Context and Participation in Machine Learning

by

Harini Suresh

B.S., Massachusetts Institute of Technology (2016)
M.Eng., Massachusetts Institute of Technology (2017)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2023

Author .....
Department of Electrical Engineering and Computer Science
January 25, 2023

Certified by .....
John V. Guttag
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by .....
Arvind Satyanarayan
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Context and Participation in Machine Learning

by

Harini Suresh

Submitted to the Department of Electrical Engineering and Computer Science
on January 25, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

ML systems are shaped by human choices and norms, from problem conceptualization to deployment. They are then used in complex socio-technical contexts, where they interact with and affect diverse populations. However, development decisions are often made in isolation, without deeply taking into account the deployment context in which the system will be used. And they are typically hidden to users in that context, who have few avenues to understand if or how they should use the system. As a result, there are numerous examples of ML systems that in practice are harmful, poorly understood, or misused.

We propose an alternate approach to the development and deployment of ML systems that is focused on incorporating the participation of the people who use and are affected by the system. We first develop two frameworks that lend clarity to the human choices that shape ML systems and the broad populations that these systems affect. These inform a prospective question: how can we shape new systems from the start to reflect context-specific needs and benefit justice and equity? We address this question through an in-depth case study of co-designing ML tools to support activists who monitor gender-related violence. Drawing from intersectional feminist theory and participatory design, we develop methods for data collection, annotation, modeling, and evaluation that prioritize sustainable partnerships and challenge power inequalities. Then, we consider an alternative paradigm where we do not have full control over the development lifecycle, e.g., where a model has already been built and made available. In these cases, we show how deployment tools can give downstream stakeholders the information and agency to understand and hold ML systems accountable. We describe the design of two novel deployment tools that provide intuitive, useful, and context-relevant insight into model strengths and

limitations. The first uses example-based visualizations and an interactive input editor to help users assess the reliability of individual model predictions. The second, Kaleidoscope, enables context-specific evaluation, allowing downstream users to translate their implicit knowledge of "good model behavior" for their context into explicitly-defined, semantically-meaningful tests.

This dissertation demonstrates several ways that context-specific considerations and meaningful participation can shape the development and use of ML systems. We hope that this is a step towards the broader goal of building ML-based systems that are grounded in societal context, are shaped by diverse viewpoints, and contribute to justice and equity.

Thesis Supervisor: John V. Guttag
Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Arvind Satyanarayan
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

This thesis is possible because of the immeasurable support, guidance, and friendship I have received from my friends, family, labmates, collaborators, and advisors.

First, I'm very thankful for my thesis committee: John Guttag, Arvind Satyanarayan and Catherine D'Ignazio.

John has been my advisor since I started my PhD in 2017. Throughout the past six years, he has never ceased to be encouraging, patient, and confident in my work. In his advice, John strikes a balance between optimism and constructive guidance that has helped encourage me and make my work better. I'm also grateful for the sense of community and support that he fosters within our lab, which has been invaluable to my PhD experience.

Arvind has been a wonderful co-advisor and mentor. Since we started working together, his investment in my research has shaped it significantly. I'm grateful for Arvind's willingness to meaningfully engage with a breadth of literature in and outside of CS, and working with him has taught me a lot about about doing and communicating interdisciplinary work.

Through working with Catherine, I have learned about running collaborative and interdisciplinary projects, sustaining meaningful partnerships, and doing technical work in a way that's grounded in justice and equity. I am grateful for her trust in letting me shape impactful projects; it has been an invaluable learning experience.

Working with John, Arvind and Catherine has allowed me to be part of three research labs during my time at MIT: the Clinical and Applied Machine Learning group (CAML), the Visualization Group, and the Data + Feminism Lab. Each of these labs feels like a supportive community, and I feel lucky to have been a part of each. I will truly miss the spontaneous conversations, last-minute paper

reading, feedback on talks, coffee breaks, holiday gatherings, and everything else. Thank you Jen, Guha, Amy, Maggie, Adrian, Joel, Davis, Divya, Katie L, Jose, Ani, Emily, Hallee, Katie M, Marianne, Victor, Andrew, Helen, Evan, Crystal, Alan, Nava, Jonathan, Aspen, Angie, Dylan, Ben, Josh, Katie, EJ, Rahul, Silvana, Helena, Eric, Natasha, Isa, Amelia, Mariel, Alessandra, Angeles, Wonyoung, Melissa, Raj, Niki, Giulia!

I'm grateful for the excitement and trust of several undergraduate students I've had the opportunity to mentor: Raj, Annie, Tiff, Helen, and Niki. And thank you to Divya Shanmugam for being a wonderful co-mentor to several.

The work in this thesis is the product of many cross-disciplinary and collaborative efforts. These collaborations have significantly shaped my work, broadened my knowledge, and impacted my future goals. The collaborators who contributed to the work in each chapter of the thesis include:

- Chapter 2: John Guttag
- Chapter 3: Kevin Nam, Steven Gomez, & Arvind Satyanarayan
- Chapter 4: Rajiv Movva, Amelia Dogan, Rahul Bhargava, Isadora Cruxên, Ángeles Martinez Cuba, Giulia Taurino, Wonyoung So, Catherine D'Ignazio, the Sovereign Bodies Institute, & the African American Policy Forum
- Chapter 5: Katie Lewis, John Guttag & Arvind Satyanarayan
- Chapter 6: Divya Shanmugam, Tiffany Chen, Annie Bryan, Alexander D'Amour, John Guttag & Arvind Satyanarayan

My work with the Data Against Feminicide initiative is described in Chapter 4, and I am thankful for the other contributors to that broader initiative who I have had

the chance to work with and learn from: Helena Suárez Val (Feminicidio Uruguay), Dawn Wilcox (Women Count USA), Silvana Fumega (ILDA), and each of the activist organizations interviewed, for their time, labor, care and dedication to righting the balance of justice.

I've also been part of several rewarding collaborative projects that are not included in the thesis, but which have undoubtedly influenced my work, and I deeply appreciate each of those collaborators as well: Jen Gong, Natalie Lao, Ilaria Liccardi, Susanne Guabe, Errol Colak, Marzyeh Ghassemi, Angie Boggust, Aspen Hopkins, Willie Boag, Bianca Lepe.

My friends have have been there through many ups and downs of the past many years, and have filled that time with support and adventures and community. Tiffany, Anne, Elliot, Natasha, Divya, Hansa, Mihika, Frankie, Kamilla, Laura, Aspen, Angie, Katie, Krisli, Davide, Natalie — I feel very lucky to have you all in my life!

Finally, I am immeasurably thankful for the love and support of my family which started long before I begun my PhD — my parents and my brothers Ravi and Sudarshan, as well as my grandparents, aunts, uncles, and cousins who have always been ready to support me on a moment's notice.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

ML systems are increasingly used across a variety of domains where they influence consequential decisions. In medicine, ML systems aim to aid with early disease detection, treatment decisions, or outcome prediction [173, 317]. In hiring, automated screening tools analyze video interviews or questionnaire answers to filter applicants [272]. In criminal justice, actuarial risk assessment tools provide scores to judges that inform bail decisions [305, 364]. In content moderation, automated tools filter comments by their likelihood of containing offensive speech [282]. In social services, algorithms allocate housing resources or predict which children may be victims of neglect or abuse [113].

Today, there is growing awareness of harmful consequences that can arise from such systems. Recent work has shown, for example, how ML systems have systematically deprioritized Black patients in the medical system [250], reinforced racist stereotypes via search engines results and advertising [246], or overlooked resumes from female job applicants [259]. In other cases, issues arise as a result of the way these systems are used by people. For example, Collins [71] describes the *off-label* use

of risk assessment tools deployed in Kentucky's criminal justice system. The term "off-label" is borrowed from medicine, where it indicates a drug being used in a way that has not been approved by the FDA. In the criminal justice example, the tools were developed to guide decisions about how to rehabilitate inmates during incarceration. However, in practice, they were also used to influence *sentencing* decisions, potentially justifying increased incarceration on the basis of personal characteristics.

While people also make mistakes or have implicit biases, machine learning systems can cause larger-scale and more systematic consequences, because they can inflict the same kind of harm "rapidly and repeatedly" [60]. Moreover, while there are avenues to contest or question human decision-makers, the broader population affected by ML systems are often left without effective means to understand, probe, or push back on important decisions [34, 328].

In this dissertation, we frame the harmful consequences of ML systems as a failure to *consider context*. Specifically, there are many *development decisions* — e.g., problem definition, sampling and annotation methods, categorization schemes, optimization functions — that are typically made without deeply considering the needs of the *deployment contexts* in which the system will be used. Different deployment contexts — e.g., geographies, populations, institutions, physical or virtual communities — have different norms, goals, and data distributions, and therefore, desired model behavior differs across them. For example, an annotation method that involves crowd workers labeling social media data does not meaningfully consider online communities where phrases or emojis take on different, context-specific meanings [260]. Or, choosing to use past diagnoses as ground-truth training labels for a diagnostic system does not meaningfully consider a context where the system will be used to predict a condition that has been systematically under-diagnosed.

This disconnect is exacerbated by the fact that there are often few avenues for

downstream populations to understand or question ML systems. For example, the typical user-facing system might simply display a probability score or the model's prediction. Abstraction is useful, and if a system is working well, we do not need to overload people with unnecessary information. However, when there *is* a concern, we are currently lacking useful deployment tools — e.g., visualizations, guidance for users, ways of probing the model, fine-grained evaluations — to surface and dig into context-relevant issues.

Together, these two issues — development decisions being disconnected from deployment contexts, and the implications of those decisions being hidden — can lead to harmful systems with few avenues for questioning or push back (Figure 1-1).

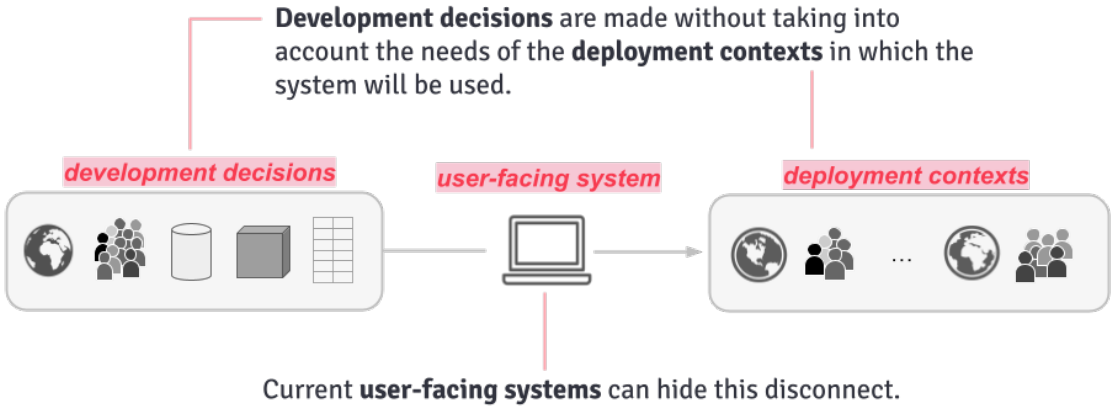


Figure 1-1: We describe two problems that, together, can lead to harmful downstream consequences of ML systems. First, development decisions (e.g., sampling, categorization schemes, annotation methods, optimization functions, et cetera) are often made without taking into account the deployment context(s) in which the system will be used. Second, the user-facing system that people in deployment contexts see often hides that disconnect, failing to surface potential issues and leaving those populations with few avenues for questioning or pushback.

