mathcal{D}=\{(x^{(i)},y^{(i)})\}_{i=1}^{N}$, a prior on our model's weights $p(\theta)$, and a likelihood function $p(\mathcal{D}|\theta)=\prod_{i=1}^{N} p(y^{(i)}|\, x^{(i)},\theta)$, the posterior distribution over the predictor's parameters $p(\theta|\mathcal{D})\propto p(\mathcal{D}|\theta)p(\theta)$ encodes uncertainty about what value θ should take. Through marginalization, this parameter uncertainty is translated into predictive uncertainty, yielding reliable

Food for Thought: Deploying CLUE

CLUEs explain which features in a data point make a model's prediction, on that data point, uncertain. CLUEs highlight what should change about each feature to increase the model's certainty at a specific data point. For instance, suppose we want to build a credit lending model. Given the financial history and income of a new customer, our model will suggest whether the customer should be granted a loan. A CLUE explanation of an uncertain model prediction suggests features to change to increase the model's certainty for a new customer. The CLUE explanation may suggest that if the customer's income were higher, then the model would be more certain in its prediction on the customer: that uncertainty could stem from data misentry by the customer, data mismatch where the customer differs greatly from the training data, etc. Explaining uncertainty with CLUEs can help practitioners understand where their model falls short and how input features may lead to uncertain predictions.

error bounds and preventing overfitting: $p(y^*|\, x^*, \mathcal{D}) = \int p(y^*|\, x^*, \theta) p(\theta|\mathcal{D})\, d\theta$. For BNNs, both the posterior over parameters and predictive distribution are intractable. Fortunately, there is rich literature concerning approximations of these objects [Gal, 2016; Hernández-Lobato and Adams, 2015; MacKay, 2003]. Herein, we use scale-adapted Stochastic Gradient Hamiltonian Monte Carlo (SG-HMC) [Springenberg et al., 2016]. For regression, we use Gaussian likelihood functions, quantifying uncertainty using their standard deviation, $\sigma(y|x)$. For classification, we take the entropy $H(y|x)$ of categorical distributions as uncertainty. In the rest of this chapter, we use $\mathcal{H}$ to refer to any uncertainty metric, be it σ or H. Bayesian methods enable neural networks (NNs) to be used for uncertainty-aware tasks, such as OOD detection [Daxberger and Hernández-Lobato, 2019], continual learning [Nguyen et al., 2018], active learning [Depeweg et al., 2018], and Bayesian optimization [Springenberg et al., 2016].

Predictive uncertainty can be separated into two components, as shown in Figure 4.10. Each conveys different information to practitioners [Depeweg et al., 2018]. Irreducible or *aleatoric uncertainty* is caused by inherent noise in the generative process of the data, usually manifesting as class overlap. Model or *epistemic uncertainty* represents our lack of knowledge about θ. Stemming from a model being under-specified by the data, epistemic uncertainty arises when we query points off the training manifold or points in sparse regions. Capturing model uncertainty is the main advantage of BNNs over regular NNs.

To the best of our knowledge, the only existing method for explaining uncertainty estimates is Uncertainty Sensitivity Analysis [Depeweg et al., 2017]. This method quantifies the global importance of an input dimension to a chosen metric of uncertainty $\mathcal{H}$ using a sum of linear

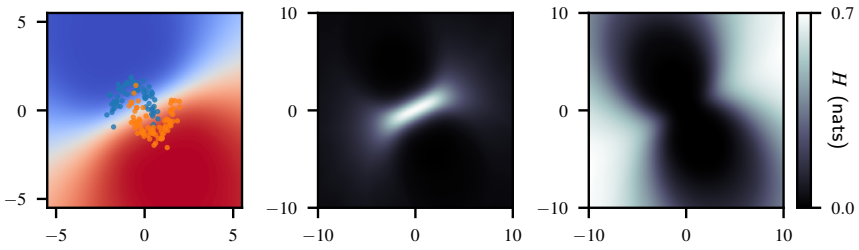


Fig. 4.10 Left: Training points and predictive distribution for variational Bayesian Logistic Regression on the Moons dataset. Center: Aleatoric entropy H_a matches regions of class non-separability. Right: Epistemic entropy H_e grows away from the data. Both uncertainties are detailed in C.1.

approximations centered at each test point:

$$I_i = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{n=1}^{|\mathcal{D}_{\text{test}}|} \left| \frac{\partial \mathcal{H}(y^{(n)}|x^{(n)})}{\partial x_i^{(n)}} \right|. \tag{4.2}$$

As discussed by Rudin [2019], linear explanations of non-linear models, such as BNNs, can be misleading. Even generalized linear models, which are often considered to be "inherently interpretable" (e.g. logistic regression), produce non-linear uncertainty estimates in input space, as seen in Figure 4.10. High-dimensional input spaces also limit the actionability of these explanations, as $\nabla_x \mathcal{H}$ will likely not point in the direction of the data manifold. CLUE leverages the latent space of a DGM to avoid working with high-dimensional input spaces and to ensure explanations are in-distribution without relying on crude linear approximations. The counterfactual nature of CLUE guarantees explanations have tangible meaning.

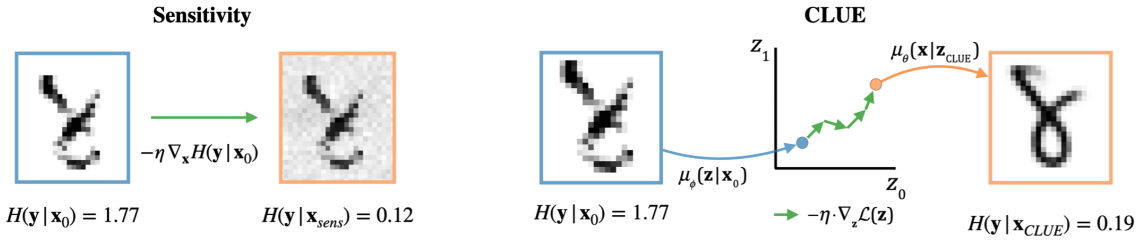


Fig. 4.11 Left: Taking a step in the direction of maximum sensitivity leads to a seemingly noisy input configuration for which H is small. Right: Minimizing CLUE's uncertainty-based objective in terms of a DGM's latent variable z produces a plausible digit with a corrected lower portion.

The term "counterfactual" captures notions of what would have happened if something had been different. ML subcommunities tend to bifurcate in their use of the term. [A] Those in causal inference make causal assumptions about interdependencies among variables and

use these assumptions to incorporate appropriate consequential adjustments when particular variables are set to new values [Kusner et al., 2017; Pearl, 2019]. [B] The explainability community recently used "counterfactual explanations" to explore how input variables must be modified to change a model's output *without* making explicit causal assumptions [Wachter et al., 2017]. We use "counterfactual" in the sense of [B]: we seek small changes to an input in order to reduce its uncertainty without explicit causal assumptions.

Multiple counterfactual explanations can exist for any given input, as the functions we are interested in explaining are often non-injective [Russell, 2019]. Generally, we are concerned with counterfactual input configurations that are close to the original input according to some pairwise distance metric. Naively optimizing for the objective of Wachter et al. [2017] in high-dimensional input spaces may result in the creation of adversarial examples which are not actionable [Goodfellow et al., 2014]. Telling a person that they would have been approved for a loan had their age been -10 is of very little use. To right this, recent work introduces DGMs to ensure explanations are in-distribution [Chang et al., 2019; Dhurandhar et al., 2018a; Joshi et al., 2018; Poyiadzi et al., 2020], a technique commonly used for search in high dimensional spaces [Gómez-Bombarelli et al., 2018; Tripp et al., 2020]. We dub these *auxiliary DGMs*. Others define linear constraints on explanations [Russell, 2019; Sharma et al., 2020]. CLUE avoids the above issues by searching for counterfactuals in the lower-dimensional latent space of an auxiliary DGM. This choice is well suited for uncertainty, as the DGM effectively constrains CLUE's search space to the data manifold. When faced with an OOD input, CLUE returns the nearest in-distribution analog, as shown in Figure 4.11.

4.2.2 Method: Generating CLUEs

Without loss of generality, we use $\mathcal{H}$ to refer to any differentiable estimate of uncertainty (σ or H). We introduce an auxiliary latent variable DGM: $p_\theta(x) = \int p_\theta(x|z)p(z)\,dz$ parameterized by θ. In the rest of this chapter, we will use the decoder from a variational autoencoder (VAE). Its encoder is denoted as $q_\phi(z|x)$. We write these models' predictive means as $\mathbb{E}_{p_\theta(x|z)}[x]{=}\mu_\theta(x|z)$ and $\mathbb{E}_{q_\phi(z|x)}[z]{=}\mu_\phi(z|x)$ respectively.

CLUE aims to find points in latent space that generate inputs similar to an original observation x_0 but are assigned low uncertainty. This is achieved by minimizing Equation 4.3. CLUEs are then decoded as Equation 4.4.

$$\mathcal{L}(z) = \mathcal{H}(y|\mu_\theta(x|z)) + d(\mu_\theta(x|z), x_0), \tag{4.3}$$

$$x_{\text{CLUE}} = \mu_\theta(x|z_{\text{CLUE}}) \;\text{ where }\; z_{\text{CLUE}} = \arg\min_z \mathcal{L}(z). \tag{4.4}$$

The pairwise distance metric takes the form $d(x,x_0) = \lambda_x d_x(x,x_0) + \lambda_y d_y(f(x),f(x_0))$ such that we can enforce similarity between uncertain points and CLUEs in both input and prediction space. The hyperparameters (λ_x,λ_y) control the trade-off between producing low uncertainty CLUEs and CLUEs close to the original inputs. We take $d_x(x,x_0) = \|x-x_0\|_1$ to encourage sparse explanations. For regression, $d_y(f(x),f(x_0))$ is mean squared error. For classification, we use cross-entropy. Note that the best choice for $d(\cdot,\cdot)$ will be task-specific.

The CLUE algorithm and a diagram of our procedure are provided in Algorithm 2 and Figure 4.12 respectively. CLUE can be applied to batches of inputs simultaneously, allowing us to leverage GPU-accelerated matrix computation. The hyperparameter λ_x is selected for each dataset and type of uncertainty by cross-validation. We set λ_y to 0 for our main experiments but explore different values in [Antorán et al., 2021]. We minimize Equation 4.3 with Adam. To facilitate optimization, the initial value of z is chosen to be $z_0 = \mu_\phi(z|x_0)$. Optimization runs for a minimum of three iterations and a maximum of 35 iterations, with a learning rate of 0.1. If the decrease in $\mathcal{L}(z)$ is smaller than $\mathcal{L}(z_0)/100$ for three consecutive iterations, we apply early stopping.

As noted by Wachter et al. [2017], individual counterfactuals may not shed light on all important features. Fortunately, we can exploit the non-convexity of CLUE's objective to address this. We initialize CLUE with $z_0 = \mu_\phi(z|x_0) + \varepsilon$, where $\varepsilon = \mathcal{N}(z;0,\sigma_0 I)$, and perform Algorithm 2 multiple times to obtain different CLUEs. We find $\sigma_0 = 0.15$ to give a good trade-off between optimization speed and CLUE diversity. We show examples of different CLUEs for the same inputs in Appendix C.2. Using similar machinery to Chapter 4.1, we propose mechanisms for selecting a diverse set of CLUEs in [Ley et al., 2022].

Algorithm 2 CLUE

Input: Original data point x_0, Distance function $d(\cdot,\cdot)$, BNN uncertainty estimator $\mathcal{H}$, DGM decoder $\mu_\theta(\cdot)$, DGM encoder $\mu_\phi(\cdot)$
Set initial value of $z = \mu_\phi(z|x_0)$
while Loss $\mathcal{L}$ is not converged **do**
    Decode: $x = \mu_\theta(x|z)$
    Use BNN to obtain $\mathcal{H}(y|x)$
    $\mathcal{L} = \mathcal{H}(y|x) + d(x,x_0)$
    Update z with $\nabla_z \mathcal{L}$
end while
Decode explanation: $x_{\text{CLUE}} = \mu_\theta(x|z)$
Output: Counterfactual example x_{CLUE}

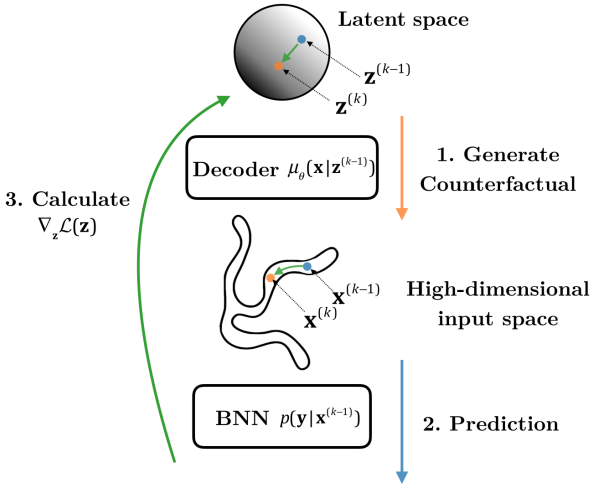
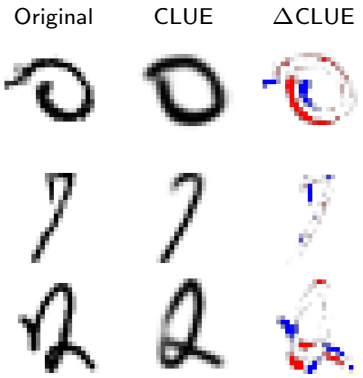


Fig. 4.12 BNN generates uncertainty estimates for inputs decoded from latent codes; its gradients are backpropagated to latent space.

We want to ensure noise from auxiliary DGM reconstruction does not affect CLUE visualization. For tabular data, we use the change in percentile of each input feature with respect to the training distribution as a measure of importance. We only highlight continuous variables for which CLUEs are separated by 15 percentile points or more from their original inputs. All changes to discrete variables are highlighted. For images, we report changes in pixel values by applying a sign-preserving quadratic function to the difference between CLUEs and original samples: $\Delta\text{CLUE} = |\Delta x| \cdot \Delta x$ with $\Delta x = x_{\text{CLUE}} - x_0$. This is showcased in Figure 4.13 and in Figure C.2.



(a) MNIST CLUE

	Original
LSAT	41.0
UGPA	3.7
Race	Asian
Sex	Female

	CLUE
LSAT	41.0
UGPA	3.0
Race	White
Sex	Female

(b) LSAT CLUE

Fig. 4.13 Example image and tabular CLUEs.

4.2.3 Computational Evaluation

We validate CLUE on 5 datasets: LSAT academic performance regression [Kusner et al., 2017], UCI Wine quality regression, UCI Credit classification [Dua and Graff, 2017], a 7 feature variant of COMPAS recidivism classification [Angwin et al., 2016], and MNIST image classification [LeCun, 1998]. For each, we select roughly the 20% most uncertain test points as those for which we reject our BNNs' decisions. We only generate CLUEs for "rejected" points. We also devise a computational evaluation framework for validating CLUEs in [Antorán et al., 2021], wherein we also detail rejection thresholds, model architectures, and hyperparameter settings.

As a baseline, we use a localized version of Uncertainty Sensitivity Analysis. It produces counterfactuals by taking a single step in the direction of the gradient of an input's uncertainty estimates $x_c = x_0 - \eta \nabla_x \mathcal{H}(y|x_0)$. Averaging $|x_0 - x_c|$ across a test set, we recover Equation 4.2. As a second baseline, we adapt FIDO [Chang et al., 2019], a counterfactual feature importance method, to explain uncertainty. We dub it U-FIDO. This method places a binary mask $\mathbf{b}$ over

the set of input variables x_U. The mask is modeled by a product of Bernoulli random variables: $p_\rho(\mathbf{b}) = \prod_{u \in U} \text{Bern}(b_u; \rho_u)$. The set of masked inputs x_B is substituted by its expectation under an auxiliary conditional generative model $p(x_B|x_{U\setminus B})$, fixed to be a VAEAC. U-FIDO finds the masking parameters ρ which minimize Equation 4.5:

$$\mathcal{L}(\rho) = \mathbb{E}_{p_\rho(\mathbf{b})}[\mathcal{H}(y|x_c(\mathbf{b})) + \lambda_b \|\mathbf{b}\|_1], \tag{4.5}$$

$$x_c(\mathbf{b}) = \mathbf{b} \odot x_0 + (1 - \mathbf{b}) \odot \mathbb{E}_{p(x_B|x_{U\setminus B})}[x_B]. \tag{4.6}$$

Counterfactuals are generated by Equation 4.6, where $\odot$ is the Hadamard product. We also compare and contrast CLUE with existing non-counterfactual feature importance methods [Lundberg et al., 2018; Ribeiro et al., 2016] in Appendix C.2.

We compare CLUE, Localized Sensitivity, and U-FIDO following [Antorán et al., 2021]. We would like counterfactuals to explain away as much uncertainty as possible while staying as close to the original inputs as possible. We manage this *informativeness* (large $\Delta\mathcal{H}_{\text{gt}}$ – which captures the average difference in uncertainty between the original and counterfactual examples) and *relevance* (small $\|\Delta\bar{x}\|_1$ – which captures the average distance between the original and counterfactual examples) trade-off with the hyperparameters η, λ_x, and λ_b for Local Sensitivity, CLUE, and U-FIDO respectively. We perform a logarithmic grid search over hyperparameters and plot Pareto-like curves. Our two metrics of interest take minimum values of 0 but their maximum is dataset and method dependent. For Sensitivity, $\|\Delta\bar{x}\|_1$ grows linearly with η. For CLUE and U-FIDO, these metrics saturate for large and small values of λ_x (or λ_b). Thus, the values obtained by these methods do not overlap. As shown in Figure 4.14, CLUE is able to explain away more uncertainty ($\Delta\mathcal{H}_{\text{gt}}$) than U-FIDO, and U-FIDO always obtains smaller values of $\|\Delta\bar{x}\|_1$ than CLUE.

To construct a single performance metric, we scale all measurements by the maximum values obtained between U-FIDO or CLUE, e.g. $(\sqrt{2} \cdot \max(\Delta\mathcal{H}_{\text{gt U-FIDO}}, \Delta\mathcal{H}_{\text{gt CLUE}}))^{-1}$, linearly mapping them to $[0, 1/\sqrt{2}]$. We then negate $\Delta\mathcal{H}_{\text{gt}}$, making its optimum value 0. We consider each method's best performing hyperparameter configuration, as determined by its curve's point nearest the origin, or *knee-point*. The Euclidean distance from each method's knee-point to the origin acts as a metric of relative performance. The best value is 0 and the worst is 1. Knee-point distances, computed across three runs, are shown for both uncertainty types in Table 4.1.

Table 4.1 Relative performance measure obtained by all methods on all datasets under consideration. Lower is better. The dimensionality of each dataset is beside its name. *e* and *a* indicate results for epistemic and aleatoric uncertainty respectively.

Method	LSAT (4)		COMPAS (7)		Wine (11)		Credit (23)		MNIST (784)	
	e	a	e	a	e	a	e	a	e	a
Sensitivity	0.70	0.67	**0.71**	**0.13**	0.69	0.03	0.63	0.50	0.66	0.68
CLUE	0.52	0.64	**0.71**	0.18	**0.01**	0.14	0.52	**0.29**	**0.26**	**0.27**
U-FIDO	**0.36**	**0.51**	**0.71**	0.31	0.22	**0.02**	**0.45**	0.63	0.38	0.50

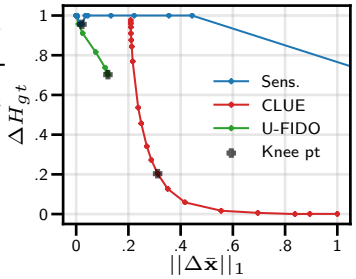


Fig. 4.14 MNIST knee-points for various methods.

Local Sensitivity performs poorly on all datasets except COMPAS. We attribute this to only two features being necessary to predict targets [Dressel and Farid, 2018]. U-FIDO's input space masking mechanism allows for counterfactuals that leave most features unchanged. It performs well in low dimensional problems but leads to high variance as dimensionality grows. CLUE performs best on higher dimensional datasets. Latent space optimization makes CLUE more robust to input space complexity. We can also simply replace input space similarity $\|\Delta\bar{x}\|_1$ with proximity to the data manifold. Therein, CLUE produces the most in-distribution counterfactuals, performing best in 8 out of 10 experiments.

We can also study CLUE's reliance on its auxiliary DGM. We compare Algorithm 2's encoder-based initialization $z_0 = \mu_\phi(z|x_0)$ with $z_0 = \mathbf{0}$. As shown in Figure 4.15, for high dimensional datasets, like MNIST, initializing z with the encoder's mean leads to CLUEs that require smaller changes in input space to explain away similar amounts of uncertainty (i.e. more *relevant*). Similar behavior is observed for Credit, our second highest dimensional dataset, in [Antorán et al., 2021]. On other datasets, both approaches yield indistinguishable results. This shows CLUEs can be generated with differentiable DGMs that lack an encoding mechanism, such as GANs. These could prove useful when dealing with more complex data.

Figure 4.15 also shows how auto-encoding uncertain MNIST samples with low-capacity VAEs significantly reduces these points' predictive entropy. CLUEs generated with these VAEs highlight features that the VAEs are unable to reproduce but are not reflective of our BNN's uncertainty. This results in large values of $\|\Delta x\|_1$; although our counterfactual examples are indeed more certain than the original samples, they contain unnecessary changes. As our auxiliary DGMs' capacity increases, the amount of uncertainty preserved when auto-encoding inputs increases as well. $\|\Delta x\|_1$ decreases while the predictive entropy of our CLUEs stays the same. More expressive DGMs allow for generating sparser CLUEs. Fortunately, even in scenarios where our predictor's training dataset is limited, we can train powerful DGMs by leveraging unlabeled data.

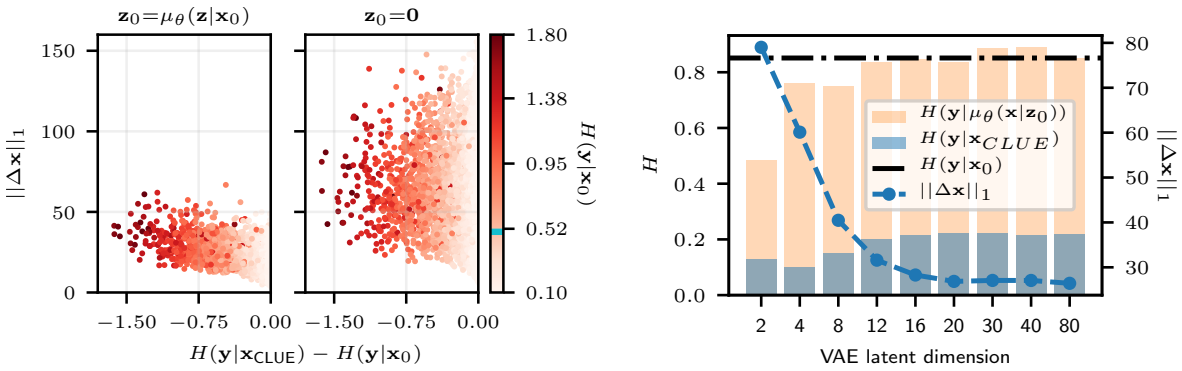


Fig. 4.15 Left: CLUEs are similarly *informative* under encoder-based and encoder-free initializations. The color bar indicates the original samples' uncertainty. Its horizontal blue line denotes our rejection threshold. Right: Auxiliary DGMs with more capacity result in more *relevant* CLUEs.

Takeaway: In terms of informativeness and relevance, CLUE performs well on high-dimensional datasets compared to baselines. Altering the underlying DGM can improve the quality of the CLUEs we generate.

4.2.4 User Study: Forward Simulation

As suggested by Chapter 2, in-context human subject evaluation is a key step in validating the utility of tools for explainability [Hoffman et al., 2018]. We want to assess the extent to which CLUEs help machine learning practitioners identify sources of uncertainty in ML models compared to using simple linear approximations (Local Sensitivity) or human intuition. To do this, we propose a forward-simulation task [Doshi-Velez and Kim, 2017], focusing on an appropriate local test to evaluate CLUEs. We show practitioners one data point below our "rejection" threshold and one data point above. The former is labeled as "certain" and the latter as "uncertain;" we refer to these as *context points*. The certain *context point* serves as a local explanation for the uncertain *context point*. Using both *context points* for reference, practitioners are asked to predict if a test point will be above or below our threshold (i.e. will our BNN's uncertainty be high or low for the new point). Our user study compares the helpfulness of the certain *context points* generated by CLUE relative to those from baselines.

In our study, we compare four different methods, varying how we select certain *context points*. We either (1) select a certain point at random from the test set as a control, generate a counterfactual certain point with (2) Local Sensitivity or with (3) CLUE, or (4) display a human-selected certain point (*Human CLUE*). To generate a *Human CLUE*, we ask participants (who will not participate in the main study) to pair uncertain *context points* with similar certain points. We select the points for our main study using a pilot procedure similar to Grgic-Hlaca

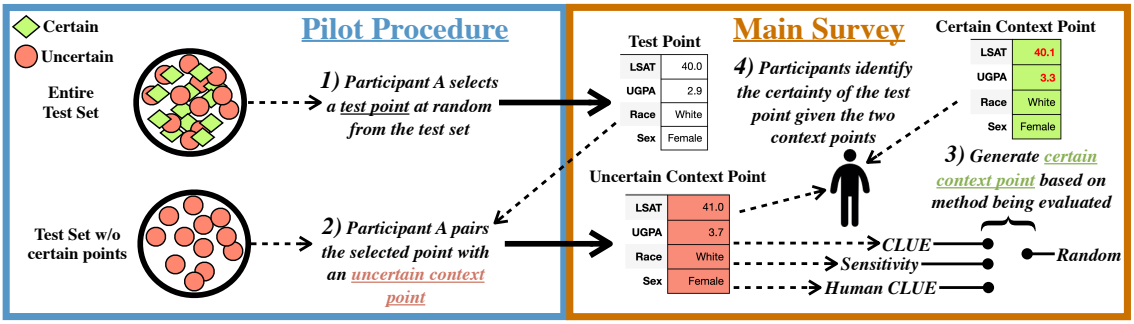


Fig. 4.16 Experimental workflow for our tabular data user study.

	Uncertain
Age	Less than 25
Race	Caucasian
Sex	Male
Current Charge	Misdemeanour
Reoffended Before	Yes
Prior Convictions	1
Days Served	0

	Certain
Age	Less than 25
Race	African-American
Sex	Male
Current Charge	Misdemeanour
Reoffended Before	No
Prior Convictions	0
Days Served	0

	?
Age	Less than 25
Race	Hispanic
Sex	Male
Current Charge	Misdemeanour
Reoffended Before	No
Prior Convictions	0
Days Served	0

Fig. 4.17 Example question shown to main study participants for the COMPAS dataset: *Given the uncertain example on the left and the certain example in the middle, will the model be certain on the test example on the right?* The red text highlights the features that differ between *context points*.

et al. [2018]. This procedure, shown in Figure 4.16, prevents us from injecting biases into point selection and ensures *context points* are relevant to test points. In our procedure, a participant is shown a pool of randomly selected certain and uncertain points. We ask this participant to select points from this pool: these will be test points. We then ask the participant to map each selected test point to a similar uncertain point without replacement. In this way, we obtain uncertain *context points* that are relevant to test points.

We use the LSAT and COMPAS datasets in our user study. Ten different participants take each variant of the main study: our participants are ML graduate students, who serve as proxies for industry practitioners. The main study consists of 18 questions, 9 per dataset. An example question is shown in Figure 4.17. The average participant accuracy by variant is CLUE (82.22%), *Human CLUE* (62.22%), Random (61.67%), and Local Sensitivity (52.78%). We measure the statistical significance of CLUE's superiority with unpaired Wilcoxon signed-rank tests [Demšar, 2006] of CLUE vs. each baseline. We obtain the following p-values: Human-CLUE ($2.34e{-}5$), Random ($1.47e{-}5$), and Sensitivity ($2.60e{-}9$). Our between-group experimental design could potentially lead to dependent observations, which violates the tests'

assumptions. However, the p-values obtained are extremely low, providing confidence in rejecting the null hypothesis of equal performance among approaches.

We also consider the breakdown of accuracy by dataset and by test point certainty in Table 4.2. CLUE outperforms all baselines on both datasets. We find that sensitivity does significantly worse in higher dimensions (on COMPAS). When splitting by the certainty of test points, we immediately notice that accuracy for uncertain test points is quite high for all methods. This similarity is expected since certain context points are the only factor that varies between each variant. Study participants seemed to ignore the certain context points to identify uncertain test points. This is probably due to the pilot procedure, wherein Participant A carefully paired test points with relevant uncertain context points. Indeed, the random baseline, which controls for the possibility that our task can be solved without access to a relevant counterfactual, performs best on uncertain test points. However, we note a large difference between the methods' results when identifying certain test points. CLUE's accuracy almost doubles the second best method's (*Human CLUE*). When generating *Human CLUE*s Participant B had knowledge of the uncertain context point, but not the test point (just like other methods). For this reason, we expect to see dissimilarity in methods' performance on certain test points. CLUE's ability to bring about the most relevant contrast is one possible explanation for why it does so much better than baselines for certain context points.

Table 4.2 Accuracy (%) of participants on the Tabular main study broken down by dataset and by the certainty of test points.

	Combined	LSAT	COMPAS	Certain Test	Uncertain Test
CLUE	**82.22**	**83.33**	**81.11**	**71.00**	96.25
Human CLUE	62.22	61.11	63.33	38.00	92.50
Random	61.67	62.22	61.11	31.00	**100**
Local Sensitivity	52.78	56.67	48.89	20.90	92.50

We find that linear explanations (Local Sensitivity) of a non-linear function (BNN) mislead practitioners and perform worse than random. While *Human CLUE* explanations are real data points, CLUE generates explanations from a VAE. We conjecture that CLUE's increased flexibility produces relevant explanations in a broader range of cases. In our tabular data user study, we only show one pair of *context points* per test point. We find that otherwise, the study is difficult for practitioners to follow, due to non-expertise in college admissions or criminal justice. Using MNIST, we run a smaller scale experiment, wherein we show participants larger sets of *context points*. Results are in C.3.

Takeaway: When compared to exisiting methods, CLUE helps decision-makers better assess whether a BNN will be certain in its prediction on an unseen test point.

4.2.5 Limitations

Both uncertainty estimation and explainability are important factors for trustworthy machine learning. To date, there is little work at the intersection of these two areas. In this chapter, we address this gap by proposing a novel method for interpreting uncertainty estimates from differentiable probabilistic models, like Bayesian Neural Networks (BNNs). Our method, Counterfactual Latent Uncertainty Explanations (CLUE), indicates how to change an input while keeping it on the data manifold, such that a BNN becomes more confident about the input's prediction. The success of CLUE hinges on the fidelity of the DGM used and the calibration of the uncertainty estimates from the probabilistic model: each of these strong assumptions is a research area in its own right. Nonetheless, our computational and human subject experiments show that CLUE outperforms baselines and enables practitioners to better understand which input patterns are responsible for predictive uncertainty.

4.3 Summary

In this chapter, we looked at how algorithmic transparency can be used in decision-making contexts. In Chapter 4.1, we propose DIVINE points as an example-based explanation of which training points, across input space, affect a model's loss the most. We show that these points help decision-makers simulate the model's decision boundary better than points from existing methods. In Chapter 4.2, we devise CLUE, a feature-based explanation method that identifies which input features, if perturbed, would reduce the model's uncertainty on a given input. We demonstrate how these explanations can help decision-makers identify the uncertainty of unseen data points. While each method is successful in its own right, crafting each method can be laborious and may not be useful in general settings; thus in the subsequent chapter, we call into question the need for always providing algorithmic transparency. We consider showing decision-makers model predictions and/or explanations (both of which are forms of decision support) based on what would improve each decision-maker's outcomes [Bhatt et al., 2023]. Our goal is to personalize access to decision support in a manner that provides algorithmic transparency when and as needed.

Chapter 5

Personalizing Access to Decision Support

In the last two chapters, we focused on the impact of algorithmic transparency, specifically explainability, on organizations (Chapter 3) and individual decision-making (Chapter 4). We find that not only are organizations struggling to use algorithmic transparency but also individuals may only benefit from transparency in settings where the transparency is hand tailored to their needs. More generally, we call into question whether algorithmic transparency is a good form of decision support. We are interested in understanding the settings under which decision-making improves after a decision-maker leverages a form of support, be it algorithmic transparency (e.g. explanations or uncertainty estimates) or model predictions.

Individual human decision-makers may benefit from different forms of support to improve decision outcomes, but *which* form of support will yield better outcomes? In this chapter, we propose the general problem of learning a *decision support policy* that, for a given input, chooses which form of support to provide to decision-makers for whom we initially have no prior information. Using techniques from stochastic contextual bandits, we introduce `THREAD`, an online algorithm to personalize a decision support policy for each decision-maker. We further propose a variant of `THREAD` for the multi-objective setting to account for auxiliary objectives like the cost of support. We find that `THREAD` can learn a personalized policy that outperforms offline policies and reduces the incurred cost with minimal degradation to performance in the cost-aware setting. Our experiments include various realistic forms of support (e.g. expert consensus, output of a large language model) on vision and language tasks. We deploy `THREAD` with real users to show how personalized policies can be learned online and illustrate nuances of learning decision support policies in practice.

5.1 Introduction

To improve decision outcomes, human decision-makers can use various forms of support to inform their opinions before making a final decision [Keen, 1980]. Decision-makers with differing expertise may benefit from different forms of support on a given input. For example, one radiologist may provide a better diagnosis of a chest X-ray by leveraging model predictions [Kahn Jr, 1994], while another may excel after viewing suggestions from senior radiologists [Briggs et al., 2008] or seeing a summary of relevant medical records from a large language model (LLM) [Yang et al., 2023a]. In this chapter, we study how to improve decision outcomes by *personalizing* which form of support we provide to a decision-maker on a case-by-case basis.

We formalize learning a *decision support policy* that dictates for each individual decision-maker what additional support, if any, should be presented for a given input (Figure 5.1). While prior work has assumed access to offline human decisions under support [Charusaie et al., 2022; Laidlaw and Russell, 2021] or oracle queries of human behavior [De-Arteaga et al., 2021; Mozannar and Sontag, 2020] to learn decision support policies, we argue that this data is unrealistic to obtain in practice across all available forms of support *for a new decision-maker*. Thus, for individuals for whom we have no prior information initially, we propose learning how to personalize support *online*. We introduce `THREAD`, an algorithm for learning decision support policies online, to sequentially estimate the prediction error of a decision-maker under each form of support leveraging techniques from the stochastic contextual bandit literature [Li et al., 2010].

Since providing support may be expensive [Arbiser et al., 2001; Mimra et al., 2016], it may also be important to consider a second objective regarding the cost of the selected support. We show how `THREAD` can be easily modified to account for a trade-off parameter between performance and cost. However, since it can be challenging to determine an appropriate choice of the trade-off parameter for an unseen decision-maker, we develop a hyper-parameter tuning strategy to identify a trade-off parameter using data from a population of decision-makers. This strategy involves finding the set of optimal parameters that minimize the incurred cost of support for each decision-maker while ensuring that a specified threshold is satisfied: we then select a parameter from this set to deploy on a new decision-maker.

We conduct computational experiments to explore the utility of `THREAD` in both the standard and cost-aware settings. In the standard cost-agnostic setting, we characterize the expertise of decision-makers for whom `THREAD` successfully learns policies that outperform offline policies, which are not personalized to the strengths of a new decision-maker, and, if there is no benefit of personalization, `THREAD` recovers fixed policies, which always provide one form of support.

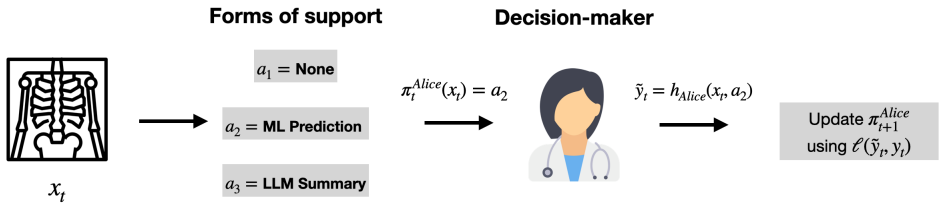


Fig. 5.1 We illustrate the process of learning a decision support policy π_t online to improve a decision-maker Alice's performance. For every decision x_t, our policy selects a form of support $\pi_t(x_t)$ from a set $\mathcal{A} = \{a_1, a_2, a_3\}$. The policy is updated using $\ell(\tilde{y}_t, y_t)$, the loss incurred after observing Alice's decision $\tilde{y}_t$. Without assuming access to prior offline data, our formulation learns a personalized policy *online*. This means Alice's learned policy may differ from that of another decision-maker, Bob, if they have different expertise. We also consider a cost-aware version of this learning problem (e.g. it is more expensive to elicit an LLM summary).

In the cost-aware setting, we show that `THREAD` can reduce the incurred cost with minimal degradation to performance.

We also conduct human subject experiments ($N = 125$) to test our proposed method on Prolific crowdworkers. We develop `Modiste`,[1] an interactive tool that gives `THREAD` an interface. In contrast to prior work that only tests offline policies or evaluates in simulation, we demonstrate how `Modiste` can be used to learn personalized decision support policies online on both vision and language tasks. We explore forms of support that include expert consensus, outputs from an LLM, or predictions from a classification model. Our findings highlight the importance of running human subject experiments to validate any proposed decision support algorithm and discuss the implications of deploying decision support policies in practice. We emphasize our main contributions:

1. Formalizing decision support policies and their associated algorithms. We propose a formulation for learning a personalized decision support policy that selects the form of support that maximizes a given decision-maker's performance. We introduce `THREAD`, an online algorithm to learn such a policy, and provide a multi-objective extension to incorporate the cost of support. We introduce a hyper-parameter tuning strategy, which finds a trade-off parameter that minimizes cost and achieves a specified performance threshold. In this chapter, we also begin to characterize under which settings we would expect `THREAD` to improve performance and/or cost.

2. Evaluating decision support policies in realistic settings. We demonstrate the importance of online learning to personalize policies to new decision-makers through both computational and human subject experiments on vision and language tasks. Our human

[1] While a "modiste" usually refers to someone who uses thread to tailor clothing and make dresses/hats, we use the term to capture our tool's ability to alter a policy to a decision-maker.

subject experiments, where real users interact with our new tool `Modiste`, validate our findings from the computational experiments on synthetic decision-makers, demonstrating that `THREAD` can yield benefits in practice.

Food for Thought: Deploying Modiste

Modiste is a tool to learn a decision support policy online. Modiste helps users know when AI assistance is appropriate for a specific individual. For instance, suppose we have a general purpose AI system like a large language model (LLM). We would like to provide users with access to the LLM selectively. Given repeated interactions with a previously unseen user for a certain task, Modiste uses our algorithm, THREAD, to decide when LLM support is worthwhile for that user. By learning from user interactions, Modiste can optimize for joint human-LLM team performance on tasks like writing emails, answering quiz questions, or proving theorems. Providing the LLM as assistance to all users all of the time may undermine user expertise/agency and expose users to potential errors in the LLM outputs. Modiste tailors seamless human-machine teams.

5.2 Related Work

Decision Support. While various forms of decision support have been proposed, such as expert consensus [Scheife et al., 2015] and changes to machine interfaces [Roda, 2011], more recent forms of support focus on algorithmic tools where decision-makers are aided by machine learning (ML) models [Bastani et al., 2022; Gao et al., 2021; Phillips-Wren, 2012]. In some prior work, the human does not always make the final decision, such as those that learn to defer decisions from a model to a single decision-maker [Madras et al., 2018; Mozannar and Sontag, 2020] or others that jointly learn an allocation function between a model and a pool of decision makers [Hemmer et al., 2022; Keswani et al., 2021]. In our setting, the *human* is always the decision-maker, even after viewing any form of support: this reflects many real-world applications. This setting includes ones where humans make the final decisions with support from ML models [Green and Chen, 2019; Lai et al., 2023] , as well as ones where humans make decisions when provided with additional information beyond a model prediction (e.g. explanations [Bansal et al., 2021b], uncertainty estimates [Zhang et al., 2020b], conformal sets [Babbar et al., 2022]). However, these studies *always* show a single form of support. We not only consider the forms of support in these studies but also formalize *learning* in what contexts each form of support should be provided. A detailed comparison to prior work is in Appendix D.1.

Prior Assumptions About Decision-Maker Information. We briefly survey the assumptions made about the decision-maker when learning decision support policies. The model of the decision maker is either synthetic, thus lacking grounding in actual human behavioral data, or learned from a batch of *offline* annotations [Charusaie et al., 2022; Gao et al., 2022; Madras et al., 2018; Okati et al., 2021]. For a new decision-maker or a new form of support, this set of data would not be available in practice. Instead, we propose to learn a decision support policy *online* to circumvent these limitations. There are few works that use some aspect of online learning for different decision-making settings or under strict theoretical conditions, as we describe in Appendix D.1.

Online Learning. Learning decision support policies can be done with bandit feedback, where at each time step the learner receives a reward based on a selected action. Our set-up is a case of stochastic contextual bandits [Li et al., 2010], where the reward depends on an i.i.d sampled context. In our setting, the contexts are the input, the actions represent the forms of support, and the reward depends on a decision-maker's decision, which in turn provides bandit feedback online. We propose an algorithm to learn decision support policies that can use existing techniques from the contextual bandits literature and evaluate two existing methods (LinUCB [Li et al., 2010] and KNN-UCB [Guan and Jiang, 2018]). While we extend the set-up to include a multi-objective problem to incorporate both performance and cost, we cannot directly apply prior works on multi-objective contextual bandits [Tekin and Turğay, 2018; Turgay et al., 2018] because they make theoretical assumptions that are not applicable in our setting, described in Appendix D.1. Instead, we reduce the multi-objective to a single-objective problem and propose a practical hyper-parameter tuning strategy.

5.3 Formulating and Learning Decision Support Policies

General Problem Formulation. We consider a human decision-making process with different forms of decision support. In particular, we focus on a multi-class classification problem with observation/feature space $\mathcal{X} \subseteq \mathbb{R}^p$ and outcome/label space $\mathcal{Y} = [K]$. We concentrate on a stochastic setting where the data $(x, y) \in \mathcal{X} \times \mathcal{Y}$ are drawn iid from a fixed, unknown data generating distribution $\mathcal{P}$, an assumption that reflects typical decision-making settings [Bastani and Bayati, 2020; Bastani et al., 2022]. In addition, we consider an action set $\mathcal{A}$ corresponding to the forms of support available.[2] Given an observation $x \in \mathcal{X}$, the human attempts to predict the corresponding label $y \in \mathcal{Y}$ using the support prescribed by an action $a \in \mathcal{A}$, i.e. the human makes the prediction using an unknown function $h : \mathcal{X} \times \mathcal{A} \to \mathcal{Y}$. *Our problem formulation and*

[2] A form of support may consist of an individual piece of information (e.g. model prediction) or a particular combination of multiple pieces of information (e.g. model prediction and explanation).

algorithms to solve the formulation are agnostic to the specific forms of support; we provide multiple instantiations in later human subject experiments. The quality of the final prediction made by the human is measured by a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0,1]$ with respect to the true label. We consider a 0-1 loss function, where $\ell(y,y') = 1$ for $y \neq y'$ and $\ell(y,y') = 0$ for $y = y'$. Other loss functions may be more appropriate in different settings.