My thesis address both of these issues. First, we ask: when we approach building new systems, how can we shape development decisions to be better suited to down-

stream deployment contexts? we then ask: for existing systems, how can we design deployment tools that communicate useful and relevant information about model limitations to people in a particular deployment context?

My overall approach to addressing these questions is centered around increasing the participation of broader populations — particularly those who interact with and are affected by a system in a particular deployment context — throughout the development, evaluation and use of ML systems (Figure 1-2).

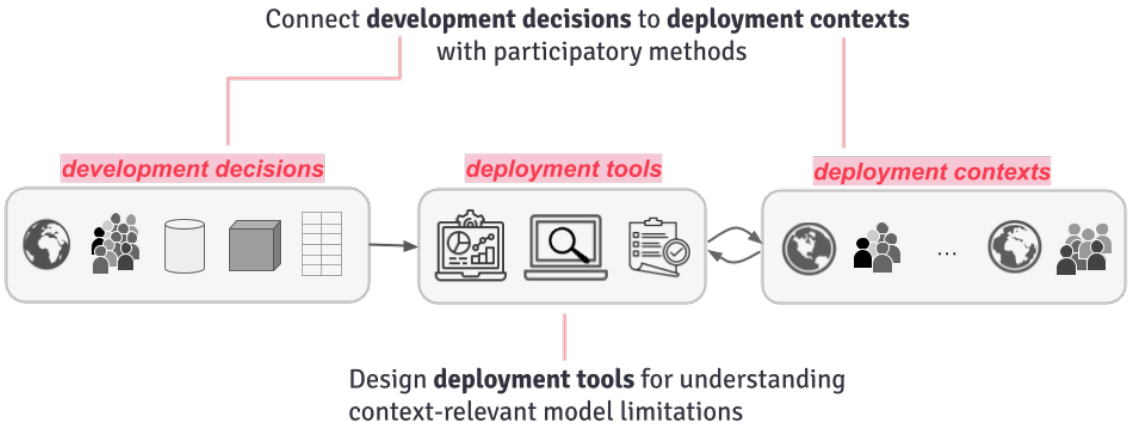


Figure 1-2: In the thesis, we address the two issues described in Figure 1-1 through increasing participation at different parts of the ML lifecycle. First, we show how, with participatory design, we can make development decisions in a way that is grounded in the needs of downstream contexts and stakeholders. Then, we show how we can design deployment tools that work with downstream users to surface semantically-meaningful, context-relevant insights into model strengths and limitations.

First, Chapter 2 provides a framework to better understand the many development decisions that are made throughout the ML lifecycle, and their downstream implications. Chapter 3 switches to focus on deployment contexts, characterizing the diverse populations who interact with ML systems and the heterogeneity of their needs.

Chapter 4 then considers the case of conceptualizing and building a new system, asking the more prospective question of how each development decision could be shaped to serve specific downstream contexts. The case study we consider is one where the team developing the system (including myself) has a large amount of control to shape the development process, and a collaborative relationship with downstream stakeholders across different deployment contexts. In this ideal scenario, we show how ML systems can be designed from the start with participatory methods to reflect context-specific needs as well as benefit justice and equity.

However, this process also requires significant time, effort and customization to design (or to update, if/when user needs evolve over time). Increasingly, existing datasets or models are publicly available and ready to use in different applications or systems. In these cases, we argue that we can still intervene at deployment, building *deployment tools* that give downstream stakeholders the tools and agency to understand context-relevant strengths and limitations of an ML system.

In Chapters 5 and 6, we describe the design, implementation and evaluation of two such deployment tools that work with users in deployment contexts to surface relevant insights into an existing ML system. The interactive mechanisms and visualizations in these tools are designed to align with users' existing conceptual models of their context and enable more direct and intuitive analyses. we show how certain techniques—e.g., grounding model evaluation in real data with which users are familiar, or semantically grouping examples in a learned embedding space—help facilitate insights that are grounded in higher-level context-relevant concepts. We evaluate both systems with real-world case studies and users, while also distilling the broader conceptual workflows and methods that are widely applicable to a range of application domains.

1.1 Overview

This section provides a more in-depth description of the work in the dissertation.

1.1.1 Frameworks for formalizing development decisions and deployment contexts

Chapter 2 presents a novel framework of the development decisions that are made throughout the ML lifecycle. For each kind of decision, we illustrate downstream implications it can have and ways in which, if not suited to a deployment context, it can cause harm. The framework is shaped by two key observations: 1) data is not a pre-existing artifact, but the product of a complex process driven by human choices and norms, and 2) there are many other choices outside of the data (e.g., model definition and deployment decisions) that have important implications. As a result, the framework's depiction of choices throughout the ML lifecycle is more granular and comprehensive than prior work, which often groups issues with the data into one category ("dataset bias") or excludes decisions about things such as model architecture or output visualizations. The framework helps us understand the context and choices that are often hidden when a system is deployed, and informs analyses and design in the chapters that follow.

Chapter 3 explores how to design for specific deployment contexts by focusing on the people who interact with and are affected by an ML system. We introduce a framework that characterizes these diverse stakeholders. The framework is comprised of two components. First, it decomposes stakeholder expertise into two dimensions that describe the types of knowledge a stakeholder may possess (i.e., formal, instrumen-

tal, and personal knowledge), and the contexts in which this knowledge manifests (i.e., machine learning, the data domain, and the milieu). Second, it distills stakeholder needs using a three-level typology of long-term goals (e.g., building trust in the model), shorter-term objectives that target these goals (e.g., justifying actions influenced by the model), and immediate tasks that stakeholders perform to meet their objectives (e.g., assessing prediction reliability). While prior frameworks often infer users' needs from their expertise (e.g., ML experts need to debug models), this chapter generates a rich *intersection* of users and design needs. We use the framework to analyze existing work, revealing groups of stakeholders (e.g., people with deep personal knowledge/lived experience) who are underserved by existing deployment tools, and explore how this might inform future designs.

1.1.2 Prospectively incorporating context with participatory methods

Chapter 4 considers the scenario where a new system is being conceptualized, and asks how each step of the ML lifecycle can be proactively shaped to fit the deployment context. We also ask how these systems, beyond simply serving their intended context, can actively benefit justice and equity. To approach these prospective design questions, we draw from the theoretical framework of intersectional feminism, which provides a conceptual model for how inequality is structure and reinforced. This conceptual model informs several concrete design principles (e.g., embrace pluralism or challenge power), methods (e.g., sustained participatory design), and choices (e.g., prioritizing the participation of groups who sit at the intersection of multiple forms of oppression).

We explore this approach through a case study of co-designing datasets and

ML models to support the efforts of activists who collect and monitor data about feminicide—gender-based killings of women and girls. We characterize different feminicide monitoring organizations as distinct contexts that sit at the intersection of different systems of power and oppression. We then use the framework of data feminism [95] as guide for how to conduct each stage of the ML lifecycle in a way that centers these different contexts. We illustrate several resulting methodological contributions, including 1) a data collection process in which we iteratively identify and incorporate context-specific examples that target model weaknesses, 2) annotation and modeling methods that incorporate sociohistorical context and explicitly focus on intersectional identities, and 3) a three-stage evaluation process—with quantitative, qualitative and participatory steps—focused on real-world, context-specific usefulness.

1.1.3 Designing better deployment tools

While the case study in Chapter 4 considers a scenario where there is a large amount of control over the whole ML lifecycle, this is often not the case in practice. Collecting high-quality, representative datasets can be prohibitively difficult or time-consuming, and pre-trained models are increasingly available. Or, we might develop a system for a particular context as in Chapter 4, but the needs of the context might change over time.

In these cases, current user-facing systems leave people in a specific deployment context without the information or agency to understand if a particular system meets their needs, and how it should be used. Chapters 5 and 6, propose deployment tools to bridge this gap. We present two novel systems that provide useful, intuitive insights into model strengths and limitations tailored for specific groups of users.

Chapter 5 presents two interface modules that help users with formal domain knowledge intuitively assess a model's reliability on a case-by-case basis. To help users better characterize and reason about a model's uncertainty, the system displays raw and aggregate information about a given input's nearest neighbors. Using an interactive editor, users can manipulate this input in semantically-meaningful ways, determine the effect on the output, and compare against their prior expectations. In a user study with physicians, participants were better able to align the model's uncertainty with domain-relevant factors and build intuition about its capabilities and limitations (as compared to a baseline feature importance visualization).

Chapter 6 presents Kaleidoscope, a system for performing user-driven, context-specific evaluations of ML models. Kaleidoscope helps users translate implicit knowledge of "good model behavior" for their context into explicitly defined tests. Using Kaleidoscope's iterative workflow, users *identify* important examples using data from a chosen context, *generalize* them into semantically-meaningful concepts, and *specify and test* model behavior on those concepts. This workflow enables a bottom-up, iterative, and exploratory approach–starting with a small number of concrete examples and then iteratively generalizing to a larger set—rather than a top-down one that requires formally defining sets of interest upfront. In doing so, it enables users with personal knowledge to understand models in terms of concepts that are more conceptually-meaningful to their context than existing methods of grouping examples.

1.1.4 Conclusion

Chapter 7 summarizes the contributions of the thesis and discusses areas for future work to explore.

1.2 Contributions

In summary, the contributions of this thesis include:

- A novel framework of development decisions and their implications throughout the machine learning lifecycle [310]. The framework provides a terminology and conceptual model to tease apart problems that have different underlying sources, which then informs if and when different mitigation techniques are appropriate. The framework provides a granular and comprehensive view on where harm can arise throughout the ML lifecycle, allowing us to recognize sources of harm that may have previously been overlooked.

- A novel framework characterizing the stakeholders of ML systems [313]. The framework highlights stakeholders that may be underserved with existing deployment tools — e.g., users with deep personal knowledge — as well as the heterogeniety of their needs. The framework lends clarity to how we might start to design systems and deployment tools to better serve broader populations.