Decision-Making Protocol. We aim to learn a policy $\pi : \mathcal{X} \to \Delta(\mathcal{A})$ that picks the appropriate form of support to assist a new decision-maker. Let Π denote the class of all stochastic decision support policies. Let $\mathcal{A} = \{\mathrm{A}_1, \ldots, \mathrm{A}_k\}$, and $\pi(x)_{\mathrm{A}_i}$ denote $\mathbb{P}[\mathrm{A}_i \sim \pi(x)]$ for each $\mathrm{A}_i \in \mathcal{A}$. When the policy π prescribes the support A_i, the human decision-maker makes the prediction $\widetilde{y}$ based on the observation x and support A_i, i.e. the final prediction $\widetilde{y}$ is given by $\widetilde{y} = h(x, \mathrm{A}_i)$. The human decision-making process with different forms of support is described below. For $t = 1, 2, \ldots, T$:

1. A data point $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ is drawn iid from $\mathcal{P}$.
2. A form of support $a_t \in \mathcal{A}$ is selected using a decision support policy $\pi_t : \mathcal{X} \to \Delta(\mathcal{A})$.
3. The human decision-maker makes the final prediction $\widetilde{y}_t = h(x_t, a_t)$ based on x_t and a_t.
4. The human decision-maker incurs a loss $\ell(y_t, \widetilde{y}_t) = 1$ if $y_t \neq \widetilde{y}_t$ and $\ell(y_t, \widetilde{y}_t) = 0$ otherwise.

A static decision support policy would not change throughout the interaction with the decision-maker. This reflects many prior user studies, per Lai et al. [2023]; for example, decision-makers may *always* be presented with a feature importance explanation alongside a model prediction. It is possible to update a policy π_t iteratively at each time step based on the most recent interaction with the decision-maker using $\{(x_t, y_t, a_t, \widetilde{y}_t)\}$. We propose a method to make such an update online in the subsequent section.

Evaluation of π via Expected Loss. The quality of a policy π can be evaluated using the expected loss incurred by the decision-maker across the input space:

$$L_h(\pi) \;=\; \mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\mathbb{E}_{\mathrm{A}_i\sim\pi(x)}[\ell(y, h(x, \mathrm{A}_i))]\right]. \tag{5.1}$$

We distinguish this metric from the more standard notion of regret, which is typically used to analyze policies in an online learning setting [Li et al., 2010]; however, we cannot realize π^* for an unseen decision-maker in practical scenarios. Thus, we rely on $L_h(\cdot)$ as a proxy metric for evaluating the effectiveness of π.

In the standard setting, our goal is to find an optimal decision support policy π^* that minimizes $L_h(\pi)$. We first rewrite Equation 5.1 as follows: $L_h(\pi) = \mathbb{E}_x\left[\sum_{i=1}^{k} \pi(x)_{\mathrm{A}_i} \cdot r_{\mathrm{A}_i}(x;h)\right]$, where $r_{\mathrm{A}_i}(x;h) = \mathbb{E}_{y|x}[\ell(y, h(x, \mathrm{A}_i))]$ is the human prediction error for input x and support A_i. Then, it can be shown that the optimal policy takes the form $\pi^*(x) = \arg\min_{\mathrm{A}_i \in \mathcal{A}} r_{\mathrm{A}_i}(x;h)$ (see

Appendix D.2). This simplified form of π^* enables us to propose an efficient strategy to learn the policy online by estimating $r_{\mathrm{A}_i}(x;h)$ values via interaction with the decision-maker h. This approach is more tractable than identifying the appropriate modeling assumptions for each h under all forms of support [Kim et al., 2008].

5.3.1 THREAD: Learning Personalized Policies

The human decision-making process with various forms of support can be modeled as a stochastic contextual bandit problem, where the forms of support are the arms and $\mathcal{X}$ is the context space. Leveraging this insight, we introduce THREAD, outlined in Algorithm 3. THREAD is an online algorithm for learning the policy π^* that minimizes the expected loss.

Algorithm 3 THREAD

Input: human decision-maker h
Initialization: data buffer $\mathcal{D}_0 = \{\}$; human error values $\{\widehat{r}_{\mathrm{A}_i,0}(x;h) = 0.5 : x \in \mathcal{X}, \mathrm{A}_i \in \mathcal{A}\}$; initial policy π_1
for $t = 1, 2, \ldots, T$ **do**
  data point $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ is drawn iid from $\mathcal{P}$
  support $a_t \in \mathcal{A}$ is selected using policy π_t
  human makes the prediction $\widetilde{y}_t$ based on x_t and a_t
  human incurs the loss $\ell(y_t, \widetilde{y}_t)$
  update the buffer $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(x_t, a_t, \ell(y_t, \widetilde{y}_t))\}$
  update the decision support policy:

$$\widehat{r}_{\mathrm{A}_i,t}(x;h) \leftarrow \mathcal{U}_r(\widehat{r}_{\mathrm{A}_i,t-1}(x;h), \mathcal{D}_t), \quad \forall \mathrm{A}_i \in \mathcal{A} \qquad \text{(Step 1)}$$

$$\pi_{t+1}(x) \leftarrow \mathcal{U}_\pi(\{\widehat{r}_{\mathrm{A}_i,t}\}_i) \qquad \text{(Step 2)}$$

end for
Output: policy $\pi_\lambda^{\mathrm{alg}} \leftarrow \pi_{T+1}$

$\mathcal{U}_r$: Update Step 1. We discuss two approaches to estimate $r_{\mathrm{A}_i}(x;h)$ for all $x \in \mathcal{X}$ and $\mathrm{A}_i \in \mathcal{A}$, but note that any online learning algorithm can be used. We first consider **LinUCB** [Li et al., 2010], a common online learning algorithm that approximates the expected loss $r_{\mathrm{A}_i}(x;h)$ by a linear function $\widehat{r}_{\mathrm{A}_i}(x;h) := \langle \theta_{\mathrm{A}_i}, x \rangle$. Although the linearity assumption may not hold in general, we learn the parameters $\{\theta_{\mathrm{A}_i} : \mathrm{A}_i \in \mathcal{A}\}$ using LinUCB with the instantaneous reward function $R(x, y, \mathrm{A}_i; h) := -\ell(y, h(x, \mathrm{A}_i))$. We then normalize the resulting $\widehat{r}_{\mathrm{A}_i}(x;h)$ values to lie in the range $[0, 1]$. The second algorithm we use is an intuitive K-nearest neighbor (**KNN**) approach, which is a simplified variant of KNN-UCB [Guan and Jiang, 2018]. Here, we maintain an evolving data buffer $\mathcal{D}_t$, which accumulates the history of interactions with the decision-maker.

For any new observation x, we estimate the $\widehat{r}_{\mathrm{A}_i}(x;h)$ values by finding its K-nearest neighbors in $\mathcal{D}_t$ and computing the average error of these neighbors.

$\mathcal{U}_\pi$: Update Step 2. While we could use a pure exploratory policy π_{t+1} in Step 2 of Algorithm 3, where $\pi_{t+1}(x)_{\mathrm{A}_i} = 1/|\mathcal{A}|$ for all $\mathrm{A}_i \in \mathcal{A}$, this approach would require a large number of interactions T to achieve accurate error estimates. In practical settings, where interactions are limited, as in our human subject experiments, T tends to be relatively small, making this approach less feasible. Thus, we guide exploration with the policy $\pi_{t+1}(x) = \arg\min_{\mathrm{A}_i \in \mathcal{A}} \widehat{r}_{\mathrm{A}_i,t}(x;h)$ in Step 2. We present specific implementations of `THREAD` (Algorithm 3) using LinUCB and online KNN in Algorithm 4 and 5, respectively.

Algorithm 4 `THREAD` using LinUCB

Input: trade-off parameter λ; human decision-maker h; cost function $c : \mathcal{A} \to [0,1]$; UCB parameter α
Initialization: data buffer $\mathcal{D}_0 = \{\}$; $\mathbf{A}_a = \mathbf{I}_{p\times p}$ and $\mathbf{b}_a = \mathbf{0}_{p\times 1}$ for all $a \in \mathcal{A}$; $\{\theta_a = (\mathbf{A}_a)^{-1}\mathbf{b}_a : a \in \mathcal{A}\}$; human prediction error values $\{\widehat{r}_{a,0}(x;h) = \langle \theta_a, x\rangle : x \in \mathcal{X}, a \in \mathcal{A}\}$; initial policy $\pi_1(x)_a = 1/|\mathcal{A}|$ for all $a \in \mathcal{A}$
for $t = 1, 2, \ldots, T$ **do**
  data point $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ is drawn iid from $\mathcal{P}$ (normalized s.t. $\|x_t\|_2 \leq 1$)
  support $a_t \in \mathcal{A}$ is selected using policy π_t
  human makes the prediction $\widetilde{y}_t$ based on x_t and a_t
  human incurs the loss $\ell(y_t, \widetilde{y}_t)$
  update the buffer $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(x_t, a_t, \ell(y_t, \widetilde{y}_t))\}$
  update the decision support policy:

$$\begin{aligned}
\mathbf{A}_{a_t} &\leftarrow \mathbf{A}_{a_t} + x_t x_t^\top \\
\mathbf{b}_{a_t} &\leftarrow \mathbf{b}_{a_t} + r_t x_t \\
\theta_a &\leftarrow (\mathbf{A}_a)^{-1}\mathbf{b}_a \text{ for all } a \in \mathcal{A} \\
\widehat{r}_{a,t}(x;h) &\leftarrow \langle \theta_a, x\rangle \text{ for all } a \in \mathcal{A} && \text{(Step 1)} \\
\pi_{t+1}(x) &\leftarrow \arg\min_{a\in\mathcal{A}} \lambda \cdot \widehat{r}_{a,t}(x;h) + (1-\lambda)\cdot c(a) - \alpha \cdot \sqrt{x^\top (\mathbf{A}_a)^{-1} x} && \text{(Step 2)}
\end{aligned}$$

end for
Output: policy $\pi_\lambda^{\mathrm{alg}} \leftarrow \pi_{T+1}$

5.3.2 Extension to Cost-Aware Decision Support

Providing decision support can be expensive [Arbiser et al., 2001; Mimra et al., 2016], so in addition to minimizing loss, it may be desirable to minimize the cost of providing support. We quantify the expected cost of a policy π using $c(\pi) = \mathbb{E}_x\left[\sum_{i=1}^k \pi(x)_{\mathrm{A}_i} \cdot c(\mathrm{A}_i)\right]$, where $c(\mathrm{A}_i)$

Algorithm 5 THREAD using online KNN

Input: trade-off parameter λ; human decision-maker h; cost function $c : \mathcal{A} \to [0,1]$; warm-up steps W; number of neighbours K; exploration parameter γ
Initialization: data buffer $\mathcal{D}_0 = \{\}$; human prediction error values $\{\widehat{r}_{a,0}(x;h) = 0.5 : x \in \mathcal{X}, a \in \mathcal{A}\}$; initial policy $\pi_1(x)_a = 1/|\mathcal{A}|$ for all $a \in \mathcal{A}$
for $t = 1, 2, \ldots, T$ **do**
  data point $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ is drawn iid from $\mathcal{P}$
  support $a_t \in \mathcal{A}$ is selected using policy π_t
  human makes the prediction $\widetilde{y}_t$ based on x_t and a_t
  human incurs the loss $\ell(y_t, \widetilde{y}_t)$
  update the buffer $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(x_t, a_t, \ell(y_t, \widetilde{y}_t))\}$
  update the decision support policy:

$$\begin{aligned}
\mathcal{N}(x) &\leftarrow \text{K neighbouring data points for } x \text{ in } \mathcal{D}_t \\
\mathcal{N}_a(x) &\leftarrow \{(x_i, a_i, \ell(y_i, \widetilde{y}_i)) : (x_i, a_i, \ell(y_i, \widetilde{y}_i)) \in \mathcal{N}(x) \text{ and } a_i = a\} \quad \text{for all } a \in \mathcal{A} \\
\widehat{r}_{a,t}(x;h) &\leftarrow \frac{1}{|\mathcal{N}_a(x)|} \cdot \sum_{(x_i, a_i, \ell(y_i, \widetilde{y}_i)) \in \mathcal{N}_a(x)} \ell(y_i, \widetilde{y}_i) \quad \text{for all } a \in \mathcal{A} \text{ with } |\mathcal{N}_a(x)| > 0
\end{aligned} \tag{Step 1}$$

$$\begin{aligned}
\pi_{\text{rand}}(x)_a &\leftarrow \frac{1}{|\mathcal{A}|} \quad \text{for all } a \in \mathcal{A} \\
\pi_{\text{knn}}(x) &\leftarrow \arg\min_{a \in \mathcal{A}} \lambda \cdot \widehat{r}_{a,t}(x;h) + (1-\lambda) \cdot c(a) \\
\pi_{t+1}(x) &\leftarrow \pi_{\text{rand}}(x) \quad \text{if } t \leq W \qquad \text{(Step 2)} \\
\pi_{t+1}(x) &\leftarrow \gamma \cdot \pi_{\text{rand}}(x) + (1-\gamma) \cdot \pi_{\text{knn}}(x) \quad \text{if } t > W \qquad \text{(Step 2)}
\end{aligned}$$

  end for
  Output: policy $\pi_\lambda^{\text{alg}} \leftarrow \pi_{T+1}$

is the cost of providing support A_i. Using both objectives, we consider the following multi-objective optimization (MOO) problem: $\min_\pi \mathcal{R}_h(\pi) = [L_h(\pi), c(\pi)]^\top$. We can reformulate this MOO problem into a single-objective optimization (SOO) problem: $\pi_\lambda^* = \arg\min_{\pi \in \Pi} \lambda \cdot L_h(\pi) + (1-\lambda) \cdot c(\pi)$, where $\lambda \in [0,1]$. It can be shown that the solutions to the SOO problem can fully characterize the Pareto front of the MOO problem [Branke et al., 2008; Mas-Colell et al., 1995]. Formal statements are provided in Appendix D.2. The SOO reformulation allows us to learn the policy π_λ^* for any $\lambda \in [0,1]$ using THREAD with minor modifications. Specifically, we can update **Step 2** of Algorithm 3 to $\pi_{t+1}(x) = \arg\min_{\mathrm{A}_i \in \mathcal{A}} \lambda \cdot \widehat{r}_{\mathrm{A}_i,t}(x;h) + (1-\lambda) \cdot c(\mathrm{A}_i)$ to incorporate the cost of providing support. Since $c(\mathrm{A}_i)$ does not depend on the input x, it does not affect **Step 1**.

In practice, it may be unclear how to select λ, but one way to consider this MOO problem may be to learn a decision support policy π that achieves a certain level of performance at a

minimal cost: given a tolerance threshold $\varepsilon \in [0,1]$ for the expected loss, we can find a policy π that achieves the minimum expected cost $c(\pi)$ while maintaining the expected loss $L_h(\pi)$ within ε of the optimal loss L_h^{opt}, where $L_h^{\text{opt}} = \min_\pi L_h(\pi)$. We discuss the existence of such a λ for a given ε in Appendix D.2. Given a large number of interactions with a decision-maker h, we can simply run Algorithm 3 for multiple values of λ by sampling from $\lambda \in [0,1]$. From the set of corresponding policies learned for each λ, we can evaluate which λ yields a policy with the lowest cost within ε of L_h^{opt}. We test this workflow computationally with synthetic decision-makers in Section 5.5.

However, it is often infeasible to extensively test many choices of λ values simultaneously for a new decision-maker without repeating queries of the same input (e.g. as in our human subject experiments); thus, we propose a hyper-parameter tuning strategy using population data. We follow the above workflow to find suitable values for λ for each decision-maker in a population. From this set of candidate λ values, we select a λ to deploy by choosing the most common value (mode) of all λ values that yield policies that get decision-makers within ε of their optimal loss. Although the decision-makers in the population may differ from the unseen decision-maker, we prefer this to a random selection of λ at deployment. We test our strategy on real users in Section 5.6.

5.3.3 Expertise Profiles

Decision-makers may have different "expertise" (i.e. strengths and weaknesses) across the input space $\mathcal{X}$ under each form of support. We define notation that captures an individual's expertise – which we call *expertise profiles*. We divide $\mathcal{X}$ into regions where $r_{\text{A}_i}(x;h)$ is roughly constant given A_i, so $\mathcal{X} = \cup_{j\in[N]}\mathcal{X}_j$. These regions could be defined by class labels or by covariates. We let $r_{\text{A}_i}(\mathcal{X}_j;h)$ capture the individual h's average prediction error under support A_i across region $\mathcal{X}_j$. We use $r_{\text{A}_i}(\mathcal{X}_j;h)$ to define the following expertise profiles:

1. **Approximately Invariant** expertise across all the regions under different forms of support, i.e. $r_{\text{A}_1}(\mathcal{X}_j;h) \approx r_{\text{A}_2}(\mathcal{X}_j;h) \approx \cdots \approx r_{\text{A}_k}(\mathcal{X}_j;h), \forall j \in [N]$. Though decision support is likely unhelpful, we should recover a fixed policy that corresponds to picking the cheapest form of support.

2. **Varying** expertise that changes by region, i.e. $r_{\text{A}_1}(\mathcal{X}_j;h) \leq r_{\text{A}_2}(\mathcal{X}_j;h)$ and $r_{\text{A}_2}(\mathcal{X}_k;h) \leq r_{\text{A}_1}(\mathcal{X}_k;h)$, for some $j,k \in [N]$. Decision-makers typically have some variation in terms of expertise, as they excel in one area but may benefit from support in areas beyond their training [Schvaneveldt et al., 1985]. However, the difference between $|r_{\text{A}_1} - r_{\text{A}_2}|$ for a region will dictate how efficiently the policy can be learned.

3. **Strictly Better** expertise (e.g. $A_1 \succ A_2 \succ \cdots \succ A_k$) that is uniformly maintained across all the regions, i.e. $r_{A_1}(\mathcal{X}_j;h) \leq r_{A_2}(\mathcal{X}_j;h) \leq \cdots \leq r_{A_k}(\mathcal{X}_j;h), \forall j \in [N]$. A decision support policy should learn the fixed form of support to use for each expert, but would not yield any cost benefit.

5.4 Experimental Setup

We overview the setup for our experiments. Additional details are in Appendix D.3.

5.4.1 Decision-making Tasks

We first overview the choices of datasets and sizes of $|\mathcal{A}|$ in our experiments. We construct tasks from three datasets: (1) A *Synthetic* classification task consisting of well-defined, separable clusters; (2) *CIFAR*-10 [Krizhevsky, 2009], a 10-class image classification dataset; (3) *MMLU* [Hendrycks et al., 2020], a multi-task text-based benchmark that tests for knowledge and problem-solving ability across 57 topics spanning both the humanities and STEM. In terms of the size of $|\mathcal{A}|$, we let *kA* denote when there are k forms of support for a task. We focus on $k = 2$ or 3, which captures a buffet of real-world scenarios in prior work where decision-makers have a few tools at their disposal, as per Appendix D.1. We now describe our two main tasks, which are designed to be accessible to crowdworkers, that will be featured in both the computational and human subject experiments.[3] While these experiments cover important forms of support studied in the literature, our choices of support are not intended to exhaustively demonstrate the diverse forms of support that our framework can handle.

CIFAR-3*A***.** The three forms of support are (1) HUMAN ALONE, where the human makes the decision solely based on the input; (2) MODEL, which shows decision-makers a model's prediction for the given input, (3) CONSENSUS, which shows a distribution over labels from approximately 50 annotators Peterson et al. [2019]. We assume the cost of HUMAN ALONE is less than the cost of either form of support, and the cost of MODEL and CONSENSUS is equal. To construct a realistic setting in which the forms of support result in different strengths and weaknesses for decision-makers, we deliberately corrupt images of different classes to evoke performance differences – necessitating that a decision-maker appropriately calibrate when to rely on each form of support. We consider 5 of the animal classes in CIFAR-10; of these, we never corrupt images of Birds, do not corrupt images of Deers and Cats for the MODEL, and do not corrupt images of Horses and Frogs for the CONSENSUS.

[3]We include computational experiments for two additional tasks (Synthetic-2*A* and CIFAR-2*A*) and experiments where we vary the size of k in the Appendix.

MMLU-2*A*. The two forms of support are HUMAN ALONE and LLM, where the human is provided responses generated from InstructGPT3.5 (`text-davinci-003`) [Ouyang et al., 2022] using the same few-shot prompting scheme for MMLU as Hendrycks et al. [2020]. Again, the cost of MODEL is greater than the cost of HUMAN ALONE. There are many topics in MMLU; to select a subset, we conducted pilot studies to identify topics where the accuracy of the LLM and average human accuracy vary. We choose the following topics: Computer Science, US Foreign Policy, High School Biology, and Elementary Mathematics. This task is *"in-the-wild"* because we expect people to excel at different topics, akin to real-world settings where decision-makers may have "varying" expertise.

5.4.2 Human-informed Synthetic Decision-makers

To construct a population of decision-makers, we collect data of user decisions to obtain realistic expertise profiles per Section 5.3.3. Specifically, we showed participants similar inputs with different forms of support to estimate $r_{\mathrm{A}_i}(\mathcal{X}_j;h)$ for each support A_i in each region $\mathcal{X}_j$: see Appendix D.3. We report expertise profiles, where regions of expertise for a given form of support A_i corresponds to class labels for CIFAR and to question topics for MMLU. We write $r_{\mathrm{A}_i} = [0.7, 0.1, 0.7]$ to denote that, under support A_i, the decision-maker h incurs a loss of 0.7 on $\mathcal{X}_1$, 0.1 on $\mathcal{X}_2$, and 0.7 on $\mathcal{X}_3$. While we instantiate decision-makers in this way, `THREAD` does not assume knowledge of the regions. Our population of *human-informed synthetic decision-makers* can then be queried to simulate decision-maker behavior. We leverage these decision-makers in Section 5.5 to evaluate `THREAD` under multiple values of λ and in Section 5.6 to select λ values to deploy on unseen decision-makers.

5.4.3 Baselines and Other Parameters

Algorithms and Baselines. We compare both variants of `THREAD` (LinUCB/KNN) against the following offline policies: *Human and Support*, where the decision-maker *always* receives the same form of support: $\pi(x) = A_i$ for all x. As in most real-world settings, there may be an action in $\mathcal{A}$ that corresponds to providing no support. *Population-level,* where we assume we have access to learned decision support policies for m decision-makers. For the two tasks described in Section 5.4.1, we construct m human-informed synthetic decision-makers based on real expertise profiles per Section 5.4.2. At inference time, the policy for our new decision-maker selects a form of support based on the majority vote from the m learned policies and breaks ties at random.

Number of Interactions. While more interactions (higher T) provide more data points to estimate each r_{A_i}, we need to consider what a realistic value of T is given the constraints of

Table 5.1 In the standard setting, we evaluate the average excess loss (i.e. $L_h(\pi) - L_h^{opt}$), where lower is better, and standard deviation across 5 individuals in each expertise profile (Section 5.3.3) for both CIFAR-3*A* (Left) and MMLU-2*A* (Right). $L_h(\pi)$ is computed by averaging across the last 10 steps of 100 total time steps. We find that learning decision support policies are not helpful for decision-makers with "invariant" expertise profiles. However, for individuals who fall under the "strictly better" and "varying" profiles, we **bold** the variant with lowest excess loss, and find at least one THREAD variant, particularly KNN, outperforms offline policies.

Algorithm	Invariant	Strictly Better	Varying
H-ONLY	0.00 ± 0.01	0.09 ± 0.08	0.50 ± 0.06
H-MODEL	0.00 ± 0.01	0.22 ± 0.19	0.35 ± 0.05
H-CONSENSUS	0.00 ± 0.01	0.23 ± 0.13	0.27 ± 0.08
Population	0.00 ± 0.02	0.18 ± 0.08	0.15 ± 0.03
THREAD-LinUCB	0.00 ± 0.01	0.17 ± 0.05	0.19 ± 0.05
THREAD-KNN	0.00 ± 0.01	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.08 \pm 0.02}$

Algorithm	Invariant	Strictly Better	Varying
H-ONLY	0.01 ± 0.01	0.18 ± 0.17	0.22 ± 0.12
H-LLM	0.01 ± 0.01	0.18 ± 0.21	0.12 ± 0.17
Population	0.00 ± 0.02	0.19 ± 0.07	0.12 ± 0.09
THREAD-LinUCB	0.00 ± 0.01	0.12 ± 0.03	0.07 ± 0.04
THREAD-KNN	0.01 ± 0.01	$\mathbf{0.05 \pm 0.03}$	$\mathbf{0.05 \pm 0.03}$

working with real humans (e.g. limited attention and cognitive load). In traditional online learning, T is usually unreasonably large, on the order of thousands Guan and Jiang [2018]; Li et al. [2010]. Via pilot studies, we found that 100 CIFAR images or 60 MMLU questions were a reasonable number of decisions to make within 20-40 minutes (a typical amount of time for an online study), which we used throughout our experiments.

5.5 Computational Experiments

Characterizing THREAD's utility in the standard setting. First, we investigate how THREAD compares against offline baselines under different expertise profiles. We observe that THREAD learns a policy that is closer to the participant's optimal performance when a decision-maker's expertise profile falls is either "strictly better" or "varying" (Table 5.1). This is because THREAD identifies *which* form of support is better in each context, compared to fixed offline policies which *always* show one form of support or to the population variant, which may not provide the correct form of support to each individual. As expected, there is no benefit in the "invariant" setting, as the decision-maker performs equally well under all forms of support. We note that KNN generally outperforms LinUCB, the latter of which can be saddled by its implicit linearity assumption.

Benefits of THREAD given multiple objectives. In the cost-aware setting, we use THREAD to learn policies for human-informed synthetic decision-makers with different expertise profiles per Section 5.4.2 using the workflow described in Section 5.3.2. In Table 5.2, we focus on MMLU, where there are naturally occurring examples of the various expertise profiles. For individuals who are "strictly better," it may not be possible to further reduce the expected loss

Table 5.2 In the cost-aware setting for MMLU-2*A*, we compare the expected loss $L_h(\pi)$ and the expected cost $c(\pi)$—lower is better—averaged across the last 10 steps of 100 total time steps of 5 separate runs. From the data collected in Section 5.4.2, we select the two individuals who have a "varying" profile (corresponding to the decision-makers with the first and second lowest risk without support) and two individuals who have a "strictly better" profile (corresponding the decision-makers with the lowest and highest L_h^{opt}). We follow the workflow described in Section 5.3.2 to find a suitable λ, reported for each person in Table D.2. We **bold** the algorithm that achieves the lowest cost within ε risk of the L_h^{opt}, and find that THREAD outperforms offline policies.

	Varying: $L_h^{\text{opt}} = 0.13$ $r_{\text{H-ONLY}} = [0.3, 0.1, 0.5, 0.3]$ $r_{\text{H-LLM}} = [0.5, 0.1, 0.0, 0.1]$		Varying: $L_h^{\text{opt}} = 0.10$ $r_{\text{H-ONLY}} = [0.1, 0.3, 0.6, 0.6]$ $r_{\text{H-LLM}} = [0.4, 0.0, 0.3, 0.0]$		Strictly Better: $L_h^{\text{opt}} = 0.43$ $r_{\text{H-ONLY}} = [0.8, 0.6, 0.8, 0.6]$ $r_{\text{H-LLM}} = [0.6, 0.3, 0.7, 0.1]$		Strictly Better: $L_h^{\text{opt}} = 0.09$ $r_{\text{H-ONLY}} = [0.3, 0.9, 0.3, 0.9]$ $r_{\text{H-LLM}} = [0.0, 0.1, 0.0, 0.3]$	
Algorithm	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$
H-ONLY	0.30 ± 0.03	0.0	0.41 ± 0.04	0.0	0.68 ± 0.03	0.0	0.54 ± 0.05	0.0
H-LLM	0.21 ± 0.03	0.1	0.16 ± 0.04	0.1	0.42 ± 0.06	0.1	0.09 ± 0.02	0.1
Population	0.13 ± 0.03	0.05 ± 0.01	0.16 ± 0.04	0.05 ± 0.01	0.53 ± 0.05	0.05 ± 0.01	0.35 ± 0.07	0.05 ± 0.01
THREAD-LinUCB	0.15 ± 0.05	0.04 ± 0.01	$\mathbf{0.15 \pm 0.04}$	$\mathbf{0.06 \pm 0.01}$	0.51 ± 0.06	0.06 ± 0.01	0.20 ± 0.10	0.06 ± 0.01
THREAD-KNN	$\mathbf{0.17 \pm 0.03}$	$\mathbf{0.03 \pm 0.01}$	0.15 ± 0.04	0.07 ± 0.01	$\mathbf{0.46 \pm 0.05}$	$\mathbf{0.05 \pm 0.02}$	$\mathbf{0.12 \pm 0.03}$	$\mathbf{0.08 \pm 0.01}$

compared to one of the offline policies; however, we find that running THREAD can reduce cost while still achieving an expected loss that is within ε of L_h^{opt} For the individuals with "varying" profiles, we find that THREAD can improve both performance and cost over many of the offline baselines. The population baseline is inconsistent in terms of its ability to identify high-performing policies at low cost, in contrast to the personalized policies learned via THREAD. We report similar results for multiple datasets in Appendix D.4. We omit the "invariant" profile as the best policy is to select the cheapest form of support.

Discussion on parameter selection. In the cost-aware setting, we assumed that we could learn policies for multiple values of λ simultaneously. However, we observe that often only a subset of the learned policies fall within ε of L_h^{opt} (e.g. in Figure D.3), underscoring that selecting a suitable λ may be finicky in practice. We further study the effect of various parameters (e.g. exploration parameters, KNN parameters, embedding size, and the number of interactions) in Appendix D.4. These studies informed the parameter selection of the main text experiments, and we advise future work deploying THREAD for new datasets to consider similar experiments with human-informed synthetic decision-makers to understand the impact of these various parameters on decision outcomes.

5.6 Human Subject Validation

To validate whether our decision support policies learned via THREAD work in practice, we run a series of human subject experiments (HSE), specifically ethics-reviewed studies with real

human participants. As we overview in Appendix D.1, prior work does not run user studies that personalize policies online. We first overview the set-up of the user study with additional information in Appendix D.5.

Interactive Interface. To translate our problem formulation into an interactive tool, we create `Modiste`, which provides an interface for `THREAD`. At each time step, `Modiste` sends each user's predictions to a server, running `THREAD`, which identifies the next form of support for the next input. `Modiste` then updates the interface accordingly to reflect the selected form of support. Our tool can flexibly be linked with crowdsourcing platforms like Prolific [Palan and Schitter, 2018]. In Appendix D.5, we provide screenshots of the interface under each form of support and the parameters used to initialize `THREAD`. Participants are informed of their own correctness after each trial, as well as the correctness of the form of support (e.g. model prediction) if support was provided; we choose to reveal correctness after each trial such that participants can learn whether support ought to be relied upon.

Recruitment Details. We recruit a total of 125 crowdsourced participants from Prolific [Palan and Schitter, 2018] to interact with `Modiste` ($N = 45$ and $N = 80$ for the CIFAR and MMLU tasks, respectively). We recruit more participants for MMLU, as we expect greater individual differences in regions where support is needed (e.g. some participants may be good at mathematics and struggle in biology, whereas others may excel in biology questions), compared to CIFAR where there is an "optimal" form of support for each stimulus. Each participant is assigned to only one task and one algorithm variant.

Trade-off parameter selection. We follow the strategies specified in Section 5.3.2 to identify λ for our cost-aware experiments. We visualize this process in further detail in Figure D.4. For CIFAR, the selected parameter values were $\lambda = 0.85$ for LinUCB and $\lambda = 0.75$ for KNN. For MMLU, the selected parameter values were $\lambda = 0.95$ for LinUCB and $\lambda = 0.75$ for KNN.

CIFAR. By design, the CIFAR task compels "varying" profiles: `Modiste`'s forte. We find that `Modiste` learns to reconstruct near-optimal policies in both the standard and cost-aware settings, as depicted in Figure 5.2. This is reflected quantitatively in Table 5.3, where both LinUCB and KNN have lower expected losses than any of the offline policies. Given a non-linear and non-separable embedding space, it is possible that KNN generally outperforms LinUCB, which implicitly makes a linearity assumption. We also identify a choice of λ for each algorithm that modulates loss versus cost, by using our human-informed synthetic decision-makers. In particular, this is done by using the cheaper form of support (H-ONLY) more. However, the trade-off is not at a pre-specified increment, as the gap between the KNN variants is much smaller than that between the LinUCB variants.

Table 5.3 We report expected loss $L_h(\pi)$ and expected cost $c(\pi)$ incurred (lower is better) in the last 10 trials by Prolific participants for each Algorithm and **bold** the variant with the lowest $L_h(\pi)$. `Modiste` learns effective, low-cost policies: for CIFAR, `Modiste` outperforms all baselines. For MMLU, we find that `Modiste` learns a policy with roughly the same performance as the best offline policy but at *half* the cost. We also consider different choices of λ, where $\lambda = 1.0$ corresponds to the standard setting and $\lambda \neq 1.0$ corresponds to a cost-aware setting, where the choice of λ was selected according to Section 5.3.2.

Algorithm	$L_h(\pi)$	$c(\pi)$
H-ONLY	0.68 ± 0.13	0
H-MODEL	0.50 ± 0.11	0.5
H-CONSENSUS	0.32 ± 0.07	0.5
Population	0.66 ± 0.08	0
`Modiste` (LinUCB, $\lambda = 1.0$)	0.22 ± 0.15	0.38 ± 0.06
`Modiste` (KNN, $\lambda = 1.0$)	$\mathbf{0.1 \pm 0.06}$	$\mathbf{0.44 \pm 0.07}$
`Modiste` (LinUCB, $\lambda = 0.85$)	0.5 ± 0.15	0.14 ± 0.07
`Modiste` (KNN, $\lambda = 0.75$)	0.14 ± 0.23	0.39 ± 0.06

Algorithm	$L_h(\pi)$	$c(\pi)$
H-ONLY	0.51 ± 0.14	0
H-LLM	$\mathbf{0.22 \pm 0.14}$	**0.1**
Population	0.33 ± 0.19	0.06 ± 0.01
`Modiste` (LinUCB, $\lambda = 1.0$)	0.25 ± 0.11	0.05 ± 0.02
`Modiste` (KNN, $\lambda = 1.0$)	0.3 ± 0.05	0.07 ± 0.02
`Modiste` (LinUCB, $\lambda = 0.95$)	0.35 ± 0.14	0.04 ± 0.01
`Modiste` (KNN, $\lambda = 0.75$)	0.3 ± 0.11	0.07 ± 0.02

MMLU. Polymaths are rare; we observe no different in our Prolific data, as most participants struggle with at least one topic. Due to the task design wherein the LLM excels at three of the four topics, many individuals benefited from receiving the LLM as support. In Figure 5.3, we see that `Modiste` personalizes support to decision-maker expertise, yielding policies that provide support on different topics but yield the same task accuracy for each decision-maker. An exhaustive set of policy visualizations is in Figure D.7. In Table 5.3, `Modiste` is not statistically different from the best baselines (H-LLM) in terms of expected loss. Yet, we observe lower expected costs for both LinUCB and KNN (despite not directly optimizing for cost), and find similar results, as expected, in the cost-aware setting. This suggests that learning policies with `Modiste` has merit: participants who have no expertise will see the LLM all of the time, and those who are proficient in a topic can expect to see the LLM in cases beyond their expertise. While $L_h(\pi)$ of the `Modiste` variants are similar to that of the Population baseline in Table 5.3, the variance is significantly smaller – particularly with KNN – underscoring the benefits of personalization.

Discussion. Our HSEs demonstrate how `Modiste` can be used to learn decision support policies online. We find that `THREAD` enjoys performance and cost benefits when encountering users with "varying" expertise profiles, and recovers fixed policies for participants with "strictly better" profiles in the standard setting. While our human-informed synthetic decision-makers permit us to pick a suitable λ for participants, modeling human idiosyncrasies [Steyvers and Kumar, 2022] explicitly in synthetic decision-makers may bolster `THREAD`'s effectiveness. One such idiosyncrasy we observe in the MMLU task is a negative correlation between expected loss and reliance (Pearson r correlation $= -0.47$ and -0.59 for KNN and LinUCB, respectively), where we define reliance as the proportion of instances when a participant adheres to the LLM's

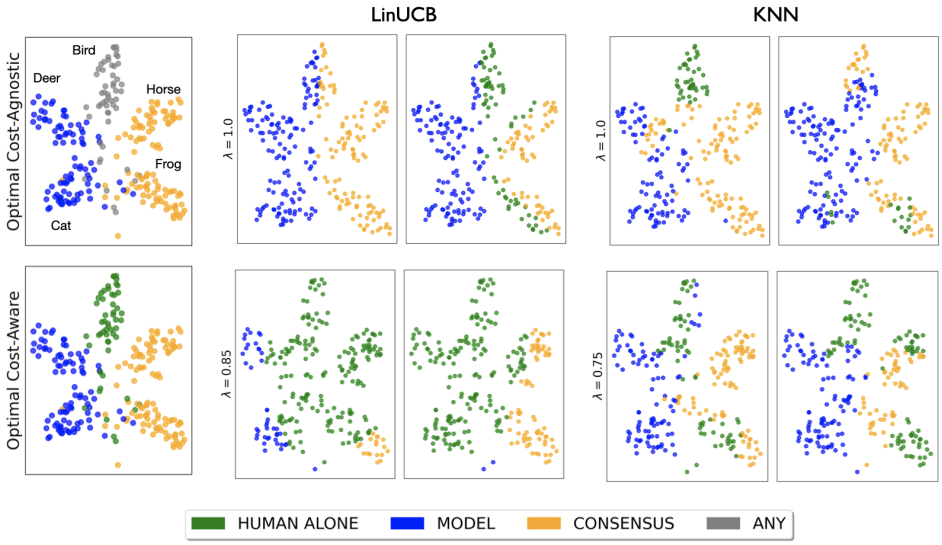


Fig. 5.2 Snapshots of the learned decision support policies computed at the end of 100 interactions with randomly sampled users for CIFAR-2*A*. The forms of support are colored in t-SNE embedding space. In the standard setting, both LinUCB and KNN generally identify the optimal form of support for a given input. However, given the task set-up, the choice of support for images of Birds does not matter (Section 5.4.1). When we introduce cost (i.e. when $\lambda \neq 1$), the learned policies favor the human alone for images of Birds, since it is cheaper than the other two forms of support. All participant plots are included in Figure D.6.

recommendation. This underscores the importance of understanding when decision-makers may over-rely on potentially fallible support [Buçinca et al., 2021; Bussone et al., 2015]. We provide further discussion on reliance and the implications of learning policies in practice in Appendix D.6.

5.7 Summary

A decision support policy captures when and which form of support should be provided to improve a decision-maker's performance. To the best of our knowledge, we are the first to consider learning such a policy online for unseen decision-makers. We propose `THREAD`, an algorithm for learning a decision support policy online, and then extend it to handle auxiliary objectives, like the cost of support, using a tuned trade-off parameter. We instantiate two variants of `THREAD` using stochastic contextual bandits. We perform computational and human subject experiments to highlight the importance – and feasibility – of personalizing decision support policies for individual decision-makers. Our human subject experiments are encouraging, as we can learn decision support policies in remarkably few iterations and tease apart differences in decision-makers' need for support. Future work can extend `THREAD` to handle potential

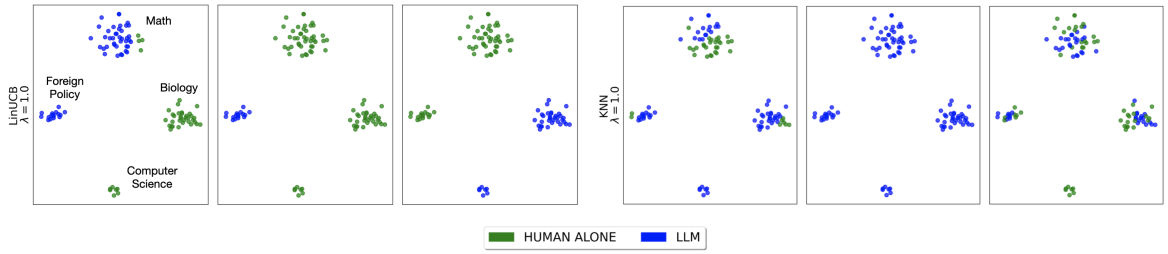


Fig. 5.3 Snapshots of the learned decision support policies computed at the end of 60 interactions for MMLU-3*A*. The forms of support are colored in t-SNE embedding space. All six participants achieve $L_h(\pi) = 0.1875$–comparable to that of H-LLM– but exhibit distinct policies across input space. `Modiste` learns policies that capture each decision-maker's strengths and weaknesses. All other participant plots are in Figure D.7.

drift in decision-maker behavior, can consider learning a policy for providing various forms of transparency atop model predictions, and can use `Modiste` to tailor decision support policies in deployed real-world settings.

Decision support is a long-standing discipline [Kahn Jr, 1994; Keen, 1980] with a history of successful deployments in practice [Spiegelhalter, 2017; Yang et al., 2023b]. However, such success does not preclude the need to carefully consider how to best leverage decision support in emerging applications. While the use of `THREAD` is promising, to encourage the thoughtful deployment of personalized decision support policies, we urge consideration of the following points. First, we acknowledge that significant issues can arise when decision-makers blindly rely on decision support [Buçinca et al., 2020], especially when the support is erroneous or ineffective; such over-reliance requires careful attention to prevent. Second, in some sense, our work hinges on domain experts defining the available forms of support (i.e. we need a clearly defined $\mathcal{A}$ to use `Modiste`). In practice, this may prove difficult, as one may not know how to define specific forms of support or decision-makers may have access to varying support sets (e.g. Bob may not be permitted to use a form of support A_i due to regulatory reasons, while Alice may have taken the requisite exam to be permitted to access support A_i). While our methods can use a deterministic rule to decide which forms of support to include in $\mathcal{A}$, it is unclear how to best administer such a pipeline and integrate `Modiste` into existing workflows [Fogliato et al., 2022]. Future work exploring the integration, encouraging rich cross-talk between domain experts and ML practitioners, paves a route towards responsible deployment of decision support.

Chapter 6

Conclusion

This thesis explores trustworthy machine learning. In Chapter 2, we start with an extensive discussion on trust, trustworthiness mechanisms, transparency, and human-machine teams, wherein decision-makers are aided by AI systems à la ML models [Bhatt and Shams, 2021; Bhatt et al., 2021a; Chen et al., 2023a; Zerilli et al., 2022]. We then explore how algorithmic transparency is used in practice in Chapter 3. We run interview studies and host a multistakeholder convening to study the needs of explainability for practitioners, policymakers, and end users. We find that many existing explainability tools are not used widely in practice [Bhatt et al., 2020a,b]. We go on to devise two explainability methods to aid decision-making in Chapter 4. We propose DIVerse INfluEntial (DIVINE) points that provide example-based explanations of which training points are not only influential to a model but also diverse [Bhatt et al., 2021b]. We use submodular diversity functions to identify a set of points that are important as per existing methods yet spread out in input space. We find that DIVINE points are quite well suited for asking decision-makers to simulate the entire decision boundary of a model. We also devise Counterfactual Latent Uncertainty Explanations (CLUE), a feature-level explanation method to explain where in input space uncertainty about an outcome lies [Antorán et al., 2021]. We achieve this by searching in the latent space of a deep generative model for a plausible data point that has lower uncertainty than the data point being explained. We show that CLUEs can help decision-makers gauge whether a probabilistic model will be uncertain on unseen examples. In Chapter 5, we propose `Modiste`, a tool for learning when it is appropriate for a decision-maker to have access to one of many forms of decision support (e.g. expert consensus or model explanation) [Bhatt et al., 2023]. Using methods from stochastic contextual bandits, we describe our algorithm, `THREAD`, that personalizes a decision support policy for each decision-maker based on their expertise. Excitingly, we demonstrate that we can efficiently learn a policy online for unseen decision-makers (e.g. deciding when a decision-maker should have access to GPT when answering quiz questions). `Modiste` helps us realize the effective

development of a human-machine team (HMT). Thus, this thesis takes trustworthy ML on a journey from algorithmic transparency to decision support.

We leverage a variety of tools in our study of trustworthy machine learning. We conducted interview studies, hosted facilitated discussions at carefully curated workshops, devised methods for algorithmic transparency, and ran user studies to check the efficacy of our methods. The diverse approaches herein suggest that future work may also require similar interdisciplinary efforts to study HMTs successfully. Our hope is that this thesis might inspire those who study HMTs to seek to forge stronger connections despite the persistence of disciplinary boundaries (in practice if not in principle). At the moment, HMT research is siloed. To take one case, the authors of a study took themselves to be challenging "the widespread assertion that people are averse to algorithms" on the basis that the participants in their study "were quite willing to rely on algorithmic advice before seeing the algorithm err" [Logg et al., 2019]. Human factors engineers would be unmoved by the finding that humans are prepared to trust—indeed overtrust—algorithms, having invested great efforts over the years in dealing with the problematic consequences of this very tendency. In our view, HMT research should comprise a unified branch of study with a basic modus operandi and *lingua franca*, albeit drawing from expertise across several autonomous subfields.

In this thesis, we sought to study how various forms of algorithmic transparency may promote algorithmic vigilance in HMTs in carefully designed contexts. Perhaps the greatest challenge in the study of HMTs, however, is simply resisting the urge to overgeneralize experimental results [Chavaillaz et al., 2016]. Indeed, we think that ecological validity is an underappreciated problem in this area. Findings in aviation and shipping contexts are of questionable value in court and law enforcement contexts, which in turn may have little bearing on how the automation of medical diagnoses should be approached [Goddard et al., 2014]. In legal and medical contexts, initial trust in automation is actually quite low, presumably due to the expertise of the users involved [Christin, 2017; Linkov et al., 2017]. This is at odds with the general findings herein (e.g. those in Chapter 5).

Insofar as ecological validity is acknowledged, too often it features as an afterthought: a mere warning to readers of the limitations of the study along with a reminder to keep those limitations in mind when applying results in real-world settings [Manzey et al., 2012]. This is a good start, but it has not prevented occasionally sweeping claims being made about how "people" using "algorithms" react in this or that situation [Carton et al., 2020; Dietvorst et al., 2015, 2018; Dzindolet et al., 2003; Logg et al., 2019]. For example, the authors of another influential study state their take-home message as follows: "observing an automated decision aid make errors leads to distrust of the automated decision aid, unless an explanation is provided explaining why the aid might err" [Dzindolet et al., 2003]. Further down the same page, one

finds the customary discussion of limitations. First, they note that "the task was very simple and artificial." Second, the study necessarily ignore "[t]he effect of one person's view of the automated aid's trustworthiness on other group members' reliance decisions," because the study limits itself to examining the dyad of a single user with an automated aid. When the findings of a branch of study are taken up with the vim and vigor typical in HMT research, ecological concerns become too important to squeeze into general disclaimers. How can we be certain that the limitations do not vitiate the generalizations entirely? Ideally, authors should premise all substantive claims so that even titles and abstracts are expressed tentatively. In this example, the take-home message cannot quite be: people distrust automated aids whose errors they witness unless an explanation is provided. Something more tentative is called for: *in very simple automated tasks involving a single person*, people *tend* to distrust automated aids whose errors they witness unless an explanation is provided. Every aid, every interface, every task, is unique after all.

Perhaps an effective way to meet the ecological challenge is for HMT research to proceed in a task-specific fashion that accounts for the precise nature of the task and its setting. Note that task specificity is distinct from domain specificity. A domain-specific investigation would confine research and its results to a more or less widely defined domain of activity (such as maritime shipping or criminal justice). Task-specific investigation, by contrast, would confine research by the nature of the task under consideration (such as adjudication between disputing parties, regardless of whether it is carried out by a court of law, a mediator, or a human resources officer). Since the basis of investigation and extrapolation in the latter case is the similarity of the tasks undertaken, regardless of domain, task-specific investigation may harness results from research conducted across distinct domains of activity. Conversely, a task-specific orientation may mean that results from one experiment are not presumed to generalize to another setting, despite the fact that both tasks occur within the same domain (e.g. results from an experiment testing the behavior of judges using recidivism risk algorithms in sentencing or bail applications may fail to generalize to a setting in which judges use algorithms to determine the likelihood of a repeat psychotic episode in a parent suing for child custody). Our impression is that sweeping claims are more typical in organizational behavior and ML than they are in ergonomics and human factors. The latter fields have always had several parallel streams of inquiry running alongside one another (e.g. one for ocean navigation, one for aviation and air traffic control, one for autonomous vehicles, another for nuclear power, etc.), and this has meant that conclusions in these fields have always been implicitly circumscribed.

An emphasis on task-specific inquiry may seem in tension with our call for HMT research to espouse greater cross-disciplinary cohesion and coordination. But what we are calling for in the latter case is simply an end to the kind of siloed research in which differences in

terminology serve no purpose, and where people from one field are unaware of discoveries in another relating to the exact same subject matter. Cross-disciplinary activity, as such, is compatible with task-specific investigation: the field of human factors itself offers an excellent model of domain- and task-specific research worth emulating at a larger scale. This transition may not be easy to achieve. HMTs researchers whose main experience is in ML may find it especially difficult. The ML community on the whole values task-independent, model-agnostic and scalable, general models to solve as many variations of a problem as possible. This work is not misconceived. Indeed there is a delicate balance to be struck between the necessity of controlled and (to a sometimes considerable extent) contrived experiments, on the one hand, and real-world applicability on the other. We appreciate that experimental conditions must strive to isolate the psychological processes underlying team behaviors, and that without a certain amount of artifice in experimental design there can be no generalizable results at all. However, our parting thought is to direct attention to the importance of real-world applicability, and more specifically, to intra-ecological generalization. Task specificity is an effective means of securing this form of generalization, and is an imperative for the ML community to contribute meaningfully to HMT research.

Arguably, we are just nearing the precipice of what will, ultimately, be a revolution in personalized HMTs, wherein decision-makers in specific settings are assisted with an appropriate form of decision support tailored to their needs. It is our responsibility, our duty, to ensure these systems are built *with* and *for* all. Future work must ensure that decision-maker autonomy is maintained while providing AI-powered decision aids. Decision support is all around us. We have our preferred forms of decision support for every task we do. You may prefer to support your decision to buy a home by consulting a family member, while someone else may support the same decision by leveraging the wealth of information on the web. Alice may need an explanation atop a model prediction to adhere to its recommendation, while the same explanation may not phase Bob's opposition to AI-based support. Preempting and tailoring access to the right form of support for a given decision-maker are ripe extensions of Chapter 5; this may require modeling and incorporating both decision-maker confidence and support helpfulness into objectives for learning when to personalize access to decision support. We end this thesis confident that trustworthy machine learning can have significant impacts on decision-making in the next half-century. With the appropriate form of support, decisions can be easy.

References

K. Aas, M. Jullum, and A. Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298: 103502, 2021.

A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515, 2018.

J. Adebayo, M. Muelly, I. Liccardi, and B. Kim. Debugging tests for model explanations. *Advances in Neural Information Processing Systems*, 33:700–712, 2020.

T. Adel, Z. Ghahramani, and A. Weller. Discovering interpretable representations for both deep generative and discriminative models. In *International Conference on Machine Learning*, pages 50–59, 2018.

S. Adel Bargal, A. Zunino, D. Kim, J. Zhang, V. Murino, and S. Sclaroff. Excitation backprop for rnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1440–1449, 2018.

A. M. Alaa and M. van der Schaar. Discriminative jackknife: Quantifying uncertainty in deep learning via higher-order influence functions. In *International Conference on Machine Learning (ICML)*, 2020.

F. A. G. Alexander McFarlane Mood and D. C. Boes. *Introduction to the theory of statistics*. McGraw-Hill New York, 3rd ed. edition, 1974. ISBN 0070428646.

S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza. Power to the people: The role of humans in interactive machine learning. *Ai Magazine*, 35(4):105–120, 2014.

S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, et al. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

H. Anahideh, A. Asudeh, and S. Thirumuruganathan. Fair active learning. *Expert Systems with Applications*, 199:116981, 2022.

M. Ancona, E. Ceolini, C. Oztireli, and M. Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *6th International Conference on Learning Representations (ICLR 2018)*, 2018.

M. Ancona, C. Oztireli, and M. Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pages 272–281. PMLR, 2019.

S. Andrist, D. Bohus, Z. Yu, and E. Horvitz. Are you messing with me? Querying about the sincerity of interactions in the open world. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 409–410. IEEE, 2016.

J. Angwin, J. Larson, S. Mattu, L. Kirchner, and ProPublica. Machine bias, 2016. URL https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

J. Antorán, U. Bhatt, T. Adel, A. Weller, and J. M. Hernández-Lobato. Getting a CLUE: A method for explaining uncertainty estimates. In *International Conference on Learning Representations*, 2021.

E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

Z. K. Arbiser, A. L. Folpe, and S. W. Weiss. Consultative (expert) second opinions in soft tissue pathology: analysis of problem-prone diagnostic situations. *American journal of clinical pathology*, 116(4):473–476, 2001.

S. Armstrong and S. Mindermann. Occam's razor is insufficient to infer the preferences of irrational agents. *Advances in Neural Information Processing Systems*, 31, 2018.

M. Arnold, R. K. Bellamy, M. Hind, S. Houde, S. Mehta, A. Mojsilović, R. Nair, K. N. Ramamurthy, A. Olteanu, D. Piorkowski, et al. Factsheets: Increasing trust in ai services through supplier's declarations of conformity. *IBM Journal of Research and Development*, 63(4/5):6–1, 2019.

A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

S. Z. Arshad, J. Zhou, C. Bridon, F. Chen, and Y. Wang. Investigating user confidence for uncertainty presentation in predictive decision making. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, pages 352–360, 2015.

A. Ashukha, A. Lyzhov, D. Molchanov, and D. Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations*, 2019.

A. Askell, M. Brundage, and G. Hadfield. The role of cooperation in responsible ai development. *arXiv preprint arXiv:1907.04534*, 2019.

A. Augustin, M. Venanzi, A. Rogers, and N. R. Jennings. Bayesian aggregation of categorical distributions with applications in crowdsourcing. In *IJCAI*, pages 1411–1417, 2017.

V. Babbar, U. Bhatt, and A. Weller. On the utility of prediction sets in human-ai teams. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2457–2463, 7 2022. Main Track.

F. Bach et al. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in machine learning*, 6(2-3):145–373, 2013.

D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11 (Jun):1803–1831, 2010.

N. Bagheri and G. A. Jamieson. Considering subjective trust and monitoring behavior in assessing automation-induced "complacency.". *Human performance, situation awareness, and automation: Current research and trends*, 1:54–59, 2004.

W. A. Bainbridge, J. W. Hart, E. S. Kim, and B. Scassellati. The benefits of interactions with physically present robots over video-displayed agents. *International Journal of Social Robotics*, 3(1):41–52, 2011.

M. A. Bakker, D. P. Tu, K. P. Gummadi, A. S. Pentland, K. R. Varshney, and A. Weller. Beyond reasonable doubt: Improving fairness in budget-constrained decision making using confidence thresholds. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 346–356, 2021.

V. A. Banks, A. Eriksson, J. O'Donoghue, and N. A. Stanton. Is partially automated driving a bad idea? observations from an on-road study. *Applied ergonomics*, 68:138–145, 2018a.

V. A. Banks, K. L. Plant, and N. A. Stanton. Driver error or designer error: Using the perceptual cycle model to explore the circumstances surrounding the fatal tesla crash on 7th may 2016. *Safety science*, 108:278–285, 2018b.

G. Bansal, B. Nushi, E. Kamar, W. S. Lasecki, D. S. Weld, and E. Horvitz. Beyond accuracy: The role of mental models in human-ai team performance. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 2–11, 2019a.

G. Bansal, B. Nushi, E. Kamar, D. S. Weld, W. S. Lasecki, and E. Horvitz. Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2429–2437, 2019b.

G. Bansal, B. Nushi, E. Kamar, E. Horvitz, and D. S. Weld. Is the most accurate AI the best teammate? optimizing AI for teamwork. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11405–11414, 2021a.

G. Bansal, T. Wu, J. Zhou, R. Fok, B. Nushi, E. Kamar, M. T. Ribeiro, and D. Weld. Does the whole exceed its parts? the effect of AI explanations on complementary team performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2021b.

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*, pages 80–89, 2020.

E. Barshan, M.-E. Brunet, and G. K. Dziugaite. Relatif: Identifying explanatory training samples via relative influence. In *International Conference on Artificial Intelligence and Statistics*, pages 1899–1909. PMLR, 2020.

P. L. Bartlett and M. H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(Aug):1823–1840, 2008.

H. Bastani and M. Bayati. Online decision making with high-dimensional covariates. *Operations Research*, 68:276–294, 2020.

H. Bastani, O. Bastani, and P. Sinchaisri. Improving human decision-making with machine learning. In *Academy of Management Proceedings*, volume 2022, page 17725. Academy of Management Briarcliff Manor, NY 10510, 2022.

S. Bates, A. Angelopoulos, L. Lei, J. Malik, and M. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6):1–34, 2021.

R. M. Battleday, J. C. Peterson, and T. L. Griffiths. Capturing human categorization of natural images by combining deep networks and cognitive models. *Nature communications*, 11(1): 1–14, 2020.

S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, page 0049124118782533, 2017.

L. Beyer, O. J. Hénaff, A. Kolesnikov, X. Zhai, and A. v. d. Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

U. Bhatt and Z. Shams. Trust in artificial intelligence: Clinicians are essential. *Healthcare Information Technology for Cardiovascular Medicine: Telemedicine & Digital Health*, pages 127–141, 2021.

U. Bhatt, P. Ravikumar, et al. Building human-machine trust via interpretability. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9919–9920, 2019.

U. Bhatt, M. Andrus, A. Weller, and A. Xiang. Machine learning explainability for external stakeholders. *ICML Workshop on Human Interpretability*, 2020a.

U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. Moura, and P. Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 648–657, 2020b.

U. Bhatt, M. B. Zafar, K. Gummadi, and A. Weller. Counterfactul accuracies for alternative models. *ICLR Workshop on Machine Learning in Real Life (ML-IRL)*, 2020c.

U. Bhatt, J. Antorán, Y. Zhang, Q. V. Liao, P. Sattigeri, R. Fogliato, G. Melançon, R. Krishnan, J. Stanley, O. Tickoo, L. Nachman, R. Chunara, M. Srikumar, A. Weller, and A. Xiang. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, page 401–413, New York, NY, USA, 2021a. Association for Computing Machinery. ISBN 9781450384735.

U. Bhatt, I. Chien, M. B. Zafar, and A. Weller. DIVINE: Diverse influential training points for data visualization and model refinement. *arXiv preprint arXiv:2107.05978*, 2021b.

U. Bhatt, A. Weller, and J. M. Moura. Evaluating and aggregating feature-based model explanations. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3016–3022, 2021c.

U. Bhatt, V. Chen, K. M. Collins, P. Kamalaruban, E. Kallina, A. Weller, and A. Talwalkar. Learning personalized decision support policies. *arXiv preprint arXiv:2304.06701*, 2023.

J. Bien and R. Tibshirani. Prototype selection for interpretable classification. *The Annals of Applied Statistics*, pages 2403–2424, 2011.

A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer, 2016.

J. M. Bland and D. G. Altman. Bayesians and frequentists. *Bmj*, 317(7166):1151–1160, 1998.

A. Blum and K. Stangl. Recovering from biased data: Can fairness constraints improve accuracy? *arXiv preprint arXiv:1912.01094*, 2019.

Board of Governors of the Federal Reserve System. Supervisory guidance on model risk management. *https://www.federalreserve.gov/supervisionreg/srletters/sr1107a1.pdf*, 2011.

L. Boels, A. Bakker, W. Van Dooren, and P. Drijvers. Conceptual difficulties when interpreting histograms: A review. *Educational Research Review*, 28:100291, 2019.

R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

E. Bondi, R. Koster, H. Sheahan, M. Chadwick, Y. Bachrach, T. Cemgil, U. Paquet, and K. Dvijotham. Role of Human-AI Interaction in Selective Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

S. Bordt and U. Von Luxburg. A bandit model for human-machine decision making with private information and opacity. In *International Conference on Artificial Intelligence and Statistics*, pages 7300–7319. PMLR, 2022.

R. Botsman. *Who can you trust?: how technology brought us together–and why it could drive us apart*. Penguin UK, 2017.

L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

J. Branke, J. Branke, K. Deb, K. Miettinen, and R. Slowiński. *Multiobjective optimization: Interactive and evolutionary approaches*, volume 5252. Springer Science & Business Media, 2008.

L. Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

G. M. Briggs, P. A. Flynn, M. Worthington, I. Rennie, and C. McKinstry. The role of specialist neuroradiology second opinion reporting: is there added value? *Clinical radiology*, 63(7): 791–795, 2008.

S. M. Brown, E. Santos, and S. B. Banks. Utility theory-based user models for intelligent interface agents. In *Advances in Artificial Intelligence: 12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI'98 Vancouver, BC, Canada, June 18–20, 1998 Proceedings 12*, pages 378–392. Springer, 1998.

M. Brundage, S. Avin, J. Wang, H. Belfield, G. Krueger, G. Hadfield, H. Khlaaf, J. Yang, H. Toner, R. Fong, et al. Toward trustworthy ai development: mechanisms for supporting verifiable claims. *arXiv preprint arXiv:2004.07213*, 2020.

Z. Buçinca, P. Lin, K. Z. Gajos, and E. L. Glassman. Proxy tasks and subjective measures can be misleading in evaluating explainable ai systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 454–464, 2020.

Z. Buçinca, M. B. Malaya, and K. Z. Gajos. To trust or to think: cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–21, 2021.

D. V. Budescu and T. S. Wallsten. Consistency in interpretation of probabilistic phrases. *Organizational behavior and human decision processes*, 36(3):391–405, 1985.

D. V. Budescu, H.-H. Por, and S. B. Broomell. Effective communication of uncertainty in the ipcc reports. *Climatic change*, 113:181–200, 2012.

J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018.

A. Bussone, S. Stumpf, and D. O'Sullivan. The role of explanations on trust and reliance in clinical decision support systems. In *2015 international conference on healthcare informatics*, pages 160–169. IEEE, 2015.

C. J. Cai, E. Reif, N. Hegde, J. Hipp, B. Kim, D. Smilkov, M. Wattenberg, F. Viegas, G. S. Corrado, M. C. Stumpe, et al. Human-centered tools for coping with imperfect algorithms during medical decision-making. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2019.

F. Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy, and K. R. Varshney. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, pages 3992–4001, 2017.

O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31, 2018.

O.-M. Camburu, E. Giunchiglia, J. Foerster, T. Lukasiewicz, and P. Blunsom. Can I trust the explainer? Verifying post-hoc explanatory methods. *arXiv:1910.02065*, 2019.

Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019.

S. Carton, Q. Mei, and P. Resnick. Feature-based explanations don't help people detect misclassifications of online toxicity. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 95–106, 2020.

D. V. Carvalho, E. M. Pereira, and J. S. Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

S. Chaiken. The heuristic—systematic. *Dual-process theories social psychology*, 73, 1999.

C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations*, 2019.

A. Chaparro, L. Groff, K. Tabor, K. Sifrit, and L. J. Gugerty. Maintaining situational awareness: The role of visual attention. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 43, pages 1343–1347. SAGE Publications Sage CA: Los Angeles, CA, 1999.

M. Chapman-Rounds, U. Bhatt, E. Pazos, M.-A. Schulz, and K. Georgatzis. Fimap: Feature importance by minimal adversarial perturbation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11433–11441, 2021.

M.-A. Charusaie, H. Mozannar, D. Sontag, and S. Samadi. Sample efficient learning of predictors that complement humans. In *International Conference on Machine Learning*, pages 2972–3005. PMLR, 2022.

A. Chavaillaz, D. Wastell, and J. Sauer. System reliability, performance and trust in adaptable automation. *Applied Ergonomics*, 52:333–342, 2016.

J. Chen, N. Kallus, X. Mao, G. Svacha, and M. Udell. Fairness under unawareness: Assessing disparity when protected class is unobserved. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 339–348. ACM, 2019a.

J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. *7th International Conference on Learning Representations (ICLR 2019)*, 2019b.

V. Chen, J. Li, J. S. Kim, G. Plumb, and A. Talwalkar. Interpretable machine learning: Moving from mythos to diagnostics. *Queue*, 19(6):28–56, 2022.

V. Chen, U. Bhatt, H. Heidari, A. Weller, and A. Talwalkar. Perspectives on incorporating expert feedback into model updates. *Patterns*, 4(7), 2023a.

V. Chen, Q. V. Liao, J. W. Vaughan, and G. Bansal. Understanding the role of human intuition on reliance in human-ai decision-making with explanations. *arXiv preprint arXiv:2301.07255*, 2023b.

H.-F. Cheng, L. Stapleton, R. Wang, P. Bullock, A. Chouldechova, Z. S. S. Wu, and H. Zhu. Soliciting stakeholders' fairness notions in child maltreatment predictive systems. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2021.

C.-K. Chow. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, 4:247–254, 1957.

P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

A. Christin. Algorithms in practice: Comparing web journalism and criminal justice. *Big Data & Society*, 4(2), 2017.

P. Cihon, M. J. Kleinaltenkamp, J. Schuett, and S. D. Baum. Ai certification: Advancing ethical practice by reducing information asymmetries. *IEEE Transactions on Technology and Society*, 2(4):200–209, 2021.

D. A. Clark. Verbal uncertainty expressions: A critical review of two decades of research. *Current Psychology*, 9(3):203–235, 1990.

A. Coden, M. Danilevsky, D. Gruhl, L. Kato, and M. Nagarajan. A method to accelerate human in the loop clustering. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 237–245. SIAM, 2017.

M. S. Cohen, R. Parasuraman, and J. T. Freeman. Trust in decision aids: A model and its training implications. In *in Proc. Command and Control Research and Technology Symp.* Citeseer, 1998.

G. W. Cole and S. A. Williamson. Avoiding resentment via monotonic fairness. *ArXiv*, abs/1909.01251, 2019.

K. M. Collins, U. Bhatt, and A. Weller. Eliciting and learning with soft labels from every annotator. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, volume 10, 2022.

K. M. Collins, M. Barker, M. E. Zarlenga, N. Raman, U. Bhatt, M. Jamnik, I. Sucholutsky, A. Weller, and K. Dvijotham. Human uncertainty in concept-based ai systems, 2023.

R. Cook and S. Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22:495–508, 1980.

R. D. Cook. Detection of influential observation in linear regression. *Technometrics*, 19(1): 15–18, 1977.

M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. Paying more attention to saliency: Image captioning with saliency and context attention. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(2):1–21, 2018.

A. Cornwall and R. Jewkes. What is participatory research? *Social science & medicine*, 41 (12):1667–1676, 1995.

C. Cortes, G. DeSalvo, and M. Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pages 67–82. Springer, 2016.

C. Cortes, G. DeSalvo, C. Gentile, M. Mohri, and S. Yang. Online learning with abstention. In *International conference on machine learning*, pages 1059–1067. PMLR, 2018.

P. Cortez and M. J. Embrechts. Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences*, 225:1–17, 2013.

L. Cosmides and J. Tooby. Are humans good intuitive statisticians after all? rethinking some conclusions from the literature on judgment under uncertainty. *cognition*, 58(1):1–73, 1996.

A. Coston, A. Rambachan, and A. Chouldechova. Characterizing fairness over the set of good models under selective labels. In *International Conference on Machine Learning*, pages 2144–2155. PMLR, 2021.

Council of European Union. European union general data protection regulation, articles 13-15. *http://www.privacy-regulation.eu/en/13.htm*, 2018.

Y. Cui, P. Koppol, H. Admoni, S. Niekum, R. Simmons, A. Steinfeld, and T. Fitzgerald. Understanding the relationship between interactions and outcomes in human-in-the-loop machine learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, Montreal, QC, Canada*, volume 10, 2021.

M. Cummings. Automation bias in intelligent time critical decision support systems. In *AIAA 1st Intelligent Systems Technical Conference*, page 6313, 2004.

T. Dao, A. Gu, A. Ratner, V. Smith, C. De Sa, and C. Ré. A kernel theory of modern data augmentation. In *International Conference on Machine Learning*, pages 1528–1537. PMLR, 2019.

A. Datta, S. Sen, and Y. Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617. IEEE, 2016.

D. R. Davies and R. Parasuraman. *The psychology of vigilance*. Academic Press, 1982.

K. R. Davies. *Inverse decision theory with medical applications*. Rice University, 2005.

B. Davis, U. Bhatt, K. Bhardwaj, R. Marculescu, and J. M. Moura. On network science and mutual information for explaining deep neural networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8399–8403. IEEE, 2020.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28 (1):20–28, 1979.

E. Daxberger and J. M. Hernández-Lobato. Bayesian variational autoencoders for unsupervised out-of-distribution detection. *arXiv preprint arXiv:1912.05651*, 2019.

A. De, N. Okati, A. Zarezade, and M. G. Rodriguez. Classification under human assistance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5905–5913, 2021.

M. De-Arteaga, R. Fogliato, and A. Chouldechova. A case for humans-in-the-loop: Decisions in the presence of erroneous algorithmic scores. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

M. De-Arteaga, A. Dubrawski, and A. Chouldechova. Leveraging expert consistency to improve algorithmic decision support. *arXiv preprint arXiv:2101.09648*, 2021.

S. Debernard, F. Vanderhaegen, and P. Millot. An experimental investigation of dynamic allocation of tasks between air traffic controller and ai systems. In *Analysis, Design and Evaluation of Man–Machine Systems 1992*, pages 95–100. Elsevier, 1993.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

S. Depeweg, J. M. Hernández-Lobato, S. Udluft, and T. A. Runkler. Sensitivity analysis for predictive uncertainty. In *ESANN*, 2017.

S. Depeweg, J.-M. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1184–1193, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

A. Der Kiureghian and O. Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

M. Desai, P. Kaniarasu, M. Medvedev, A. Steinfeld, and H. Yanco. Impact of robot failures and feedback on real-time trust. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 251–258. IEEE, 2013.

A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*, pages 592–603, 2018a.

A. Dhurandhar, K. Shanmugam, R. Luss, and P. A. Olsen. Improving simple models with confidence profiles. In *Advances in Neural Information Processing Systems*, pages 10296–10306, 2018b.

T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

B. J. Dietvorst and S. Bharti. People reject algorithms in uncertain decision domains because they have diminishing sensitivity to forecasting error. *Psychological science*, 31(10):1302–1314, 2020.

B. J. Dietvorst, J. P. Simmons, and C. Massey. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, 144(1):114, 2015.

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*, 64 (3):1155–1170, 2018.

J. J. Dijkstra. User agreement with incorrect expert system advice. *Behaviour & Information Technology*, 18(6):399–411, 1999.

B. Dimanov, U. Bhatt, M. Jamnik, and A. Weller. You shouldn't trust me: Learning models which conceal unfairness from multiple explanation methods. *Proceedings of the 2020 European Conference on AI*, 2020.

A. Dionisio, R. Menezes, and D. A. Mendes. Entropy and uncertainty analysis in financial markets. *arXiv preprint arXiv:0709.0668*, 2007.

F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

A. D. Dragan, S. Bauman, J. Forlizzi, and S. S. Srinivasa. Effects of robot motion on human-robot collaboration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 51–58, 2015.

J. Dressel and H. Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

W. DuMouchel. Data squashing: constructing summary data sets. In *Handbook of Massive Data Sets*, pages 579–591. Springer, 2002.

C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck. The role of trust in automation reliance. *International journal of human-computer studies*, 58(6):697–718, 2003.

U. Ehsan and M. O. Riedl. Explainability pitfalls: Beyond dark patterns in explainable AI. *arXiv:2109.12480*, 2021.

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*, pages 81–87, 2018.

U. Ehsan, P. Tambwekar, L. Chan, B. Harrison, and M. O. Riedl. Automated rationale generation: a technique for explainable ai and its effects on human perceptions. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 263–274, 2019.

U. Ehsan, Q. V. Liao, M. Muller, M. O. Riedl, and J. D. Weisz. Expanding explainability: Towards social transparency in ai systems. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2021.

D. Ensign, S. A. Friedler, S. Neville, C. Scheidegger, and S. Venkatasubramanian. Runaway feedback loops in predictive policing. In *Conference on Fairness, Accountability and Transparency*, pages 160–171, 2018.

N. Epley, A. Waytz, and J. T. Cacioppo. On seeing human: a three-factor theory of anthropomorphism. *Psychological review*, 114(4):864, 2007.

G. Erion, J. D. Janizek, P. Sturmfels, S. Lundberg, and S.-I. Lee. Learning explainable models using attribution priors. *arXiv preprint arXiv:1906.10670*, 2019.

M. K. Erramilli. Nationality and subsidiary ownership patterns in multinational corporations. *Journal of International Business Studies*, 27:225–248, 1996.

K. Ethayarajh. Is your classifier actually biased? measuring fairness under uncertainty with bernstein bounds. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2914–2919, 2020.

C. Etmann, S. Lunz, P. Maass, and C. Schoenlieb. On the connection between adversarial robustness and saliency map interpretability. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1823–1832, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

A. M. Evans and J. I. Krueger. The psychology (and economics) of trust. *Social and Personality Psychology Compass*, 3(6):1003–1017, 2009.

J. A. Fails and D. R. Olsen Jr. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45, 2003.

D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578, 2011.

M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

B. Fischhoff and A. L. Davis. Communicating scientific uncertainty. *Proceedings of the National Academy of Sciences*, 111(Supplement 4):13664–13671, 2014.

A. Fisher, C. Rudin, and F. Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81, 2019.

S. T. Fiske. Stereotype content: Warmth and competence endure. *Current directions in psychological science*, 27(2):67–73, 2018.

R. Fogliato, A. Chouldechova, and M. G'Sell. Fairness evaluation in presence of biased noisy labels. In *International conference on artificial intelligence and statistics*, pages 2325–2336. PMLR, 2020.

R. Fogliato, S. Chappidi, M. Lungren, P. Fisher, D. Wilson, M. Fitzke, M. Parkinson, E. Horvitz, K. Inkpen, and B. Nushi. Who goes first? influences of human-ai workflow on decision making in clinical imaging. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1362–1374, 2022.

R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017.

J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

C. Frye, C. Rowat, and I. Feige. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *Advances in Neural Information Processing Systems*, 33: 1229–1239, 2020.

Y. Gal. Uncertainty in deep learning. *University of Cambridge*, 1(3), 2016.

M. Galesic and R. Garcia-Retamero. Statistical numeracy for health: a cross-cultural comparison with probabilistic national samples. *Archives of internal medicine*, 170(5):462–468, 2010.

R. Gao, M. Saar-Tsechansky, M. De-Arteaga, L. Han, M. K. Lee, and M. Lease. Human-AI collaboration with bandit feedback. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1722–1728. International Joint Conferences on Artificial Intelligence Organization, 2021.

R. Gao, M. Saar-Tsechansky, M. De-Arteaga, L. Han, W. Sun, M. K. Lee, and M. Lease. Learning complementary policies for human-ai teams. *Conference on Information Systems and Technology (CIST)*, 2022.

S. Garg, Y. Wu, S. Balakrishnan, and Z. C. Lipton. A unified view of label shift estimation. *arXiv preprint arXiv:2003.07554*, 2020.

T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. Iii, and K. Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30, 2017.

A. M. Geoffrion. Proper efficiency and the theory of vector maximization. *Journal of mathematical analysis and applications*, 22(3):618–630, 1968.

Z. Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553): 452–459, May 2015. ISSN 1476-4687. doi: 10.1038/nature14541.

B. Ghai, Q. V. Liao, Y. Zhang, R. Bellamy, and K. Mueller. Explainable active learning (xal) toward ai explanations as interfaces for machine teachers. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):1–28, 2021.

A. Ghorbani and J. Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251, 2019.

A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, pages 9277–9286, 2019.

G. Gigerenzer, O. Wegwarth, and M. Feufel. Misleading communication of risk, 2010.

L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

E. Glikson and A. W. Woolley. Human trust in artificial intelligence: Review of empirical research. *Academy of Management Annals*, 14(2):627–660, 2020.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

K. Goddard, A. Roudsari, and J. C. Wyatt. Automation bias: empirical results assessing influencing factors. *International journal of medical informatics*, 83(5):368–375, 2014.

M. C. Gombolay, R. A. Gutierrez, S. G. Clarke, G. F. Sturla, and J. A. Shah. Decision-making authority, team efficiency and human worker satisfaction in mixed human–robot teams. *Autonomous Robots*, 39(3):293–312, 2015.

R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

M. L. Gordon, K. Zhou, K. Patel, T. Hashimoto, and M. S. Bernstein. The disagreement deconvolution: Bringing machine learning performance metrics in line with reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2021.

M. L. Gordon, M. S. Lam, J. S. Park, K. Patel, J. Hancock, T. Hashimoto, and M. S. Bernstein. Jury learning: Integrating dissenting voices into machine learning models. In *CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2022.

B. Green and Y. Chen. The principles and limits of algorithm-in-the-loop decision making. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–24, 2019.

N. Grgic-Hlaca, E. M. Redmiles, K. P. Gummadi, and A. Weller. Human perceptions of fairness in algorithmic decision making: A case study of criminal risk prediction. In *Proceedings of the 2018 World Wide Web Conference*, pages 903–912, 2018.

M. Guan and H. Jiang. Nonparametric stochastic contextual bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

L. J. Gugerty and W. C. Tirre. Individual differences in situation awareness. *Situation awareness analysis and measurement*, pages 249–276, 2000.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

V. Guimaraes Goecks. *Human-in-the-loop Methods for Data-driven and Reinforcement Learning Systems*. PhD thesis, Aerospace Engineering, Texas A & M, 2020.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017.

H. Guo, N. Rajani, P. Hase, M. Bansal, and C. Xiong. Fastif: Scalable influence functions for efficient model interpretation and debugging. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10333–10350, 2021.

M. Gupta, A. Cotter, M. M. Fard, and S. Wang. Proxy fairness. *arXiv preprint arXiv:1806.11212*, 2018.

K. S. Gurumoorthy, A. Dhurandhar, G. Cecchi, and C. Aggarwal. Efficient data representation by selecting prototypes with importance weights. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 260–269. IEEE, 2019.

S. Hajian and J. Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *IEEE transactions on knowledge and data engineering*, 25(7): 1445–1459, 2012.

M. Hall, D. Harborne, R. Tomsett, V. Galetic, S. Quintana-Amate, A. Nottle, and A. Preece. A systematic method to understand requirements for explainable ai (xai) systems. In *Proceedings of the IJCAI Workshop on eXplainable Artificial Intelligence (XAI 2019), Macau, China*, volume 11, 2019.

F. R. Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

B. Hancock, M. Bringmann, P. Varma, P. Liang, S. Wang, and C. Ré. Training classifiers with natural language explanations. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2018, page 1884. NIH Public Access, 2018.

M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

F. E. Harrell, R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982.

P. Hase and M. Bansal. Evaluating explainable ai: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552, 2020.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

T. J. Hastie and R. J. Tibshirani. *Generalized additive models*, volume 43. CRC press, 1990.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

C. Heath and A. Tversky. Preference and belief: Ambiguity and competence in choice under uncertainty. *Journal of risk and uncertainty*, 4(1):5–28, 1991.

M. E. Hellman. The nearest neighbor classification rule with a reject option. *IEEE Transactions on Systems Science and Cybernetics*, 6(3):179–185, 1970.

P. Hemmer, S. Schellhammer, M. Vössing, J. Jakubik, and G. Satzger. Forming effective human-AI teams: Building machine learning models that complement the capabilities of multiple experts. In *Proceedings of the Thirtieth International Conference on International Joint Conferences on Artificial Intelligence*, 2022.

L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *European Conference on Computer Vision*, pages 793–811. Springer, 2018.

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2020.

J. Heo, S. Joo, and T. Moon. Fooling neural network interpretations via adversarial model manipulation. *Advances in Neural Information Processing Systems*, 32:2925–2936, 2019.

D. Hernández-Lobato et al. Prediction based on averages over automatically induced learners: Ensemble methods and bayesian techniques. *Ph. D. Dissertation, UNIVERSIDAD AUTONOMA de MADRID*, 2009.

J. M. Hernández-Lobato and R. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pages 918–926, 2014.

R. Hertwig and I. Erev. The description–experience gap in risky choice. *Trends in cognitive sciences*, 13(12):517–523, 2009.

S. Hilgard, N. Rosenfeld, M. R. Banaji, J. Cao, and D. Parkes. Learning representations by humans, for humans. In *International Conference on Machine Learning*, pages 4227–4238. PMLR, 2021.

M. Hind, D. Wei, M. Campbell, N. C. Codella, A. Dhurandhar, A. Mojsilović, K. Natesan Ramamurthy, and K. R. Varshney. Ted: Teaching ai to explain its decisions. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 123–129, 2019.

J. L. Hintze and R. D. Nelson. Violin plots: a box plot-density trace synergism. *The American Statistician*, 52(2):181–184, 1998.

G. Hiranandani, S. Boodaghians, R. Mehta, and O. Koyejo. Performance metric elicitation from pairwise classifier comparisons. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 371–379, 2019.

G. Hiranandani, H. Narasimhan, and S. Koyejo. Fair performance metric elicitation. *Advances in Neural Information Processing Systems*, 33:11083–11095, 2020.

G. Hiranandani, J. Mathur, H. Narasimhan, M. M. Fard, and S. Koyejo. Optimizing black-box metrics with iterative example weighting. In *International Conference on Machine Learning*, pages 4239–4249. PMLR, 2021.

K. A. Hoff and M. Bashir. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human factors*, 57(3):407–434, 2015.

R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman. Metrics for explainable AI: Challenges and prospects. *arXiv preprint arXiv:1812.04608*, 2018.

J. M. Hofman, D. G. Goldstein, and J. Hullman. How visualizing inferential uncertainty can mislead readers about treatment effects in scientific results. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

P. W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

K. Holstein, J. Wortman Vaughan, H. Daumé III, 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*, page 600. ACM, 2019.

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*, 4(CSCW1):1–26, 2020.

G. Hooker and L. Mentch. Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:1905.03151*, 2019.

S. C. Hora. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety*, 54(2-3):217–223, 1996.

E. Horvitz and M. Barry. Display of information for time-critical decision making. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 296–305, 1995.

E. Horvitz and T. Paek. Complementary computing: policies for transferring callers from dialog systems to human receptionists. *User Modeling and User-Adapted Interaction*, 17: 159–182, 2007.

Y. T.-Y. Hou and M. F. Jung. Who is the expert? reconciling algorithm aversion and algorithm appreciation in ai-supported decision making. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–25, 2021.

N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

C. I. Hovland, I. L. Janis, and H. H. Kelley. *Communication and persuasion.* Yale University Press, 1953.

J. Huggins, T. Campbell, and T. Broderick. Coresets for scalable bayesian logistic regression. In *Advances in Neural Information Processing Systems*, pages 4080–4088, 2016.

J. Hullman. Why authors don't visualize uncertainty. *IEEE transactions on visualization and computer graphics*, 26(1):130–139, 2019.

J. Hullman, P. Resnick, and E. Adar. Hypothetical outcome plots outperform error bars and violin plots for inferences about reliability of variable ordering. *PloS one*, 10(11):e0142444, 2015.

J. Hullman, X. Qiao, M. Correll, A. Kale, and M. Kay. In pursuit of error: A survey of uncertainty visualization evaluation. *IEEE transactions on visualization and computer graphics*, 25(1):903–913, 2018.

IBM. Ibm's principles for data trust and transparency, Mar 2019. URL https://www.ibm.com/blogs/policy/trust-principles/.

A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

V. S. Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 279–288, 2002.

A. Jacovi, A. Marasović, T. Miller, and Y. Goldberg. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 624–635, 2021.

D. Janz, J. Hron, P. Mazur, K. Hofmann, J. M. Hernández-Lobato, and S. Tschiatschek. Successor uncertainties: exploration and uncertainty in temporal difference learning. *Advances in Neural Information Processing Systems*, 32, 2019.

J. V. Jeyakumar, J. Noor, Y.-H. Cheng, L. Garcia, and M. Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. *Advances in Neural Information Processing Systems*, 2020.

H. Jiang, B. Kim, M. Guan, and M. Gupta. To trust or not to trust a classifier. In *Advances in neural information processing systems*, pages 5541–5552, 2018.

S. Joshi, O. Koyejo, B. Kim, and J. Ghosh. xGEMs: Generating examplars to explain black-box models. *arXiv preprint arXiv:1806.08867*, 2018.

S. Joshi, O. Koyejo, W. Vijitbenjaronk, B. Kim, and J. Ghosh. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv preprint arXiv:1907.09615*, 2019.

I. Jubran, A. Maalouf, and D. Feldman. Introduction to coresets: Accurate coresets. *arXiv preprint arXiv:1910.08707*, 2019.

C. E. Kahn Jr. Artificial intelligence in radiology: decision support systems. *Radiographics*, 14 (4):849–861, 1994.

D. Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013.

A. Kale, M. Kay, and J. Hullman. Visual reasoning strategies for effect size judgments and decisions. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

A. Kalis, S. Kaiser, and A. Mojzisch. Why we should talk about option generation in decision-making research. *Frontiers in psychology*, 4:555, 2013.

N. Kallus, X. Mao, and A. Zhou. Assessing algorithmic fairness with unobserved protected class using data combination. *Management Science*, 68(3):1959–1981, 2022.

E. Kamar, S. Hacker, and E. Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 467–474, 2012.

A.-H. Karimi, B. Schölkopf, and I. Valera. Algorithmic recourse: from counterfactual explanations to interventions. *arXiv preprint arXiv:2002.06278*, 2020.

L. Kaufmann. Clustering by means of medoids. In *Proc. Statistical Data Analysis Based on the L1 Norm Conference, Neuchatel, 1987*, pages 405–416, 1987.

H. Kaur, H. Nori, S. Jenkins, R. Caruana, H. Wallach, and J. Wortman Vaughan. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.

D. Kaushik, E. Hovy, and Z. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2019.

M. Kay, T. Kola, J. R. Hullman, and S. A. Munson. When (ish) is My Bus? User-centered Visualizations of Uncertainty in Everyday, Mobile Predictive Systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5092–5103. Association for Computing Machinery, 2016. ISBN 978-1-4503-3362-7. doi: 10/b2pj.

P. G. Keen. Decision support systems: a research perspective. In *Decision support systems: Issues and challenges: Proceedings of an international task force meeting*, pages 23–44, 1980.

C. J. Kelly, A. Karthikesalingam, M. Suleyman, G. Corrado, and D. King. Key challenges for delivering clinical impact with artificial intelligence. *BMC medicine*, 17(1):195, 2019.

E. M. Kenny, C. Ford, M. Quinn, and M. T. Keane. Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in xai user studies. *Artificial Intelligence*, 294:103459, 2021.

V. Keswani, M. Lease, and K. Kenthapadi. Towards unbiased and accurate deferral to multiple experts. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 154–165, 2021.

R. Khanna, B. Kim, J. Ghosh, and S. Koyejo. Interpreting black box predictions using fisher kernels. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3382–3390, 2019.

B. Kim, C. Rudin, and J. A. Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in neural information processing systems*, pages 1952–1960, 2014.