- A case study of co-designing context-specific datasets and ML models for activists who monitor gender-based violence [315]. This work serves as a first example of translating sustained participation "based on mutual benefit, reciprocity, equity and justice" [300] to the context of ML—from problem conceptualization to dataset collection to system evaluation.

- A system that helps users with formal domain knowledge understand model reliability on a case-by-case basis, via example-based visualizations and an interactive input editor [314]. In contrast to existing methods that describe model behavior in terms of low-level features, the system helps users to reason

more intuitively about model relability in terms of high-level, domain-relevant concepts.

- Kaleidoscope, a system that facililtates context-specific, semantically-meaningful and user-driven evaluation [316]. Kaleidoscope allows users to answer questions about model behavior in terms of high-level concepts that would be difficult or impossible to answer with standard evaluation methods.

Together, this work demonstrates many ways in which context-specific considerations and meaningful participation can shape the development and use of ML systems.

Chapter 2

Sources of Harm throughout the ML Lifecycle

In this chapter, we identify important choices throughout the ML lifecycle and the downstream implications they can have. In particular, we highlight the harm that can arise when these choices are made without adequately taking into account the context in which the system will be used. In later chapters, we discuss how to "take into account" deployment contexts, and develop methods and systems to do so.

2.1 Introduction

A common refrain is that undesirable behaviors of ML systems happen when "data is biased." Indeed, a recent comment by a prominent ML researcher[1] set off a heated debate — not necessarily because the statement "data is biased" is *false*, but because it treats data as a static artifact divorced from the process that produced it. This

[1] https://twitter.com/ylecun/status/1274782757907030016?s=20

process is long and complex, grounded in historical context and driven by human choices and norms. Understanding the implications of each stage in the data generation process can reveal direct and meaningful ways to prevent or address harmful downstream consequences.

And it is not just the data. The ML pipeline involves a series of choices and practices, from model definition to user interfaces. Each stage involves decisions that can lead to undesirable effects. For an ML practitioner working on a new system, it is not straightforward to identify if and how problems might arise. Even once identified, it is not clear what the appropriate application- and data-specific mitigations might be, or how they might generalize over factors such as time and geography.

Consider the following simplified scenario: a medical researcher wants to build a model to help detect whether someone is having a heart attack. She trains the model on medical records from a subset of prior patients at a hospital, along with labels indicating if and when they suffered a heart attack. She observes that the system has a higher false negative rate for women (it is more likely to miss cases of heart attacks in women), and hypothesizes that the model was not able to effectively learn the signs of heart attacks in women because of a lack of examples. She seeks out additional data representing women who experienced heart attacks to augment the dataset, re-trains the model, and observes that the performance for female patients improves. Meanwhile, a co-worker hiring new lab technicians tries to build an algorithm for predicting the suitability of a candidate, using a sample of resumes and human-assigned ratings as training data. He notices that women are much less likely to be predicted as suitable candidates than men. Like his colleague, he tries to collect many more samples of women to add to the dataset, but is disappointed to see that the model's behavior does not change. Why did this happen? The *sources* of the

disparate performance were different. In the first case, it arose because of a lack of data on women, and introducing more data was helpful. In the second case, using human assessment of quality as a label to estimate true qualification allowed the model to discriminate by gender, and collecting more labelled data from the same distribution did not help.

This chapter provides a framework and vocabulary for understanding distinct *sources* of downstream harm from ML systems. This framework can be used in different ways by a variety of stakeholders, including those who build, evaluate, use, or are affected by ML systems. We demonstrate how issues arise in distinct stages of the ML life cycle, and provide corresponding terminology that avoids overly broad and/or overloaded terms. Doing so facilitates a methodical analysis of the risks of a particular system, and can inform appropriate if and how different mitigation techniques are appropriate. The framework can also help practitioners anticipate issues and design more thoughtful and contextual methods for data collection, development, evaluation, and/or deployment. Beyond those involved in model development, an understanding of how and why issues arise throughout the ML life cycle can provide a valuable guide for external stakeholders, such as regulators or affected populations.

Throughout the chapter, we refer to the concept of "harm" or "negative consequences" caused by ML systems. Barocas *et al.* [22] provide a useful framework for thinking about how these consequences actually manifest, splitting them into *allocative* harms (when opportunities or resources are withheld from certain people or groups) and *representational* harms (when certain people or groups are stigmatized or stereotyped). For example, algorithms that determine whether someone is offered a loan or a job [86, 272] risk inflicting allocative harm. This is typically the type of harm that we think and hear about, because it can be measured and is more commonly recognized as harmful. However, even if they do not directly with-

hold resources or opportunities, systems can still cause representational harm; e.g., language models that encode and replicate stereotypes.

Section 2.2 follows with a brief overview of the ML pipeline that will be useful background information as we refer to different parts of this process. Section 2.3 details each source of harm in more depth with examples. Section 2.4 provides a more rigorous presentation of our framework for formalizing and mitigating the issues we describe. Finally, Section 2.5 is a brief conclusion.

2.2 Machine Learning Overview

Machine learning is a type of statistical inference that learns, from existing data, a function that can be generalized to new, unseen data. ML algorithms make personalized Netflix or YouTube recommendations, power Siri's stilted conversation, provide live transcriptions on our video calls, auto-tag the people in our photos, decide whether we are offered job interviews, and approve (or not) tests at the doctor's office. In each of these examples, an ML algorithm has found patterns in a (usually massive) dataset, and is applying that knowledge to make a prediction about new data points (which might be photos, medical records, resumes, etc.).

In this section, we briefly describe the typical life cycle of an ML system. We describe each step generally, as well as how it might occur in a running hypothetical example: a machine learning-based loan-approval system. In the running example, we describe each step as it typically happens (not necessarily as it *should*). In the next section, we analyze the implications of each step and problems that may be introduced. Figure 2-1 depicts these steps. Later, in Section 2.4, we provide a more rigorous formalization of these steps.

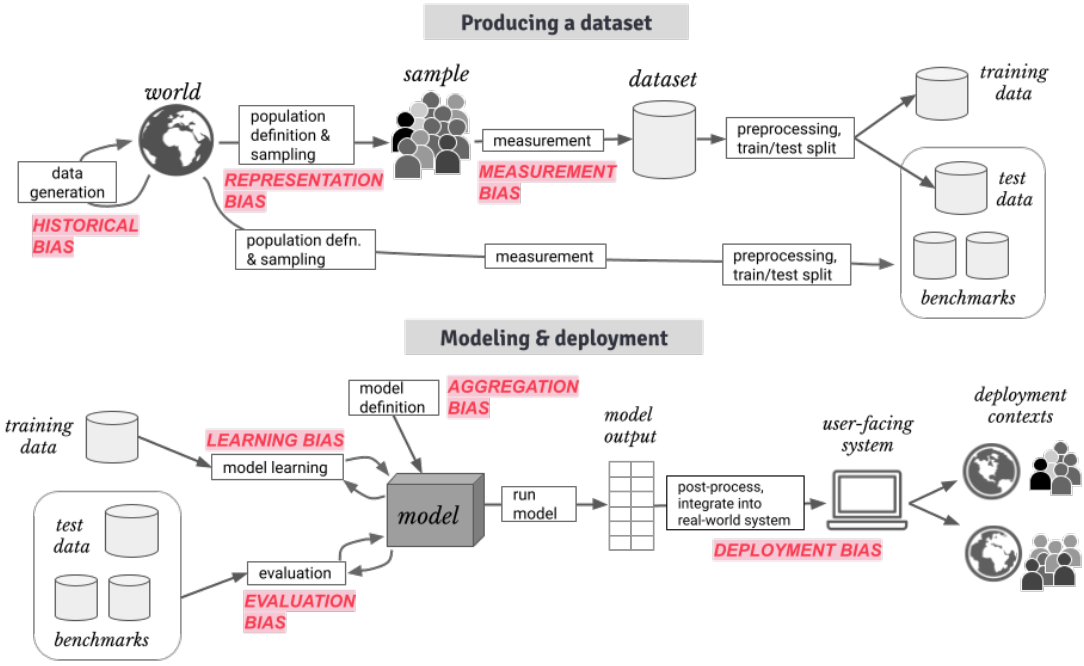


Figure 2-1: Top half: The data generation process begins with data collection. This process involves defining a target population and sampling from it, as well as identifying and measuring features and labels. This dataset is split into training and test sets. Data is also collected (perhaps by a different process) into benchmark datasets. Bottom half: A model is defined, and optimized on the training data. Test and benchmark data is used to evaluate it, and the final model is then integrated into real-world context(s). This process is naturally cyclic, and decisions influenced by models affect the state of the world that exists the next time data is collected or decisions are applied. In red, we indicate where in this pipeline different sources of downstream harm might arise.

Data Collection

Before any analysis or learning happens, data must first be collected. Compiling a dataset involves identifying a *target population* (of people or things), as well as defining and measuring *features* and *labels* from it. Typically, it is not feasible to include the entire target population, and instead, features and labels are sampled

from a subset of it (here, we refer to this subset as the *development sample*). Often, ML practitioners use existing datasets rather than going through the data collection process.

Example. For a loan approval system, a team in charge of data collection could choose the target population to be people who live in the state in which the system will be used, people who have previously applied for loans, people with credit cards, etc. The sample that ends up in the dataset will be a subset of this target population and will depend upon the sampling method (e.g., sourcing information from public records or surveying people). There is also the question of what to actually measure or collect about these people: perhaps things like their debt history, the number of credit cards they have, their income, their occupation, etc. Some of these variables will be chosen to serve as labels: for example, information about whether the person received and/or paid back a loan in the past.

Data Preparation

Depending on the data modality and task, different types of preprocessing may be applied to the dataset before using it. Datasets are usually split into *training data* used during model development, and *test data* used during model evaluation. Part of the training data may be used as *validation data* that will be used to compare different modeling techniques or hyperparameter choices against each other.

Example. For the loan approval system, preprocessing might involve dealing with missing data (e.g., imputing missing credit history values via interpolation), simplifying the feature space (e.g., grouping occupations in broader categories like "physician" rather than encoding detailed specialities), or normalizing continuous measurements (e.g., scaling income so it lies on a 0-to-1 scale). If a resulting dataset

included 1000 examples (e.g., data collected from 1000 people), 600 examples might be used for training, 100 as a validation set during training, and 300 for post-development testing.

Model Development

Models are then built using the training data (not including the held-out validation data). Typically, models are trained to optimize a specified *objective*, such as minimizing the mean squared error between its predictions and the actual labels. A number of different model types, hyperparameters, and optimization methods may be tried at this point; usually these different configurations are compared based on their performance on the validation data, and the best one chosen.