B. Kim, R. Khanna, and S. Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, 2016.

B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

C. N. Kim, K. H. Yang, and J. Kim. Human decision-making behavior and modeling effects. *Decision Support Systems*, 45(3):517–527, 2008.

J. Kleinberg, H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan. Human decisions and machine predictions. *The quarterly journal of economics*, 133(1):237–293, 2018.

F. H. Knight. *Risk, uncertainty and profit*, volume 31. Houghton Mifflin, 1921.

M. J. Kochenderfer. *Decision making under uncertainty: theory and application*. MIT press, 2015.

P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70 (ICML 2017)*, pages 1885–1894. Journal of Machine Learning Research, 2017.

P. W. Koh, J. Steinhardt, and P. Liang. Stronger data poisoning attacks break data sanitization defenses. *arXiv preprint arXiv:1811.00741*, 2018.

P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

P. W. W. Koh, K.-S. Ang, H. Teo, and P. S. Liang. On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems*, pages 5254–5264, 2019.

M. Körber. Theoretical considerations and development of a questionnaire to measure trust in automation. In *Congress of the International Ergonomics Association*, pages 13–30. Springer, 2018.

A. Krause and D. Golovin. Submodular function maximization. *Tractability*, 3(71-104):3, 2014.

A. Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

J. Kruschke. *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan*. Academic Press, 2014.

T. Kulesza, S. Stumpf, M. Burnett, S. Yang, I. Kwan, and W.-K. Wong. Too much, too little, or just right? ways explanations impact end users' mental models. In *2013 IEEE Symposium on Visual Languages and Human Centric Computing*, pages 3–10. IEEE, 2013.

B. Kulynych, D. Madras, S. Milli, I. D. Raji, A. Zhou, and R. Zemel. Participatory approaches to machine learning. International Conference on Machine Learning Workshop, July 2020.

I. E. Kumar, S. Venkatasubramanian, C. Scheidegger, and S. Friedler. Problems with shapley-value-based explanations as feature importance measures. *arXiv preprint arXiv:2002.11097*, 2020.

M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. In *Advances in neural information processing systems*, pages 4066–4076, 2017.

Y. Kwon, M. A. Rivas, and J. Zou. Efficient computation and analysis of distributional shapley values. In *International Conference on Artificial Intelligence and Statistics*, pages 793–801. PMLR, 2021.

I. Lage, A. Ross, S. J. Gershman, B. Kim, and F. Doshi-Velez. Human-in-the-loop interpretability prior. *Advances in neural information processing systems*, 31, 2018.

I. Lage, E. Chen, J. He, M. Narayanan, B. Kim, S. J. Gershman, and F. Doshi-Velez. Human evaluation of models built for interpretability. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 59–67, 2019.

V. Lai and C. Tan. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 29–38, 2019.

V. Lai, S. Carton, R. Bhatnagar, Q. V. Liao, Y. Zhang, and C. Tan. Human-AI collaboration via conditional delegation: A case study of content moderation. In *CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2022.

V. Lai, C. Chen, A. Smith-Renner, Q. V. Liao, and C. Tan. Towards a science of human-ai decision making: An overview of design space in empirical human-subject studies. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1369–1385, 2023.

C. Laidlaw and S. Russell. Uncertain decisions facilitate better preference learning. *Advances in Neural Information Processing Systems*, 34:15070–15083, 2021.

H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1675–1684, 2016.

H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154*, 2017.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

A. Lamy, Z. Zhong, A. K. Menon, and N. Verma. Noise-tolerant fair classification. In *Advances in Neural Information Processing Systems*, pages 294–306, 2019.

M. T. Lash, Q. Lin, N. Street, J. G. Robinson, and J. Ohlmann. Generalized inverse classification. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 162–170. SIAM, 2017.

N. D. Lawrence. Data science and digital systems: The 3ds of machine learning systems design. *arXiv preprint arXiv:1903.11241*, 2019.

Y. LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

B. C. G. Lee, K. Lo, D. Downey, and D. S. Weld. Explanation-based tuning of opaque machine learners with application to paper recommendation. *arXiv preprint arXiv:2003.04315*, 2020.

J. Lee and N. Moray. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*, 35(10):1243–1270, 1992.

J. D. Lee and N. Moray. Trust, self-confidence, and operators' adaptation to automation. *International journal of human-computer studies*, 40(1):153–184, 1994.

J. D. Lee and K. A. See. Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1):50–80, 2004.

K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

M. K. Lee. Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management. *Big Data & Society*, 5(1):2053951718756684, 2018.

S. Lewandowsky, M. Mundy, and G. Tan. The dynamics of trust: Comparing humans to automation. *Journal of Experimental Psychology: Applied*, 6(2):104, 2000.

R. J. Lewicki and C. Brinsfield. Trust repair. *Annual Review of Organizational Psychology and Organizational Behavior*, 4:287–313, 2017.

D. Ley, U. Bhatt, and A. Weller. Diverse, global and amortised counterfactual explanations for uncertainty estimates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7390–7398, 2022.

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

L. Li, S. Nakandala, and A. Kumar. Intermittent human-in-the-loop model selection using cerebro: a demonstration. *Proceedings of the VLDB Endowment*, 14(12):2687–2690, 2021.

X. Li, N. C. Dvornek, Y. Zhou, J. Zhuang, P. Ventola, and J. S. Duncan. Efficient interpretation of deep learning models using graph structure and cooperative game theory: Application to ASD biomarker discovery. *arXiv preprint arXiv:1812.06181*, 2018.

M. W. Libbrecht, J. A. Bilmes, and W. S. Noble. Choosing non-redundant representative subsets of protein sequence data sets using submodular optimization. *Proteins: Structure, Function, and Bioinformatics*, 86(4):454–466, 2018.

S. Lichtenstein and J. R. Newman. Empirical scaling of common verbal phrases associated with numerical probabilities. *Psychonomic science*, 9(10):563–564, 1967.

S. Lichtenstein, B. Fischhoff, and L. D. Phillips. Calibration of probabilities: The state of the art. *Decision making and change in human affairs*, pages 275–324, 1977.

H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 510–520, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

J. Lin, W.-M. Chen, Y. Lin, C. Gan, S. Han, et al. Mcunet: Tiny deep learning on iot devices. *Advances in Neural Information Processing Systems*, 33:11711–11722, 2020.

F. Linkov, A. Sanei-Moghaddam, R. P. Edwards, P. J. Lounder, N. Ismail, S. L. Goughnour, C. Kang, S. M. Mansuria, and J. T. Comerci. Implementation of hysterectomy pathway: impact on complications. *Women's Health Issues*, 27(4):493–498, 2017.

D. Linsley, D. Shiebler, S. Eberhardt, and T. Serre. Learning what and where to attend. In *International Conference on Learning Representations*, 2018.

I. M. Lipkus and J. G. Hollands. The visual communication of risk. *JNCI monographs*, 1999 (25):149–163, 1999.

P. Lipton. Understanding without explanation. *Scientific understanding: Philosophical perspectives*, pages 43–63, 2009.

Z. C. Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.

J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020.

W. Liu, B. Dai, A. Humayun, C. Tay, C. Yu, L. B. Smith, J. M. Rehg, and L. Song. Iterative machine teaching. In *International Conference on Machine Learning*, pages 2149–2158. PMLR, 2017.

J. M. Logg, J. A. Minson, and D. A. Moore. Algorithm appreciation: People prefer algorithmic to human judgment. *Organizational Behavior and Human Decision Processes*, 151:90–103, 2019.

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*, pages 150–158, 2012.

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*, pages 623–631, 2013.

J. Lu, D. Lee, T. W. Kim, and D. Danks. Good explanation for algorithmic transparency. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 93–93, 2020.

A. Lucic, M. Srikumar, U. Bhatt, A. Xiang, A. Taly, Q. V. Liao, and M. de Rijke. A multistakeholder approach towards evaluating ai transparency mechanisms. *arXiv preprint arXiv:2103.14976*, 2021.

K. Lum and W. Isaac. To predict and serve? *Significance*, 13(5):14–19, 2016.

S. M. Lundberg. Explaining quantitative measures of fairness. *Fair & Responsible AI Workshop @ CHI2020*, 2020.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K.-W. Low, S.-F. Newman, J. Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering*, 2(10):749, 2018.

D. J. MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

D. Madras, T. Pitassi, and R. Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, pages 6147–6157, 2018.

D. Manzey, J. Reichenbach, and L. Onnasch. Human performance consequences of automated decision aids: The impact of degree of automation and system experience. *Journal of Cognitive Engineering and Decision Making*, 6(1):57–87, 2012.

D. Martens and F. Provost. Explaining data-driven document classifications. *Mis Quarterly*, 38 (1):73–100, 2014.

N. Martinez, M. Bertran, and G. Sapiro. Minimax pareto fairness: A multi objective perspective. In *International Conference on Machine Learning*, pages 6755–6764. PMLR, 2020.

A. Mas-Colell, M. D. Whinston, J. R. Green, et al. *Microeconomic theory*, volume 1. Oxford university press New York, 1995.

J. Matejka and G. Fitzmaurice. Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1290–1294, 2017.

R. C. Mayer, J. H. Davis, and F. D. Schoorman. An integrative model of organizational trust. *Academy of management review*, 20(3):709–734, 1995.

R. McAllister, Y. Gal, A. Kendall, M. Van Der Wilk, A. Shah, R. Cipolla, and A. Weller. Concrete problems for autonomous vehicle safety: advantages of bayesian deep learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4745–4753, 2017.

J. M. McGuirl and N. B. Sarter. Supporting trust calibration and the effective use of decision aids by presenting dynamic system confidence information. *Human factors*, 48(4):656–665, 2006.

N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ArXiv*, abs/1908.09635, 2019.

D. A. Melis and T. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems (NeurIPS 2018)*, pages 7775–7784, 2018.

M. J. Metzger and A. J. Flanagin. Credibility and trust of information in online environments: The use of cognitive heuristics. *Journal of pragmatics*, 59:210–220, 2013.

U. Metzger and R. Parasuraman. Automation in future air traffic management: Effects of decision aid reliability on controller performance and mental workload. *Human factors*, 47 (1):35–49, 2005.

Microsoft. Our approach: Microsoft ai principles, Mar 2019. URL https://www.microsoft.com/en-us/ai/our-approach-to-ai.

K. Miettinen. Introduction to multiobjective optimization: Noninteractive approaches. In *Multiobjective optimization*, pages 1–26. Springer, 2008.

S. M. Miller. Monitoring and blunting: validation of a questionnaire to assess styles of information seeking under threat. *Journal of personality and social psychology*, 52(2):345, 1987.

T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

S. Milli, J. Miller, A. D. Dragan, and M. Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 230–239, 2019a.

S. Milli, L. Schmidt, A. Dragan, and M. Hardt. Model reconstruction from model explanations. *In Proceedings of ACM FAT* 2019*, 2019b.

W. Mimra, A. Rasch, and C. Waibel. Second opinions in markets for expert services: Experimental evidence. *Journal of Economic Behavior & Organization*, 131:106–125, 2016.

B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause. Lazier than lazy greedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229. ACM, 2019.

C. Molnar. Interpretable machine learning. 2019. *URL https://christophm. github. io/interpretable-ml-book*, 2019.

E. Montague and J. Xu. Understanding active and passive users: The effects of an active user using normal, hard and unreliable technologies on user assessment of trust in technology and co-user. *Applied ergonomics*, 43(4):702–712, 2012.

G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital signal processing*, 73:1–15, 2018.

K. L. Mosier and L. J. Skitka. 10 human decision makers and automated decision aids: Made for each other? In *Automation and human performance: Theory and applications*, page 120. CRC Press New York, NY, 1996.

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*, pages 607–617, 2020.

H. Mozannar and D. Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pages 7076–7087. PMLR, 2020.

H. Mozannar, G. Bansal, A. Fourney, and E. Horvitz. Reading between the lines: Modeling user behavior and costs in ai-assisted programming. *arXiv preprint arXiv:2210.14306*, 2022a.

H. Mozannar, A. Satyanarayan, and D. Sontag. Teaching humans when to defer to a classifier via exemplars. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5323–5331, 2022b.

E. Mylonakis, M. Paliou, T. C. Greenbough, T. P. Flaningan, N. L. Letvin, and J. D. Rich. Report of a false-positive hiv test result and the potential use of additional tests in establishing hiv serostatus. *Archives of internal medicine*, 160(15):2386–2388, 2000.

M. P. Naeini, G. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Hybrid models with deep and invertible features. In *International Conference on Machine Learning*, pages 4723–4732, 2019.

R. M. Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

A. Noriega-Campero, M. A. Bakker, B. Garcia-Bulle, and A. Pentland. Active fairness in algorithmic decision making. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 77–83, 2019.

G. Noti and Y. Chen. Learning when to advise human decision makers. *arXiv preprint arXiv:2209.13578*, 2022.

B. Nushi, E. Kamar, E. Horvitz, and D. Kossmann. On human intellect and machine failures: Troubleshooting integrative machine learning systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

B. Nushi, E. Kamar, and E. Horvitz. Towards accountable ai: Hybrid human-machine analyses for characterizing system failure. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 6, pages 126–135, 2018.

Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

OECD. Recommendation of the council on artificial intelligence, May 2019. URL https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0449.

A. O'Hagan, C. E. Buck, A. Daneshkhah, J. R. Eiser, P. H. Garthwaite, D. J. Jenkinson, J. E. Oakley, and T. Rakow. *Uncertain judgements: eliciting experts' probabilities*. John Wiley & Sons, 2006.

N. Okati, A. De, and M. Rodriguez. Differentiable learning under triage. *Advances in Neural Information Processing Systems*, 34:9140–9151, 2021.

O. O'Neill. Linking trust to trustworthiness. *International Journal of Philosophical Studies*, 26 (2):293–300, 2018.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13991–14002, 2019.

S. Palan and C. Schitter. Prolific. ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, 17:22–27, 2018.

R. Parasuraman and D. H. Manzey. Complacency and bias in human use of automation: An attentional integration. *Human factors*, 52(3):381–410, 2010.

R. Parasuraman and C. A. Miller. Trust and etiquette in high-criticality automated systems. *Communications of the ACM*, 47(4):51–55, 2004.

R. Parasuraman and V. Riley. Humans and automation: Use, misuse, disuse, abuse. *Human factors*, 39(2):230–253, 1997.

R. Parasuraman, M. Mouloua, and R. Molloy. Effects of adaptive task allocation on monitoring of automated systems. *Human factors*, 38(4):665–679, 1996.

M. Pawelczyk, K. Broelemann, and G. Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020*, pages 3126–3132, 2020.

K. Pazouki, N. Forbes, R. A. Norman, and M. D. Woodward. Investigation on the impact of human-automation interaction in maritime operations. *Ocean engineering*, 153:297–304, 2018.

J. Pearl. *Causality: models, reasoning and inference*, volume 29. Springer, 2000.

J. Pearl. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60, 2019.

B. A. Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

J. Pek and T. Van Zandt. Frequentist and bayesian approaches to data analysis: Evaluation and estimation. *Psychology Learning & Teaching*, 19(1):21–35, 2020.

C. Peter and T. Koch. When debunking scientific myths fails (and when it does not) the backfire effect in the context of journalistic coverage and immediate judgments as prevention strategy. *Science Communication*, 38(1):3–25, 2016.

E. Peters, J. Hibbard, P. Slovic, and N. Dieckmann. Numeracy skill and the communication, comprehension, and use of risk-benefit information. *Health affairs*, 26(3):741–748, 2007.

J. C. Peterson, R. M. Battleday, T. L. Griffiths, and O. Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9617–9626, 2019.

R. E. Petty and J. T. Cacioppo. The elaboration likelihood model of persuasion. In *Communication and persuasion*, pages 1–24. Springer, 1986.

G. Phillips-Wren. Ai tools in decision making support systems: a review. *International Journal on Artificial Intelligence Tools*, 21(02):1240005, 2012.

E. Pierson, C. Simoiu, J. Overgoor, S. Corbett-Davies, D. Jenson, A. Shoemaker, V. Ramachandran, P. Barghouty, C. Phillips, R. Shroff, et al. A large-scale analysis of racial disparities in police stops across the united states. *Nature human behaviour*, 4(7):736–745, 2020.

F. Pinto, M. O. Sampaio, and P. Bizarro. Automatic model monitoring for data streams. *arXiv preprint arXiv:1908.04240*, 2019.

G. Plumb, D. Molitor, and A. S. Talwalkar. Model agnostic supervised local explanations. In *Advances in Neural Information Processing Systems*, pages 2515–2524, 2018.

G. Plumb, M. Al-Shedivat, Á. A. Cabrera, A. Perer, E. Xing, and A. Talwalkar. Regularizing black-box models for improved interpretability. *Advances in Neural Information Processing Systems*, 33:10526–10536, 2020.

G. Plumb, M. T. Ribeiro, and A. Talwalkar. Finding and fixing spurious patterns with explanations. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

M. C. Politi, P. K. Han, and N. F. Col. Communicating the uncertainty of harms and benefits of medical interventions. *Medical Decision Making*, 27(5):681–695, 2007.

F. Poursabzi-Sangdeh, D. G. Goldstein, J. M. Hofman, J. W. Wortman Vaughan, and H. Wallach. Manipulating and measuring model interpretability. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–52, 2021.

R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, and P. Flach. Face: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020.

A. Prasad, S. Jegelka, and D. Batra. Submodular meets structured: Finding diverse subsets in exponentially-large structured item sets. *Advances in Neural Information Processing Systems*, 2014.

I. D. Raji and J. Yang. About ml: Annotation and benchmarking on understanding and transparency of machine learning lifecycles. *arXiv preprint arXiv:1912.06166*, 2019.

P. Rajpurkar, E. Chen, O. Banerjee, and E. J. Topol. Ai in health and medicine. *Nature medicine*, 28(1):31–38, 2022.

T. Rakow. Risk, uncertainty and prophet: The psychological insights of frank h. knight. *Judgment and Decision Making*, 5(6):458, 2010.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access, 2017.

V. F. Reyna and C. J. Brainerd. Numeracy, ratio bias, and denominator neglect in judgments of risk and probability. *Learning and individual differences*, 18:89–107, 2008.

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*, pages 1135–1144. ACM, 2016.

M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, 2018.

B. D. Roads and B. C. Love. Enriching imagenet with human similarity judgments and psychological embeddings. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 3547–3557, 2021.

P. Robinette, A. M. Howard, and A. R. Wagner. Timing is key for robot trust repair. In *International conference on social robotics*, pages 574–583. Springer, 2015.

C. E. Roda. *Human attention and its implications for human-computer interaction.* Cambridge University Press, 2011.

A. S. Ross, M. C. Hughes, and F. Doshi-Velez. Right for the right reasons: training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2662–2670. AAAI Press, 2017.

D. M. Rousseau, S. B. Sitkin, R. S. Burt, and C. Camerer. Not so different after all: A cross-discipline view of trust. *Academy of management review*, 23(3):393–404, 1998.

P. J. Rousseeuw, I. Ruts, and J. W. Tukey. The bagplot: a bivariate boxplot. *The American Statistician*, 53(4):382–387, 1999.

D.-H. Ruben. *Explaining explanation*. Routledge, 2015.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206, 2019.

I. T. Ruginski, A. P. Boone, L. M. Padilla, L. Liu, N. Heydari, H. S. Kramer, M. Hegarty, W. B. Thompson, D. H. House, and S. H. Creem-Regehr. Non-expert interpretations of hurricane forecast uncertainty visualizations. *Spatial Cognition & Computation*, 16(2):154–172, 2016.

C. Russell. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 20–28, 2019.

I. Ryazanov, A. T. Nylund, D. Basu, I.-M. Hassellöv, and A. Schliep. Deep learning for deep waters: An expert-in-the-loop machine learning framework for marine sciences. *Journal of Marine Science and Engineering*, 9(2):169, 2021.

M. Salem, G. Lakatos, F. Amirabdollahian, and K. Dautenhahn. Would you trust a (faulty) robot? Effects of error, task type and personality on human-robot cooperation and trust. In *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 1–8. IEEE, 2015.

S. Santurkar, D. Tsipras, M. Elango, D. Bau, A. Torralba, and A. Madry. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems*, 34, 2021.

J. Sauer, C.-S. Kao, and D. Wastell. A comparison of adaptive and adaptable automation under different levels of environmental stress. *Ergonomics*, 55(8):840–853, 2012.

R. T. Scheife, L. E. Hines, R. D. Boyce, S. P. Chung, J. D. Momper, C. D. Sommer, D. R. Abernethy, J. R. Horn, S. J. Sklar, S. K. Wong, et al. Consensus recommendations for systematic evaluation of drug–drug interaction evidence for clinical decision support. *Drug safety*, 38(2):197–206, 2015.

E. Schmidt, B. Work, S. Catz, S. Chien, C. Darby, K. Ford, J.-M. Griffiths, E. Horvitz, A. Jassy, W. Mark, et al. National security commission on artificial intelligence (ai). *National Security Commission on Artificial Intellegence, Tech. Rep*, 2021.

P. Schmidt, F. Biessmann, and T. Teubner. Transparency and trust in artificial intelligence systems. *Journal of Decision Systems*, 29(4):260–278, 2020.

J. Schoeffer, N. Kuehl, and I. Valera. A ranking approach to fair classification. In *ACM SIGCAS Conference on Computing and Sustainable Societies*, pages 115–125, 2021.

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

P. Schulam and S. Saria. Can you trust this prediction? auditing pointwise reliability after learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1022–1031, 2019.

D. A. Schum, G. Tecuci, D. Marcu, and M. Boicu. Toward cognitive assistants for complex decision making under uncertainty. *Intelligent Decision Technologies*, 8(3):231–250, 2014.

R. W. Schvaneveldt, F. T. Durso, T. E. Goldsmith, T. J. Breen, N. M. Cooke, R. G. Tucker, and J. C. De Maio. Measuring the structure of expertise. *International journal of man-machine studies*, 23(6):699–728, 1985.

B. Schwartz, A. Ward, J. Monterosso, S. Lyubomirsky, K. White, and D. R. Lehman. Maximizing versus satisficing: happiness is a matter of choice. *Journal of personality and social psychology*, 83(5):1178, 2002.

C. Scott, G. Blanchard, and G. Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *Conference On Learning Theory*, pages 489–511, 2013.

A. D. Selbst and S. Barocas. The intuitive appeal of explainable machines. *Fordham L. Rev.*, 87:1085, 2018.

A. D. Selbst, D. Boyd, S. A. Friedler, S. Venkatasubramanian, and J. Vertesi. Fairness and abstraction in sociotechnical systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 59–68, 2019.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

L. Semenova and C. Rudin. A study in rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning. *arXiv preprint arXiv:1908.01755*, 2019.

Y. Seong and A. M. Bisantz. The impact of cognitive feedback on judgment performance and trust with decision aids. *International Journal of Industrial Ergonomics*, 38(7-8):608–625, 2008.

G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(Mar):371–421, 2008.

D. Shahaf and E. Horvitz. Generalized task markets for human and machine computation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 986–993, 2010.

S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

L. S. Shapley. A value for n-person games. In *Contributions to the Theory of Games II*, pages 307–317, 1953.

S. Sharma, J. Henderson, and J. Ghosh. Certifai: A common framework to provide explanations and analyse the fairness and robustness of black-box models. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 166–172, 2020.

H. Shen and T.-H. Huang. How useful are the machine-generated interpretations to general users? A human evaluation on guessing the incorrectly predicted labels. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 8, pages 168–172, 2020.

R. Shokri, M. Strobel, and Y. Zick. On the privacy risks of model explanations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 231–241, 2021.

A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70 (ICML 2017)*, pages 3145–3153. Journal of Machine Learning Research, 2017.

M. Siegrist, T. C. Earle, and H. Gutscher. Test of a trust and confidence model in the applied context of electromagnetic field (emf) risks. *Risk Analysis: An International Journal*, 23(4): 705–716, 2003.

M. Siegrist, H. Gutscher, and T. C. Earle. Perception of risk: the influence of general trust, and general confidence. *Journal of risk research*, 8(2):145–156, 2005.

P. Y. Simard, S. Amershi, D. M. Chickering, A. E. Pelton, S. Ghorashi, C. Meek, G. Ramos, J. Suh, J. Verwey, M. Wang, et al. Machine teaching: A new paradigm for building machine learning systems. *arXiv preprint arXiv:1707.06742*, 2017.

H. A. Simon. The new science of management decision. *The Ford distinguished lectures.*, 1960.

S. Singla, E. Wallace, S. Feng, and S. Feizi. Understanding impacts of high-order loss approximations and features in deep learning interpretation. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5848–5856, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020a.

D. Slack, S. Hilgard, S. Singh, and H. Lakkaraju. How much should i trust you? modeling uncertainty of black box explanations. *arXiv preprint arXiv:2008.05030*, 2020b.

D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

S. Song, K. Chaudhuri, and A. D. Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE, 2013.

R. M. Sorrentino, D. R. Bobocel, M. Z. Gitta, J. M. Olson, and E. C. Hewitt. Uncertainty orientation and persuasion: Individual differences in the effects of personal relevance on social judgments. *Journal of Personality and social Psychology*, 55(3):357, 1988.

D. Spiegelhalter. Risk and uncertainty communication. *Annual Review of Statistics and Its Application*, 4:31–60, 2017.

D. Spiegelhalter, M. Pearson, and I. Short. Visualizing uncertainty about the future. *science*, 333(6048):1393–1400, 2011.

J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust bayesian neural networks. *Advances in neural information processing systems*, 29, 2016.

A. Squirrell. Conformity assessment: providing confidence in testing and calibration. *Accreditation and Quality Assurance*, 13(9):543–546, 2008.

M. Steyvers and A. Kumar. Three challenges for ai-assisted decision-making. *Perspectives on Psychological Science*, 2022. doi: 10.1177/17456916231181102.

M. Steyvers, H. Tejeda, G. Kerrigan, and P. Smyth. Bayesian modeling of human–ai complementarity. *Proceedings of the National Academy of Sciences*, 119(11), 2022.

K. Stowers, N. Kasdaglis, M. Rupp, J. Chen, D. Barber, and M. Barnes. Insights into human-agent teaming: Intelligent agent transparency and uncertainty. In *Advances in Human Factors in Robots and Unmanned Systems*, pages 149–160. Springer, 2017.

E. Straitouri, L. Wang, N. Okati, and M. G. Rodriguez. Provably improving expert predictions with conformal prediction. *arXiv preprint arXiv:2201.12006*, 2022.

M. Strathern. 'improving ratings': audit in the british university system. *European review*, 5 (3):305–321, 1997.

E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3):647–665, 2014.

M. Sugiyama, N. D. Lawrence, A. Schwaighofer, et al. *Dataset shift in machine learning*. The MIT Press, 2017.

S. S. Sundar. *The MAIN model: A heuristic approach to understanding technology effects on credibility*. MacArthur Foundation Digital Media and Learning Initiative, 2008.

S. S. Sundar and J. Kim. Machine heuristic: When we trust computers more than humans with our personal information. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–9, 2019.

M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70 (ICML 2017)*, pages 3319–3328. Journal of Machine Learning Research, 2017.

C. R. Sunstein. *Decisions about Decisions: Practical Reason in Ordinary Life*. Cambridge University Press, 2023.

H. Suresh and J. Guttag. A framework for understanding sources of harm throughout the machine learning life cycle. In *Equity and access in algorithms, mechanisms, and optimization*, pages 1–9. ACM, 2021.

H. Suresh, N. Lao, and I. Liccardi. Misplaced trust: Measuring the interference of machine learning in human decision-making. In *12th ACM Conference on Web Science*, pages 315–324, 2020.

R. J. Swartz, D. D. Cox, S. B. Cantor, K. Davies, and M. Follen. Inverse decision theory: characterizing losses for a decision rule with applications in cervical cancer screening. *Journal of the American Statistical Association*, 101(473):1–8, 2006.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

C. Tan. On the diversity and limits of human explanations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2173–2188, 2022.

S. Tang, A. Ghorbani, R. Yamashita, S. Rehman, J. A. Dunnmon, J. Zou, and D. L. Rubin. Data valuation for medical imaging using shapley value and application to a large-scale chest x-ray dataset. *Scientific reports*, 11(1):8366, 2021.

H. Tejeda, A. Kumar, P. Smyth, and M. Steyvers. Ai-assisted decision-making: A cognitive modeling approach to infer latent reliance strategies. *Computational Brain & Behavior*, pages 1–18, 2022.

C. Tekin and E. Turğay. Multi-objective contextual multi-armed bandit with a dominant objective. *IEEE Transactions on Signal Processing*, 66(14):3799–3813, 2018.

I. Thielmann and B. E. Hilbig. Trust: An integrative review from a person–situation perspective. *Review of General Psychology*, 19(3):249–277, 2015.

R. Tomsett, A. Preece, D. Braines, F. Cerutti, S. Chakraborty, M. Srivastava, G. Pearson, and L. Kaplan. Rapid trust calibration through interpretable and uncertainty-aware ai. *Patterns*, 1 (4):100049, 2020.

S. Tonekaboni, S. Joshi, M. D. McCradden, and A. Goldenberg. What clinicians want: Contextualizing explainable machine learning for clinical end use. In *Machine Learning for Healthcare Conference*, pages 359–380, 2019.

A. Tripp, E. Daxberger, and J. M. Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33:11259–11272, 2020.

A. M. Tseng, A. Shrikumar, and A. Kundaje. Fourier-transform-based attribution priors improve the interpretability and stability of deep learning models for genomics. *bioRxiv*, 2020.

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

E. Turgay, D. Oner, and C. Tekin. Multi-objective contextual bandit problem with similarity information. In *International Conference on Artificial Intelligence and Statistics*, pages 1673–1681. PMLR, 2018.

A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *science*, 185(4157):1124–1131, 1974.

A. Tversky and D. Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5(4):297–323, 1992.

A. Tversky and D. Kahneman. On the reality of cognitive illusions. *Psychological Review*, 103 (3):582–591, 1996.

A. Uma, T. Fornaciari, D. Hovy, S. Paun, B. Plank, and M. Poesio. A case for soft loss functions. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 8(1): 173–177, Oct. 2020. URL https://ojs.aaai.org/index.php/HCOMP/article/view/7478.

A. Uma, D. Almanea, and M. Poesio. Scaling and disagreements: Bias, noise, and ambiguity. *Frontiers in Artificial Intelligence*, 5, 2022.

US Department of HHS. Summary of the hipaa privacy rule, 2013.

B. Ustun, A. Spangher, and Y. Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 10–19. ACM, 2019.

A. M. Van Der Bles, S. Van Der Linden, A. L. Freeman, J. Mitchell, A. B. Galvao, L. Zaval, and D. J. Spiegelhalter. Communicating uncertainty about facts, numbers and science. *Royal Society open science*, 6(5):181870, 2019.

A. M. Van Der Bles, S. van der Linden, A. L. Freeman, and D. J. Spiegelhalter. The effects of communicating uncertainty on public trust in facts and numbers. *Proceedings of the National Academy of Sciences*, 117(14):7672–7683, 2020.

L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

J. van der Waa, E. Nieuwburg, A. Cremers, and M. Neerincx. Evaluating xai: A comparison of rule-based and example-based explanations. *Artificial Intelligence*, 291:103404, 2021.

M. Veale. A critical take on the policy recommendations of the eu high-level expert group on artificial intelligence. *European Journal of Risk Regulation*, 11(1):e1, 2020.

S. Venkatasubramanian and M. Alfano. The philosophical basis of algorithmic recourse. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 284–293, 2020.

F. Volk, S. Hauke, D. Dieth, and M. Mühlhäuser. Communicating and visualising multicriterial trustworthiness under uncertainty. In *2014 Twelfth Annual International Conference on Privacy, Security and Trust*, pages 391–397. IEEE, 2014.

J. Von Kügelgen, A.-H. Karimi, U. Bhatt, I. Valera, A. Weller, and B. Schölkopf. On the fairness of causal algorithmic recourse. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 9584–9594, 2022.

V. Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56(2):153–173, 1998.

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, 01 2005. doi: 10.1007/b106715.

S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gpdr. *Harv. JL & Tech.*, 31:841, 2017.

S. Wan, Y. Hou, F. Bao, Z. Ren, Y. Dong, Q. Dai, and Y. Deng. Human-in-the-loop low-shot learning. *IEEE Transactions on Neural Networks and Learning Systems*, 32(7):3287–3292, 2020.

D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016.

H. Wang, B. Ustun, and F. Calmon. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. In *International Conference on Machine Learning*, pages 6618–6627. PMLR, 2019.

L. Wang, G. A. Jamieson, and J. G. Hollands. Trust and reliance on an automated combat identification system. *Human factors*, 51(3):281–291, 2009.

P. Wang, J. Wang, P. Paranamana, and P. Shafto. A mathematical theory of cooperative communication. *Advances in Neural Information Processing Systems*, 33:17582–17593, 2020.

S. Wang and M. Gupta. Deontological ethics by monotonicity shape constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 2043–2054. PMLR, 2020.

W. Wang and J. Shen. Deep visual attention prediction. *IEEE Transactions on Image Processing*, 27(5):2368–2378, 2017.

Z. J. Wang, A. Kale, H. Nori, P. Stella, M. E. Nunnally, D. H. Chau, M. Vorvoreanu, J. W. Vaughan, and R. Caruana. Interpretability, Then What? Editing Machine Learning Models to Reflect Human Knowledge and Values. In *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2022.

J. S. Warm, W. N. Dember, and P. A. Hancock. Vigilance and workload in automated systems. In *Automation and human performance: Theory and applications*, pages 183–200. CRC Press, 2018.

H. J. Weerts, W. van Ipenburg, and M. Pechenizkiy. A human-grounded evaluation of shap for alert processing. *Proceedings of KDD Workshop on Explainable AI*, 2019.

J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, and Y. Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022.

E. Weinberger, J. Janizek, and S.-I. Lee. Learning deep attribution priors based on prior knowledge. *Advances in Neural Information Processing Systems*, 33:14034–14045, 2020.

A. Weller. Transparency: motivations and challenges. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 23–40. Springer, 2019.

J. Whitehill, T.-f. Wu, J. Bergsma, J. Movellan, and P. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in neural information processing systems*, 22, 2009.

Y. Wiener and R. El-Yaniv. Agnostic selective classification. In *Advances in neural information processing systems*, pages 1665–1673, 2011.

B. Wilder, E. Horvitz, and E. Kamar. Learning to complement humans. In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 1526–1533. International Joint Conferences on Artificial Intelligence Organization, 7 2020.

C. O. Wilke. *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*. O'Reilly Media, 1st edition edition, 2019. ISBN 978-1-4920-3108-6.

N. Wolczynski, M. Saar-Tsechansky, and T. Wang. Learning to advise humans by leveraging algorithm discretion. *arXiv preprint arXiv:2210.12849*, 2022.

T. Wu, M. T. Ribeiro, J. Heer, and D. S. Weld. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 747–763, 2019.

D. U. Wulff, M. Mergenthaler-Canseco, and R. Hertwig. A meta-analytic review of two modes of learning and the description-experience gap. *Psychological bulletin*, 144(2):140, 2018.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Y. Xiao, P. P. Liang, U. Bhatt, W. Neiswanger, R. Salakhutdinov, and L.-P. Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7273–7284, 2022.

M.-g. Xie and K. Singh. Confidence distribution, the frequentist distribution estimator of a parameter: A review. *International Statistical Review*, 81(1):3–39, 2013.

D. Xu, S. Yuan, L. Zhang, and X. Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE, 2018.

F. Yang, Z. Huang, J. Scholtz, and D. L. Arendt. How do visual explanations foster end users' appropriate trust in machine learning? In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 189–201, 2020.

M. Yang and B. Kim. BIM: Towards quantitative evaluation of interpretability methods with ground truth. *arXiv:1907.09701*, 2019.

Q. Yang, Y. Hao, K. Quan, S. Yang, Y. Zhao, V. Kuleshov, and F. Wang. Harnessing biomedical literature to calibrate clinicians' trust in ai decision support systems. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2023a.

S. Yang, O. Nachum, Y. Du, J. Wei, P. Abbeel, and D. Schuurmans. Foundation models for decision making: Problems, methods, and opportunities. *arXiv preprint arXiv:2303.04129*, 2023b.

Y. Yang, E. Kandogan, Y. Li, P. Sen, and W. S. Lasecki. A study on interaction in human-in-the-loop machine learning for text analytics. In *IUI Workshops*, 2019.

N. Ye and Z. Zhu. Bayesian adversarial learning. *Advances in Neural Information Processing Systems*, 31, 2018.

C.-K. Yeh, J. Kim, I. E.-H. Yen, and P. K. Ravikumar. Representer point selection for explaining deep neural networks. In *Advances in neural information processing systems*, pages 9291–9301, 2018.

C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. I. Inouye, and P. K. Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

M. B. Zafar, I. Valera, M. Rodriguez, K. Gummadi, and A. Weller. From parity to preference-based notions of fairness in classification. In *Advances in Neural Information Processing Systems*, pages 229–239, 2017a.

M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017b.

M. R. Zafar and N. M. Khan. Dlime: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *arXiv preprint arXiv:1906.10263*, 2019.

O. F. Zaidan, J. Eisner, and C. Piatko. Machine learning with annotator rationales to reduce annotation cost. In *Proceedings of the NIPS* 2008 workshop on cost sensitive learning*, pages 260–267, 2008.

J. Zerilli, A. Knott, J. Maclaurin, and C. Gavaghan. Algorithmic decision-making and the control problem. *Minds and Machines*, 29(4):555–578, 2019a.

J. Zerilli, A. Knott, J. Maclaurin, and C. Gavaghan. Transparency in algorithmic and human decision-making: is there a double standard? *Philosophy & Technology*, 32(4):661–683, 2019b.

J. Zerilli, U. Bhatt, and A. Weller. Transparency Modulates Trust in Artificial Intelligence. *Patterns*, 2022.

H. Zhang and L. T. Maloney. Ubiquitous log odds: a common representation of probability and frequency distortion in perception, action, and cognition. *Frontiers in neuroscience*, 6:1, 2012.

H. Zhang, E. Horvitz, Y. Chen, and D. C. Parkes. Task routing for prediction tasks. In *11th International Conference on Autonomous Agents and Multiagent Systems 2012: Innovative Applications Track, AAMAS 2012*, pages 824–831. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2012.

H. Zhang, X. Ren, and L. T. Maloney. The bounded rationality of probability distortion. *BioRxiv*, page 662429, 2020a.

J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

W. Zhang and J. C. Weiss. Fair decision-making under uncertainty. In *2021 IEEE international conference on data mining (ICDM)*, pages 886–895. IEEE, 2021.

Y. Zhang, R. K. Bellamy, and W. A. Kellogg. Designing information for remediating cognitive biases in decision-making. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 2211–2220, 2015.

Y. Zhang, K. Song, Y. Sun, S. Tan, and M. Udell. "why should you trust my explanation?" understanding uncertainty in lime explanations, 2019.

Y. Zhang, Q. V. Liao, and R. K. E. Bellamy. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 295–305, New York, NY, USA, 2020b. Association for Computing Machinery. ISBN 9781450369367. doi: 10.1145/3351095.3372852.

S. Zhao, M. M. Fard, H. Narasimhan, and M. Gupta. Metric-optimized example weights. In *International Conference on Machine Learning*, pages 7533–7542. PMLR, 2019.

J. Zhou, Z. Li, H. Hu, K. Yu, F. Chen, Z. Li, and Y. Wang. Effects of influence on user trust in predictive decision making. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.

Y. Zhou and D. Danks. Different "intelligibility" for different folks. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 194–199, 2020.

B. J. Zikmund-Fisher, D. M. Smith, P. A. Ubel, and A. Fagerlin. Validation of the subjective numeracy scale: effects of low numeracy on comprehension of risk communications and utility elicitations. *Medical Decision Making*, 27(5):663–671, 2007.

Appendix A

Publications

Below is a list of publications that are included in this thesis.

1. **Perspectives on Incorporating Expert Feedback into Model Updates**
   Patterns. Cell Press 2023
   Valerie Chen*, **UB***, Hoda Heidari, Adrian Weller, Ameet Talwalkar

2. **How Transparency Modulates Trust in Artificial Intelligence**
   Patterns. Cell Press 2022
   John Zerilli, **UB**, Adrian Weller

3. **Uncertainty as a Form of Transparency: Measuring, Communicating, and Using Uncertainty**
   AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society (AIES) 2021
   UB, Javier Antorán, Yunfeng Zhang, Q. Vera Liao, Prasanna Sattigeri, Riccardo Fogliato, Gabrielle Melançon, Ranganath Krishnan, Jason Stanley, Omesh Tickoo, Lama Nachman, Rumi Chunara, Madhulika Srikumar, Adrian Weller, Alice Xiang

4. **Trust in Artificial Intelligence: Clinicians are Essential**
   Healthcare Information Technology for Cardiovascular Medicine. Springer 2021
   UB, Zohreh Shams

5. **Getting a CLUE: A Method for Explaining Uncertainty Estimates**
   International Conference on Learning Representations (ICLR) 2021 **(Oral)**
   Javier Antorán, **UB**, Tameem Adel, Adrian Weller, José Miguel Hernández-Lobato

6. **Machine Learning Explainability for External Stakeholders**
   ICML Workshop on Human Interpretability 2020
   UB, McKane Andrus, Adrian Weller, Alice Xiang

7. **Explainable Machine Learning in Deployment**
   ACM Conference on Fairness, Accountability, and Transparency (FAccT) 2020
   UB, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M.F. Moura, Peter Eckersley

8. **Learning Personalized Decision Support Policies**
   Under Review.
   UB*, Valerie Chen*, Katherine Collins, Parameswaran Kamalaruban, Emma Kallina, Adrian Weller, Ameet Talwalkar

9. **DIVINE: Diverse Influential Training Points**
   Forthcoming.
   UB, Isabel Chien, Muhammad Bilal Zafar, Adrian Weller

Collating across the acknowledgements of the papers above, I would like to thank the following individuals for their advice, comments, and/or edits on versions of the papers that appear in this thesis: Karina Alexanyan, James Allingham, Kasun Amarasinghe, Varun Babbar, Gagan Bansal, Kemi Bello, Alex Cabrera, Rich Caruana, Kartik Chandra, Amit Dhurandhar, Krishna Gade, Konstantinos Georgatzis, Krishna Gummadi, Jette Henderson, Ken Holstein, Hudson Hongo, Jessica Hullman, Nari Johnson, Bahador Kaleghi, Matthew Kay, Joon Sik Kim, Himabindu Lakkaraju, Katherine Lewis, Jeffrey Li, Alan Liu, Leqi Liu, Weiyang Liu, Peter Lo, Bradley Love, Terah Lyons, Saayeli Mukherji, Aditya Nori, Elre Oldewage, Erik Pazos, Gregory Plumb, Inioluwa Deborah Raji, Charvi Rastogi, Nicole Rigillo, Francesca Rossi, Elena Spitzer, Richard Tomsett, Jay Turcot, Richard E. Turner, Kush Varshney, Jennifer Wortman Vaughan, Carroll Wainwright, Dennis Wei, Timothy Ye, Edward Zhong, Gabi Zijderveld, and many other individuals, including reviewers.

Below is a list of papers that I was involved in during my PhD but *are not* in this thesis.