Example. The team developing the loan approval model would first need to instantiate a particular model (e.g., a dense, feed-forward neural network) and define an objective function (e.g., minimizing the cross-entropy loss between the model's predictions and the label defined in the training data). Then, in the optimization process, the model will try to learn a function that goes from the inputs (e.g., income, occupation, etc.) to the output (e.g., whether the person paid back a previous loan). They might also train a number of different models (e.g., with varying architectures or training procedures) and choose the one that performs best on the validation set.

Model Evaluation

After the final model is chosen, the performance of the model on the test data is reported. The test data is not used before this step, to ensure that the model's performance is a true representation of how it performs on unseen data. Aside from the test data, other available datasets — also called *benchmark datasets* — may

be used to demonstrate model robustness or to enable comparison to other existing methods. The particular *performance metric(s)* used during evaluation are chosen based on the task and data characteristics.

Example. Here, the model developed in the previous step would be evaluated by its performance on the test set. There might be several performance metrics to consider — for example, loan applicants might be concerned with false negatives (i.e., being denied a loan when they actually are deserving), while lenders might care more about false positives (i.e., recommending loans to people who don't pay them back). In addition, the model might be evaluated on existing datasets used for similar tasks (e.g., the dataset from the U.S. Small Business Association described in Li *et al.* [213]).

Model Postprocessing

Once a model has been trained, there are various post-processing steps that might be needed. For example, if the output of a model performing binary classification is a probability, but the desired output to display to users is a categorical answer, there remains a choice of what threshold(s) to use to round the probability to a hard classification.

Example. The resulting model for predicting loan approval likely outputs a continuous score between 0 and 1. The team might choose to transform this score into discrete buckets (e.g., low-risk of defaulting, unsure, high-risk of defaulting) or a binary recommendation (e.g., should/should not receive a loan).

Model Deployment

There are many steps that arise when deploying a model in a real-world setting. For example, the model may need to be changed based on requirements for explainability or apparent consistency of results, or there might need to be built-in mechanisms to integrate real-time feedback. Importantly, there is no guarantee that the population a model sees as input after it is deployed (here, we will refer to this as the *use population*) looks the same as the population in the development sample.

Example. In order to deploy the loan approval system, the team will likely need to develop a user interface that displays the result and the recommended action. They might need to develop different visualizations of the model's reasoning and results for lenders, applicants, regulatory agencies, or other relevant stakeholders. And they may need to incorporate mechanisms for applicants to seek recourse if they believe the model recommendation was inaccurate or discriminatory.

2.3 Seven Sources of Harm in ML

In this section, we will go into more depth on potential sources of harm. There are several possible organizational principles for creating such a taxonomy. For example, Ntoutsi *et al.* [247] distinguish issues caused by data generation, data collection, or institutional bias; and Mehrabi *et al.* [228] group types of bias based on how they interact with the data, the algorithm, or the user. Here, with the goal of focusing on *sources* of harm, we choose to use the different stages in the ML life cycle for organizational structure; the sources of harm we describe roughly map to the processes described in Figure 1. Each subsection will detail where and how in the ML life cycle problems might arise, as well as a characteristic example. These

categories are not mutually exclusive; however, identifying and characterizing each one as distinct makes them less confusing and easier to tackle.

We use the term "bias" to describe these problems primarily because of precedence, acknowledging that it is a heavily overloaded term that is used to describe a range of issues across different fields. Here, the biases we describe refer to distinct sources of harm in an ML system, and can be thought of as breaking down vague terms like "algorithmic bias" or "data bias" into more useful and granular concepts. Types of bias described in other works might map onto our framework depending on where in the ML life cycle they manifest. For example, "cognitive bias," in crowd annotators of a dataset would fall under the umbrella of our "measurement bias" (Section 2.3.3), because it describes an issue that arises during the process of measuring labels in a dataset. Similarly, Friedman and Nissenbaum's "preexisting bias" [126] might map to our "historical bias" (Section 2.3.1) when it describes existing societal stereotypes that are reflected in datasets.

2.3.1 Historical Bias

Historical bias arises even if data is perfectly measured and sampled, if the world *as it is* or *was* leads to a model that produces harmful outcomes. Such a system, even if it reflects the world accurately, can still inflict harm on a population. Considerations of historical bias often involve evaluating the representational harm (such as reinforcing a stereotype) to a particular group.

Example: Word Embeddings

Word embeddings are learned vector representations of words that encode semantic meaning, and are widely used for natural language processing (NLP) applications.

Recent research has shown that word embeddings, which are learned from large corpora of text (e.g., Google news, web pages, Wikipedia), reflect human biases. One such study [130] demonstrates that word embeddings reflect real-world biases about women and ethnic minorities, and that an embedding model trained on data from a particular decade reflects the biases of that time. For example, gendered occupation words like "nurse" or "engineer" are highly associated with words that represent women or men, respectively. A range of NLP applications (e.g., chatbots, machine translation, speech recognition) are built using these types of word embeddings, and as a result can encode and reinforce harmful stereotypes.

2.3.2 Representation Bias

Representation bias occurs when the development sample under-represents some part of the population, and subsequently fails to generalize well for a subset of the use population. Representation bias can arise in several ways:

1. **When defining the target population, if it does not reflect the use population.** Data that is representative of Boston, for example, may not be representative if used to analyze the population of Indianapolis. Similarly, data representative of Boston 30 years ago will likely not reflect today's population.

2. **When defining the target population, if it contains under-represented groups.** Say the target population for a particular medical dataset is defined to be adults aged 18-40. There are, of course, minority groups within this population. For example, people who are pregnant may make up only 5% of the target population. Even we sample perfectly, and even if the use population is the same (adults 18-40), the model will likely be less robust for those 5% of pregnant people because it has less data to learn from.

3. **When sampling from the target population, if the sampling method is limited or uneven.** For example, the target population for modeling an infectious disease might be all adults, but data might be available only for the sample of people whose condition was considered severe enough to bring in for further screening. As a result, the development sample will represent a skewed subset of the target population. In statistics, this is typically referred to as *sampling bias*.

Example: Geographic Diversity in Image Datasets

ImageNet is a widely-used image dataset consisting of 1.2 million labeled images [89]. ImageNet is intended to be used widely (i.e., its target population is "all natural images"). However, ImageNet does not evenly sample from this target population. Approximately 45% of the images in ImageNet were taken in the United States, and the majority of the remaining images are from the rest of North America or Western Europe. Only 1% and 2.1% of the images come from China and India, respectively. As a result, Shankar *et al.* [296] show that the performance of a classifier trained on ImageNet is significantly worse at classifying images containing certain objects or people (such as "bridegroom") when the images come from under-sampled countries such as Pakistan or India.

2.3.3 Measurement Bias

Measurement bias occurs when choosing, collecting, or computing features and labels to use in a prediction problem. Typically, a feature or label is a *proxy* (a concrete measurement) chosen to approximate some *construct* (an idea or concept) that is not directly encoded or observable. For example, "creditworthiness" is an abstract

construct that is often operationalized with a measureable proxy like a credit score. Proxies become problematic when they are poor reflections or the target construct and/or are generated differently across groups, which can when:

1. **The proxy is an oversimplification of a more complex construct.** Consider the prediction problem of deciding whether a student will be successful (e.g., in a college admissions context). Fully capturing the outcome of "successful student" in terms of a single measurable attribute is impossible because of its complexity. In cases such as these, algorithm designers may resort to a single available label such as "GPA" [190], which ignores different indicators of success present in different parts of the population.

2. **The method of measurement varies across groups.** For example, consider factory workers at several different locations who are monitored to count the number of errors that occur (i.e., observed number of errors is being used as a proxy for work quality). If one location is monitored much more stringently or frequently, there will be more errors observed for that group. This can also lead to a feedback loop wherein the group is subject to further monitoring because of the apparent higher rate of mistakes [21, 111].

3. **The accuracy of measurement varies across groups.** For example, in medical applications, "*diagnosed* with condition X" is often used as a proxy for "has condition X." However, structural discrimination can lead to systematically higher rates of misdiagnosis or underdiagnosis in certain groups [237, 266, 162]. For example, there are both gender and racial disparities in diagnoses for conditions involving pain assessment [57, 161].

Example: Risk Assessments in the Criminal Justice System

ML-based risk assessment tools have been deployed at several points within criminal justice settings [157]. For example, risk assessments such as Northpointe's COMPAS predict the likelihood that a defendant will re-offend, and may be used by judges or parole officers to make decisions around pre-trial release [11]. The data for models like these often include proxy variables such as "arrest" to measure "crime" or some underlying notion of "riskiness." Because minority communities are more highly policed, this proxy is *differentially mismeasured* — there is a different mapping from crime to arrest for people from these communities. Many of the other features used in COMPAS (e.g., "rearrest" to measure "recidivism" [102]) were also differentially measured proxies. The resulting model had a significantly higher false positive rate for black defendants versus white defendants (i.e., it was more likely to predict that black defendants were at a high-risk of reoffending when they actually were not).

2.3.4 Aggregation Bias

Aggregation bias arises when a one-size-fits-all model is used for data in which there are underlying groups or types of examples that should be considered differently. Underlying aggregation bias is an assumption that the mapping from inputs to labels is consistent across subsets of the data. In reality, this is often not the case. A particular dataset might represent people or groups with different backgrounds, cultures or norms, and a given variable can mean something quite different across them. Aggregation bias can lead to a model that is not optimal for any group, or a model that is fit to the dominant population.

Example: Social Media Analysis.

Patton *et al.* [260] describe analyzing Twitter posts of gang-involved youth in Chicago. By hiring domain experts from the community to interpret and annotate tweets, they were able to identify shortcomings of more general, non-context-specific NLP tools. For example, certain emojis or hashtags convey particular meanings that a nonspecific model trained on all Twitter data would miss. In other cases, words or phrases that might convey aggression elsewhere are actually lyrics from a local rapper [124]. Ignoring this group-specific context in favor of a single, more general model built for all social media data would likely lead to harmful misclassifications of the tweets from this population.

2.3.5 Learning Bias

Learning bias arises when modeling choices amplify performance disparities across different examples in the data [168]. For example, an important modeling choice is the objective function that an ML algorithm learns to optimize during training. Typically, these functions encode some measure of accuracy on the task (e.g., cross-entropy loss for classification problems or mean squared error for regression problems). However, issues can arise when prioritizing one objective (e.g., overall accuracy) damages another (e.g., disparate impact) [189]. For example, minimizing cross-entropy loss when building a classifier might inadvertently lead to a model with more false positives than might be desirable in many contexts.

Example: Optimizing for Privacy or Compactness.

Recent work has explored training models that maintain *differential privacy* (i.e., preventing them from inadvertently revealing excessive identifying information about

the training examples during use). However, Bagdasaryan *et al.* [16] show that differentially private training, while improving privacy, reduces the influence of underrepresented data on the model, and subsequently leads to a model with worse performance on that data (as compared to a model without differentially private training). Similarly, Hooker *et al.* [169] show how prioritizing compact models (e.g., with methods such as *pruning*) can amplify performance disparities on data with underrepresented attributes. This happens because, given limited capacity, the model learns to preserve information about the most frequent features.

2.3.6 Evaluation Bias

Evaluation bias occurs when the ways in which a model is evaluated (e.g., the benchmark datasets or metrics used) hide harmful effects. Evaluation bias can shape research directions, encouraging the development and deployment of models that continue to propagate those harmful effects [91].

Evaluation bias often arises because of a desire to quantitatively compare models against each other. Applying different models to a set of external datasets attempts to serve this purpose, but is often extended to make general statements about how good a model is. Such generalizations are often not statistically valid [286], and can lead to overfitting to a particular benchmark. This is especially problematic if the benchmark suffers from historical, representation or measurement bias.

The metrics used to report performance can also contribute to evaluation bias. For example, aggregate measures can hide subgroup underperformance, but single measures are often used because they make it more straightforward to compare models and make a judgment about which one is "better." Looking at only one type of metric (e.g., accuracy) can also hide other types of errors (e.g., a high false negative

rate).

Example: Commercial Facial Analysis Tools

Buolamwini and Gebru [46] point out the drastically worse performance of commercial facial analysis algorithms (performing tasks such as gender- or smiling- detection) on images of dark-skinned women. Images of dark-skinned women comprise only 7.4% and 4.4% of the common benchmark datasets Adience and IJB-A, and thus benchmarking on them failed to discover and penalize underperformance on this part of the population. Since this study, other algorithms have been benchmarked on more balanced face datasets, changing the development process itself to encourage models that perform well across groups [285].

2.3.7 Deployment Bias

Deployment bias arises when there is a mismatch between the problem a model is intended to solve and the way in which it is actually used. This often occurs when a system is built and evaluated as if it were fully autonomous, while in reality, it operates in a complicated sociotechnical system moderated by institutional structures and human decision-makers (Selbst *et al.* [294] refers to this as the "framing trap"). In some cases, for example, systems produce results that must first be interpreted by human decision-makers. Despite good performance in isolation, they may end up causing harmful consequences because of phenomena such as automation or confirmation bias.

Example: Risk Assessment Tools in Practice

Algorithmic risk assessment tools in the criminal justice context (also described in Section 3.3.1) are models intended to predict a person's likelihood of committing a future crime. In practice, however, these tools may be used in "off-label" ways, such as to help determine the length of a sentence. Collins [71] describes the harmful consequences of risk assessment tools for actuarial sentencing, including justifying increased incarceration on the basis on personal characteristics. Stevenson [305] builds on this idea, and through an in-depth study of the deployment of risk assessment tools in Kentucky, demonstrates how evaluating the system in isolation created unrealistic notions of its benefits and consequences.

2.3.8 Identifying Sources of Harm

Knowledge of a model's context and intended uses can inform identifying and addressing sources of harm. Recognizing historical bias, for example, requires an understanding of how structural oppression and discrimination has manifested in a particular domain over time. Issues that arise in image recognition are frequently related to representation or evaluation bias since many large publicly-available image datasets and benchmarks are collected via web scraping, and thus do not equally represent different groups, objects, or geographies. When features or labels represent human decisions (e.g., diagnoses in the medical context, human-assigned ratings in the hiring context), they typically serve as proxies for some underlying, unmeasurable concepts, and can introduce measurement bias. Identifying aggregation bias usually requires some understanding of meaningful underlying groups in the data and reason to think they have different conditional distributions with respect to the prediction label. Medical applications, for example, often risk aggregation bias because patients

of different sexes with similar underlying conditions may present and progress in different ways. Deployment bias is often a concern when systems are used as decision aids for people, since the human intermediary may act on predictions in ways that are typically not modeled in the system.

2.4 Formalization and Mitigations

2.4.1 Formalizing the framework

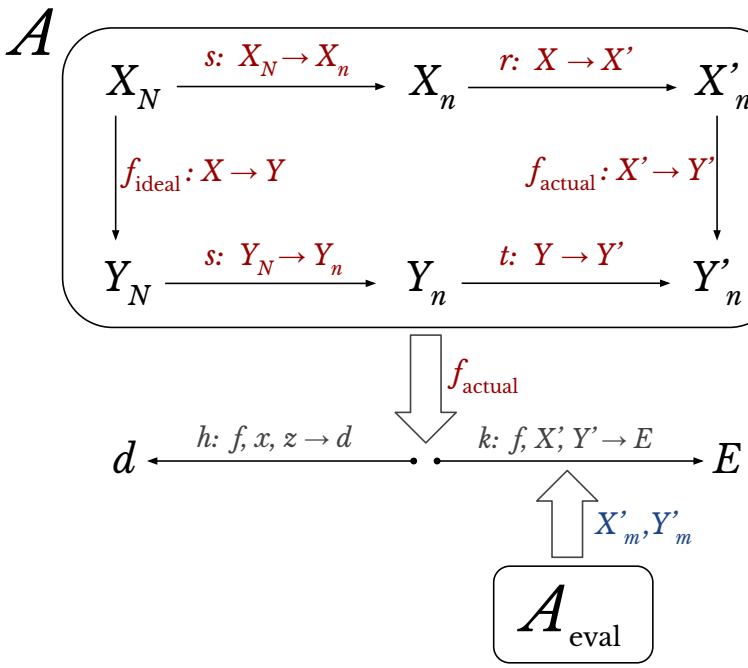


Figure 2-2: A data generation and ML pipeline viewed as a series of mapping functions. The upper part of the diagram deals with data collection and model building, while the bottom half describes the evaluation and deployment process. See the text for a detailed walk-through.

We now take a step towards formalizing some of the notions introduced in the

previous sections. We do this by describing the ML pipeline as a series of data transformations. This formalization provides a context we then use to discuss targeted mitigations for specific sources of bias.

Consider the data transformations for a dataset as depicted in Figure 2-2. This data transformation sequence can be abstracted into a general process A. Let X and Y be the underlying feature and label constructs we wish to capture. The subscript indicates the size of the populations, so X_N indicates these constructs over the target population and X_n indicates the smaller development sample, where $s : X_N \to X_n$ is the sampling function. X' and Y' are the measured feature and label proxies that are chosen to build a model, where r and t are the projections from constructs to proxies, i.e., $X \to X'$ and $Y \to Y'$. The function $f_{\text{ideal}} : X \to Y$ is the target function (i.e., if it could learn using the ideal constructs from the whole target population) but $f_{\text{actual}} : X' \to Y'$ is the actual function that is learned using proxies measured from the development sample. Then, the function k computes some evaluation metric(s) E for f_{actual} on data X'_m, Y'_m (possibly generated by a different process, e.g., A_{eval} in Figure 2-2). Finally, given the learned function f_{actual}, a new input example x, and any external, environmental information z, a function h governs the actual action d that is taken (e.g., a human decision-maker taking a model's prediction and making a final decision).

2.4.2 Designing Mitigations

There is a growing body of work on "fairness-aware algorithms" that modify some part of the modeling pipeline to satisfy particular notions of "fairness." Interested readers are referred to Narayanan [243] for a detailed overview of different fairness definitions typically found in this literature, and Friedler *et al.* [125] for a comparison

of several of these techniques on a number of different datasets. Finocchiaro *et al.* [120] further discuss potential issues and mitigation mechanisms in the context of a range of application domains. Here, our aim is to understand and motivate mitigation techniques in terms of their ability to target different *sources* of harm. In doing so, we can get a better understanding of when and why different approaches might help, and what hidden assumptions they make. Understanding where intervention is necessary and how feasible it is can also inform discussions around when harm can be mitigated versus when it is better not to deploy a system at all.

To avoid implying that there is a comprehensive or generalizable set of solutions, we do not include a table or checklist of mitigations for different problems here. Instead, we intend this framework to provide a useful organizational structure for thinking through potential problems, understanding if and what mitigation techniques are appropriate, and/or motivating new ones.

As an example, measurement bias is related to how features and labels are generated (i.e., how r and t are instantiated). Historical bias is defined by inherent problems with the distribution of X and/or Y across the entire population. Therefore, solutions that try to adjust s by collecting more data (that then undergoes the same transformation to X') will likely be ineffective for either of these issues. However, it might be possible to combat historical bias by designing s to systematically over- or under-sample X and Y, leading to a development sample with a distribution that does not reflect the same undesirable historical biases. In the case of measurement bias, changing r and t through more thoughtful, context-aware measurement or annotation processes (e.g., as in Patton *et al.* [260]) might help.

In contrast, representation bias stems either from the target population definition (X_N, Y_N) or the sampling function (s). In this case, methods that adjust r or t (e.g., choosing different features or labels) or f (e.g., changing the objective function)

are probably misguided. Importantly, solutions that *do* address representation bias by adjusting s implicitly assume that r and t are acceptable and that therefore, improving s will mitigate the harm.

Aggregation bias is a limitation on the learned function f that stems from an assumption about the homogeneity of $p(Y'|X')$, and can result in an f that is disproportionately worse for some group(s). Therefore, aggregation bias could be targeted through 1) parameterizing f so that it better models the data complexities (e.g., coupled learning methods, such as multitask learning, that take into account group differences [109, 311]), or 2) transforming the training data such that f is now better suited to it (e.g., projecting data into a learned representation space where $p(Y'|X')$ is the same across groups [360]). Note that methods that attempt to make predictions independently of group membership [74] likely will not address aggregation bias.

Learning bias is an issue with the way f is optimized, and mitigations should target the defined objective(s) and learning process [168]. In addition, some sources of harm are connected: e.g., learning bias can exacerbate performance disparities on under-represented groups, so changing s to more equally represent different groups/examples could also help prevent it.

Evaluation bias is an issue with E, which is a measure of the quality of the learned function, f. Tracing the inputs to E, we can see that addressing evaluation bias could involve 1) redefining k (the function that computes evaluation metrics) and/or 2) adjusting the data X' and Y' on which metrics are computed. We might improve k through computing and reporting a broader range of metrics on more granular subsets of the data (e.g. as in Buolamwini and Gebru [46]). The best groups and metrics to use are often application-specific, requiring intersectional analysis and privacy considerations; they should be chosen with domain specialists and affected

populations that understand the usage and consequences of the model. In a predictive policing application, for example, law enforcement may prioritize a low false negative rate (not missing any high-risk people) while affected communities may value a low false positive rate (not being mistakenly classified as high-risk). See Mitchell *et al.* [232] for a more in-depth discussion. Issues with evaluation data X'_m and Y'_m stem from problems within the data generation process in A_{eval}, e.g., an unrepresentative sampling function s_{eval}. Improving s_{eval} could involve targeted data augmentation to populate parts of the data distribution that are underrepresented [64, 66]. In other cases, it might be better to develop entirely new benchmarks that are more representative and better suited to the task at hand [91, 177, 14]. Beyond better metrics or benchmark data, we could also expand evaluation paradigms to include other ways of measuring and assessing success — e.g., longer-term field studies, or qualitative feedback from users.