1. **Selective Concept Models: Permitting Stakeholder Customisation at Test-Time**
   AAAI Conference on Human Computation and Crowdsourcing (HCOMP) 2023
   Matthew Barker, Katherine Collins, Krishnamurthy Dvijotham, Adrian Weller, **UB**

2. **FeedbackLogs: Recording and Incorporating Stakeholder Feedback into Machine Learning Pipelines**
   ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO) 2023
   Matthew Barker, Emma Kallina, Dhananjay Ashok, Katherine Collins, Ashley Casovan, Adrian Weller, Ameet Talwalkar, Valerie Chen, **UB**

3. **GeValDi: Generative Validation of Discriminative Models**
   ICLR Workshop on Pitfalls of limited data and computation for Trustworthy ML 2023
   Vivek Palaniappan, Matt Ashman, Katherine Collins, Juyeon Heo, Adrian Weller, **UB**

4. **Harms from Increasingly Agentic Algorithmic Systems**
   ACM Conference on Fairness, Accountability, and Transparency (FAccT) 2023
   Alan Chan, Rebecca Salganik, Alva Markelius, Chris Pang, Nitarshan Rajkumar, Dmitrii Krasheninnikov, Lauro Langosco, Zhonghao He, Yawen Duan, Micah Carroll, Michelle Lin, Alex Mayhew, Katherine Collins, Maryam Molamohammadi, John Burden, Wanru Zhao, Shalaleh Rismani, Konstantinos Voudouris, **UB**, Adrian Weller, David Krueger, Tegan Maharaj

5. **Human-in-the-Loop Mixup**
   Uncertainty in Artificial Intelligence (UAI) 2023 **(Oral)**
   Katherine Collins, **UB**, Weiyang Liu, Vihari Piratla, Ilia Sucholutsky, Bradley Love, Adrian Weller

6. **On the Informativeness of Supervision Signals**
   Uncertainty in Artificial Intelligence (UAI) 2023
   Ilia Sucholutsky, Ruairidh McLennan Battleday, Katherine Collins, Raja Marjieh, Joshua Peterson, Pulkit Singh, **UB**, Nori Jacoby, Adrian Weller, Thomas Griffiths

7. **Human Uncertainty in Concept-Based AI Systems**
   AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society (AIES) 2023
   Katherine Collins, Matthew Barker, Mateo Espinosa Zarlenga, Naveen Raman, **UB**, Mateja Jamnik, Ilia Sucholutsky, Adrian Weller, Krishnamurthy Dvijotham

8. **Iterative Teaching by Data Hallucination**
   International Conference on Artificial Intelligence and Statistics (AISTATS) 2023
   Zeju Qiu, Weiyang Liu, Tim Xiao, Zhen Liu, Yucen Luo, **UB**, Adrian Weller, Bernhard Schölkopf

9. **Approximating Full Conformal Prediction at Scale via Influence Functions**
   AAAI Conference on Artificial Intelligence (AAAI) 2023
   Javier Abad Martinez, **UB**, Adrian Weller, Giovanni Cherubin

10. **Towards Robust Metrics for Concept Representation Evaluation**
    AAAI Conference on Artificial Intelligence (AAAI) 2023
    Mateo Espinosa Zarlenga, Pietro Barbiero, Zohreh Shams, Dmitry Kazhdan, **UB**, Adrian Weller, Mateja Jamnik

11. **Uncertainty Quantification with Pre-trained Language Models: A Large-Scale Empirical Analysis**
Conference on Empirical Methods in Natural Language Processing (EMNLP) 2022
Yuxin Xiao, Paul Pu Liang, **UB**, Willie Neiswanger, Ruslan Salakhutdinov, Louis-Philippe Morency

12. **Human-Centric Risk Controlling Predictive Sets**
NeurIPS Workshop on Human in the Loop Learning 2022
Varun Babbar, **UB**, Miri Zilka, Adrian Weller

13. **Eliciting and Learning with Soft Labels from Every Annotator**
AAAI Conference on Human Computation and Crowdsourcing (HCOMP) 2022
Katherine Collins*, **UB***, Adrian Weller

14. **On the Utility of Prediction Sets in Human-AI Teams**
International Joint Conference on Artificial Intelligence (IJCAI) 2022 **(Oral)**
Varun Babbar, **UB**, Adrian Weller

15. **Towards the Use of Saliency Maps for Explaining Low-Quality Electrocardiograms to End Users**
IJCAI Workshop on Explainable Artificial Intelligence 2022
Ana Lucic, Sheeraz Ahmad, Amanda Furtado Brinhosa, Vera Liao, Himani Agrawal, **UB**, Krishnaram Kenthapadi, Alice Xiang, Maarten de Rijke, Nicholas Drabowski

16. **Social Explainability of AI: The Impact of Non-Technical Explanations on Trust**
IJCAI Workshop on Explainable Artificial Intelligence 2022
Frens Kroeger, Bianca Slocombe, Isa Inuwa, Baker Kagimu, Beate Grawemeyer, **UB**

17. **Diverse and Amortised Counterfactual Explanations for Uncertainty Estimates**
AAAI Conference on Artificial Intelligence (AAAI) 2022
Daniel Ley, **UB**, Adrian Weller

18. **On the Fairness of Causal Algorithmic Recourse**
AAAI Conference on Artificial Intelligence (AAAI) 2022 **(Oral)**
Julius von Kügelgen, Amir-Hossein Karimi, **UB**, Isabel Valera, Adrian Weller, Bernhard Schölkopf

19. **Fast Conformal Classification using Influence Functions**
Symposium on Conformal and Probabilistic Prediction with Applications 2021
UB, Adrian Weller, Giovanni Cherubin

20. **A Multistakeholder Approach Towards Evaluating AI Transparency Mechanisms**
ACM CHI Workshop on Operationalizing Human-Centered Perspectives in Explainable AI 2021
Ana Lucic, Madhulika Srikumar, **UB**, Alice Xiang, Ankur Taly, Q. Vera Liao, Maarten de Rijke

21. **Do Concept Bottleneck Models Learn As Intended?**
ICLR Workshop on Responsible AI 2021
Andrei Margeloiu*, Matt Ashman*, **UB***, Yanzhi Chen, Mateja Jamnik, Adrian Weller

22. **FIMAP: Feature Importance by Minimal Adversarial Perturbation**
AAAI Conference on Artificial Intelligence (AAAI) 2021
Matt Chapman-Rounds, **UB**, Erik Pazos, Marc-Andre Schulz, Kostas Georgatzis

23. **Effects of Uncertainty on the Quality of Feature Importance Estimates**
AAAI Workshop on Explainable Agency in AI 2021
Torgyn Shaikhina, **UB**, Roxanne Zhang, Kostas Georgatzis, Alice Xiang, Adrian Weller

24. **Counterfactual Accuracies for Alternative Models**
ICLR Workshop on Machine Learning in Real Life 2020
UB, Adrian Weller, Muhammad Bilal Zafar, Krishna Gummadi

25. **You Shouldn't Trust Me: Learning Models Which Conceal Unfairness From Multiple Explanation Methods**
European Conference on Artificial Intelligence (ECAI) 2020
Botty Dimanov, **UB**, Mateja Jamnik, Adrian Weller

26. **Evaluating and Aggregating Feature-based Explanations**
International Joint Conference on Artificial Intelligence (IJCAI) 2020
UB, Adrian Weller, José M.F. Moura

27. **On Network Science and Mutual Information for Explaining Deep Neural Networks**
IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP) 2020
Brian Davis*, **UB***, Kartikeya Bhardwaj*, Radu Marculescu, José M.F. Moura

Appendix B

DIVINE Supplementary Material

B.1 Practitioner Guide

Throughout the paper, we use the word "practitioners" to refer to data scientists who can use DIVINE in practical ML settings where explainability is valued or those who hope to refine their models by better understanding their training data. In this guide, we explain how practitioners can select the parameters used in DIVINE: importance measure $\mathcal{I}$, evaluation function $\mathcal{L}$, diversity function $\mathcal{R}(\mathcal{S})$, influence-diversity tradeoff γ, and the number of DIVINE points, m. Our code can be extended to support additional influence measures $\mathcal{I}$, submodular diversity functions $\mathcal{R}$, evaluation functions $\mathcal{L}$, and γ selection strategies.

Selecting Parameters

Influence Measure $\mathcal{I}$. Within our work, the influence measure $\mathcal{I}$ assigns importance to individual data points and to groups of data points. We aim to find an importance score I_i for the i-th training point. Under our additivity assumption per Koh et al. [2019], we let the importance of a set of points $\mathcal{S}$ be $\mathcal{I}(\mathcal{S}) = \sum_{x_i \in \mathcal{S}} I_i$. We can obtain importance scores from various methods, like influence functions [Koh and Liang, 2017], Data Shapley [Ghorbani and Zou, 2019], counterfactual prediction [Bhatt et al., 2020c], or leave-one-out (LOO) [Hastie et al., 2009]. In Table B.1, we compare various methods for finding valuable training points with respect to a model θ or a diversity function $\mathcal{R}$.

One would select influence functions if they prefer a fast computation (assuming the Hessian computation is done once for a reasonably sized set of parameters). One could use Data Shapley if they want importance scores that adhere to the game-theoretic guarantees of the Shapley value. LOO scores are also possible to compute but note these may be computationally expensive as retraining is required for every point that is dropped. We show how all methods

Method	θ	$\mathcal{R}(\mathcal{S})$	$\mathcal{L}(\theta)$
Prototypes/Criticisms [Kim et al., 2016]	✗	✓	N/A
ProtoDash [Gurumoorthy et al., 2019]	✗	✓	N/A
Influential Points [Koh and Liang, 2017]	✓	✗	Loss
Representer Points [Yeh et al., 2018]	✓	✗	Loss
Seq. Bayesian Quadrature [Khanna et al., 2019]	✓	✗	Loss
Data Shapley [Ghorbani and Zou, 2019]	✓	✗	Accuracy and AUC
RelatIF [Barshan et al., 2020]	✓	✗	Loss
DIVINE (Ours)	✓	✓	Any $\mathcal{L}$

Table B.1 Practitioners can leverage DIVINE to value data points based on their contributions to model-specific evaluation metrics and then select a diverse, influential subset of points as a summary. We list other methods based on (1) their dependence on model parameters (e.g. prototypes are model-independent), (2) the diversity in the points to which they assign high importance, and (3) the metric with respect to which the data points are valued.

behave in comparison to each other on our synthetic data in Appendix B.4. Our package supports influence functions, Data Shapley, LOO, and counterfactual prediction.

Moreover, when applying an importance scoring measure, a practitioner may have to select an evaluation function $\mathcal{L}$ with respect to which one wants to value data points. While we primarily let $\mathcal{L}$ be $\mathcal{L}_{\text{loss}}$ or $\mathcal{L}_{\text{unf}}$ in our experiments, we report additional evaluation functions $\mathcal{L}$ that can be used to score data points in Table B.2. For every evaluation function, we write the function such that a large $\mathcal{L}(\theta)$ is undesirable: a negative importance score (if our score I is taken to be the difference between $\mathcal{L}$ evaluated at the new and old parameters – trained with and without the point respectively) implies that a point is harmful to $\mathcal{L}$. Moreover, we can also consider functions that do not simply take a difference between the evaluation function's value at the new and old parameters. These functions might be independent of $\mathcal{L}$ (e.g. $||\theta_{\text{new}} - \theta_{\text{old}}||_p$). However, from a model debugging perspective, these definitions may not be relevant when the dimensionality of θ is even moderately large. We can also define importance scores that jointly consider unfairness and loss: $\mathcal{L}(\theta) = \mathcal{L}_{\text{train}}(\theta) + \mathcal{L}_{\text{unf}}(\theta)$ would tell us how important a point is for unfairness and loss. The evaluation functions in Table B.2 represent other group fairness metrics. We hope that future work considers finding DIVINE points with respect to robustness and privacy, which requires devising a new $\mathcal{L}$.

Practitioners should select the $\mathcal{L}$ that captures the property, for which they wish to test their model. If one wants to see the impact of data points on performance, $\mathcal{L}_{\text{loss}}$ would be a good option, but if the impact of data points on fairness is desired, $\mathcal{L}_{\text{unf}}$ would be better.

Metric	Evaluation Function ($\mathcal{L}$)
Loss (e.g. wrt $\mathcal{D}_{\text{train}}$)	$\mathcal{L}_{\text{train}}(\theta) = \sum_i^n \ell(x_i, y_i; \theta)$
Equal Accuracy [Berk et al., 2017]	$\mathcal{L}_{\text{ea}}(\theta) = \mathcal{L}_{\text{unf}}(\theta) = \sum_{j\in\{-1,1\}} \lvert P_{j,a,j} - P_{j,b,j}\rvert$
Equal Opportunity [Hardt et al., 2016]	$\mathcal{L}_{\text{eq}}(\theta) = P_{1,a,1} - P_{1,b,1}$
Equalized Odds [Hardt et al., 2016]	$\mathcal{L}_{\text{eo}}(\theta) = \sum_{j\in\{-1,1\}} \lvert P_{1,a,j} - P_{1,b,j}\rvert$

Table B.2 Various candidate evaluation functions $\mathcal{L}$; note $P_{j,a,j} = P(\hat{y} = j | A = a, y = j)$.

Diversity Function $\mathcal{R}$. One main ingredient of DIVINE is a submodular function $\mathcal{R}$. This allows us to perform greedy selection when adding points to our DIVINE set. While other non-submodular diversity functions $\mathcal{R}$ are possible, they would benefit from the ease of using greedy selection. Future work might benefit from more clever set selection methodologies. Throughout the paper, we mostly let our diversity function be the sum-redundancy function, $\mathcal{R}_{\text{SR}}$. This function ensures that our selected points differ from each other, i.e., have low similarity [Libbrecht et al., 2018; Lin and Bilmes, 2011]. In Appendix B.4, we demonstrate how the facility location function, $\mathcal{R}_{\text{FL}}$, and maximum mean discrepancy, $\mathcal{R}_{\text{MMD}}$, perform. While the equations for each $\mathcal{R}$ appear in Section 4.1.2, $\mathcal{R}_{\text{FL}}$ is the submodular facility location function [Krause and Golovin, 2014] from the sensor-placement literature, selects points that minimize are similar to the most number of points in the entire dataset and does not explicitly prohibit redundancy between the points selected. When γ is large and $\mathcal{R}_{\text{MMD}}$ is maximized, the prototypes of Kim et al. [2016] are recovered. If a practitioner does not mind some potential redundancy in the points selected and wants a set of points representative of the dataset, then $\mathcal{R}_{\text{FL}}$ may be suitable. On the other hand, $\mathcal{R}_{\text{MMD}}$ from Kim et al. [2016] selects a set of points that summarize the entire dataset and penalizes similarity between the chosen points. When γ is large and $\mathcal{R}_{\text{MMD}}$ is maximized, the prototypes of Kim et al. [2016] are recovered. If a practitioner wants representativeness without much redundancy, $\mathcal{R}_{\text{MMD}}$ might suffice. All three submodular diversity functions are implemented in our package.

Influence-Diversity Tradeoff γ. In Section 4.1.3, we introduce the influence-diversity tradeoff curve. This illustrates how γ controls how much influence to forgo in favor of diversity with respect to $\mathcal{R}$. While practitioners can select any γ along the curve. We suggest two ways to pick γ. A practitioner could specify the maximum % of influence they are willing to sacrifice. A practitioner could also find the γ that optimizes the average pairwise distance between the m selected points. By default, our package favors the latter. Practitioners can implement other γ selection strategies as they see fit.

Number of DIVINE points m. Selecting the number of DIVINE points to find and visualize will be use-case dependent. If the goal of finding DIVINE points is to display them as an explanation of model behavior, we suggest displaying at most 5 points, which aligns with the number of cognitive chunks a user can handle at any given moment [Doshi-Velez and Kim, 2017]. We suggest curating the size of the explanation to the needs of the stakeholders who will be analyzing the DIVINE points [Bhatt et al., 2020b]. When selecting points to remove, a practitioner may consider checking our additivity assumption, i.e., see if removing a large number of low-value points at once does not affect other metrics of interest. We discuss how one would go about doing this analysis in Appendix B.2. We find that, for small m, our additivity assumption is valid. Therefore, one might consider recalculating importance scores after removing a few batches of m. We hope future work develops additional tools for choosing m.

Using Our Code

Our code is publicly available at https://github.com/umangsbhatt/divine-release, with a comprehensive README describing our implementation of DIVINE. We intend our code to be usable out of the box. We describe typical use cases for our DIVINE codebase in the README. Practitioners can use our codebase by importing necessary files (as shown in **tutorial.py**) into their own code. All use cases are runnable from **tutorial.py**. More details are available in our README.

All use-cases utilize the `DivineRunner`, implemented in **divine_runner.py**. In the README in, we detail the `ScorerData` object (which holds data throughout the DIVINE codebase) and how to use it. Additionally, all use cases use configuration files. Examples are found in **sample_configs/** and the tutorial code shows how they are used. Note that instead of using the logistic regression classifier we support in the current codebase, a practitioner can also supply their own importance scores to the DIVINE framework; however, this would need to be accompanied by the predicted labels from their trained model and the training data itself. With these three ingredients, a practitioner can run DIVINE in a lightweight fashion on top of their existing workflows. For our experiments with image datasets, we opt for this route: train and run influence outside the DIVINE framework but then leverage our DIVINE codebase to identify DIVINE points.

If a practitioner decides to use our codebase for tabular datasets, we support them with training a logistic regression classifier for their dataset, finding influential points with respect to a specified influence measure $\mathcal{I}$ and a specified evaluation function $\mathcal{L}$, finding DIVINE points after further specifying a diversity function $\mathcal{R}$, tuning γ for the desired set of DIVINE points, visualizing DIVINE points either on scatter plots or parallel coordinate plots, refining their

classifiers by removing low importance training points. We think our codebase will make it easy for practitioners to use DIVINE end-to-end or to integrate DIVINE into their existing codebase (e.g., using their own influence calculation methods or using their own visualization methods). We now discuss some useful elements of the DIVINE codebase.

Gamma Tuning. We can tune the influence-diversity tradeoff parameter, gamma, with two methods implemented in **utils/gamma_tuning.py**. One approach is to maximize the average pairwise distance in the selected set of DIVINE points. The method `get_best_apd_gamma` runs a gamma sweep over the specified `ScorerData data_object` for the specified `set_size`, also using a `DivineRunner`, and returns the optimal gamma value.

```
def example_apd_gamma_tuning(config_yml):
    config = DivineConfig(config_yml)
    runner = DivineRunner(config)
    sample_data = runner.load_sample_data(seed=1)

    set_size = 5
    best_gamma = get_best_apd_gamma(set_size, sample_data, runner)
```

Another approach is for the practitioner to specify the amount of influence they are willing to sacrifice to achieve improved diversity. The method `get_influence_sacrifice_gamma` runs a gamma sweep over the specified `ScorerData data_object` for the specified `set_size` and `p_inf` (influence sacrifice percentage value) and returns the optimal gamma value.

```
def example_sacrifice_influence_gamma_tuning(config_yml):
    config = DivineConfig(config_yml)
    runner = DivineRunner(config)
    sample_data = runner.load_sample_data(seed=1)

    # This value is set at -0.1, meaning 10% sacrifice of influence
    inf_sacrifice_percent = -0.1
    set_size = 5
    best_gamma = get_influence_sacrifice_gamma(set_size,
     ↪  inf_sacrifice_percent, sample_data, runner)
```

We can also generate gamma tradeoff plots (as seen in Figure 4.1.3 of the paper). These show tradeoff curves at different set sizes, with market points where 10% influence is sacrificed and the average pairwise difference is maximized. These curves can aid a practitioner in deciding how they want to tune gamma, or they can select gamma directly from the curves. The method `gamma_tradeoff_plot` creates such a tradeoff plot with the specified `ScorerData data_object`.

```
def example_gamma_tradeoff_plot(config_yml):
    config = DivineConfig(config_yml)
    runner = DivineRunner(config)
    sample_data = runner.load_sample_data(seed=1)
    output_filename = "gamma_tradeoff_plot.png"
    gamma_tradeoff_plot(sample_data, runner, output_filename)
```

Selecting DIVINE Points. We can select DIVINE points easily, using the `DivineRunner`. Here, we load sample data as specified in a config file (seen at **sample_configs/divine_config.yml**) and then the `DivineRunner` is able to return the indices of the selected data points. The selected indices apply to the training data set, accessible from the `ScorerData` object at: `sample_data.x_train`, `sample_data.y_train`, `sample_data.x_control_train`. Note that a practitioner could provide their own influence scores to the DIVINE algorithm for selection.

```
def example_select_divine_points(config_file):
    config = DivineConfig(config_yml)
    runner = DivineRunner(config)

    # Load sample data into ScorerData object that is used throughout
     ↪  the codebase
    sample_data = runner.load_sample_data(seed=10)
    # Influence scores - note that a practitioner could provide their
     ↪  own influence scores here
    influence_scores = runner.get_influence_scores(sample_data)
    # Select the top 10 DIVINE points from the sample data
    selected_indices = runner.select_divine_points(sample_data,
     ↪  influence_scores, num_selected=10)
```

```
    return selected_indices
```

Removing DIVINE Points. We can remove DIVINE points as well, using the `DivineRunner`. Here, we load sample data as specified in the config file at **sample_configs/divine_config.yml**. Then, we remove the top 10 most harmful points as determined by the DIVINE algorithm, according to the parameters set in the configuration file. The result is a new `ScorerData` object with the harmful points removed from the training data.

```
def example_remove_divine_points(config_file):
    config = DivineConfig(config_yml)
    runner = DivineRunner(config)

    # Load sample data into ScorerData object that is used throughout
     ↪  the codebase
    sample_data = runner.load_sample_data(seed=10)
    # Remove the 10 most harmful points (as determined by DIVINE) from
     ↪  the sample data
    modified_data = runner.remove_harmful_points(sample_data,
     ↪  num_to_remove=10)
    return modified_data
```

Visualizations. We provide methods to visualize data in the context of DIVINE-selected points. A parallel coordinates plot can be used to visualize a smaller set of points. The method `parallel_coords_plots` defined in **utils/parcoords.py** can be used to generate a parallel coordinates plot. The method `get_train_data_frame` is used to get data into the correct format to use for the plots. Figure B.1 shows an example parallel coordinates plot. Parallel coordinate plots can be intuitive ways to display how well

```
def example_parcoords_plot(config_yml):
    '''
    Create a parallel coordinates plot with select points. Useful for
 ↪  visualising the diversity of your dataset.
    '''
```

```
    num_pts = 10
    config = DivineConfig(config_yml)
    runner = DivineRunner(config)
    sample_data, sample_frame = runner.load_sample_data_frame(seed=10)

    weights, train_loss = runner.train_simple_model(sample_data)
    train_predicted_labels = runner.get_train_predictions(weights,
     ↪  sample_data)

    influence_scores = runner.get_influence_scores(sample_data)
    selected_indices = runner.select_divine_points(sample_data,
     ↪  influence_scores)
    # Transforms the data into a format useable for the parallel
     ↪  coordinates plot
    train_frame = get_train_data_frame(config.dataset_type,
     ↪  sample_data, sample_frame, selected_indices, influence_scores,
     ↪  train_predicted_labels)

    output_filename = "parcoords_plot.png"
    parallel_coords_plot(config.dataset_type, train_frame, num_pts,
     ↪  output_filename)
```

A scatter plot can be helpful to visualize the data as a whole and see where DIVINE points lie within it. The method `scatterplot`, defined in **utils/scatterplot.py**, can be used to plot the entire dataset with selected points shown in a different shape and color. If the number of data features is 4 or greater, TSNE is used for visualization. The `weights` parameter shows a decision boundary generated by a simple trained classifier.

```
def example_scatterplot(config_yml):
    config = DivineConfig(config_yml)
    runner = DivineRunner(config)
    sample_data = runner.load_sample_data(seed=1)
    influence_scores = runner.get_influence_scores(sample_data)
    selected_indices = runner.select_divine_points(sample_data,
     ↪  influence_scores, num_selected=5)
    weights, _ = runner.train_simple_model(sample_data)
```

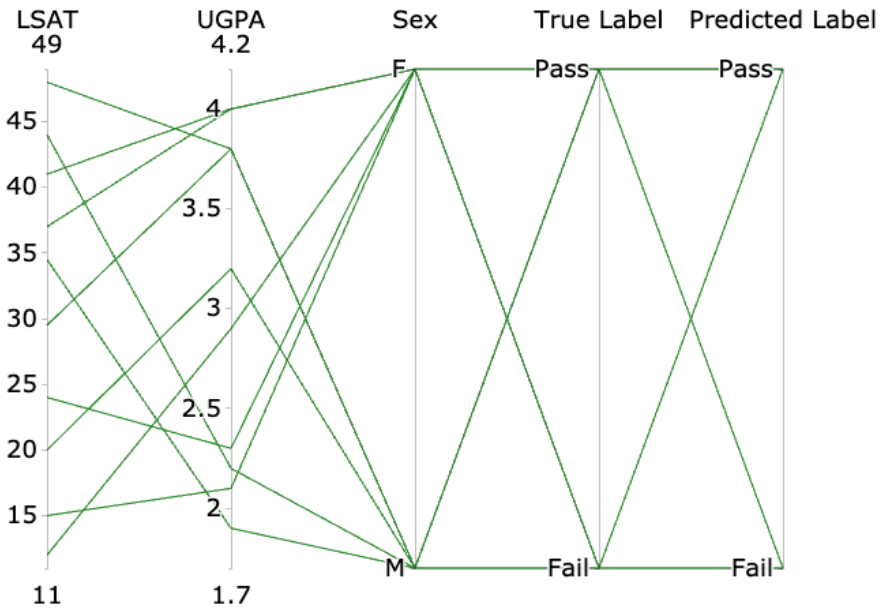


Fig. B.1 Parallel coordinates plot for 10 LSAT data points

```
output_filename = "scatterplot.png"
scatterplot(sample_data, selected_indices, weights, output_filename)
```

B.2 Experimental Setup

Dataset Metadata

We employ 6 datasets in our experiments, 4 tabular and 2 image. All are publicly available, with details given in Table B.3. For all datasets, we use a 70% train, 20% validation, and 10% test split. We use the LSAT loading script from Cole and Williamson [2019]'s github page. The raw data can be downloaded from https://raw.githubusercontent.com/throwaway20190523/MonotonicFairness/master/data/law_school_cf_test.csv and https://raw.githubusercontent.com/throwaway20190523/MonotonicFairness/master/data/law_school_cf_train.csv. We let "sex" be our protected attribute and drop "race" from the dataset when running our experiments. Features used are undergraduate grade point average, LSAT score, and sex. The predicted label is first-year law school performance.

Table B.3 Summary of datasets used in our experiments. (*)We use a 7 feature version of COMPAS; however, other versions exist.

Name	Targets	Input Type	# Features	# Total Samples
LSAT	Continuous	Continuous & Categorical	3	21791
COMPAS	Binary	Continuous & Categorical	7*	5278
Adult	Binary	Continuous & Categorical	11	45222
Bank	Binary	Continuous & Categorical	21	41188
MNIST	Categorical	Image (greyscale)	28×28	70000
FashionMNIST	Categorical	Image (greyscale)	28×28	70000

For the COMPAS criminal recidivism prediction dataset, we use a modified version of Zafar et al. [2017b]'s loading and pre-processing script. It can be found at https://github.com/mbilalzafar/fair-classification/blob/master/disparate_mistreatment/propublica_compas_data_demo/load_compas_data.py. We add an additional feature: "days served" which we compute as the difference, measured in days, between the "c_jail_in" and "c_jail_out" variables. The raw data is found at https://github.com/propublica/compas-analysis/blob/master/compas-scores-two-years.csv. We let "race" be our protected attribute. Other features used are age, sex, charge degree (felony or misdemeanor), and priors count. The predicted label is recidivism within 2 years.

The adult dataset [Dua and Graff, 2017] can be obtained from and is described in detail at https://archive.ics.uci.edu/ml/datasets/adult. The features we used are age, work class, education, education number, marital status, occupation, relationship, capital gain, capital loss, hours per week, and native country. We let "sex" be our protected attribute. The predicted label is whether the person makes more than 50K a year.

The bank marketing dataset [Dua and Graff, 2017] can be obtained from and is described in detail at https://archive.ics.uci.edu/ml/datasets/Bank+Marketing. The features we used are described in detail at the link. We let "age" be our protected attribute. The predicted label is whether a client will subscribe to a term deposit. The MNIST handwritten digit image dataset [LeCun, 1998] can be obtained from http://yann.lecun.com/exdb/mnist/. The FashionMNIST image dataset [Xiao et al., 2017] can be obtained from https://github.com/zalandoresearch/fashion-mnist.

Models

In Section 4.1.3, we primarily use logistic regression for our tabular data experiments. For all tabular datasets, we append an intercept to the input features before learning our parameters, $\theta \in \mathbb{R}^d$: this is customary in such settings (i.e. Zafar et al. [2017b] has a similar setup). We learn classifier's parameters using **scipy.optimize**, using the SLSQP (Sequential Least SQuares

Programming) solver. For image datasets, we use **tensorflow** to learn a three-layered multilayer perceptron (for MNIST) and a three-layered convolutional neural network (for FashionMNIST).

B.3 DIVINE for Fairness

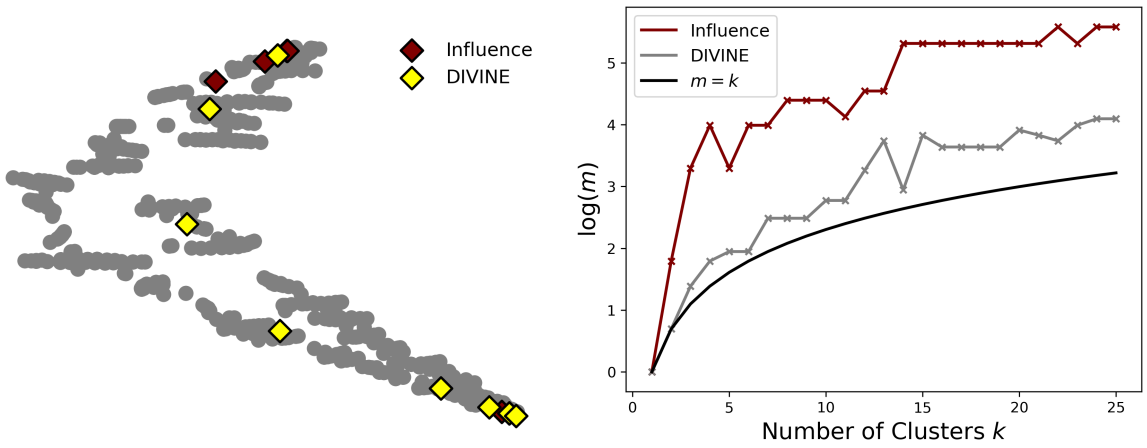
Existing approaches to valuing data points usually take $\mathcal{L}$ to be a model's training loss [Barshan et al., 2020; Koh and Liang, 2017], accuracy, or AUC [Ghorbani and Zou, 2019]. We let $\mathcal{L}$ be any group fairness criteria. This allows us to value training data based on their helpfulness or harmfulness for achieving various notions of fairness. This can be seen as the inverse of fair data augmentation or fair active learning [Anahideh et al., 2022], approaches that identify additional data to collect. Datta et al. [2016] use quantitative input influence to evaluate the effect of removing a feature on a model's fairness. Lundberg [2020] use Shapley values to decompose demographic parity in terms of feature contributions. Our work can be seen as an influential point analog. While they extend feature importance to identify features that contribute to unfairness, we extend training point importance to identify which points contribute to unfairness.

Studying the diverse modes of data that contribute to canonical model behavior, herein with respect to unfairness, can help practitioners analyze their models. In Table B.4 for LSAT, we qualitatively compare the top-7 diverse unfairness-inducing points found by Equation 4.1 (left) and the top-7 most influential points (right). We use IF with respect to $\mathcal{L}_{\text{unf}}$ for $\mathcal{I}$ and $\mathcal{R}_{\text{SR}}$. A is the sensitive attribute: male (M) or female (F). FYA is first-year average, binarized as pass/fail. The maximum LSAT score is 48, and maximum GPA is 4.0. Notice the lack of diversity in the points on the right: most points are males with poor LSAT test scores and low GPA grades yet pass their course. DIVINE points include not only points with poor LSAT scores and low GPAs that pass but also points with high LSAT test scores and high GPAs (which are mostly female) yet fail. By trading off influence and diversity (left), we identify an unfairness-inducing "mode" of the dataset—female participants with high LSAT scores and GPAs, but fail—which is not identified by influence alone (right). We visualize this diversity in low dimensions via TSNE [Van der Maaten and Hinton, 2008] in Figure B.2a: notice how all the IF points are clustered. This ability to detect modes missed by the top IF points highlights the utility of DIVINE. With DIVINE, we find multiple modes that lead to unfairness in our model. Quantitatively, we show that DIVINE does a better job of covering clusters of data in input space. In Figure B.2b, we cluster the full dataset using KMeans into k clusters and then find the m such that one point from each cluster is in the top-m points of IF and DIVINE. The black line is a lower-bound, $m = k$. DIVINE points requires a smaller m than IF to represent all k clusters of the data in the top points. The redundancy of the top IF points make it difficult to

get a holistic picture of model behavior, as top IF points lie in clusters where we already have important points identified. *Unlike IF points, DIVINE points allow us to identify various modes of data that contribute to a model's unfairness.*

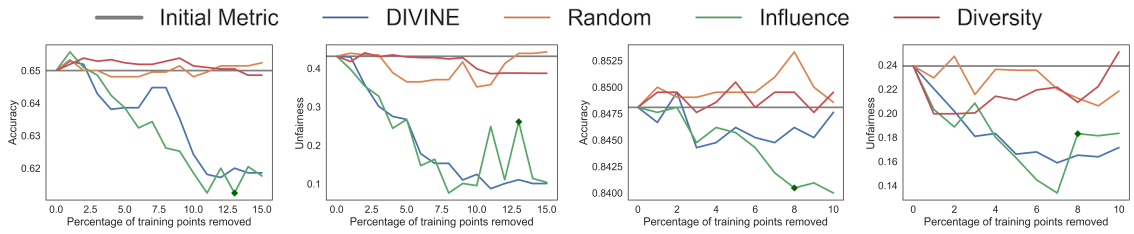
$\mathcal{I}(\mathcal{S})+\gamma\mathcal{R}_{\text{SR}}(\mathcal{S})$				$\mathcal{I}(\mathcal{S})$			
LSAT	GPA	*A*	FYA	LSAT	GPA	*A*	FYA
14	2.9	F	Pass	20	2.8	M	Pass
25	3.6	M	Pass	25	3.6	M	Pass
20	2.8	M	Pass	20	3.2	M	Pass
41	3.8	F	Fail	14	2.9	F	Pass
33	4.0	F	Fail	21	3.1	M	Pass
45	3.9	F	Fail	23	2.8	M	Pass
29	3.7	F	Fail	22	2.9	M	Pass

Table B.4 LSAT Unfairness Inducing Points



(a) TSNE (b) LSAT Clustering

Fig. B.2 DIVINE and dataset redundancy



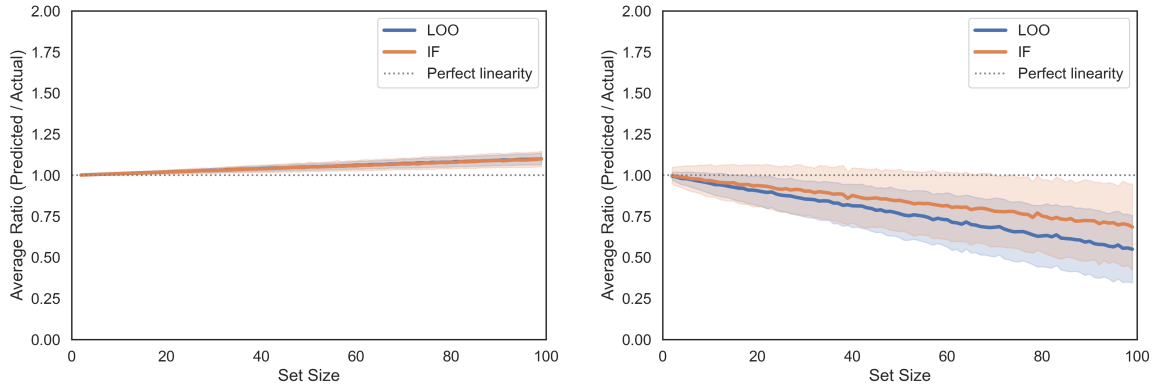
(a) COMPAS Accuracy (b) COMPAS $\mathcal{L}_{\text{unf}}$ (c) Adult Accuracy (d) Adult $\mathcal{L}_{\text{unf}}$

Fig. B.3 Training data performance after removal of unfairness-inducing points. Line color denotes method. Blue selects DIVINE points (Equation 4.1) with IF, $\mathcal{R}_{\text{SR}}$, and γ *tuned* via pairwise distance. Orange denotes randomly selected points. Green uses IF to select points ($\gamma = 0$). Red chooses points by maximizing $\mathcal{R}_{\text{SR}}$ alone. Grey indicates original performance. Diamond indicates the point all unfairness-inducing points have been removed, after which we expect unfairness to increase, as we remove low importance (but helpful) points. In IF and DIVINE, valuing data with respect to $\mathcal{L}_{\text{unf}}$ identifies harmful data to remove; after removal, fairness outcomes improve greatly, but accuracy drops slightly.

B.4 Additional Experiments

Additivity implies Modularity

We briefly comment on the modularity of $\mathcal{I}(\mathcal{S})$. If $I_i > 0$ holds $\forall i$, then $\mathcal{I}(\mathcal{S})$ is monotone. To show the modularity of $\mathcal{I}(\mathcal{S})$, it suffices to show each importance measure with a selected $\mathcal{L}$, is additive, which implies importance scores are linear. When $\mathcal{L}$ is loss, Koh et al. [2019] find that influence functions are approximately linear, $I_{ij} \approx I_i + I_j$. Furthermore, irrespective of $\mathcal{L}$, Shapley values, by construct, satisfy linearity: see [Ghorbani and Zou, 2019; Shapley, 1953] for a thorough treatment. While we know that $\mathcal{I}(\mathcal{S})$ will be modular when $\mathcal{L}$ is a function of loss, we consider linearity using influence functions with $\mathcal{L}_{\text{unf}}$. In Figure B.4, we plot how a linearity approximation of importance (calculated using $\mathcal{L}_{\text{unf}}$) performs as we increase the number of points we remove from the dataset. The average difference between the predicted importance score $I_i + I_j$, and the actual importance score $I_{\{i,j\}}$, over 1000 samples of m sized sets. On LSAT, shown in Figure B.4a, all three scoring maintain linearity as set sizes increases, implying that $\mathcal{I}(\mathcal{S})$ is modular. Linearity is also satisfied on Adult (Figure B.4b) with LOO scores. For IF, the importance score is linear for small values of m. As m increases, our approximation no longer maintains linearity. If we keep m relatively small, we can use our linearity approximation, as we desire simple explanations with few cognitive chunks [Doshi-Velez and Kim, 2017]. However, as we can assume additivity with larger set sizes when we use LOO, we expect it to perform better overall in identifying large sets of important points on high-dimensional data. Practitioners might find using these graphs when deciding how large to make m; if m is too large, then DIVINE points for unfairness might not be trustworthy.

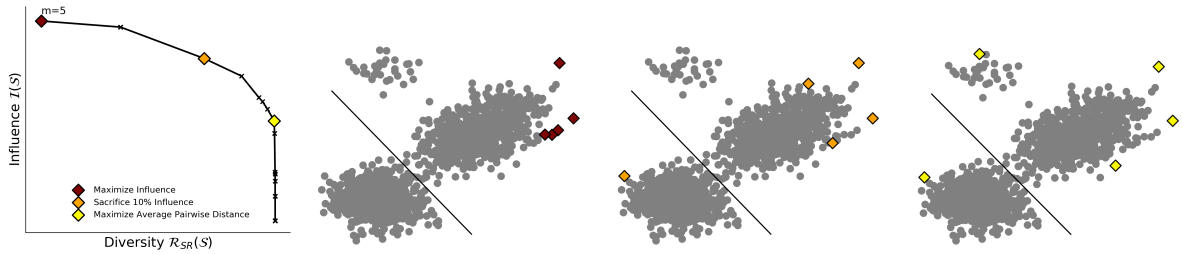


(a) LSAT (b) COMPAS

Fig. B.4 We show $\frac{I_i + I_j}{I_{\{i,j\}}}$ when $\mathcal{L}$ is equal accuracy.

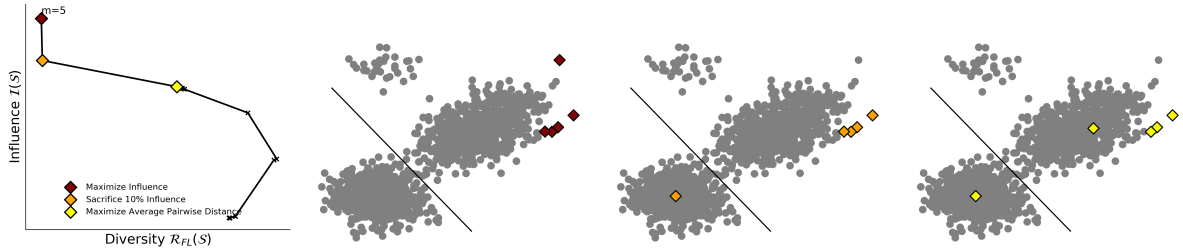
Results

In Figure B.8, we show how varying m affects our trade-off curves for various $\mathcal{R}$.



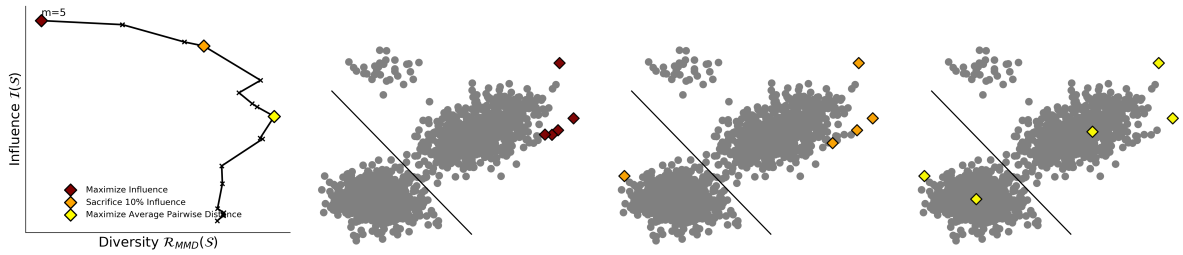
(a) $\mathcal{I}$-$\mathcal{R}$ Tradeoff (b) DIVINE $\gamma = 0$ (c) DIVINE $\gamma = 161$ (d) DIVINE $\gamma = 434$

Fig. B.5 $\mathcal{R}_{\text{SR}}$ – note this figure is the same as Figure 4.2 from the main text.



(a) $\mathcal{I}$-$\mathcal{R}$ Tradeoff (b) DIVINE $\gamma = 0$ (c) DIVINE $\gamma = 0.526$ (d) DIVINE $\gamma = 0.628$

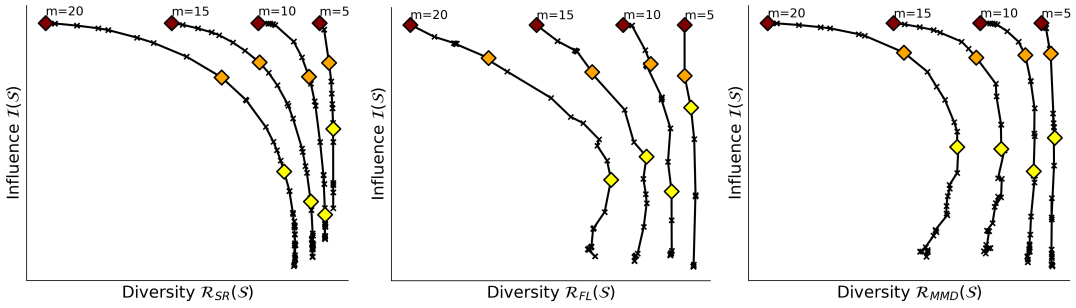
Fig. B.6 $\mathcal{R}_{\text{FL}}$. Some points are representative (near cluster center), but others are redundant (top right).