Deployment bias arises when h introduces unexpected behavior affecting the final decision d. Dealing with deployment bias is challenging since the function h is usually determined by complex real-world institutions or human decision-makers. Mitigating deployment bias might involve systems that help users balance their faith in model predictions with other information and judgements [172]. This might be facilitated by choosing an f that is human-interpretable, and/or by developing intuitive interfaces that help users understand model uncertainty and how predictions should be used.

2.5 Conclusion

This chapter provides a framework for understanding the sources of downstream harm caused by ML systems. We do so in a way that we hope will facilitate productive communication around these issues; we envision future work being able to state

upfront which particular type of bias they are addressing, making it immediately clear what problem they are trying to solve and what assumptions they are making about the data and domain.

By framing sources of downstream harm through the data generation, model building, evaluation, and deployment processes, we encourage application-appropriate solutions rather than relying on broad notions of what is fair. Fairness is not one-size-fits-all; knowledge of an application and engagement with its stakeholders should inform the identification of these sources.

Finally, we illustrate that there are important choices being made throughout the data generation and ML pipeline that extend far beyond just model training. In practice, ML is an iterative process with a long and complicated feedback loop. We highlight problems that manifest through this loop, from historical context to the process of benchmarking models to their final integration into real-world processes.

Chapter 3

Characterizing ML Stakeholders and their Needs

In the prior chapter, we show how harm arises when development decisions are made without considering the deployment context in which a system will be used. This issue is exacerbated when the implications of these decisions are hidden, leaving downstream users with few avenues to understand, probe, or push back on the system. We propose an alternate approach to the development and deployment of ML systems that is focused on the people in a particular deployment context who use and are affected by the system. In this chapter, we explore the broad and diverse range of ML stakeholders, and the heterogeneity of their needs.

3.1 Introduction

To design effective deployment tools, we need to first consider the question: who, exactly, are the stakeholders involved, and what are they trying to achieve?

Take, for example, an ML-based medical decision-making system. The physicians using the system need to be able to align the output with their own clinical expectations and justify their recommendations to patients. Patients, then, need to have some confidence in the validity of these recommendations, and may want to explain decisions to family members. Other medical staff need to understand the decision-making processes insofar as it affects the treatment they administer to the patient. The developers who created the system should be able to monitor its performance and understand how to make improvements. Physicians and patients, as well, may want and be well-suited to provide feedback about on-the-ground errors the system makes. And there are undoubtedly other people involved in or affected by this system, from external legal agencies to all the people whose data went into training the ML model. When we think about designing deployment tools for this context, it is critical to understand each of these different stakeholders and what they need out of the tool.

Throughout the thesis, we use the term "deployment tool" to indicate any insight into an ML system that is provided to downstream users — including, for instance, a visualization of the training data, representations of prediction uncertainty, interactive tools for probing the model, et cetera. Much of the existing work in designing deployment tools has been framed as "model interpretability." Therefore, in this chapter, we also use the term "interpretability" to describe these tools. However, we note that we adopt a broader definition of interpretability (that aligns with our notion of deployment tools) than some prior work, which mostly focuses on mechanistic explanations of a model.

Existing interpretability methods often do not explicitly identify or describe their intended user. As a result, their outputs inadvertently end up being most understandable to the people that build them (i.e., ML researchers or developers). In other

cases, the recipient of the interpretability system is described generally as a "layperson" or "end user"; resulting methods may produce simpler visuals, but experimental studies have shown that too often these tools are not useful for people in practice [268, 206, 47, 312]. In our prior example, doctors, patients, and medical staff may all be considered "end users," but have significantly different needs and goals when interpreting, understanding, and reacting to the output of the ML model. Indeed, when it comes down to it, many organizations say they want to give users insight into ML systems to improve trust and prevent misuse, but these methods are only actually used internally by developers [32].

Part of this disconnect stems from the difficulty in identifying and characterizing different stakeholders and what they need to understand about the system. A growing body of work has engaged with this problem, proposing an ecosystem of stakeholders [323, 269], and conducting literature surveys [357, 164, 117, 235] and interview studies [55, 166, 325] to understand their goals. Resultant frameworks typically adopt one of two approaches: they either categorize stakeholders by their expertise (using labels such as "experts," "novices," or "non-experts" [357, 164, 235]) or by their functional role in the ecosystem (e.g., "executives" and "engineers" [32], model "breakers" and "consumers" [166], or model "operators" and "executors" [323]). Stakeholder needs and goals then follow from these categories. While usefully advancing our understanding of the stakeholders involved, these initial frameworks are limited in their descriptive and generative powers [27]. For instance, role-based frameworks implicitly conflate a person's expertise with what they need from the system, with roles often depicted as a relatively static constructs. And expertise-based categories typically portray stakeholders lying on roughly linear scales that only account for cognitive notions of expertise and, thus, do not acknowledge the rich tacit knowledge and lived experience they may possess.

In response, we introduce a framework with a more granular and composable vocabulary to characterize the stakeholders of machine learning systems, and their needs. Our framework comprises two components. First, we decompose stakeholder expertise into two dimensions that describe the types of knowledge a stakeholder might possess (i.e., formal, instrumental, and personal knowledge), and the contexts in which this knowledge manifests (i.e., machine learning, the data domain, and the milieu). Second, we define stakeholder needs using a three-level typology of long-term goals, shorter-term objectives that target these goals, and immediate tasks that stakeholders perform to meet their objectives.

To understand the implications of our framework, we assess its descriptive, evaluative, and generative powers — three properties of interaction models first described by Michel Beaudouin-Lafon [27]. We code 58 papers describing interpretability systems or users, and find that our framework is consistently able to describe stakeholders' knowledge and interpretability needs while adding granularity and drawing new connections between them. We describe how our framework's abstractions can allow us to design more precise application-grounded evaluations [98], including bringing precision to participant recruiting and providing a structure to operationalize comparative studies. And finally, we demonstrate that our framework generates a rich intersection of user expertise and needs for study, and can also be turned inwards to facilitate a more reflexive design process.

3.2 Background and Motivation

Researchers have recognized that precisely defining "interpretability" or "explainability" is a key challenge for the field [220]. Although some work seeks to develop formal or technical definitions of interpretability [98, 134, 92], and though the bur-

geoning set of interpretability techniques often do not name specific target users or tasks [278, 365, 359, 15, 106], there is a growing recognition for the need to approach this problem space in a human-oriented manner. In this section, we motivate our contribution by surveying prior work and describing the limitations we observe with current approaches for defining the *why* and *who* of machine learning interpretability.

In early definitions, Lipton [220] and Doshi-Velez & Kim [98] identified that the need for interpretability primarily stems from a mismatch between the formal definition of the machine learning model, and its output and real-world impact. Lipton further expanded on this need by enumerating a set of desiderata for interpretability including building trust in the model, inferring causal relationships between the input and output, improving model tranferability and generalizability, providing introspection, and finally to facilitate fair and ethical decision-making. Others have since contributed to this list in a variety of ways including detailing how different interpretability methods can be chosen to mitigate particular cognitive biases [334], proposing taxonomies of questions used to arrive at an appropriate interpretability method [12], discussing how applications with different contexts or levels of automation might necessitate different design decisions [298, 216], and grounding the need for explanations in the social sciences [231].

Most relevant to this chapter is a body of work that seeks to better define interpretability by studying the specific users involved. In surveying this work, we identified two distinct approaches to doing so. First are a group of papers that characterize users based on their expertise. For instance, both Yu & Shi [357] and Hohman et al. [164] classify users on roughly linear scales of machine learning expertise (from beginner to expert for Yu & Shi, while Hohman et al. adopt the terms "model developers and builders," "model users," and "non-experts"). Similarly, Mohseni et al. [235] identify "AI Novices" and "AI Experts" and add "Data Experts"

to the mix. With all of these schemes, a user's needs then stem from their expertise. For instance, novices are typically described as needing educational or teaching tools, whereas experts require tools for debugging or deploying models, or assessing model performance.

The second category of papers characterize users based on their functional role instead. For instance, Tomsett et al. [323] posit an ecosystem of stakeholders including model creators, operators, executors and examiners, as well as the decision and data subjects that are affected by the model or whose data the model was trained with, respectively. Similarly, through semi-structured interviews, Bhatt et al. [32] identify four categories of stakeholders (executives, ML engineers, end users, and others) while Hong et al. [166] identify model builders, breakers, and consumers. Across this work, the role a person inhabits within an organization (or the role they play during the human-AI interaction) determines their interpretability needs. For example, model creators/builders/engineers are said to want introspection of the level of individual instances and features, model operators/breakers may wish to monitor the performance of the model including authoring test cases, and finally model executors/executives/consumers want to be able to have confidence and trust in the model. Cai et al. [55] and Tonekaboni et al. [325] follow this approach of role-based needfinding as well by interviewing clinicians.

While both expertise-oriented and role-oriented frameworks have usefully brought further definition to the problem of machine learning interpretability, we can observe limitations to their descriptive and generative powers (i.e., the degree to which they describe *existing* points, and help us identify *new* points in the problem space, respectively [27]). Role-based frameworks, for instance, do not break the problem space down into sufficiently granular and composable units. As a result, several roles appear to implicitly conflate expertise and interpretability needs — for example, model

"creators" are likely most expert with machine learning, and thus need debugging tools at the level of individual instances or features; but, one could imagine "auditors" appreciating insight at this level of abstraction even if they lack an equivalent level of machine learning expertise. Similarly, consider the domain of clinical diagnoses: model "consumers" could equally describe doctors and patients despite these users likely requiring different explanations of the model's output as a result of different levels of medical expertise. Here, model "executors" does not provide much more precision as both doctors and patients are tasked with making decisions informed by the model — doctors about what treatment to prescribe, and patients about whether they do indeed wish to proceed with the treatment. Finally, although most role-based frameworks explicitly note that roles are not mutually exclusive (i.e., a single role may map to more than one individual, and one person may play several roles), roles are nevertheless depicted as relatively static constructs. Not only might an individual user's role change over time but, even if they remain in the same role(s), their interpretability needs may change through repeated exposure to and increased familiarity with the models they are working with, or the situations in which these models are deployed.