(a) $\mathcal{I}$-$\mathcal{R}$ Tradeoff (b) DIVINE $\gamma = 0$ (c) DIVINE $\gamma = 201$ (d) DIVINE $\gamma = 614$

Fig. B.7 $\mathcal{R}_{\text{MMD}}$. Our selected points are representative, though the top cluster is missed. There are no redundant points, in contrast to points selected with $\mathcal{R}_{\text{FL}}$.

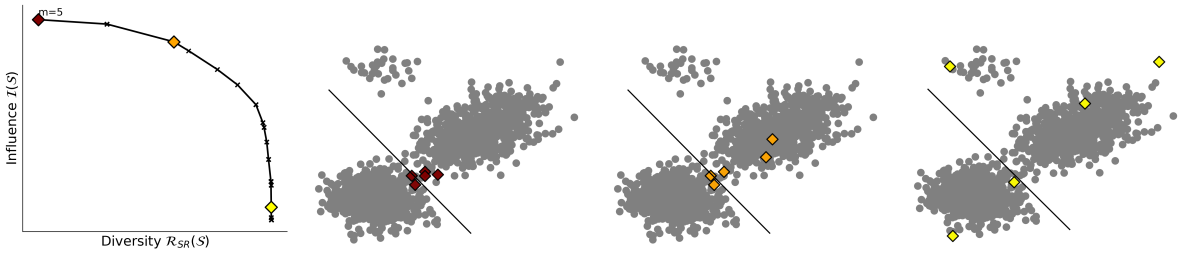
We next consider the effects of varying the underlying influence measure $\mathcal{I}$ but use $\mathcal{L}_{\text{loss}}$ for all experiments herein. In Figure B.9 and B.10, we show how DIVINE points are selected for DataShapley and Leave-one-out, respectively. Note that when $\gamma = 0$, the DIVINE points are simply the highest-scoring points from each method alone. Every method selects similar points (all high importance are located in a small cluster) when no diversity is considered. Then, we



(a) $\mathcal{R}_{\text{SR}}$ (b) $\mathcal{R}_{\text{FL}}$ (c) $\mathcal{R}_{\text{MMD}}$

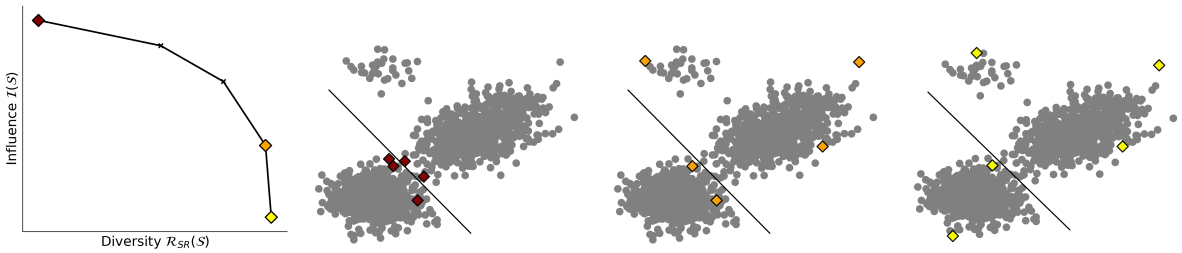
Fig. B.8 The DIVINE trade-off as a function of m for various $\mathcal{R}$ with our synthetic data.

trade-off $\mathcal{R}_{\text{SR}}$ with influence, and obtain similar trade-off plots. In Figure B.11, we find that as we increase m we get similar behavior for other influence measures that we obtained for influence functions before. For all influence measures, we use $\mathcal{L}_{\text{loss}}$ as our evaluation function.



(a) $\mathcal{I}$-$\mathcal{R}$ Tradeoff (b) DIVINE $\gamma = 0$ (c) DIVINE $\gamma = 0.041$ (d) DIVINE $\gamma = 12.42$

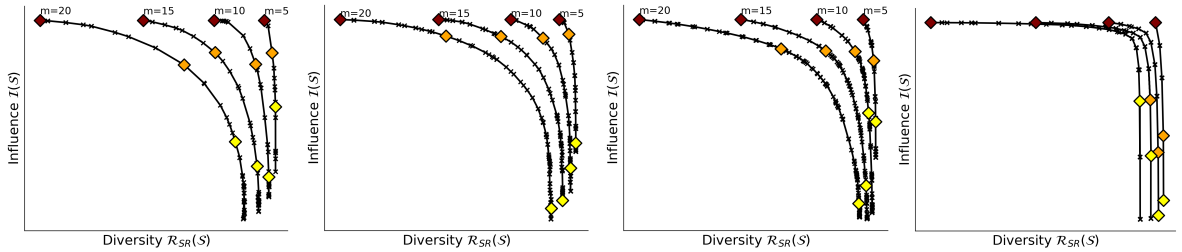
Fig. B.9 Results when using DIVINE atop Data Shapley.



(a) $\mathcal{I}$-$\mathcal{R}$ Tradeoff (b) DIVINE $\gamma = 0$ (c) DIVINE $\gamma = 0.97$ (d) DIVINE $\gamma = 7.97$

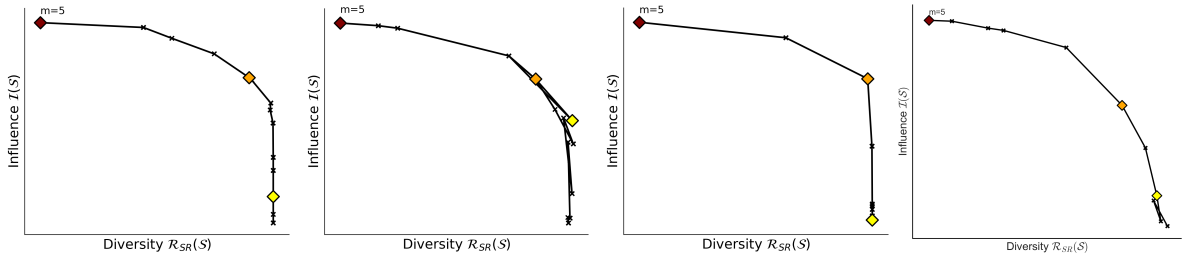
Fig. B.10 Results when using DIVINE atop leave-one-out.

We show how the trade-off curves look for various m from various datasets: LSAT, COMPAS, Adult, and FashionMNIST. We use IF as our influence measure, $\mathcal{R}_{\text{SR}}$ as our diversity function, and $\mathcal{L}_{\text{loss}}$ as our evaluation function. In Figure B.12, we report the trade-off curves for when $m = 5$.



(a) Influence Functions (b) DS (c) CFP (d) LOO

Fig. B.11 The DIVINE trade-off as a function of m for various $\mathcal{I}$ with our synthetic data. Note CFP is the method described in [Bhatt et al., 2020c].

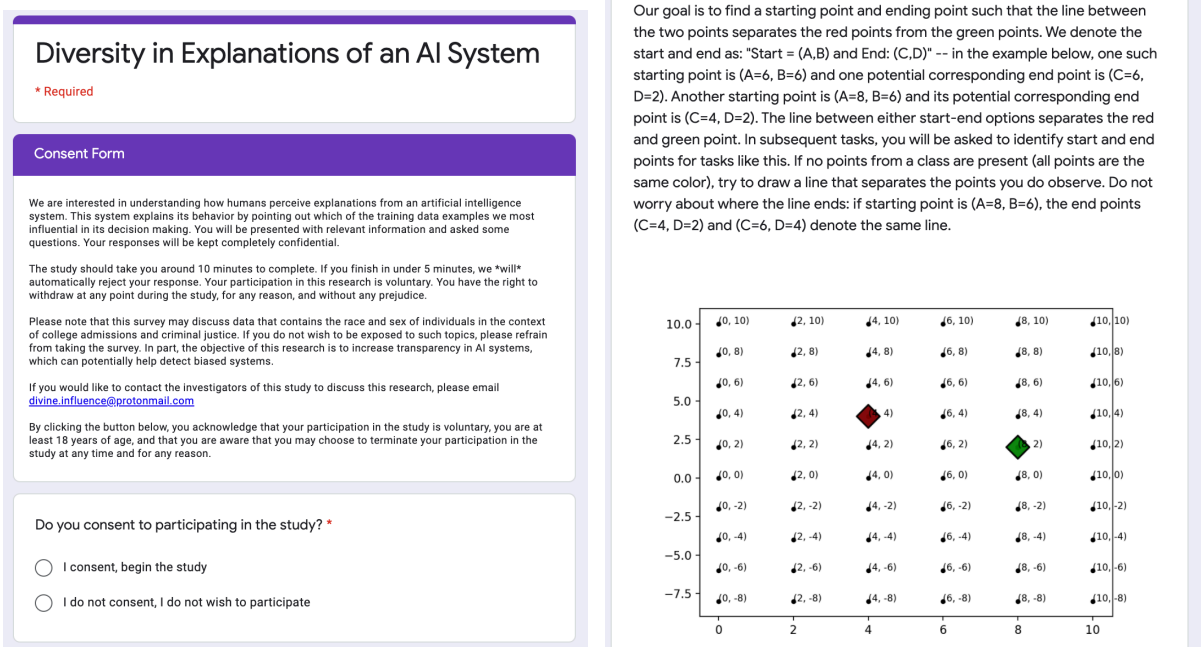


(a) LSAT (b) COMPAS (c) Adult (d) FashionMNIST

Fig. B.12 For four datasets and $m = 5$, we characterize the influence-diversity trade-off. In B.12d, we show a trade-off curve for FashionMNIST. Notice that it has a similar shape to 4.2a, even though the model type is a CNN, not LR, and the data type is image not tabular.

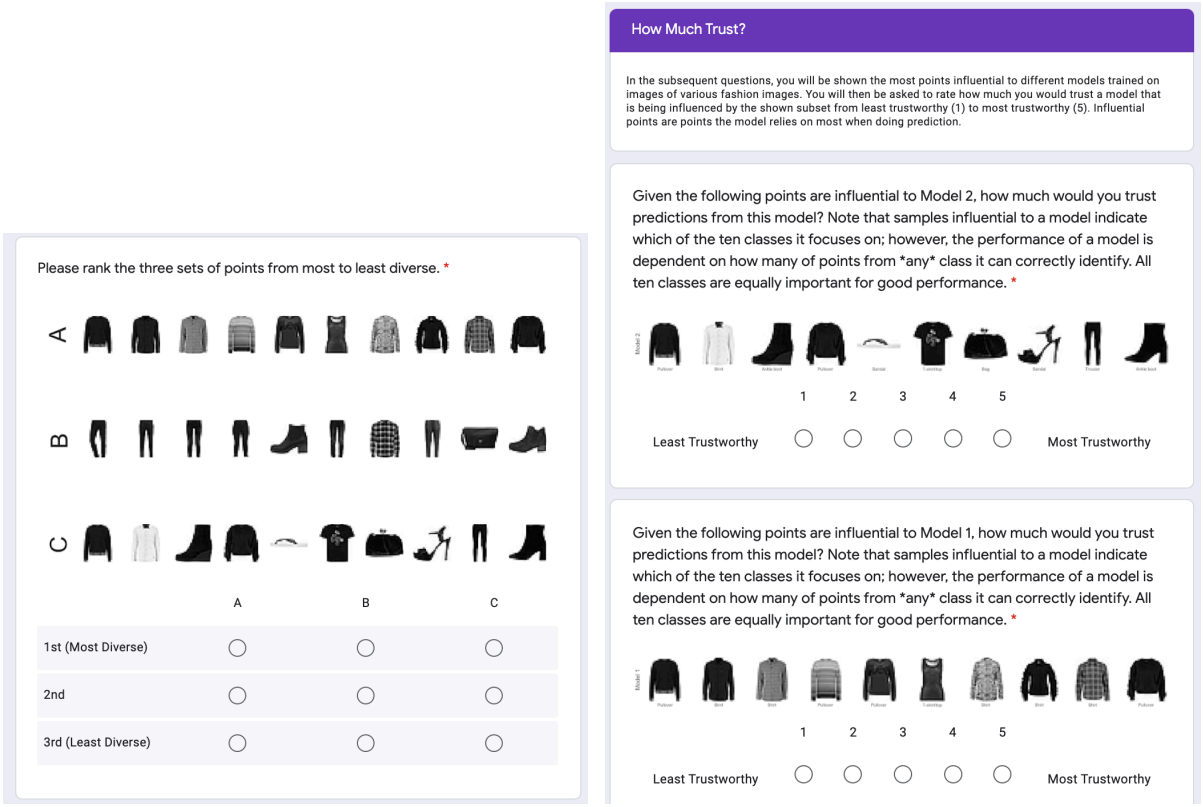
B.5 Explanatory Figures for User Studies

Below are some additional details and figures that elaborate upon the user studies. See user study prose for details.



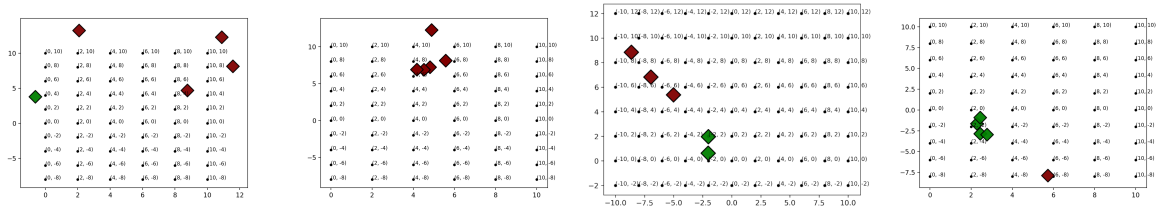
(a) Consent Form for the Survey (b) Attention Check for Simulatability

Fig. B.13 Setup of DIVINE user studies.



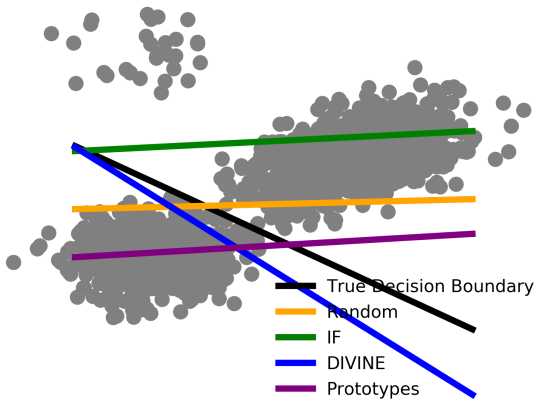
(a) Diversity in top-10 points (b) Trustworthiness Question

Fig. B.14 FashionMNIST User Study with DIVINE.

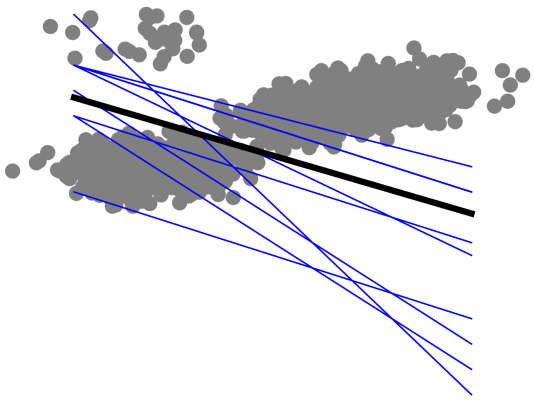


(a) Top DIVINE Points (b) Top IF Points (c) Prototypes (d) Random Points

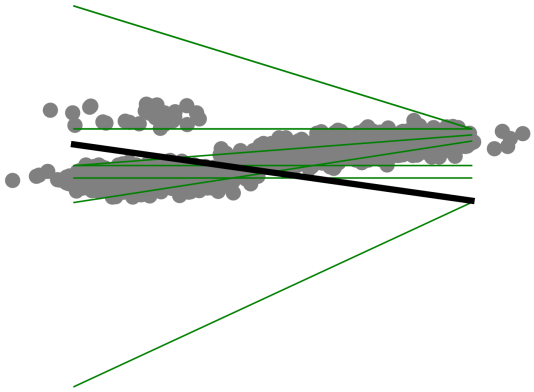
Fig. B.15 Simulatability task. Points are shown to users on the coordinate grid for $m = 5$.



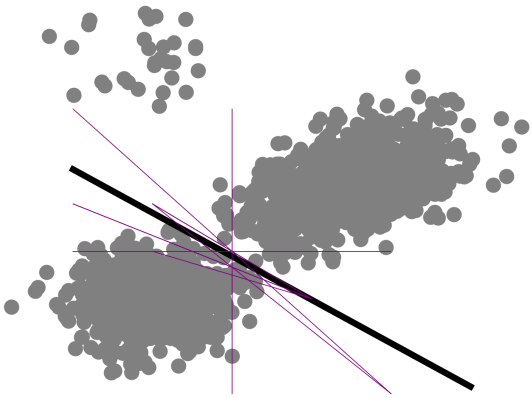
(a) Average



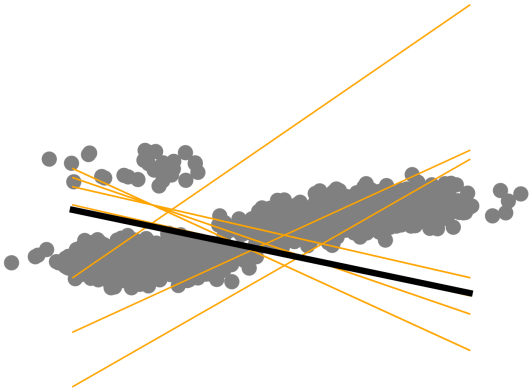
(b) DIVINE



(c) IF



(d) Prototypes



(e) Random

Fig. B.16 User-drawn decision boundaries when $m = 5$. Notice that the user-drawn decision boundaries are most similar to the true decision boundary upon seeing DIVINE points.

Appendix C

CLUE Supplementary Material

C.1 Experimental Details

Datasets

We employ 5 datasets in our experiments, 4 tabular and one composed of images. All of them are publicly available. Their details are given in Table C.1.

Table C.1 Summary of datasets used in our experiments. (*) We use a 7-feature version of COMPAS, however, other versions exist.

Name	Targets	Input Type	N. Inputs	N. Train	N. Test
LSAT	Continuous	Continuous & Categorical	4	17432	4358
COMPAS	Binary	Continuous & Categorical	7*	5554	618
Wine (red)	Continuous	Continuous	11	1438	160
Credit	Binary	Continuous & Categorical	24	27000	3000
MNIST	Categorical	Image (greyscale)	28×28	60000	10000

We use the LSAT loading script from Cole and Williamson [2019]'s github page. The raw data can be downloaded from (https://raw.githubusercontent.com/throwaway20190523/MonotonicFairness/master/data/law_school_cf_test.csv) and (https://raw.githubusercontent.com/throwaway20190523/MonotonicFairness/master/data/law_school_cf_train.csv). Unlike with our DIVINE experiments, we keep "race" as a feature in our CLUE experiments on LSAT. For the COMPAS criminal recidivism prediction dataset, we use a modified version of Zafar et al. [2017a]'s loading and pre-processing script. It can be found at (https://github.com/mbilalzafar/fair-classification/blob/master/disparate_mistreatment/propublica_compas_data_demo/load_compas_data.py). We add an additional feature: "days served" which we compute as the difference, measured in days, between the "c_jail_in" and "c_jail_out" variables. The raw data is found

at (https://github.com/propublica/compas-analysis/blob/master/compas-scores-two-years.csv). The red wine quality prediction dataset can be obtained from and is described in detail at (https://archive.ics.uci.edu/ml/datasets/wine+quality). The default of credit card clients dataset, which we refer to as "Credit" in this work, can be obtained from and is described in detail at (https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients). Note that this dataset is different from the also commonly used German credit dataset. The MNIST handwritten digit image dataset can be obtained from (http://yann.lecun.com/exdb/mnist/).

Computing Uncertainty Estimates

In this work, we consider NNs that parametrize two types of distributions over target variables: the categorical for classification problems and the Gaussian for regression. For classification, our networks output a probability vector with elements $f_\theta(x)$, corresponding to one of K classes. The likelihood function is $p(y|\, x,\theta) = Cat(y; f_\theta(x))$. Given a posterior distribution over weights $p(\theta|\mathcal{D})$, we use marginalization to translate uncertainty in θ into uncertainty in predictions. Unfortunately, this operation is intractable for BNNs. We resort to approximating the predictive posterior with M Monte Carlo samples:

$$\begin{aligned} p(y^*|\, x^*,\mathcal{D}) &= \mathbb{E}_{p(\theta|\mathcal{D})}[p(y^*|\, x^*,\theta)] \\ &\approx \frac{1}{M}\sum_{m=0}^{M} f(\, x^*,\theta); \quad \theta \sim p(\theta|\mathcal{D}). \end{aligned}$$

The resulting predictive distribution is categorical. We quantify its uncertainty using entropy:

$$H(y^*|\, x^*,\mathcal{D}) = \sum_{k=1}^{K} p(y^*{=}c_k|\, x^*,\mathcal{D}) \log p(y^*{=}c_k|\, x^*,\mathcal{D}).$$

This quantity contains aleatoric and epistemic components (H_a, H_e). The former is estimated as:

$$H_a = \mathbb{E}_{p(\theta|\mathcal{D})}[H(y^*|\, x^*,\theta)] \approx \frac{1}{M}\sum_{m}^{M} H(y^*|\, x^*,\theta); \quad \theta \sim p(\theta|\mathcal{D}).$$

The epistemic component can be obtained as the difference between the total and aleatoric entropies. This quantity is also known as the mutual information between y^* and θ:

$$H_e = I(y^*,\theta|\, x^*,\mathcal{D}) = H(y^*|\, x^*,\mathcal{D}) - \mathbb{E}_{p(\theta|\mathcal{D})}[H(y^*|\, x^*,\theta)].$$

For regression, we employ heteroscedastic likelihood functions. Their mean and variance are parametrized by our NN: $p(y^*|\, x^*, \theta) = \mathcal{N}(y; f_\mu(\, x^*, \theta), f_{\sigma^2}(\, x^*, \theta))$. Marginalizing over θ with MC induces a Gaussian mixture distribution over outputs. Its mean is obtained as:

$$\mu_a \approx \frac{1}{M} \sum_{m=0}^{M} f_\mu(\, x^*, \theta); \quad \theta \sim p(\theta|\mathcal{D}).$$

There is no closed-form expression for the entropy of this distribution. Instead, we use the variance of the GMM as an uncertainty metric. It also decomposes into aleatoric and epistemic components (σ_a^2, σ_e^2):

$$\sigma^2(y^*|\, x^*, \mathcal{D}) = \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}[\sigma^2(y^*|\, x^*, \theta)]}_{\sigma_a^2} + \underbrace{\sigma^2_{p(\theta|\mathcal{D})}[\mu(y^*|\, x, \theta)]}_{\sigma_e^2}.$$

These are also estimated with MC:

$$\sigma^2(y^*|\, x^*, \mathcal{D}) \approx \underbrace{\frac{1}{M}\sum_m^M \mu(y^*|\, x^*, \theta)^2 - (\frac{1}{M}\sum_m^M \mu(y^*|\, x^*, \theta))^2}_{\sigma_e^2} + \underbrace{\frac{1}{M}\sum_m^M \sigma^2(y^*|\, x^*, \theta)}_{\sigma_a^2};\ \theta \sim p(\theta|\mathcal{D}).$$

Here, σ_e^2 reflects model uncertainty - our lack of knowledge about θ - while σ_a^2 tells us about the irreducible uncertainty or noise in our training data.

C.2 Additional Examples

We exploit the non-convexity of CLUE's objective to generate diverse CLUEs. We initialize CLUE with $z_0 = \mu_\phi(z|x_0) + \varepsilon$, where $\varepsilon = \mathcal{N}(z; \mathbf{0}, \sigma_0 \mathbf{I})$, and run Algorithm 2 multiple times to obtain different CLUEs. We choose $\sigma_0 = 0.15$. In Figure C.2, we showcase different CLUEs for the same original MNIST inputs. Different counterfactuals represent digits of different classes. Despite this, all explanations resemble the original data points being explained. Being exposed to this multiplicity could potentially inform practitioners about similarities of an original input to multiple classes that lead their model to be uncertain.

Different initializations lead to CLUEs that explain away different amounts of uncertainty. In a few rare cases, CLUE fails: the algorithm does not produce a feature configuration that has significantly lower uncertainty than the original input. This is the case for the third CLUE in the bottom 2 rows of Figure C.2. We attribute this to a disadvantageous initialization of z.

In Figure C.1, we show multiple CLUEs for a single individual from the COMPAS dataset. In this case, uncertainty can be reduced by changing the individual's prior convictions and charge degree, or by changing their sex and age range. Making both sets of changes simultaneously also reduces uncertainty.

	Original
Age	Greater than 45
Race	African-American
Sex	Female
Current Charge	Felony
Reoffended Before	No
Prior Convictions	1
Days Served	0

	CLUE
Age	Greater than 45
Race	African-American
Sex	Female
Current Charge	Misdemeanour
Reoffended Before	No
Prior Convictions	0
Days Served	0

	CLUE
Age	25-45
Race	African-American
Sex	Male
Current Charge	Misdemeanour
Reoffended Before	No
Prior Convictions	0
Days Served	0

	CLUE
Age	25-45
Race	African-American
Sex	Male
Current Charge	Felony
Reoffended Before	No
Prior Convictions	1
Days Served	0

	CLUE
Age	25-45
Race	Asian
Sex	Female
Current Charge	Felony
Reoffended Before	No
Prior Convictions	1
Days Served	0

Fig. C.1 The leftmost entry is an uncertain COMPAS test sample. To its right are four candidate CLUEs. The first three successfully reduce uncertainty past our rejection threshold, while the rightmost does not.

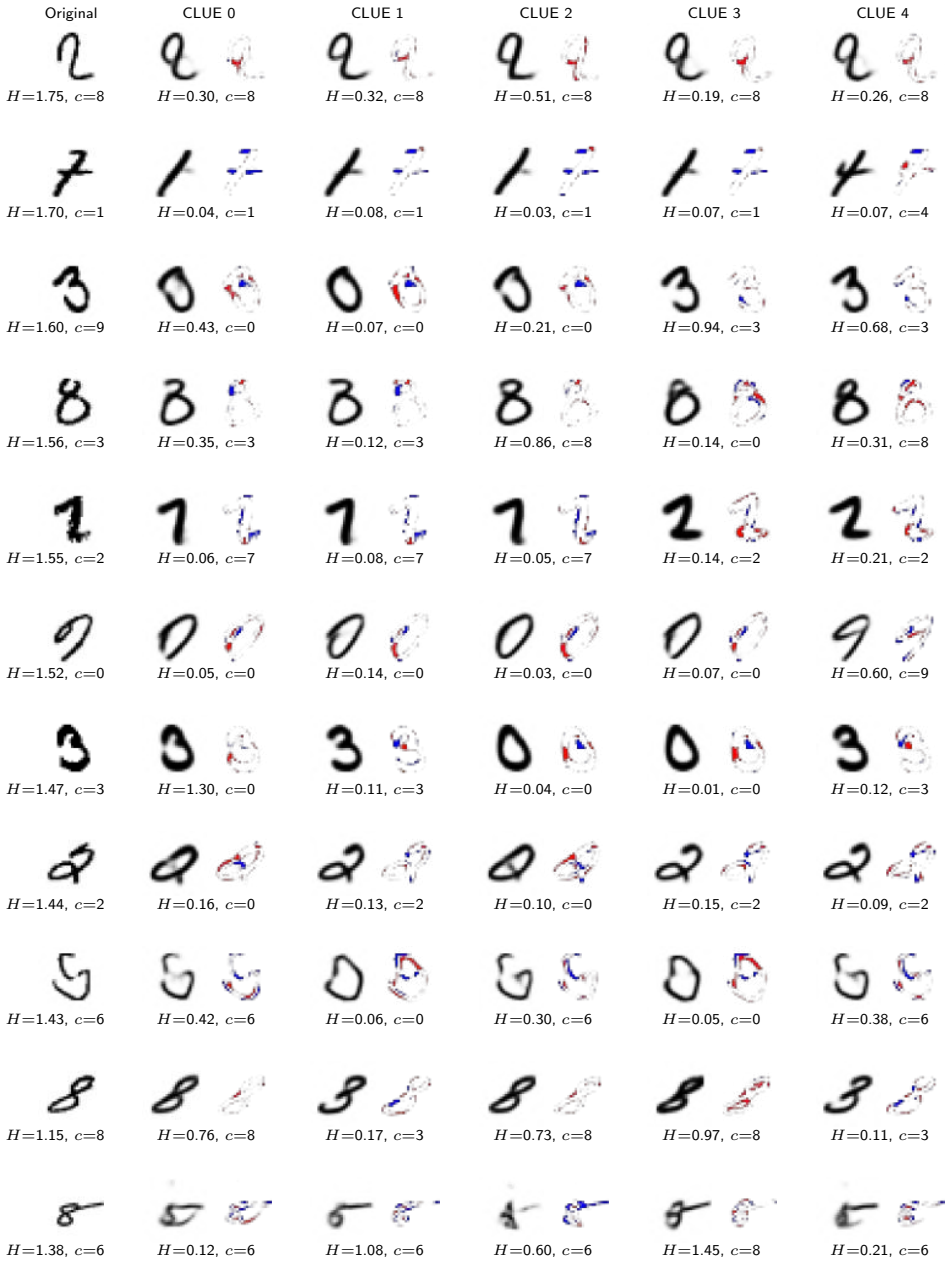


Fig. C.2 We generate 5 possible CLUEs for 11 MNIST digits score above the uncertainty rejection threshold. Below each digit or counterfactual is the predictive entropy it is assigned H and the class of maximum probability c.

Comparing CLUE to Feature Importance Estimators

Among machine learning practitioners, two of the most popular approaches for determining feature importance from back-box models are LIME and SHAP Bhatt et al. [2020b]. LIME locally approximates the back-box model of interest around a specific test point with a surrogate linear model [Ribeiro et al., 2016]. This surrogate is trained on points sampled from nearby the input of interest. The surrogate model's weights for each class can be interpreted as each feature's contribution towards the prediction of said class. Kernel SHAP extends lime by introducing a kernel such that resulting explanations have desirable properties [Lundberg and Lee, 2017]. For SHAP, a reference input is chosen. It allows importance to be only assigned where the inputs are different from the reference. For MNIST, the reference is an entirely black image. Note that alternative versions of SHAP exist that incorporate information about internal NN dynamics into their explanations. However, they produce very noisy explanations when applied to our BNNs. We conjecture that this high variance might be induced by disagreement among the multiple weight configurations from our BNNs.

Figure C.3 shows examples of LIME and Kernel SHAP being applied to a BNN for high-confidence MNIST test digits. We use the default LIME hyperparameters for MNIST: the "quickshift" segmentation algorithm with kernel size 1, maximum distance 5, and a ratio of 0.2. We plot the top 10 segments with weight greater than 0.01. We draw 1000 samples with both methods. Using the same configuration, we generate LIME and SHAP explanations for some MNIST digits to which our BNN assigns predictive entropy above our rejection threshold. The results are displayed in Figure C.4.

A positive CLUE attribution means that the addition of that feature will make our model more certain. A positive feature importance attribution means the presence of that feature serves as evidence towards a predicted class. A negative CLUE attribution means that the absence of that feature will make the model more certain. A negative feature importance attribution means the absence of that feature would serve as evidence for a particular prediction. While CLUE and feature importance techniques solve similar problems and both provide saliency maps, CLUE highlights regions that need to be added or removed to make the input certain to a predictive model. In some cases, we see that feature importance negative attribution aligns with CLUE negative attribution, suggesting the features that negatively contribute to the model's predicted probability are the features that need to be removed to increase the models' certainty. CLUE's ability to suggest the addition of unobserved features (positive CLUE attribution) is unique.

The feature importance methods under consideration are difficult to retrofit for uncertainty. They are unable to add features; they are limited to explaining the contribution of existing

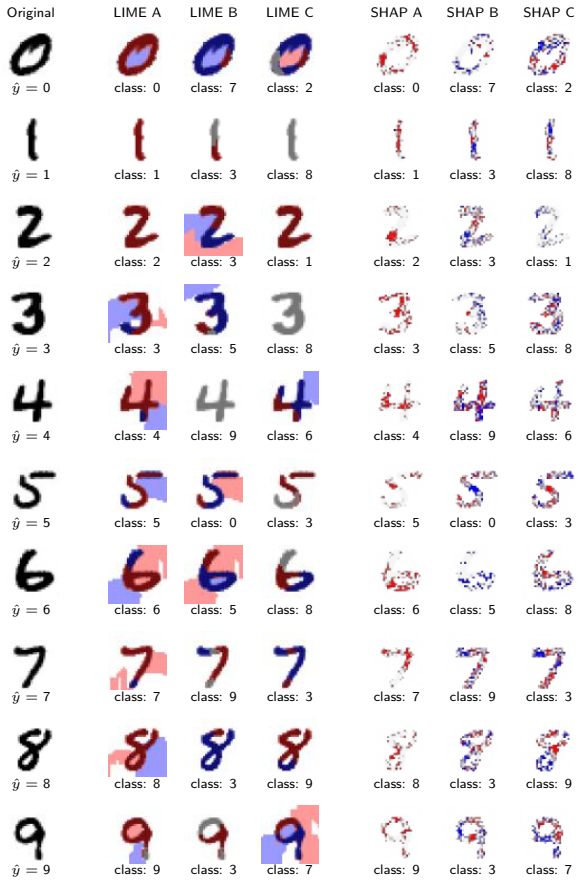


Fig. C.3 High confidence MNIST test examples together with LIME and SHAP explanations for the top 3 predicted classes. The model being investigated is a BNN with the architecture described in Antorán et al. [2021]. The highest probability class is denoted by $\hat{y}$.

features. This may suffice if our input contains all the information needed to make a prediction for a certain class but otherwise results in noisy, potentially meaningless, explanations.

Generative-model-based methods are counterfactual because they do not assign importance to the observed features but rather propose alternative features based on the data manifold [Chang et al., 2019]. This is the case for FIDO and CLUE. Generative modeling allows for increased flexibility, which is required when dealing with uncertain inputs. Quantitatively contrasting feature importance and uncertainty explanations under existing evaluation criteria [Hase and Bansal, 2020] is an interesting direction for future work.

Methods like LIME and SHAP require a choice of class to produce explanations. This complicates their use in scenarios where our model is uncertain and multiple classes have a similarly high predictive probability. On the other hand, CLUEs are class agnostic.



Fig. C.4 Ten MNIST test digits for which our BNN's predictive entropy is above the rejection threshold. A single CLUE is provided for each one. For each digit, the top scoring class is denoted by $\hat{y}$. LIME and SHAP explanations are provided for the three most likely classes.

Additional CLUE and U-FIDO Examples

We provide additional examples of CLUEs generated for high uncertainty MNIST digits in Figure C.5. U-FIDO counterfactuals generated for the same inputs are shown in Figure C.6. Both methods often attribute importance to the same features. However, in almost all cases, CLUE is able to reduce the original input's uncertainty significantly more than U-FIDO. The latter method suggests smaller changes. We attribute this to U-FIDO's input masking mechanism being less flexible than CLUE's latent space generation mechanism.

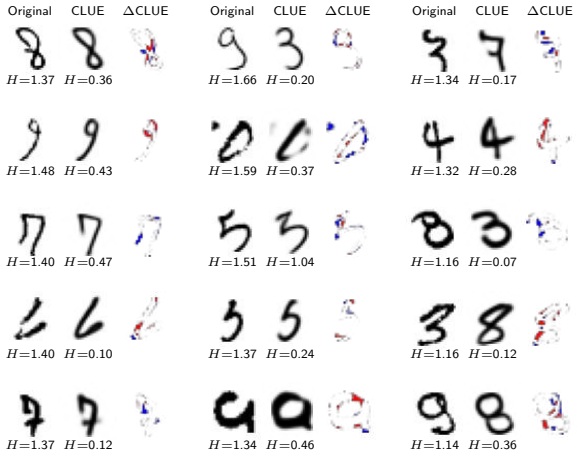


Fig. C.5 CLUEs generated for MNIST digits for which our BNN's predictive entropy is above the rejection threshold. The BNNs predictive entropy for both original inputs and CLUEs is shown under the corresponding images.

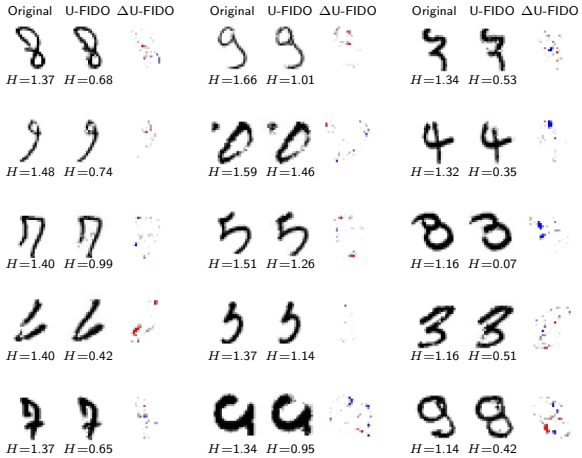
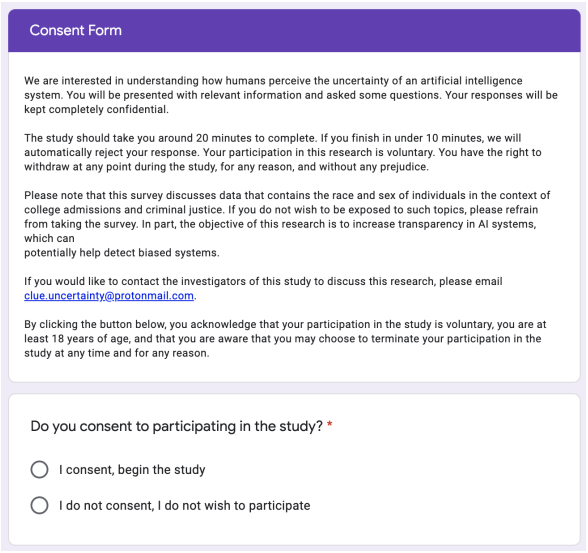


Fig. C.6 U-FIDO counterfactuals generated for MNIST digits for which our BNN's predictive entropy is above the rejection threshold. The BNNs predictive entropy for both original inputs and counterfactuals is shown under the corresponding images.

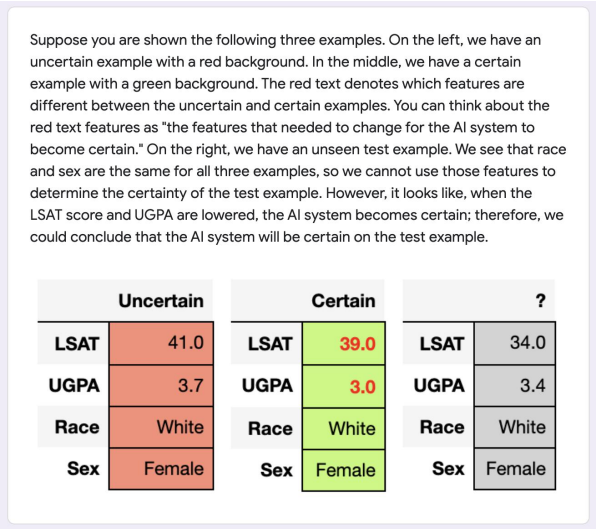
C.3 Details on User Study

Additional Details on Tabular User Study

For our pilot point selection procedure, we take points from each dataset's test set that score above the uncertainty rejection thresholds described in Antorán et al. [2021] as uncertain points. Points below the thresholds are labeled as certain points. Pilot procedure participants, referred to as Participant A in the main text, were not informed that the pools were split up by the points' certainty with respect to the BNN being explained.



(a) Consent Form for the Tabular Main Survey



(b) Attention Check for the Tabular Main Survey

Fig. C.7 Setup of tabular user studies.

In Figure C.7a, we include the consent form used in our user studies. This user study was performed with the approval of the University of Cambridge's Department of Engineering Research Ethics Committee. Only three participants who were asked to take the survey did not provide consent and thus exited the form. We still ensured that at least ten participants took each of the four survey variants.

We then include an example question for each dataset, called an "attention check." An example is shown in Figure C.7b. Note that the answer to this example question is provided in line. Later in the survey, we ask participants this exact same question. We ask one attention check per dataset. If participants get the attention check wrong for both datasets, we void their results. We only had to void one result. This did not affect our criteria of ten completed surveys per variant. The consent form and attention check questions were the same for all survey variants. The main survey participants were first asked the ten LSAT questions followed



(a) Two LSAT questions with certain points generated by CLUE

(b) Two COMPAS questions with certain points generated by CLUE

Fig. C.8 Example Tabular Main Survey questions

by the ten COMPAS questions: we made this design decision since the dimensionality of LSAT is lower than that of COMPAS, easing participants into the task. Examples of questions from the CLUE survey variant are shown in Figure C.8.

MNIST User Study

To validate CLUE on image data, we create a modified MNIST dataset with clear failure modes for practitioners to identify. We first discard all classes except four, seven, and nine. We then manually identify forty sevens from the training set which have dashes crossing their stems. Using K-nearest-neighbors, we identify the twelve sevens closest to each of the ones manually selected. We delete these 520 sevens from our dataset. We repeat the same procedure for fours which have a closed, triangle-shaped top. We do not delete any digits from the test set. We train a BNN on this new dataset. Our BNN presents high epistemic uncertainty when tested on

dashed sevens and closed fours as a consequence of the sparsity of these features in the train set.

We evaluate the test set of fours, sevens, and nines with our BNN. Data points that surpass our uncertainty threshold are selected as candidates to be shown in our user study as uncertain context examples or test questions. We show example CLUEs for a four and a seven that display the characteristics of interest in Figure C.9.

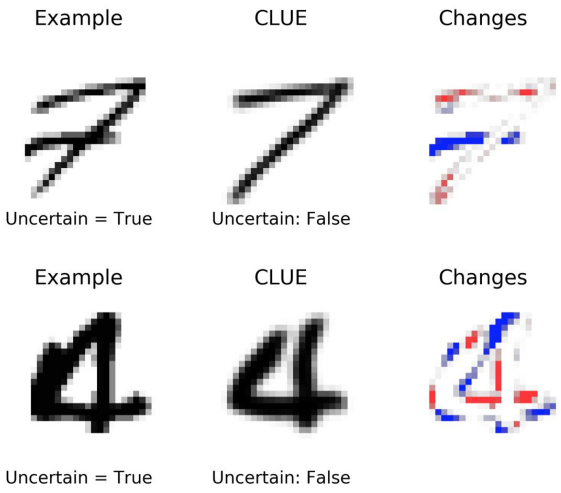
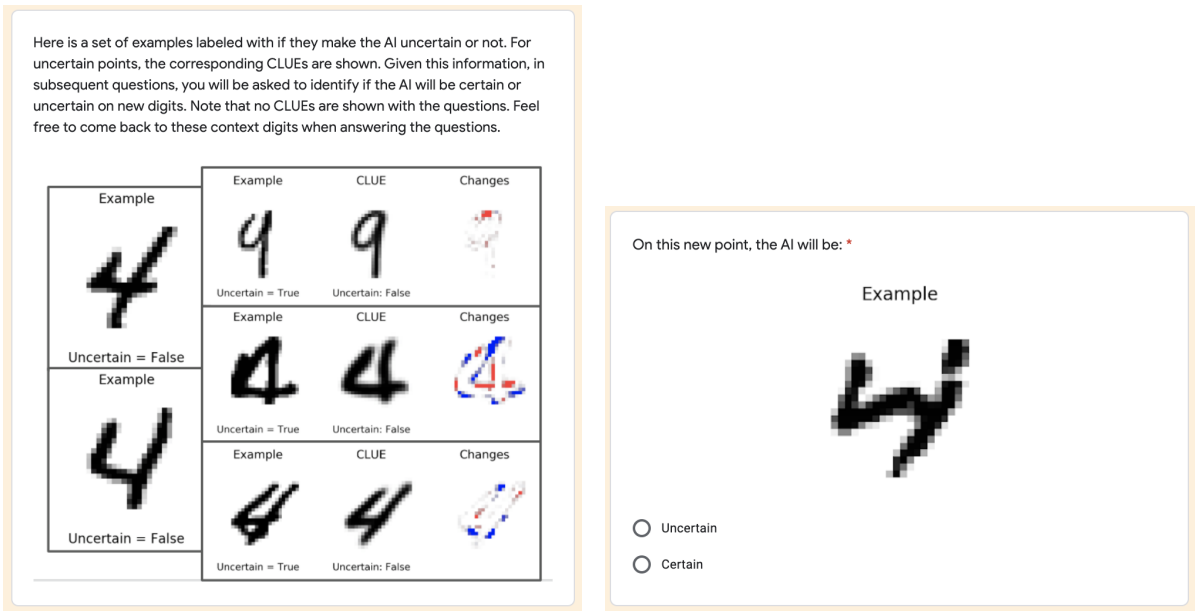


Fig. C.9 Examples of high uncertainty digits containing characteristics that are uncommon in our modified MNIST dataset. Their corresponding CLUEs and ΔCLUEs are displayed beside them.

Leveraging the modified MNIST dataset, we run another user study with 10 questions and two variants. Unlike our tabular experiments, we show practitioners a set of five *context points* to start, as opposed to a pair. This set of *context points* is chosen at random from the training set. The first variant involves showing users the set of *context points*, labeled with if their uncertainty surpasses our predefined threshold. We then ask users to predict if new test points will be certain or uncertain to the BNN. The second variant contains the same labeled context points and test data points. However, together with uncertain context points, practitioners are shown CLUEs of how the input features can be changed such that the BNN's uncertainty falls below the rejection threshold. The practitioners are then asked to decide if new points' predictions will be certain or not. If CLUE works as intended, practitioners taking the second variant should be able to identify points on which the BNN will be uncertain more accurately.

The first variant was shown to 5 graduate students with machine learning expertise who only received context points and rejection labels (uncertain or not). This group was able to correctly classify 67% of the new test points as high or low uncertainty. The second variant was



(a) Example Context Set with CLUEs

(b) Example question

Fig. C.10 MNIST User Study Setup

shown to 5 other graduate students with machine learning expertise who received context points together with CLUEs in cases of high uncertainty. This group was able to reach an accuracy of 88% on new test points. This user study suggests CLUEs are useful for practitioners in image-based settings as well.

Appendix D

Learning Personalized Decision Support Policies

We provide further details on decision support policies, additional computational experiments with THREAD, and extensive information on our human subject experiments with Modiste.

D.1 Comparison Against Prior Work

Most papers on human-AI collaboration have considered clever ways of abstaining from predicting on specific inputs [Cortes et al., 2016, 2018], learning deferral functions based on multiple experts [Keswani et al., 2021; Vovk, 1998], or teaching decision-makers when to rely [Mozannar et al., 2022b]. There are also a number of papers from the HCI literature (see survey by [Lai et al., 2023]) that evaluate the two-action setting of our formulation using a *static* policy (e.g. always showing a model prediction or always showing some form of explanation with the prediction).

To clarify how our set-up and assumptions differ from prior work, we overview work that we believe could be considered most similar to ours. We decompose our comparisons along a few dimensions: **Decision-support set-up:** Does the human make the final decision, or is it a different set-up? **Assumptions about decision-maker information:** What does prior work assume about access to a decision-maker when learning a policy? **Evaluation:** Does prior work simulate humans? Does prior work run user studies?

Mozannar and Sontag [2020]:

- **Decision-support set-up:** This work's set-up can be considered a two-action setting of our formulation, where $\mathcal{A} = \{\texttt{DEFER}, \texttt{MODEL}\}$. Extending the work of Madras et al. [2018], this

work proposes the learning to defer paradigm, where the decision-maker may not always make a final decision (i.e. sometimes the decision is deferred entirely to an algorithmic-based system). In our set-up, deferring to a MODEL is equivalent to always adhering to a label-based form of support. The human is always the final decision-maker in our work, which is representative of many decision-making set-ups in practice [Lai et al., 2022], but not captured in this line of prior work.

- **Assumptions about decision-maker information:** This work assumes oracle query access to the decision-maker, for whom they are learning a policy.
- **Evaluation:** This work evaluates their approach using human simulations (no real human user studies). Mozannar and Sontag [2020] define synthetic experts in the following way: "if the image belongs to the first k classes the expert predicts perfectly, otherwise the expert predicts uniformly over all classes."

Gao et al. [2021] and Gao et al. [2022]:

- **Decision-support set-up:** This work defines two actions: $\mathcal{A} = \{\texttt{DEFER}, \texttt{MODEL}\}$. They do not consider the, more practical assumption that the decision-maker will view a model prediction before making a decision themselves. Their formulation is similar to the above but they use offline bandits to learn a suitable policy.
- **Assumptions about decision-maker information:** Gao et al. [2021] assume that understanding decision-maker's expertise (at a population-level, not at a individual-level) can help learn better routing functions (i.e. defer only when appropriate). Gao et al. [2022] assume access to a decision history for each decision-maker.
- **Evaluation:** They run a human subject experiment to collect offline annotations, which can be used to learn when to defer to decision-makers. Gao et al. [2022] goes further to personalize a deferral policy based on offline annotations for each decision-maker.

Bordt and Von Luxburg [2022]:

- **Decision-support set-up:** This work's set-up can be considered a two-action setting of our formulation, where $\mathcal{A} = \{\texttt{DEFER}, \texttt{SHOW}\}$; however, they are concerned with the learnability of such a set up. They do not devise algorithms for this setting, as they are only focused on its theoretical formulation.
- **Assumptions about decision-maker information:** They assume the decision-maker has access to information not contained in the input but still important to the task.

- **Evaluation:** This is a theory paper, containing neither computational experiments nor human subject experiments.

Noti and Chen [2022]:

- **Decision-support set-up:** This work's set-up can be considered as a two-action setting of our formulation, where $\mathcal{A} = \{\texttt{DEFER}, \texttt{SHOW}\}$. This is not an online algorithm and as such, the policy does not update.
- **Assumptions about decision-maker information:** They assume access to a dataset of human decisions and that all decision-makers are similar (i.e. they deploy one policy for all decision-makers).
- **Evaluation:** This work is one of few that runs a user study to evaluate their (fixed) policy on unseen decision-makers.

Babbar et al. [2022]:

- **Decision-support set-up:** This work considers the two-action setting of our formulation, where $\mathcal{A} = \{\texttt{DEFER}, \texttt{CONFORMAL}\}$. Their policy is learned offline, is the same for all decision-makers, and is not updated in real-time based on decision-maker behavior.
- **Assumptions about decision-maker information:** They use CIFAR-10H Peterson et al. [2019] to learn a population-level deferral policy. This assumes that we have annotations for each decision-maker for every data point and assumes that all new decision-makers have the same expertise profiles as the population average.
- **Evaluation:** This work runs a user study to evaluate their (fixed) policy. They show the benefits of `DEFER+CONFORMAL` over `CONFORMAL` or `SHOW` alone.

Wolczynski et al. [2022]:

- **Decision-support set-up:** This work considers two actions per our formulation, where $\mathcal{A} = \{\texttt{DEFER}, \texttt{SHOW}\}$. They learn a rule-based policy offline for each decision-maker.
- **Assumptions about decision-maker information:** They simulate human behavior by considering explicit functions of how human expertise may vary in input space.
- **Evaluation:** While this work does consider the human to be the final decision-maker, they only validate their proposal in simulation, not on actual human subjects.

Prior work on multi-objective contextual bandits. We summarize why some theoretical work on multi-objective contextual bandits cannot be directly applied to our problem formulation. Tekin and Turğay [2018] addressed a contextual multi-armed bandit problem with two objectives, where one objective dominates the other. Their aim was to maximize the total reward in the non-dominant objective while ensuring that the dominant objective's total reward is also maximized. However, our specific case requires the minimization of the total expected cost of the support policy while ensuring that the expected accuracy of the decision-maker under the policy remains above a certain threshold. Therefore, the techniques presented cannot be directly applied to our scenario. Turgay et al. [2018] investigated the multi-objective contextual bandit problem with similarity information. Their approach relies on the assumption that a Lipschitz condition holds for the set of feasible context-arm pairs concerning the expected rewards for all objectives and that the learning algorithm has knowledge of the corresponding distance function (see Assumption 1 of their paper).

D.2 Additional Details on Problem Formulation

Standard (cost-agnostic) Setting. The optimization problem in the standard setting, where the only objective relates to expected loss, can be written as:

$$\begin{aligned}
\pi^* &= \arg\min_{\pi\in\Pi} L_h(\pi) \\
&= \arg\min_{\pi\in\Pi} \mathbb{E}_x\left[\sum_{i=1}^{k} \pi(x)_{\mathrm{A}_i} \cdot \mathbb{E}_{y|x}[\ell(y,h(x,\mathrm{A}_i))]\right] \\
&= \arg\min_{\pi\in\Pi} \sum_{i=1}^{k} \pi(x)_{\mathrm{A}_i} \cdot \mathbb{E}_x\left[\mathbb{E}_{y|x}[\ell(y,h(x,\mathrm{A}_i))]\right] \\
&= \arg\min_{\pi\in\Pi} \sum_{i=1}^{k} \pi(x)_{\mathrm{A}_i} \cdot \mathbb{E}_x\left[r_{\mathrm{A}_i}(x;h)\right],
\end{aligned}$$

Thus $\pi^*(x) = \arg\min_{\mathrm{A}_i\in\mathcal{A}} r_{\mathrm{A}_i}(x;h)$ is a minimizer of the above.

Cost-Aware Setting. We consider the following multi-objective optimization (MOO) problem:

$$\min_{\pi} \mathcal{R}_h(\pi) = [L_h(\pi), c(\pi)]^\top. \tag{D.1}$$

We can reformulate this MOO problem into a single-objective optimization (SOO) problem:

$$\pi^*_\lambda = \arg\min_{\pi\in\Pi} \lambda \cdot L_h(\pi) + (1-\lambda)\cdot c(\pi), \tag{D.2}$$

where $\lambda \in [0,1]$. We observe that we can rewrite the SOO problem as follows:

$$\pi^*_\lambda(x) \;=\; \arg\min_{\mathrm{A}_i\in\mathcal{A}} \lambda \cdot r_{\mathrm{A}_i}(x;h) + (1-\lambda)\cdot c(\mathrm{A}_i), \tag{D.3}$$

To show that the solution to the SOO problem in Equation D.2 can be written as Equation D.3, consider the following:

$$\begin{aligned}
\pi^*_\lambda &= \arg\min_{\pi\in\Pi} \lambda \cdot L_h(\pi) + (1-\lambda)\cdot c(\pi) \\
&= \arg\min_{\pi\in\Pi} \lambda \cdot \mathbb{E}_x\left[\sum_{i=1}^{k} \pi(x)_{\mathrm{A}_i} \cdot \mathbb{E}_{y|x}[\ell(y,h(x,\mathrm{A}_i))]\right] + (1-\lambda)\cdot \mathbb{E}_x\left[\sum_{i=1}^{k} \pi(x)_{\mathrm{A}_i} \cdot c(\mathrm{A}_i)\right] \\
&= \arg\min_{\pi\in\Pi} \sum_{i=1}^{k} \pi(x)_{\mathrm{A}_i} \cdot \mathbb{E}_x\left[\lambda \cdot \mathbb{E}_{y|x}[\ell(y,h(x,\mathrm{A}_i))] + (1-\lambda)\cdot c(\mathrm{A}_i)\right] \\
&= \arg\min_{\pi\in\Pi} \sum_{i=1}^{k} \pi(x)_{\mathrm{A}_i} \cdot \mathbb{E}_x\left[\lambda \cdot r_{\mathrm{A}_i}(x;h) + (1-\lambda)\cdot c(\mathrm{A}_i)\right],
\end{aligned}$$

where the objective in the last step is minimized by the solution in Equation D.3.

Pareto Optimality

We use dominance [Miettinen, 2008] to define Pareto optimality for the MOO problem in Equation D.1.

Definition D.2.1 (Dominant policy)**.** A policy π is said to dominate another policy π', noted as $\pi \prec \pi'$, if the following holds: (i) $L_h(\pi) \leq L_h(\pi')$ and $c(\pi) \leq c(\pi')$; and (ii) either $L_h(\pi) < L_h(\pi')$ or $c(\pi) < c(\pi')$. Likewise, we denote $\pi \preceq \pi'$ if $\pi' \not\prec \pi$.

Definition D.2.2 (Pareto front and Pareto optimality)**.** Given a set of policies Π, the set of Pareto front policies is $\mathcal{P}_\Pi = \{\pi \in \Pi : \nexists \pi' \in \Pi \text{ s.t. } \pi' \prec \pi\}$. The corresponding Pareto front is given by $\mathcal{P}^{\mathcal{R}_h}_\Pi = \{(v_1, v_2) \in \mathbb{R}^2 : \exists \pi \in \mathcal{P}_\Pi \text{ s.t. } v_1 = L_h(\pi) \text{ and } v_2 = c(\pi)\}$. A policy π is a Pareto optimal solution to the MOO problem in Equation D.1 iff $\pi \in \mathcal{P}_\Pi$.

Definition D.2.3 (Convex Pareto front)**.** A Pareto front $\mathcal{P}^{\mathcal{R}_h}_\Pi$ is convex if $\forall v, v' \in \mathcal{P}^{\mathcal{R}_h}_\Pi, \lambda \in [0,1]$, $\exists v_\lambda \in \mathcal{P}^{\mathcal{R}_h}_\Pi$ such that $v_\lambda \preceq \lambda \cdot v + (1-\lambda)\cdot v'$.

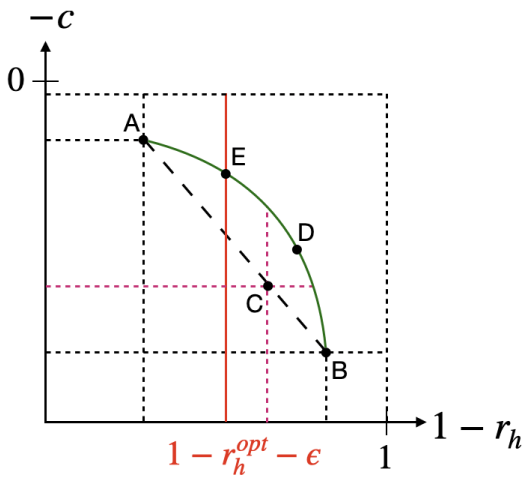


Fig. D.1 The green curve corresponds to the (unknown) Pareto front of the MOO problem in Equation D.1, e.g. $\mathsf{A} = (1 - L_h(\pi_0^*), -c(\pi_0^*))$, $\mathsf{B} = (1 - L_h(\pi_1^*), -c(\pi_1^*))$, and $\mathsf{D} = (1 - L_h(\pi_\lambda^*), -c(\pi_\lambda^*))$ for some $\lambda \in (0,1)$. The point $\mathsf{E} = (1 - L_h(\pi^\varepsilon), -c(\pi^\varepsilon))$ corresponds to the constrained optimisation problem Equation D.4, which is the point on the curve that we would like our decision support policy to achieve. The dashed line corresponds to the set $\{\pi_\mu := \mu \cdot \pi_0^* + (1-\mu) \cdot \pi_1^* : \mu \in [0,1]\}$, e.g. $\mathsf{C} = (1 - L_h(\pi_\mu), -c(\pi_\mu))$ for some $\mu \in (0,1)$. While the set of Pareto optimal policies Π_h^{opt} may be difficult to compute without further assumptions, we propose a tractable proxy $\{\pi_\mu\}$, that can be recovered if one is able to compute π_0^* and π_1^*, which would already require accurate estimation of $L_h(x,a)$ values for all x and a.

Note that optimal solutions to the SOO problem in Equation D.2 are Pareto optimal solutions to the MOO problem in Equation D.1. In the following proposition, we show that the Pareto front $\mathcal{P}_\Pi^{\mathcal{R}_h}$ of the MOO problem in Equation D.1 can be fully characterized by the solutions of the SOO problem in Equation D.2, i.e. any Pareto optimal policy $\pi \in \mathcal{P}_\Pi$ is a solution to the SOO problem in Equation D.2 for some choice of $\lambda \in [0,1]$.

Proposition D.2.4. *The Pareto front of the MOO problem in Equation D.1 is convex:* $\forall v, v' \in \mathcal{P}_\Pi^{\mathcal{R}_h}, \lambda \in [0,1], \exists v'' \in \mathcal{P}_\Pi^{\mathcal{R}_h} : v'' \preceq \lambda \cdot v + (1-\lambda) \cdot v'$. *Every Pareto solution of the MOO problem in Equation D.1 is a solution to the SOO problem in Equation D.2:* $\forall v \in \mathcal{P}_\Pi^{\mathcal{R}_h}, \exists \lambda :$ $v = [L_h(\pi_\lambda^*), c(\pi_\lambda^*)]^\top$.

Proof. First, we note that the set of stochastic policies Π is a convex set, and both $L_h : \Pi \to [0,1]$ and $c : \Pi \to [0,1]$ are convex functions. Then, the first statement of the proposition follows from Theorem 4.1 of [Martinez et al., 2020]. The second statement is a direct application of the results in [Geoffrion, 1968]. □

Learning Policy with a Tolerance Threshold

Given a tolerance threshold $\varepsilon \in [0,1]$ for the expected loss, we want to learn a decision support policy π that achieves minimum expected cost while maintaining the expected loss that is within ε of the optimal loss L_h^{opt}:

$$\pi^{\varepsilon} \;=\; \arg\min_{\pi\in\Pi} c(\pi) \quad \text{such that} \quad L_h(\pi) \le L_h^{\text{opt}} + \varepsilon, \tag{D.4}$$

where $L_h^{\text{opt}} = \min_\pi L_h(\pi)$. We observe that there exists a $\lambda(\varepsilon) \in [0,1]$ such that $\pi^*_{\lambda(\varepsilon)} = \pi^{\varepsilon}$ (i.e. $\pi^{\varepsilon} \in \Pi_{\text{opt}} = \{\pi^*_\lambda : \lambda \in [0,1]\}$ where Π_h^{opt} is the complete set of Pareto optimal policies for a given human h).

Human Misspecification

We upper bound the misspecification error of applying a policy π_h (originally learned for human h) on human $\widetilde{h}$. Inspired by [Ben-David et al., 2006], we define $\Pi\Delta\Pi$-divergence to bound the human misspecification error. Let $\ell_h(\pi,\pi') = \mathbb{E}_x[\ell(h(x,\pi(x)), h(x,\pi'(x)))]$ be the expected disagreement between two policies π and π' for human h. The $\Pi\Delta\Pi$-divergence between two humans h and $\widetilde{h}$ is defined as $d_{\Pi\Delta\Pi}(h,\widetilde{h}) = \sup_{\pi,\pi'\in\Pi} \left|\ell_h(\pi,\pi') - \ell_{\widetilde{h}}(\pi,\pi')\right|$. Based on this divergence measure, we bound the risk $L_{\widetilde{h}}(\pi_h) = \mathbb{E}_{(x,y)\sim\mathcal{P}}[\ell(y,\widetilde{h}(x,\pi_h(x)))]$ of using policy π_h on human $\widetilde{h}$ as follows:

$$L_{\widetilde{h}}(\pi_h) \;\le\; L_h(\pi_h) + d_{\Pi\Delta\Pi}(h,\widetilde{h}) + \lambda_\Pi,$$

where $\lambda_\Pi = \inf_{\pi'\in\Pi}[L_h(\pi') + L_{\widetilde{h}}(\pi')]$.

λ Selection Strategies

There are many ways to select the λ to use on unseen decision-makers using a population of decision-makers. We outline three such strategies below. Note we use B in the main text.

A. **Most likely λ.** For each simulator, we identify the set of $\{\lambda_{\text{sim-}j}\}$, which yield policies that meet $L^{\text{opt}}_{h_{\text{sim-}j}} + \varepsilon$. We then take the λ that occurs the *most often* across simulators. If no policy meets the threshold for $h_{\text{sim-}j}$, we select the policy closest to $L^{\text{opt}}_{h_{\text{sim-}j}}$, which can be computed given r_{A_i} for all x for each simulated decision-maker.

B. **Most likely λ with lowest cost.** For each simulator, we identify the set of $\{\lambda_{\text{sim-}j}\}$, which yield policies that meet $L^{\text{opt}}_{h_{\text{sim-}j}} + \varepsilon$, and select the λ whose policy has the least expected cost. Then, we identify the most common value across simulators. While the

set of values here is a subset of the selection strategy A, a cost-aware selection strategy may identify a parameter that leads to a lower cost for new decision-makers.

C. **Conservative λ.** For each simulator, we identify $\lambda_{\text{sim-}j}$ which is the minimum parameter that meets $L^{\text{opt}}_{h_{\text{sim-}j}} + \varepsilon$. We then take $\max_j \lambda_{\text{sim-}j}$ over all simulated decision-makers. In the absence of suitable population-level information, this strategy is similar to picking a conservative value of λ that prioritizes performance.

D.3 Additional Experimental Details

Datasets

Computational Only

Two tasks (Synthetic-2*A* and CIFAR-2*A*) are used only in the computational experiments, so we describe them abstractly. Both have the same set-up, but have different underlying data generating distributions. Consider learning a decision support policy in a setting where there are *two* forms of support available: $\mathcal{A} = \{\text{A}_1, \text{A}_2\}$, hence CIFAR-2*A*. Let $c(\text{A}_1) = 0$ and $c(\text{A}_2) = 0.5$. We simulate a decision-maker's behavior under each r_{A_i} across 3 classes for Synthetic/CIFAR. We instantiate the m decision-makers from the population data using human simulations, where we randomly sample $r_{\text{A}_i}(x;h)$ from a distribution for each h.

CIFAR

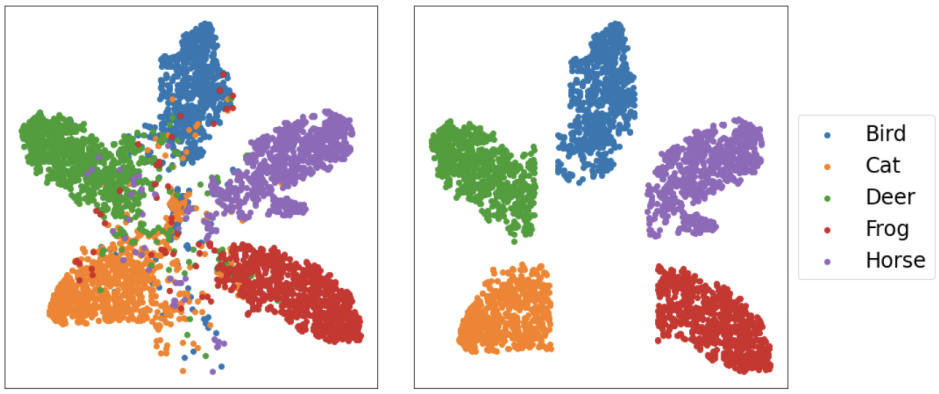


Fig. D.2 We depict the latent space of the CIFAR subset used. Embeddings without filtering (left) and after filtering out points that violate separability of classes (right). We use the embeddings on the right for our Synthetic-2*A* task and the left for the natural CIFAR tasks.

To explore the performance of our algorithms in both separable and non-separable settings, we construct two subvariants of the CIFAR-10 [Krizhevsky, 2009] image dataset. As depicted in Figure D.2, we subsample from the CIFAR embeddings to create linearly separable classes. Embeddings are constructed by running t-SNE on the 512-dimensional latent codes extracted by the penultimate layer of a variant of the VGG architecture (VGG-11) [He et al., 2016] trained on the animal class subset of CIFAR; the model attained an accuracy of 89.5%. We filter the original 10 CIFAR classes to only include those involving animals (Birds, Cats, Deers, Dogs, Frogs, and Horses). We consider at most 5 such classes in our experiments (dropping Dog).

MMLU

We consider questions from four topics of MMLU (elementary mathematics, high school biology, high school computer science, and US foreign policy). Topic names match those proposed in [Hendrycks et al., 2020]. We select topics to cover span an array of disciplines across the sciences and humanities, as discussed in Appendix D.5. We use questions in the MMLU test set for each topic that are at most 150 characters long. We limit the length of questions to facilitate readability in our human user studies and wanted to maximize parity between our computational and human experiments by considering the same set of questions across both set-ups. This yields 264, 197, 98, and 47 questions for the elementary mathematics, high school biology, US foreign policy, and high school computer science topics, respectively. We emphasize that in CIFAR, our forms of support operate in label space – in contrast, with MMLU, support selection is in *covariate* (topic) space – further highlighting the flexibility of our adaptive support paradigm.

We leverage OpenAI's LLM-based embedding model (`text-embedding-ada-003`), to produce embeddings over the question prompts. Embedding vectors extracted via OpenAI's API are by default length 1536; we apply t-SNE, like in the CIFAR tasks, to compress the embeddings into two-dimensional latent codes per example. Nicely, these latents are already separable (see Figure 5.3; no post-filtering is applied to ensure separability).

CIFAR Task Set-up

We consider a 5-class subset of CIFAR (Bird, Cat, Deer, Frog, and Horse), as discussed in Appendix D.4[1]. We instantiate two forms of support: 1) a simulated AI model that provides a

[1]We explored a 3-class variant for HSEs to directly match the computational experiments; however, we realized that participants were able to figure out that which classes were impoverished, raising the base rate of correctly categorizing such images. Such behavior again highlights the need to carefully consider real human behavior in adaptive decision support systems.

prediction for the image class, and 2) a consensus response derived from real humans, derived from the approximately 50 human annotators from CIFAR-10H [Battleday et al., 2020; Peterson et al., 2019], presented as a distribution over the 5 classes.

We treat the original CIFAR-10 test set labels as the "true" labels and only include images for which the original CIFAR-10 label matches the label deemed most likely from the CIFAR-10H annotators (discrepancy was a rare occurrence, only 1.1% of the 5000 images considered). We sample a pool of 300 such images, and construct three different batches of 100 images. Participants are assigned to one batch of 100 images. Images per batch are sampled in a class-balanced fashion (i.e. 20 images for each of the 5 categories). Images are shuffled for each participant. The same latent codes (z_t) used in the computational experiments are employed for the user studies. For CIFAR, we define the support costs as: $c(\text{HUMAN ALONE}) = 0$ and $c(\text{MODEL}) = c(\text{CONSENSUS}) = 0.5$.

Corruptions and Support Design

We want to study whether our algorithms can properly learn which forms of decision support are actually needed by real humans. We therefore need to ensure that the task is sufficiently rich such that humans do *need* support (and the existing forms of support which can be of value). To mimic such a setting in CIFAR, we deliberately corrupt images of all classes except one (i.e. Birds[2]) when presenting images to the user. Corruptions are formed via a composition of natural adversarial transformations proposed in Hendrycks et al. [2020], specifically shot noise ontop of glass blur. This enables us to check that our algorithms are able to properly recommend support for all other classes, as a human will be unable to decide on image category unassisted. Example corrupted images can be seen in Figures D.9, D.10, and D.11

To ensure that the available forms of support have different regions of strength, we enforce that each form of support is only good at two of the five classes. In the case of the simulated AI model, we return the "true" class whenever the image is a Deer or Cat classes, and return one of the incorrect classes for all other images. For the consensus labels, we use the CIFAR-10H distribution derived from 50 annotators when the image is a Horse or Frog. As the CIFAR-10H labels were originally collected over all 10 CIFAR classes, sometimes an annotation was endorsed for a class outside of the 5 we consider; in that case, we discard the annotations assigned to those classes and renormalize the remaining distribution (on average, $< 2\%$ of the original labeling mass was discarded per image). For all other classes, we intend the consensus

[2]We further disambiguate the non-corrupted bird class by upsampling to 160x160 images via Lanczos-upsampling following [Battleday et al., 2020; Collins et al., 2022; Peterson et al., 2019] before presenting them to participants.

to be an unhelpful form of support, as such, for images in the Bird, Deer, and Horse class, we return a uniform distribution (i.e. providing no information to the participant).

MMLU Task Set-up

Participants respond to 60 multiple choice questions, which were balanced by topic (15 questions per topic). Participants are assigned to one of three possible batches of 60 such questions. Order is shuffled. Participants are informed of the topic associated with each question (e.g. that the question was about biology). We implement a 10 second delay between when the question is presented and when the participant is allowed to submit their response to encourage participants to try each problem in earnest. For MMLU, we let the support costs be $c(\text{HUMAN ALONE}) = 0$ and $c(\text{MODEL}) = 0.1$.

Topic Selection

We ran several pilot studies with `Modiste` to determine people's base performance on a subset of MMLU topics. We intended to select a diverse set of topics such that it was unlikely any one participant would excel at all topics – as many of the topics were specialized and challenging [Hendrycks et al., 2020] – but also varied enough that participants may be strong in at least one area. We found that a large number of participants achieved reasonable performance on elementary mathematics, and that a sufficient number of participants also excelled at questions in the high school biology, US foreign policy, and high school computer science topics – though usually not all together.

We also factored in `InstructGPT-3.5`'s performance while deciding on the topics. To best check whether our adaptive decision support algorithms are effective at learning good policies – like with CIFAR – it is helpful to have support available that is effective, should someone be unable to answer adequately alone. As a result, we looked more sympathetically on categories where model accuracy was already high.

However, the real-world is not so perfect: there may not be high-performing support available when a human struggles to make a decision. As such, we also deliberately forced down the accuracy of the LLM form of support in the mathematics topic. While we expect that most participants would be able to solve elementary mathematics problems without the aid of the LLM support, should they be unable to, the LLM was not able to help them. We leave further impoverishing studies which mimic real-world support settings for future work.

Question Selection and Model Accuracy

The resulting accuracy of the LLM form of support on the questions shown to humans per topic is 29% for elementary mathematics, 89% for high school biology, 87% for US foreign policy, and 91% for high school computer science. We upweighted selection of foreign policy questions that participants in our pilot had gotten correct when constructing the question subset, as we wanted to ensure that, should a participant have foreign policy experience, they would be able to answer them.

Constructing the Population Baseline

We recruited 20 participants from Prolific (via the same recruitment scheme as Section D.5) to estimate the population-level parameters (10 participants for MMLU and CIFAR, each). On each trial, participants are randomly assigned a form of support; trials are approximately balanced by the type of support and grouping (i.e. topic or class). We compute participant accuracy averaged over all trials (100 trials for CIFAR, 60 trials for MMLU).

Compute Resources

All computational experiments were run using CPUs (either on local machines or Google Colab), with the exception of training the VGG used for CIFAR embeddings. Here, we accessed a group compute cluster and trained the model for roughly five hours on a Nvidia A100-SXM-80GB GPU.

D.4 Additional Computational Experiments

Implications of selecting the trade-off parameter λ (with varying cost structure). We follow the strategies specified in Section 5.3.2 to identify λ for our cost-aware experiments. In Figure D.4, we visualized the expected excess loss and cost for each value of λ that is used for our human-informed synthetic decision-makers. Further, as shown in Figure D.3, we highlight how only a subset of λ values falls within $\varepsilon = 0.05$ of the L_h^{opt} of that individual (e.g. in Figure D.3). We also find that `THREAD`-KNN has more monotonic and controllable behavior as λ increases, while λ seems to have less of an effect on `THREAD`-LinUCB. We note that the set of λ values that fall within the red region is affected by the choice of cost structure, as shown in Figure D.3. Thus, any changes to the cost structure would imply the need to redo the hyper-parameter tuning process.

Additional Experiments on Datasets. In Table D.1, we provide additional computational results for CIFAR-2A and MMLU-2A in the cost-aware setting. We find that THREAD successfully personalizes decision support policies that trade-off cost and performance effectively. We note that in all of our experiments, we fix the exploration parameters in LinUCB and KNN (i.e. $\alpha = 1$ and $\gamma = 0.1$ respectively). However, we do explore varying these exploration parameters in Table D.3.

Varying KNN Parameters. While LinUCB does not require identifying additional parameters, KNN has two: K which is the number of nearest neighbors to select when estimating the risk of a form of support and W which is the length of the warm-up period. In Table D.4, we show that as long as K is reasonably sized (i.e. $K > 3$), the performance of KNN does not vary too much across datasets.

Varying Embedding Size. In the main text, we run computational experiments using a two-dimension t-SNE embedding. In Table D.5 for CIFAR-2A, we consider how THREAD would behave when we vary dimensionality of $\mathcal{X}$. We extract higher dimensional embeddings from animal-class trained VGGs (see above) with varied penultimate layer widths, where width matches embedding size. While we opt for t-SNE embeddings, in the main text, to provide a clearer visualizations, we find that strong performance holds even for high-dimensional embeddings. We expect that even larger embedding dimensions will permit THREAD to work in a variety of real-world contexts, but may come at the added cost of an increase in the number of required interactions, if we learn with no prior assumptions on the decision-maker's expertise profile.

$|\mathcal{A}| > 3$ **Experiments.** We now demonstrate how THREAD behaves when we increase the number of forms of support available to a decision-maker. Using the same set-up and algorithm-specific hyperparameters as Table 5.1, we convert the CIFAR setting to five forms of support (one for each class), resulting in CIFAR-5A, and convert the MMLU setting to four forms of support (one for each topic) resulting in MMLU-4A. In Table D.6, we report $L_h(\pi)$ at three time steps: $t = 50, 100, 250$. For CIFAR, we find that KNN significantly outperforms given more interactions, which is expected when we increase $|\mathcal{A}|$. We believe that LinUCB struggles as the t-SNE embedding space is likely not rich enough to capture the intricacies of all five forms of support. Given a larger embedding space, we expect that LinUCB would excel when the number of forms of support is increased. For MMLU, both LinUCB and KNN perform well by the 250th interaction: KNN is very good at lowering $L_h(\pi)$ on this task. In all, this

experiment assures us that careful design of decision support can permit us to use THREAD when we have a larger set of potential forms of support.

While we include experiments with larger $|\mathcal{A}|$ here, we note that higher set sizes may also be problematic for two reasons: 1) decision-makers may struggle under many forms of support [Kalis et al., 2013], and 2) as size increases, we find that the number of interactions required to learn an accurate personalized policy also increases.

Sensitivity to choice of T. In Figure D.5, we consider a setting of learning a policy with three forms of support: $\mathcal{A} = \{A_1, A_2, A_3\}$. Let $c(A_1) < c(A_2) = c(A_3)$. This is an analogous setup to what we do in our human subject experiments in Section 5.6. We indicate the simulated human expertise profiles in the caption. We show how performance on a random sample of 100 points from a held-out set varies with time while learning a decision support policy. In general, we find that the more forms of support to learn, the longer it may take for the online algorithms to "converge." For various simulations, KNN is more consistent in learning a decision support policy efficiently.

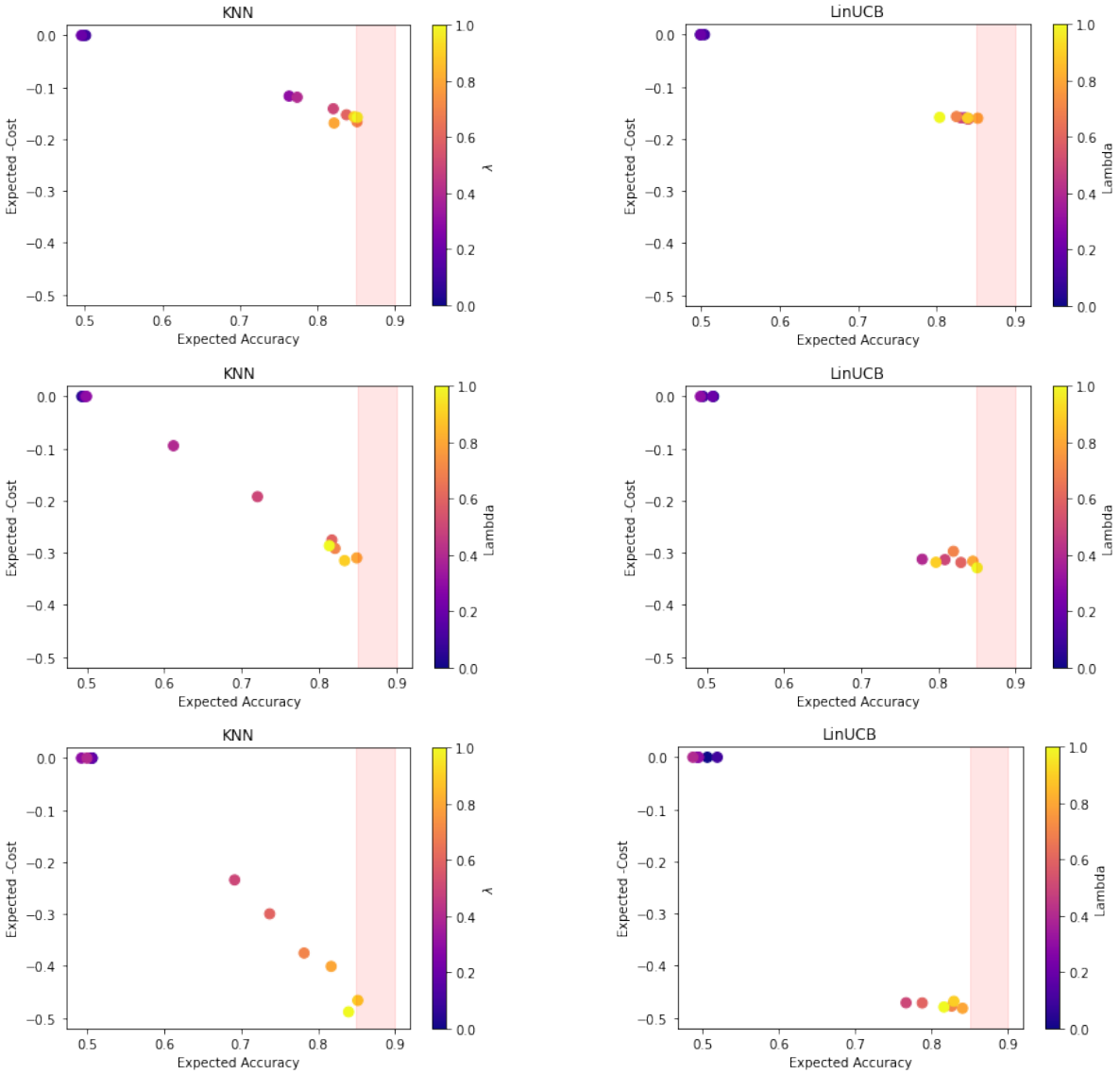


Fig. D.3 We plot the expected accuracy and the negative of the expected cost for an individual sampled from the Synthetic-2A dataset for $\lambda \in [0,1]$. The ideal policy would lie in the far right corner. The red region denotes policies that fall within ε of the best risk for the sampled individual. We observe that there exists a policy that lies in the red region for both THREAD-KNN and THREAD-LinUCB. We also vary the cost structure (i.e. the cost of $A_1 = 0$ and the cost of $A_2 = 0.25, 0.5, 0.75$ in the first, second, and third row respectively).

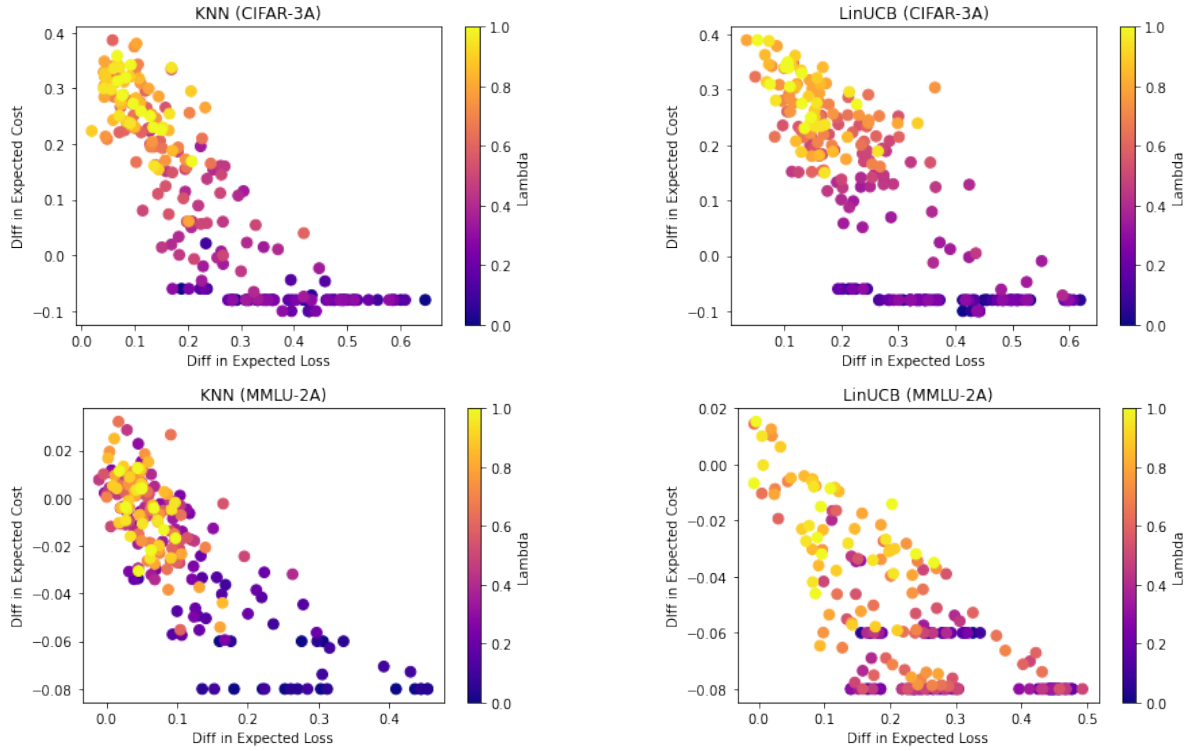
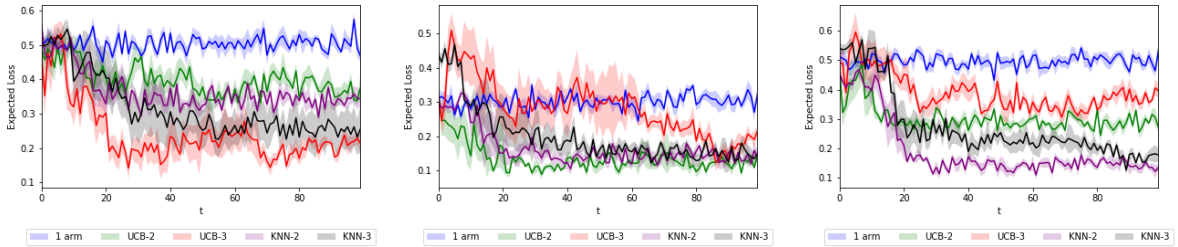


Fig. D.4 Top is CIFAR-3*A* and Bottom is MMLU-2*A*. For each of the $N = 10$ simulators, we instantiate using the population data, we compute the difference in expected loss between the optimal policy for that simulator and the learned policy for all values of $\lambda \in [0,1]$ in increments of 0.05, and compute the difference in expected cost in the same manner. For KNN, we observe overlap between many values of λ that all come within ε of the Best Risk: this implies we can use lower values of λ to potentially achieve the desired level of performance. For LinUCB, we find that large values of λ are required to achieve a suitable level of performance.