Expertise-based frameworks exhibit similar limitations. In particular, a key concern is how these frameworks portray expertise as a linear scale from "novice" to "expert." Several external literatures have articulated concerns with this framing of expertise. For instance, in critiquing the influential Dreyfus linear model of skill acquisition [103], Dall'Alba & Sandberg note that *"[s]tage models of development appear to assume we know what skillful performance entails for each area of skill"* and that the *"focus on stages veils more fundamental aspects of development; it directs attention away from the skill that is being developed"* [83]. Moreover, Dall'Alba & Sandberg point to the fact that such models are primarily concerned with cognitive

development and fail to acknowledge expertise gained through embodied practice of a skill [83]. We see a form of this latter critique in the literature on participatory design as well, which advocates that all stakeholders in a design process possess valuable expertise through their lived experience and tacit knowledge [304, 200]. Finally, although recent frameworks usefully consider domain expertise in addition to machine learning expertise, such a clean decoupling does not account for the ways expertise may transfer. For example, Cai et al. find that while medical practitioners express a desire for an "AI primer", they are nevertheless able to bring some of their training and experience working with other clinical technologies to bear — for instance, in understanding that the output of a model will not be perfect, or in enumerating "test cases" for an AI assistant [55]. Similarly, as AI/ML-enabled technologies increasingly permeate every day life, this ubiquity and familiarity will shape users' interpretability needs in ways that current expertise-based frameworks leave unaddressed.

And, across the two types of frameworks, interpretability needs or goals are determined primarily by the category a user falls within. While many frameworks allow for categories to overlap, this approach nevertheless obscures the fact that many goals can cut across several roles or expertise. For instance, almost every stakeholder involved will likely want to have trust in the model, and want to be able to assess the degree to which it may be biased. We see explicit evidence for this for machine learning experts [164] and data experts [235], model creators and breakers [166], model operators [55, 325], as well as for decision- and data-subjects who may wish to contest a decision or otherwise seek recourse [8, 200]. Similarly, while current frameworks primarily pose debugging and improving the model as goals model creators, builders, or any other traditionally-"expert" stakeholders may have, one could imagine that activists and other groups with non-traditional expertise may also wish to assess the

outcome of domain-specific test cases.

In summary, recent work has recognized that better defining the problem space is a key challenge for machine learning interpretability, and has advanced our understanding by contributing frameworks for describing the stakeholders involved and their goals or needs. However, in analyzing the descriptive and generative powers of these frameworks, we see several limitations. In particular, by not providing a sufficiently granular or composable vocabulary, existing frameworks poorly distinguish the rich intersection that exists between attributes of the stakeholder (e.g., their expertise), the role that they may play (e.g., model creator or consumer), and their ultimate goals or needs with regards to interpretability (e.g., debugging the model, or building trust).

3.3 A Framework to Characterize the Stakeholders of Interpretable ML

To develop a more granular and composable vocabulary for describing the stakeholders of interpretable machine learning, we engaged in an iterative process with alternating phases to diverge and converge our thinking. In particular, we began by surveying the literature on interpretability summarized in the previous section, and extracting passages that described users and stakeholders, as well as their needs, actions, and goals. To diverge our thinking, we looked to domains outside of interpretability and computer science, including the literatures on expertise and pedagogy [121, 112, 355, 354, 341, 33, 187, 153, 336, 154, 83], critical theory [252, 320, 233, 8, 208], law [332, 358, 159, 69, 100], and participatory action research [142, 175]. To converge our thinking, we reflected on how concepts from these

external domains could be adapted within interpretability. This reflection process involved alternating phases of open coding to map external concepts to the passages we had initially extracted, affinity diagramming to identify recurring groupings and patterns between codes, analytic memo writing, and weekly hour-long conversations between all authors.

Our framework comprises two halves. First, it describes the knowledge stakeholders may possess and the contexts in which this knowledge may manifest. And, second, it enumerates the long-term goals stakeholders may have, and breaks these goals down into shorter-term objectives and specific tasks they can perform.

3.3.1 Decomposing Stakeholder Expertise into Knowledge and Context

Prior work has identified, either explicitly [164] or implicitly via roles [323, 166, 235], that expertise is a defining attribute of interpretable ML stakeholders. To provide a more granular treatment of expertise, we adapt models of expertise from Fleck [121] and Eraut [112] to decompose a singular notion of expertise into three constituent types of *knowledge*. **Formal** knowledge comprises an understanding of codified theories, embodied in text or diagrams such as those found in textbooks, and is acquired through a prolonged educational process. **Instrumental** knowledge is an understanding of how to "apply" formal knowledge. It is embodied in the use of tools or other instruments, and is learnt through demonstration and practice. Finally, **personal** knowledge describes information that is entirely embodied in individuals, and is gained through their participation in specific domains. It is difficult to codify [112] as it consists of a person's lived experience (e.g., memories of specific events, self-knowledge about the way they may react in certain scenarios, etc.) as well as values

Table 3.1: Examples of how the three types of knowledge manifest in the three contexts described by our framework.

	Knowledge		
Context	**Formal**	**Instrumental**	**Personal**
ML	The math behind model architectures, optimization and training processes, etc.	Familiarity with ML toolkits, off-the-shelf models, etc.	Tricks of the trade (e.g., hyperparameter values, feature engineering, etc.)
Data Domain	Theories relevant to the data domain (e.g., symptoms and treatments, case law and legal precedent, etc.)	Experience working with other related technology (e.g., medical devices, document mining tools, etc.)	Lived experience (e.g., prior memories of similar events)
Milieu	Sociocultural theories (e.g., redlining, gerrymandering, mass incarceration, etc.)	Familiarity with broader ML-enabled systems (e.g., virtual assistants, recommendation algorithms, etc.)	Lived experience and cultural knowledge (e.g., values, attitudes)

that may be distributed in the cultures and societies to which they belong.

These types of knowledge manifest in *contexts*, or the domains or situations that determine what knowledge is relevant. We identify three contexts: **machine learning**, or the knowledge required to research, develop, operate, or deploy machine learning models; the **data domain**, or the knowledge necessary to collect, organize, analyze, and communicate the data the model was trained with or makes decisions about; and **milieus**, which refer to the environments that the human-AI interaction might be occurring within. These environments include both the physical surroundings (e.g., a home, bank, courthouse, doctor's office, etc.) as well as the broader sociocultural context (e.g., mass incarceration, redlining, gerrymandering, etc.).

Our framework provides a expansive yet precise treatment of expertise in interpretable ML. While prior expertise- or role-based approaches latently encode notions of formal and instrumental knowledge, by explicitly articulating these concepts, our framework facilitates teasing apart differences and understanding the implications on

interpretability design. For example, "model users" [164] and "model breakers" [166] cover an extremely broad range of possible stakeholders including model architects, trainers, engineers, data scientists, and machine learning artists [164], as well as domain experts, product managers, and auditors [166], respectively. These categories appear primarily focused on stakeholders' instrumental knowledge and, by analyzing contexts, we can separate machine learning instrumentalists (model architects, trainers) from data domain instrumentalists (artists, domain experts, product managers), and those that may span the two (data scientists, auditors). Doing so suggests that these stakeholder groups may have different interpretability needs that the broader categories of "model users" or "model breakers" obscure. For instance, perhaps interfaces for machine learning instrumentalists should be articulated in terms of the components exposed by popular toolkits. Similarly, for data domain instrumentalists, how might we analogize interpretability to tools and systems that they already work with in order to enable expertise transfer (akin to how Cai et al. found medical practitioners reasoning about uncertainty [55])?

Moreover, our framework explicitly recognizes the personal knowledge stakeholders may have — including "tricks of the trade" a person may have acquired, their experiences and memories, or the more distributed values of the cultures and societies they are a member of — as an important consideration when designing for interpretability. Critically, by placing it alongside formal and instrumental knowledge, our framework identifies it as an *equally important* form of knowledge. As a result, one might consider designing *for* stakeholders' personal knowledge — for instance, using example-based explanations such that a stakeholder might better "see themselves" in the data [3, 275, 262]. But our framework also suggests designing *with* stakeholders, to better account for personal knowledge that designers do not have — a position advocated for by various communities including participatory ac-

tion research [142] & design [304, 200]. For instance, members of the general public might have different notions of what constitutes an "error" based on their personal knowledge [150].

Our framework also highlights that interpretability design must attend to more than the immediate contexts of machine learning and the data domain — explanations must be situated in stakeholders' milieu. Here, we draw an analogy to data visualization. Researchers and data journalists consider annotations to be a crucial component of effective visualization design because it helps readers understand the broader context associated with the visualized data. As Amanda Cox, former Data Editor for The New York Times, says, *"the annotation layer is the most important thing we do ... otherwise it's a case of here it is, you go figure it out* [77]. We believe this property holds true for interpretability as well — it is insufficient for an explanation to be articulated purely in terms of the model or data if it misses critical aspects of the milieu. For instance, consider an ML-backed loan evaluation system: explanations in the ML context would articulate the output decision in terms of model components, while explanations in the data domain might also discuss distributions in the training or test set and how this may lead to biased output. However, under our framework, we would consider these explanations to be incomplete if they were not situated the broader sociocultural milieu — for instance, how disparities in data distributions have occurred through policies such as redlining, or in the difficulty ex-offenders have in finding employment.

Finally, by decoupling knowledge and context as two orthogonal dimensions of the problem space, our framework enables a more systematic analysis of the stakeholders of interpretability. It eschews prior easily-quantifiable linear scales in favor of more descriptive treatments of expertise. Designers can work with each dimension individually — for example, how might interpretability help stakeholders formalize

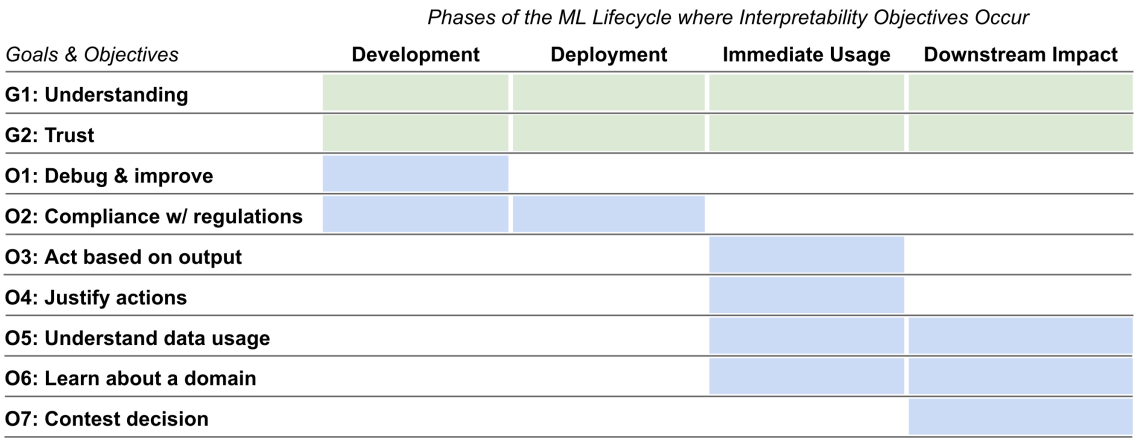


Figure 3-1: A visualization of the latent chronology in goals and objectives. Categories along the horizontal axis are relevant phases of the ML process. Colored cells indicate the phase in which a particular goal or objective typically occurs. Phases need not occur linearly, may be iteratively revisited, and many different stakeholders may be involved at any given phase.

their personal knowledge in the data domain by scaffolding example-based explanation with featured-based saliency methods akin to faded worked examples [13]; or, as described previously, how might instrumental knowledge in the data domain transfer to the machine learning context [55]. And, by considering the intersection of the two dimensions as well, our framework can help us identify the ways in which expertise recurs in the interpretability ecosystem.