$r_{A_1} = [0.1, 0.7, 0.7]$
$r_{A_2} = [0.7, 0.1, 0.7]$
$r_{A_3} = [0.7, 0.7, 0.1]$

$r_{A_1} = [0.1, 0.7, 0.1]$
$r_{A_2} = [0.7, 0.1, 0.1]$
$r_{A_3} = [0.7, 0.7, 0.7]$

$r_{A_1} = [0.1, 0.7, 0.7]$
$r_{A_2} = [0.7, 0.1, 0.1]$
$r_{A_3} = [0.7, 0.7, 0.7]$

Fig. D.5 Loss plots over time for human simulations (denoted by r_{A_i}) in the CIFAR-3*A* setting.

Table D.1 Unlike Table 5.2 where risks come from pilot studies, we instantiate simulated **synthetic** humans, each specified by the human's loss on each form of support H-ONLY (Human only) and H-MODEL (Human+Model), for MMLU-2*A*. We report the expected loss $r_h(\pi)$ and the expected cost $c(\pi)$—for both metrics, lower is better—averaged across the last 10 steps of 100 total time steps along with their standard deviations (across 5 runs). We bold the algorithm that achieves the lowest cost within ε of the Best Risk for each human simulation and find that our algorithm outperforms baselines on various simulated humans. For each form of support, we specify risk r over each of the 4 topics for MMLU-2*A*. We fix the cost structure: $c(\text{H-ONLY}) = 0.0$ and $c(\text{H-MODEL}) = 0.1$. The selected λ values used for THREAD are in Table D.2. For THREAD, we select the best policy over a sweep of λ values that minimizes Eq. D.4 under $\varepsilon = 0.05$. We also indicate the values of λ chosen beneath the results table.

		$r_{A_1}=[0.7,0.1,0.7]$ $r_{A_2}=[0.1,0.1,0.1]$		$r_{A_1}=[0.7,0.1,0.7]$ $r_{A_2}=[0.1,0.7,0.1]$		$r_{A_1}=[0.7,0.1,0.7]$ $r_{A_2}=[0.7,0.7,0.1]$	
	Algorithm	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$
Synthetic-2*A*	A_1 Only	0.51 ± 0.05	0.0	0.50 ± 0.06	0.0	0.50 ± 0.06	0.0
	A_2 Only	0.10 ± 0.03	0.5	0.30 ± 0.05	0.5	0.50 ± 0.04	0.5
	Population	0.37 ± 0.04	0.13 ± 0.03	0.34 ± 0.05	0.13 ± 0.02	0.49 ± 0.06	0.12 ± 0.03
	THREAD-LinUCB	$\mathbf{0.15 \pm 0.05}$	$\mathbf{0.31 \pm 0.03}$	0.12 ± 0.04	0.32 ± 0.03	0.34 ± 0.11	0.28 ± 0.07
	THREAD-KNN	0.11 ± 0.04	0.36 ± 0.03	$\mathbf{0.13 \pm 0.04}$	$\mathbf{0.30 \pm 0.03}$	$\mathbf{0.33 \pm 0.06}$	$\mathbf{0.21 \pm 0.03}$
	Best Risk	0.1	0.33	0.1	0.33	0.3	0.17

		$r_{A_1}=[0.7,0.1,0.7]$ $r_{A_2}=[0.1,0.1,0.1]$		$r_{A_1}=[0.7,0.1,0.7]$ $r_{A_2}=[0.1,0.7,0.1]$		$r_{A_1}=[0.7,0.1,0.7]$ $r_{A_2}=[0.7,0.7,0.1]$	
	Algorithm	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$
CIFAR–2*A*	A_1 Only	0.50 ± 0.05	0.0	0.50 ± 0.05	0.0	0.47 ± 0.05	0.0
	A_2 Only	0.1 ± 0.03	0.5	0.31 ± 0.04	0.5	0.49 ± 0.05	0.5
	Population	0.46 ± 0.07	0.03 ± 0.04	0.43 ± 0.06	0.07 ± 0.03	0.50 ± 0.06	0.03 ± 0.02
	THREAD-LinUCB	$\mathbf{0.15 \pm 0.04}$	$\mathbf{0.31 \pm 0.02}$	$\mathbf{0.14 \pm 0.05}$	$\mathbf{0.32 \pm 0.02}$	0.34 ± 0.07	0.25 ± 0.07
	THREAD-KNN	0.14 ± 0.04	0.33 ± 0.06	$\mathbf{0.14 \pm 0.04}$	$\mathbf{0.32 \pm 0.03}$	$\mathbf{0.33 \pm 0.05}$	$\mathbf{0.18 \pm 0.03}$
	Best Risk	0.1	0.33	0.1	0.33	0.3	0.17

		$r_{A_1}=[0.8,0.8,0.1,0.1]$ $r_{A_2}=[0.2,0.1,0.1,0.2]$		$r_{A_1}=[0.2,0.8,0.1,0.1]$ $r_{A_2}=[0.8,0.1,0.1,0.2]$		$r_{A_1}=[0.2,0.5,0.5,0.8]$ $r_{A_2}=[0.8,0.1,0.1,0.8]$	
	Algorithm	$r_h(\pi)$	$c(\pi)$	$r_h(\pi)$	$c(\pi)$	$r_h(\pi)$	$c(\pi)$
MMLU-2*A*	H-ONLY ($\pi^*_{\lambda=0}$)	0.45 ± 0.04	0.0	0.33 ± 0.06	0.0	0.51 ± 0.04	0.0
	H-MODEL	0.14 ± 0.02	0.1	0.30 ± 0.05	0.1	0.45 ± 0.04	0.1
	Human Pop	0.45 ± 0.07	0.05 ± 0.01	0.22 ± 0.08	0.06 ± 0.01	0.41 ± 0.04	0.05 ± 0.01
	THREAD-LinUCB	0.15 ± 0.04	0.05 ± 0.06	0.19 ± 0.07	0.05 ± 0.02	0.39 ± 0.08	0.06 ± 0.01
	THREAD-KNN	$\mathbf{0.14 \pm 0.06}$	$\mathbf{0.04 \pm 0.01}$	$\mathbf{0.13 \pm 0.04}$	$\mathbf{0.03 \pm 0.04}$	$\mathbf{0.33 \pm 0.05}$	$\mathbf{0.05 \pm 0.01}$
	Best Risk ($\pi^*_{\lambda=1}$)	0.13	0.05	0.13	0.03	0.3	0.05

Selected λ Values

	Algorithm	$r_{A_1}=[0.7,0.1,0.7]$ $r_{A_2}=[0.1,0.1,0.1]$	$r_{A_1}=[0.7,0.1,0.7]$ $r_{A_2}=[0.1,0.7,0.1]$	$r_{A_1}=[0.7,0.1,0.7]$ $r_{A_2}=[0.7,0.7,0.1]$
Synthetic-2*A*	THREAD-LinUCB	0.4	0.6	0.6
	THREAD-KNN	0.9	0.6	0.6
CIFAR-2*A*	THREAD-LinUCB	0.4	0.9	0.5
	THREAD-KNN	0.7	0.9	0.6

	Algorithm	$r_{A_1}=[0.8,0.8,0.1,0.1]$ $r_{A_2}=[0.2,0.1,0.1,0.2]$	$r_{A_1}=[0.2,0.8,0.1,0.1]$ $r_{A_2}=[0.8,0.1,0.1,0.2]$	$r_{A_1}=[0.2,0.5,0.5,0.8]$ $r_{A_2}=[0.8,0.1,0.1,0.8]$
MMLU-2*A*	THREAD-LinUCB	1.0	0.9	1.0
	THREAD-KNN	0.3	0.2	0.3

Table D.2 The best value of λ that is selected from doing a sweep over $\lambda \in [0,1]$ that corresponds to the human simulators in Table 5.2. The column ordering is the same as the main text.

	Algorithm	Best Alone	Second Best Alone	Highest Best Risk	Lowest Best Risk
MMLU-2A	THREAD-LinUCB	0.7	0.65	1.0	0.95
	THREAD-KNN	0.25	0.55	0.35	0.6

Table D.3 Vary exploration parameters (α for LinUCB and γ for KNN). We find that LinUCB generally performs better when exploration increases and KNN generally performs better when exploration decreases.

		$r_{A_1} = [0.7,0.1,0.7]$ $r_{A_2} = [0.1,0.1,0.1]$		$r_{A_1} = [0.7,0.1,0.7]$ $r_{A_2} = [0.1,0.7,0.1]$		$r_{A_1} = [0.7,0.1,0.7]$ $r_{A_2} = [0.7,0.7,0.1]$	
	Algorithm	$r_h(\pi)$	$c(\pi)$	$r_h(\pi)$	$c(\pi)$	$r_h(\pi)$	$c(\pi)$
Synthetic-2A	THREAD-LinUCB ($\alpha = 0.1$)	0.14 ± 0.04	0.33 ± 0.02	0.19 ± 0.09	0.32 ± 0.02	0.34 ± 0.10	0.21 ± 0.07
	THREAD-LinUCB ($\alpha = 10$)	0.13 ± 0.04	0.32 ± 0.03	0.12 ± 0.03	0.33 ± 0.02	0.30 ± 0.05	0.23 ± 0.07
	THREAD-KNN ($\gamma = 0.01$)	0.12 ± 0.04	0.35 ± 0.06	0.19 ± 0.08	0.26 ± 0.07	0.33 ± 0.04	0.20 ± 0.03
	THREAD-KNN ($\gamma = 0.2$)	0.16 ± 0.05	0.33 ± 0.04	0.21 ± 0.06	0.24 ± 0.03	0.34 ± 0.06	0.15 ± 0.02
CIFAR-2A	THREAD-LinUCB ($\alpha = 0.1$)	0.15 ± 0.08	0.30 ± 0.04	0.14 ± 0.05	0.32 ± 0.02	0.31 ± 0.05	0.26 ± 0.07
	THREAD-LinUCB ($\alpha = 10$)	0.14 ± 0.04	0.32 ± 0.02	0.18 ± 0.04	0.31 ± 0.02	0.32 ± 0.05	0.26 ± 0.08
	THREAD-KNN ($\gamma = 0.01$)	0.12 ± 0.03	0.33 ± 0.02	0.14 ± 0.06	0.31 ± 0.03	0.35 ± 0.06	0.14 ± 0.03
	THREAD-KNN ($\gamma = 0.2$)	0.17 ± 0.05	0.32 ± 0.06	0.20 ± 0.09	0.31 ± 0.03	0.35 ± 0.06	0.16 ± 0.04

		$r_{A_1} = [0.8,0.8,0.1,0.1]$ $r_{A_2} = [0.2,0.1,0.1,0.2]$		$r_{A_1} = [0.2,0.8,0.1,0.1]$ $r_{A_2} = [0.8,0.1,0.1,0.2]$		$r_{A_1} = [0.2,0.5,0.5,0.8]$ $r_{A_2} = [0.8,0.1,0.1,0.8]$	
	Algorithm	$r_h(\pi)$	$c(\pi)$	$r_h(\pi)$	$c(\pi)$	$r_h(\pi)$	$c(\pi)$
MMLU-2A	THREAD-LinUCB ($\alpha = 0.1$)	0.26 ± 0.06	0.03 ± 0.01	0.22 ± 0.09	0.04 ± 0.01	0.47 ± 0.08	0.03 ± 0.02
	THREAD-LinUCB ($\alpha = 10$)	0.15 ± 0.05	0.04 ± 0.01	0.18 ± 0.07	0.03 ± 0.01	0.40 ± 0.06	0.05 ± 0.01
	THREAD-KNN ($\gamma = 0.01$)	0.16 ± 0.05	0.05 ± 0.00	0.15 ± 0.06	0.03 ± 0.01	0.39 ± 0.06	0.03 ± 0.01
	THREAD-KNN ($\gamma = 0.2$)	0.15 ± 0.03	0.05 ± 0.01	0.12 ± 0.03	0.03 ± 0.00	0.38 ± 0.05	0.03 ± 0.01

Table D.4 We vary the K and W parameters used to instantiate KNN across three datasets and report the expected risk $L_h(\pi)$ (lower is better) and expected cost $c(\pi)$ (lower is better) across 5 runs. The human simulation used here is the same as the one used in the second setting of Table D.1.

	$K = 3, W = 10$		$K = 5, W = 10$		$K = 5, W = 25$		$K = 8, W = 40$	
	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$
Synthetic-2A	0.20 ± 0.11	0.25 ± 0.08	0.15 ± 0.05	0.30 ± 0.03	0.14 ± 0.05	0.31 ± 0.03	0.12 ± 0.04	0.00 ± 0.00
CIFAR-2A	0.25 ± 0.15	0.24 ± 0.12	0.14 ± 0.04	0.34 ± 0.03	0.17 ± 0.04	0.30 ± 0.04	0.14 ± 0.04	0.32 ± 0.03
MMLU-2A	0.18 ± 0.07	0.03 ± 0.01	0.14 ± 0.04	0.03 ± 0.01	0.15 ± 0.03	0.04 ± 0.01	0.15 ± 0.04	0.03 ± 0.01

Table D.5 On the CIFAR-2*A* dataset, we explore the effect of varying the embedding size for KNN, with $K = 8$, $W = 25$ (Top), and LinUCB (Bottom). We report the expected risk $L_h(\pi)$ (lower is better) and expected cost $c(\pi)$ (lower is better) averaged over 5 runs. Recall that in our main text experiments we used two-dimensional t-SNE embeddings, not these model embeddings, for ease of visualization.

Embedding Size	$r_{A_1} = [0.7, 0.1, 0.7]$, $r_{A_2} = [0.1, 0.7, 0.1]$				$r_{A_1} = [0.7, 0.1, 0.7]$, $r_{A_2} = [0.7, 0.7, 0.1]$			
	$T = 100$		$T = 200$		$T = 100$		$T = 200$	
	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$
2	0.18 ± 0.08	0.28 ± 0.06	0.14 ± 0.04	0.32 ± 0.03	0.32 ± 0.05	0.24 ± 0.06	0.33 ± 0.06	0.23 ± 0.05
4	0.18 ± 0.04	0.29 ± 0.04	0.16 ± 0.04	0.32 ± 0.03	0.33 ± 0.05	0.20 ± 0.05	0.33 ± 0.05	0.21 ± 0.04
8	0.15 ± 0.05	0.30 ± 0.04	0.15 ± 0.03	0.33 ± 0.03	0.32 ± 0.05	0.24 ± 0.04	0.33 ± 0.05	0.24 ± 0.06
512	0.15 ± 0.05	0.30 ± 0.04	0.15 ± 0.03	0.33 ± 0.03	0.35 ± 0.05	0.24 ± 0.07	0.35 ± 0.05	0.23 ± 0.04

Embedding Size	$r_{A_1} = [0.7, 0.1, 0.7]$, $r_{A_2} = [0.1, 0.7, 0.1]$				$r_{A_1} = [0.7, 0.1, 0.7]$, $r_{A_2} = [0.7, 0.7, 0.1]$			
	$T = 100$		$T = 200$		$T = 100$		$T = 200$	
	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$	$L_h(\pi)$	$c(\pi)$
2	0.31 ± 0.05	0.19 ± 0.03	0.32 ± 0.04	0.20 ± 0.02	0.49 ± 0.05	0.10 ± 0.08	0.49 ± 0.05	0.11 ± 0.08
4	0.14 ± 0.04	0.36 ± 0.02	0.14 ± 0.04	0.36 ± 0.02	0.32 ± 0.04	0.23 ± 0.06	0.31 ± 0.50	0.22 ± 0.05
8	0.13 ± 0.04	0.35 ± 0.02	0.13 ± 0.03	0.36 ± 0.02	0.31 ± 0.03	0.20 ± 0.03	0.31 ± 0.04	0.20 ± 0.04
512	0.17 ± 0.03	0.31 ± 0.03	0.16 ± 0.04	0.31 ± 0.03	0.34 ± 0.04	0.24 ± 0.05	0.33 ± 0.06	0.25 ± 0.04

Table D.6 We evaluate $|\mathcal{A}| > 3$ in the cost-agnostic setting, $L_h(\pi_t)$ for $t = 50, 100, 250$ for the same setup as Table 5.1, but with more arms. We find that `THREAD` works well over time. The success of LinUCB is highly dependent on the embedding space and its geometry.

	Algorithm	$T = 50$	$T = 100$	$T = 250$
CIFAR-5*A*	`THREAD`-LinUCB	0.59 ± 0.10	0.29 ± 0.17	0.34 ± 0.10
	`THREAD`-KNN	0.19 ± 0.15	0.14 ± 0.17	0.11 ± 0.15
MMLU-4*A*	`THREAD`-LinUCB	0.13 ± 0.10	0.10 ± 0.07	0.17 ± 0.09
	`THREAD`-KNN	0.11 ± 0.12	0.08 ± 0.12	0.02 ± 0.04

D.5 Additional User Study Details and Results

Additional Recruitment Details

Within a task, participants are randomly assigned to an algorithm variant; an equal number of participants are included per variant (i.e. 10 for MMLU and 5 for CIFAR)[3]. Participants are required to reside in the United States and speak English as a first language. Participants are paid at a base rate of $9/hr, and are told they may be paid an optional bonus up to $10/hr based on the number of correct responses. We allot 25-30 minutes for the CIFAR task and 30-40 for MMLU, as each MMLU question takes more effort. We applied the bonus to all participants.

CIFAR Algorithm Parameters

For KNN, we take $K = 8$, $W = 25$, and $\varepsilon = 0.1$; for both algorithms, we let $\alpha = 1$. The cost of either model arm is 0.5. For each algorithm, we consider two different settings of λ ($\lambda \in \{0.75, 1.0\}$ and $\in \{0.85, 1.0\}$ for KNN and LinUCB respective), selected using the hyper-parameter selection methods per Section 5.3.2.

MMLU Algorithm Parameters

We use the same parameters as in the CIFAR task, with the exception that we set the cost of support (i.e. the `LLM` arm) to 0.1. Additionally, we employ different λ values – $\lambda \in \{0.75, 1.0\}$ for KNN and $\lambda \in \{0.95, 1.0\}$ for LinUCB – selected using the hyper-parameter selection methods per Section 5.3.2.

Additional HSE Results

Visualizing Learned Policies. We provide additional details on how the policy generalization snapshots in Figures 5.2 and 5.3 are constructed. We save out the parameters learned by each algorithm while a participant interacts with `Modiste`. After the 60 or 100 trials (depending on which task they participated in), we load in the final state of parameters. We sample the embeddings of *unseen points* from the respective task dataset (as described in Appendix D.3) and pass these through the respective algorithm (LinUCB or KNN, depending on which variant the participant was assigned to) – yielding a recommended form of support for said embedding. We color the latent by the form of support. We draw randomly 250 such unseen examples for CIFAR, and 100 for MMLU, when computing the policy recommendations.

[3]Due to a server-side glitch, 6 of the 125 recruited participants received incorrect feedback on $\leq 2\%$ of trials.

We include snapshots for participants, across both tasks, in Figures D.6 and D.7. We depict all CIFAR examples and a random sample of 7 of the 10 participants per variant for MMLU.

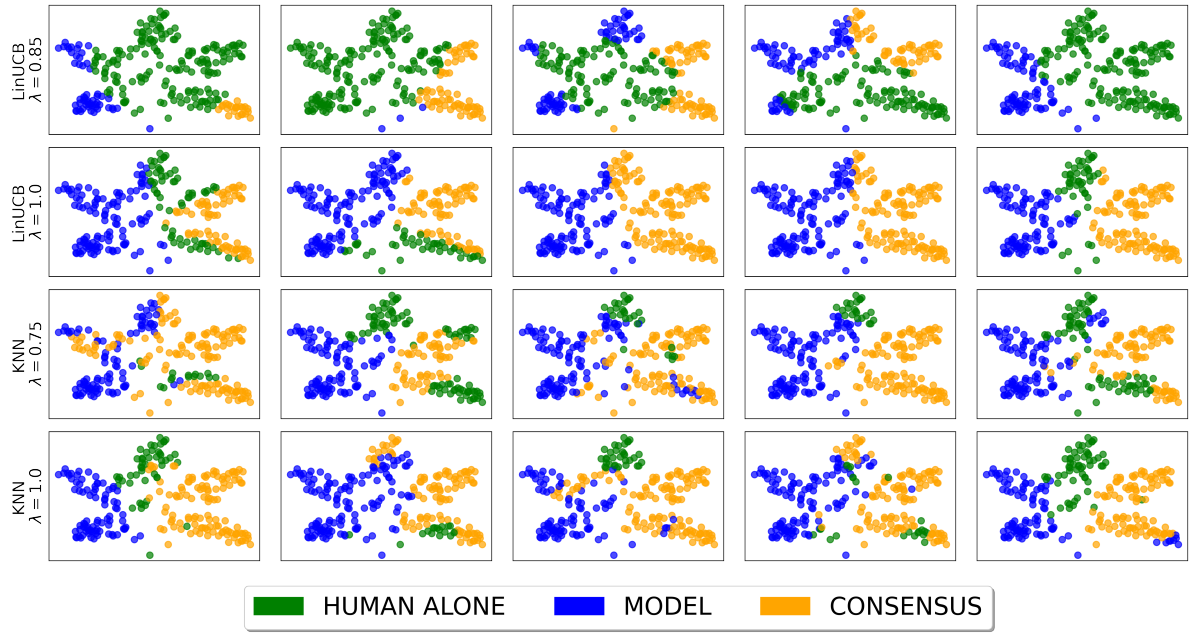


Fig. D.6 Snapshots of the recommended forms of support learned via `Modiste` for all participants in the CIFAR task, for different λ. Policies learned using KNN get closer to optimal; see Figure 5.2.

Cumulative results. We also report in Table D.7 the expected loss and cost across all trials. In general, we find similar trends as Table 5.3.

Interface

We include screenshots for the CIFAR (Figures D.8 D.9, D.10, and D.11) task. Here, we highlight the differences in image quality presented to participants during the task; images are drawn from the *same* dataset (i.e. CIFAR-10 [Krizhevsky, 2009]), but in Figure D.9, the image is *corrupted* (see Section D.3). We include screenshots from the MMLU task in Figures D.12 and D.13.

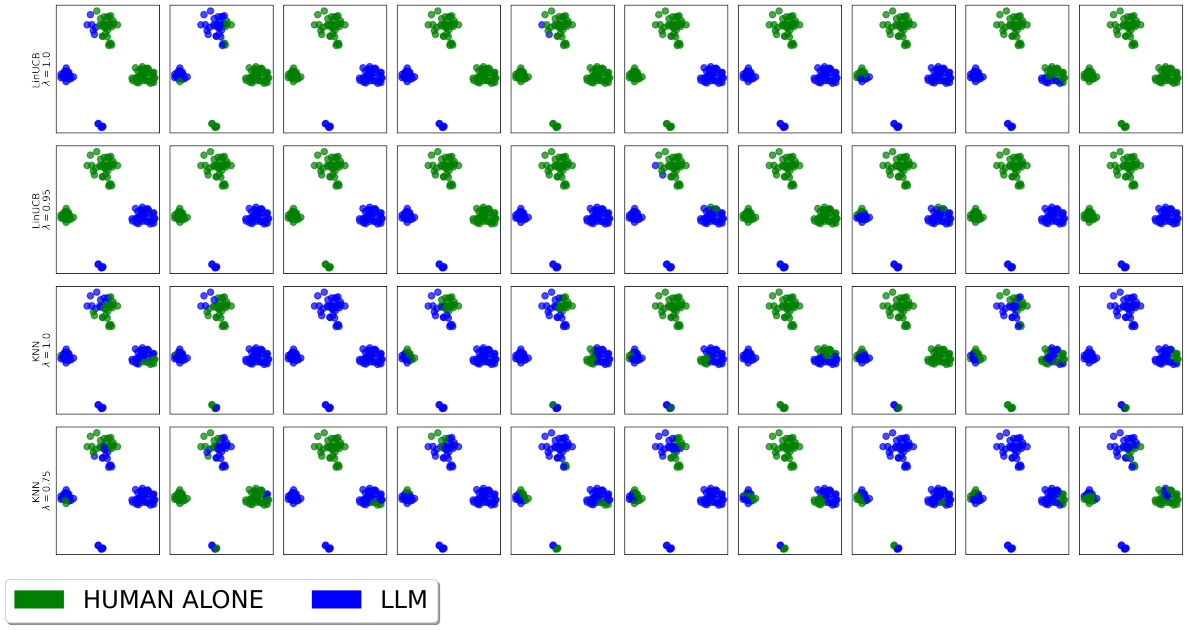


Fig. D.7 Highlighting individual differences in recommended support policies learned via `Modiste` for participants in the MMLU task. Topics are labeled as in Figure 5.3; clockwise from the top: math (M), biology (B), computer science (CS), foreign policy (FP). Adjacent to each policy visualization, we depict each participants' *unassisted* accuracy per topic; i.e. the accuracy for `DEFER` arms. We sample 7 participants out of the 10 per variant. Just as in our other experiments, we also visualize t-SNE embeddings here [Van der Maaten and Hinton, 2008].

Table D.7 We report expected loss $L_h(\pi)$ and expected cost $c(\pi)$ incurred (lower is better) across *all* trials by Prolific participants for each Algorithm and **bold** the variant with the lowest $L_h(\pi)$. `Modiste` learns effective, low-cost policies: for CIFAR, `Modiste` outperforms all baselines. For MMLU, we find that `Modiste` learns a policy with roughly the same performance as the best offline policy but at *half* the cost. We also consider different choices of λ, where $\lambda = 1.0$ corresponds to the standard setting and $\lambda \neq 1.0$ corresponds to a cost-aware setting, where the choice of λ was selected according to Section 5.3.2.

Algorithm	$L_h(\pi)$	$c(\pi)$
H-ONLY	0.57 ± 0.06	0
H-MODEL	0.44 ± 0.04	0.5
H-CONSENSUS	0.32 ± 0.06	0.5
Population	0.6 ± 0.02	0
`Modiste` (LinUCB, $\lambda = 1.0$)	0.28 ± 0.03	0.35 ± 0.01
`Modiste` (KNN, $\lambda = 1.0$)	$\mathbf{0.24 \pm 0.03}$	$\mathbf{0.39 \pm 0.04}$
`Modiste` (LinUCB, $\lambda = 0.85$)	0.37 ± 0.07	0.25 ± 0.01
`Modiste` (KNN, $\lambda = 0.75$)	0.24 ± 0.03	0.4 ± 0.03

Algorithm	$L_h(\pi)$	$c(\pi)$
H-ONLY	0.44 ± 0.07	0
H-LLM	$\mathbf{0.22 \pm 0.06}$	**0.1**
Population	0.34 ± 0.08	0.05 ± 0.00
`Modiste` (LinUCB, $\lambda = 1.0$)	0.26 ± 0.07	0.05 ± 0.00
`Modiste` (KNN, $\lambda = 1.0$)	0.32 ± 0.1	0.06 ± 0.01
`Modiste` (LinUCB, $\lambda = 0.95$)	0.31 ± 0.09	0.06 ± 0.01
`Modiste` (KNN, $\lambda = 0.75$)	0.32 ± 0.1	0.06 ± 0.01

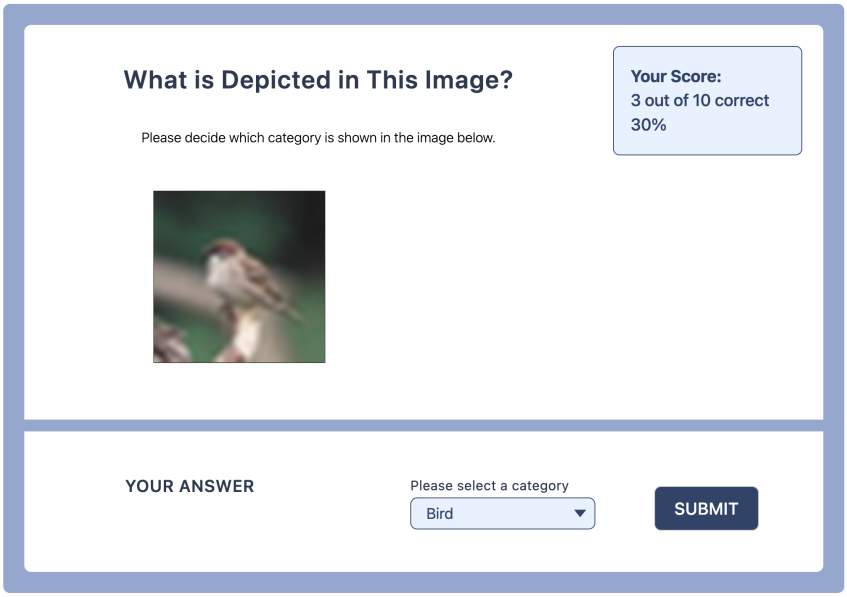


Fig. D.8 Example CIFAR interface for the HUMAN ONLY form of support.

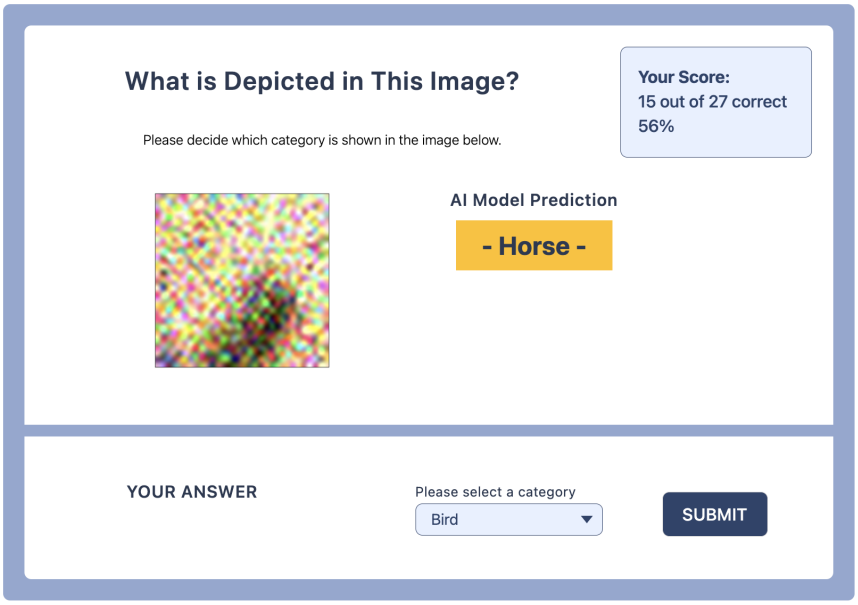


Fig. D.9 Example CIFAR interface for a MODEL form of support. Note that the image shown in this example is corrupted to create settings where the decision-maker needs to calibrate when to rely on each form of support as described in Section 5.4.1.

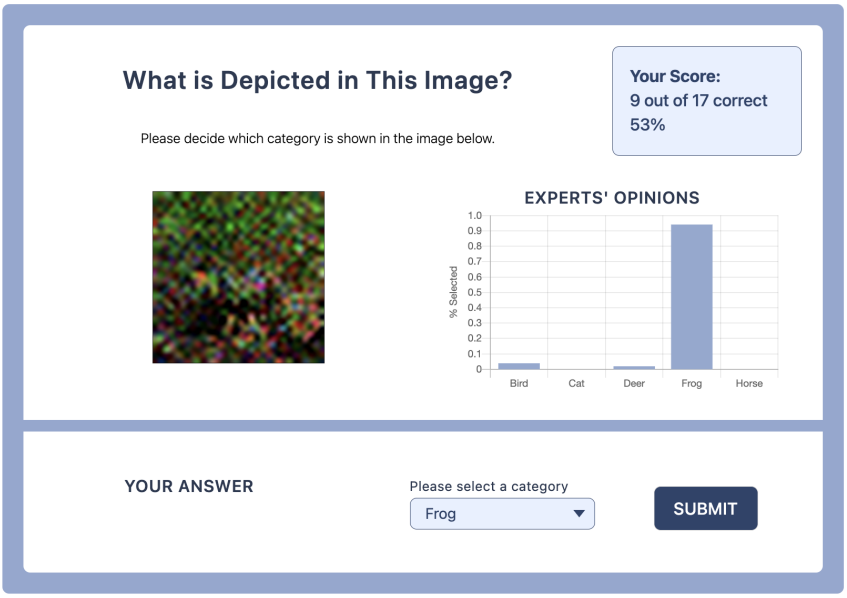


Fig. D.10 Example CIFAR interface for a CONSENSUS form of support, using CIFAR-10H labels. Note that the image shown in this example is corrupted to create settings where the decision-maker needs to calibrate when to rely on each form of support as described in Section 5.4.1.

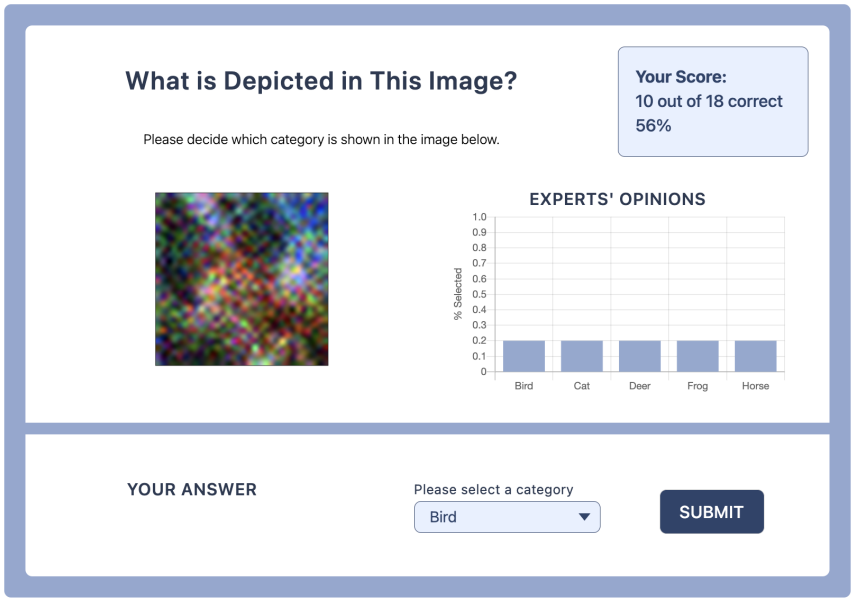


Fig. D.11 Example CIFAR interface for a CONSENSUS form of support when the form of support is intended to be unhelpful.

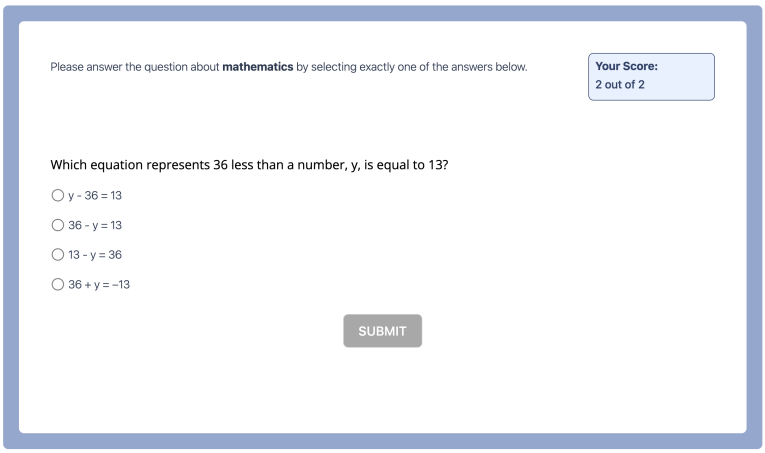


Fig. D.12 Example MMLU interface for the `HUMAN ONLY` form of support.

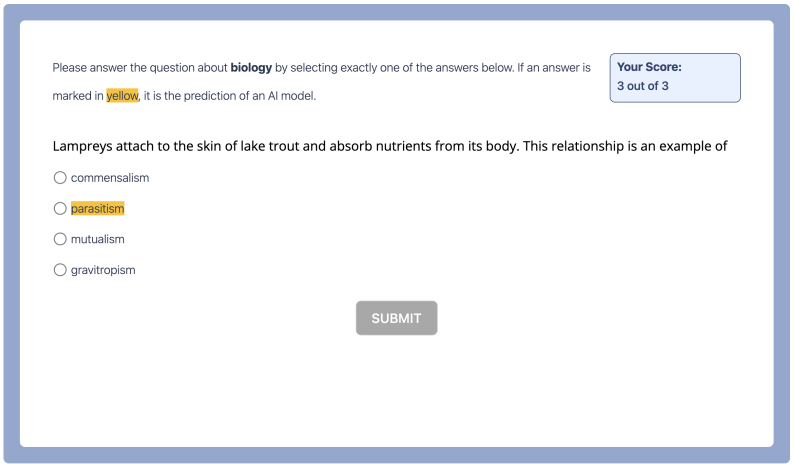


Fig. D.13 Example MMLU interface for the `LLM` form of support.

D.6 Discussion on Reliance

Additional Reliance Investigations

We include further details into participant reliance. We clarify our definition of reliance, and expand our investigation into participant reliance and the incurred cost and loss of the learned policy. We include qualitative responses from users hinting at their mental models of the AI-based support.

Defining "Reliance Sensibility"

We introduced participant reliance in Section 5.6. Specifically, we define a measure of "reliance sensibility": i.e. the proportion of trials the participant agreed with the support when correct and responding differently when the support was incorrect, out of all trials where the participant received support. The higher the proportion, the more "sensible" a participant's degree of reliance on the provided support is, relative to what would be most beneficial for decision outcomes.[4]

Reliance versus Policy Performance

We plot the performance of the learned policy (i.e. incurred loss and cost) against a participants' reliance sensibility in Figures D.14 and D.15, respectively. We in noted a moderate negative correlation between incurred loss and reliance sensibility for MMLU (see Section 5.6). In the CIFAR task, participants largely had calibrated reliance, and such reliance was not a strong predictor of incurred loss for KNN (Pearson $r = -0.1$), though may be for LinUCB (Pearson $r = -0.64$) albeit the minimal variance in reliance obfuscates the possible relationship. We observe no strong relationship between reliance and incurred cost.

Participant Comments Hint at Mental Models of AI

We observed that several participants in our MMLU task noted[5] that their responses were biased by the AI. We include a sampling of comments provided by participants which we found particularly revealing. We believe that these responses motivate the need for further study of how users' mental models of AI systems inform their decision-making performance,

[4]Note that we cannot directly deduce whether, or to what degree, an individual participant relied on the form of support when presented - since we do not have access to what an individually would have said without support; alternative reliance inference schemes like [Tejeda et al., 2022] could be considered in future work.

[5]We included a post-experiment survey where we permitted participants to leave general comments.

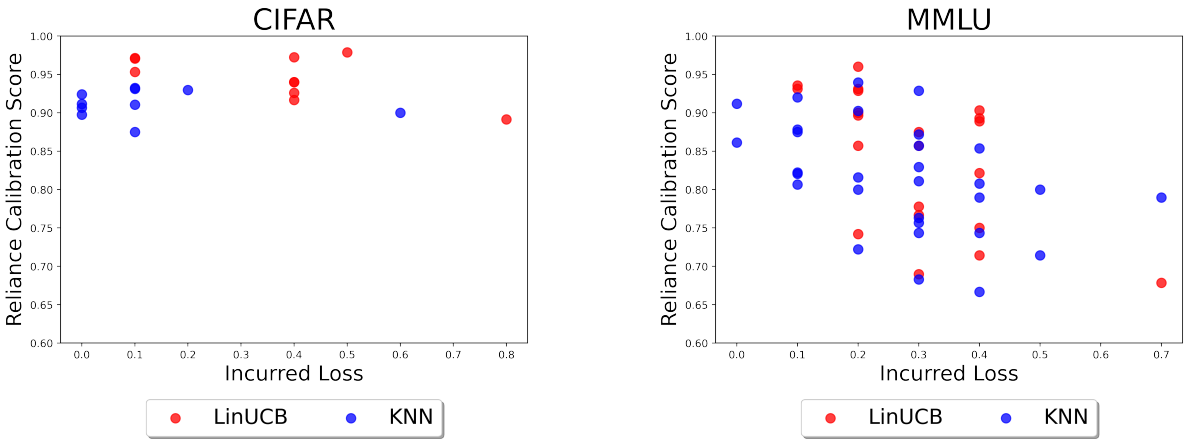


Fig. D.14 Relationship between a participants' sensibility of reliance (measured as the proportion of times they correctly agreed or disagreed with the form of support's prediction) and the loss incurred by the learned policy for the participant. Reliance is computed over all trials (100 for CIFAR, 60 for MMLU), and loss is averaged over the final 10 timesteps. We uncover an inverse relationship between reliance sensibility and the performance of learned policies for MMLU.

particularly in light of the rapidly advancing and public-facing foundation models [Bommasani et al., 2021].

- "I thought the AI answers went against my previous notions about what an AI would be proficient at answering. I thought an AI would excel at anything where there is a set answer such as math, computer science coding question, and biology textbook definitions. Then I expected it to absoutly fail at political, and historical questions. Since they can be more opinionated and not have a definitive answer. However, the AI struggled with math and excelled in the other topics. So I was actively not choosing to agree with the AI on any math where I could also easily check its work and was heavily relying on it for everything else [...] I knew nothing about any question on computer science, so I mainly relied on AI for answers on that topic."
- "AI seemed to me more helpful in topics I didn't know, but sometime I trusted it too much, instead of thinking the answer through"
- "The AI had me second guessing myself sometimes so it affected my answers occasionally."

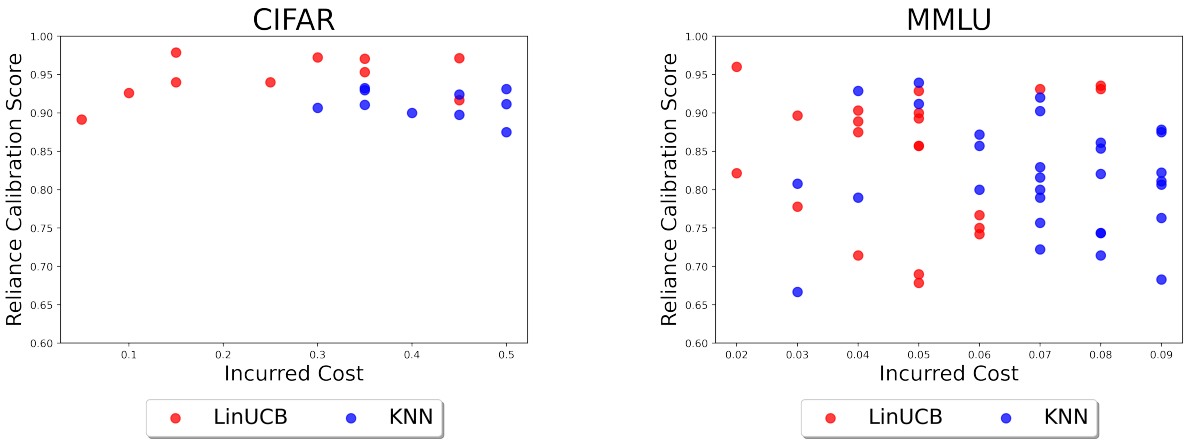


Fig. D.15 There is not a strong relationship between a participants' sensibility of reliance (measured as the proportion of times they correctly agreed or disagreed with the form of support's prediction) and the cost incurred by the learned policy for the participant.

- "`Anything highlighted with the AI I was confident that would be the right answer.`"