3.3.2 Distilling Stakeholder Needs into Goals, Objectives, and Tasks

Through our open coding and reflection process, we distilled a three-level typology of interpretability needs. The first level identifies two long-term *goals*: **understanding the model (G1)** and **building trust in the model (G2)**. These goals are

high-level and difficult to define precisely, but we include them in our framework because they underlie almost every single piece of work we read. We do not expect stakeholders to be able to directly accomplish these goals, nor do we imagine that future methods or systems will address them squarely. Rather, these goals function like substrates which inform and influence the two lower levels of needs we describe below. For instance, we expect stakeholders to develop an understanding of models over time — through repeated exposure and interactions. And much work has framed trust as something society as a whole needs in order to accept new technologies [332, 42, 358] — indeed, trust may grow as stakeholders better understand the model, but may also develop in a proxied or deferred fashion through increased regulations and standards.

The second level of our typology describes the shorter-term *objectives* stakeholders might target to achieve their longer-term goals. We give real examples for each objective, and demonstrate how they can be relevant to a diverse range of stakeholders. These objectives are grounded in stakeholders' current real-world needs, but as ML tools continue to be deployed in new domains, we expect this typology will continue to evolve.

(O1) Debug or improve a model. The objective of improving a system or correcting its mistakes appears frequently in the literature, and is often posed in terms of the needs of developers [42, 215, 158]. For example, Bhatt *et al.* [32] describe how internal members of organizations try to use interpretability techniques to uncover inconsistencies between the model's logic and their intuition or expectations, in order to guide further improvements. However, it is critical to acknowledge that developers are not the only stakeholder group to which this need applies. For instance, Tonekaboni *et al.* [325] and Zarsky [358]

both highlight the value of allowing a larger group of stakeholders, including the general public, to provide feedback for improving systems. Indeed, theories from Participatory Action Research also hold that people on the ground in a specific context are often much better suited to realize errors and devise appropriate fixes, as opposed to developers for whom the errors typically have no direct consequence [142].

(O2) Ensure compliance with standards or regulations. Auditing, or ensuring that the development, deployment, and results of a certain system are compliant with a particular set of standards (whether they are legal, ethical, safety, or other) is already necessary in other areas such as finance or aerospace, and is emerging as an important objective for ML systems as well. The introduction of the GDPR, for example, has established a set of legal standards that automated systems must comply with. And it is not only external watchdog agencies or governments that are interested in ensuring such compliance. Individuals or groups within an organization may also have their own internal standards they want to ensure are met — for example, Raji *et al.* [273] describe the design of an internal auditing pipeline.

(O3) Understand how to incorporate the model's output into downstream actions. Several prior papers mention the need for guidance on whether and how to incorporate model predictions into further actions — whether that involves relating the model's output to relevant and actionable decisions, or understanding how much weight to place on the model's prediction [166, 325, 42, 32, 215]. This objective emerged as important for a number of different types of stakeholders, such as doctors using a diagnostic aid [166, 325], people applying for health insurance that involves automated screening [32], or for

those subjected to automated decisions more generally [332].

(O4) Justify or explain actions influenced by a model's output. Through interviews with Intensive Care Unit and Emergency Department clinicians, Tonekaboni *et al.* [325] describe clinicians' desire to justify decisions influenced by a model's output to patients or colleagues. Similarly, by interviewing the head of AI at a bank using automated credit evaluations, Hong *et al.* [166] identify the need to justify to customers decisions that were influenced by the model. In addition to the immediate stakeholders acting on the model output (*executors* according to Tomsett *et al.* [323]), this goal can also stem from people about whom a decision was made (*decision subjects*). For example, Zarsky [358] frames the need to provide someone with an explanation of a decision or action that affects them as one that is necessary in order to respect their autonomy.

(O5) Understand how one's data is being used. Zarsky [358] grounds this objective in the theoretical premise of information privacy rights, framing it as an extension of individual autonomy. The need to have control over one's personal data has also been broadly accepted in European data protection law. Hildebrandt [159] makes a further argument that people should understand not only what data about them is being utilized, but the potential consequences of this usage as well. And Buneman *et al.* [45] distinguish between different types of data provenance users may be interested in. Clearly, given the prevalence of data mining, this objective is relevant to a wide range of stakeholders.

(O6) Learn about a domain. Through interview studies, both Hong *et al.* [166] and Liao *et al.* [215] describe how stakeholders across different domains use

interpretability to generate new hypotheses or insights about a domain. For example, one participant aimed to use a model predicting surgeons' future performances as a tool to better understand what factors drive good performance, rather than using it as a predictive system. Hohman *et al.* [163] focus specifically on data scientists, describing how interpretability helped them find "valuable nuggets of information" in the data. Similarly, Doshi-Velez and Kim [98] identify the use of interpretability to advance scientific understanding. Indeed, there is a growing subfield of machine learning investigating how interpretability mechanisms can aid in scientific discovery [1].

(O7) Contest a decision made based on the model's output. Citron and Pasquale [69] posit that the right to challenge a decision affecting oneself should be ensured under due process, and Doshi-Velez *et al.* [100] draw a comparison to the legal system, where mechanisms for redress serve as a powerful form of accountability. Wachter *et al.* [332] also notes that the right to contest an automated decision is provided in Article 22(3) of the European General Data Protect Regulation (GDPR). An individual affected by an algorithmic system may not be the only one who wishes to contest it, either. We can imagine that external stakeholders like lawyers, judges, or activists may also be interested in pushing back against model outputs that seem incorrect, arbitrary, or unfair.

The third and final level of our typology identifies the specific *tasks* a stakeholder can perform to achieve the goals described above. We break tasks out as a separate level of the typology to make clear that tasks do not map to objectives in a one-to-one fashion; rather, the same task may be used to accomplish several different objectives. For instance, detecting discrimination or other undesirable behavior in a model's prediction is likely to be a necessary task for both contesting a decision

(O7) but also for understanding whether or how to incorporate model output in downstream actions (O3). Although the task is shared across these objectives, the specific type of discriminatory behavior a stakeholder may wish to detect, and the manner in which is it exposed and communicated, may differ based on the domain, the higher-level objective, and the stakeholder's knowledge. Here, we describe several such underlying tasks that we found to recur in the literature and give examples of how they can be relevant for multiple objectives. As with objectives, we expect this level of the typology to grow as ML continues to be deployed in new situations.

(T1) Assess reliability of a given prediction. Understanding the reliability of a given prediction is important for deciding how (or whether) to incorporate the model's output into further actions (O3), to prevent harmful outcomes or over-reliance [362, 47]. Similarly, the ability to assess a given prediction and show, for example, that it may not have been reliable, is likely to provide important evidence for contesting a decision (O7).

(T2) Detect mistaken, discriminatory, or arbitrary behavior. The ability to detect discrimination or other unwanted logic codified in a model is considered a crucial tool for being able to contest an automated decision (O7) [100]. Similarly, ensuring that predictions are not being made arbitrarily is likely necessary to ensure compliance with ethical or legal standards (O2). In other cases, detecting incorrect reasoning was a way to guide model debugging and elucidate areas for improvement (O1) [55]. Some papers also frame this task as its converse, i.e., verifying that predictions are sensible and/or fair (by some definition) [281].

(T3) Understand the extent of the information the model is using. Understanding details and extent of features used emerged as important for explaining

actions influenced by the model (O4). Tonekaboni *et al.* [325], for example, describe how doctors felt that understanding the clinically relevant model features that were used was critical to first rationalizing the predictions to themselves, and then explaining them to patients. Depending on the context, recognizing higher-level groups of features (e.g., "demographic information," "patient medical history") may be more understandable and feasible than individual features. We can also imagine that developing this understanding will also be an important way for stakeholders to identify what aspects of their personal data are being incorporated into a specific system (O5) [159].

(T4) Understand the influence of different factors on the model's output. For stakeholders who are interested in generating new insights about a domain (O6), understanding how different factors influence the output is key. Roscher *et al.* [284] provide several examples of deriving scientific or medical insights by investigating the impact of scientifically-meaningful factors on predictive outcomes. This task is also important for ensuring compliance with particular standards or regulations (O2), which may detail when/how it is acceptable to use certain features. Unlike T3, this task might not provide a comprehensive view of the features used since it is more focused on the ways that the most salient features influence the output.

(T5) Understand model strengths and limitations. Understanding the model's overall potential weaknesses is critical for understanding how to incorporate its output into further actions (O3). For example, Cai *et al.* [55] describe how doctors consistently wanted to know the proposed AI tool's specific limitations so that they could anticipate and account for them during decision-making. Understanding areas of weakness is likely to also be useful for debugging and

improving the model (O1), e.g., by guiding additional data collection or training.

Note that specific implementations (e.g., counterfactual explanations [332]) are not included at the task-level; rather, they are used to *implement* a particular task. For example, counterfactual explanations might be one way to implement the task "detect discriminatory behavior" (T2), and might be more or less appropriate depending on the stakeholder's knowledge, their overarching objective, and the surrounding context. Section 3.4.3 (Generative Power) further discusses the implications our framework might have on choosing particular methods.

While several prior literature surveys have sought to collate and organize a list of interpretability needs, our framework makes some key advances to provide a more nuanced understanding these needs. First, where prior surveys focus primarily on computer science subdisciplines [164, 235, 117], our framework incorporates these insights and extends them by looking to the legal literature [332, 159, 358, 69] and research on participatory action and design [142, 304]. As a result, our framework is able to surface objectives such as "contesting a decision" (O7) or "understanding how one's data is being used" (O5) that prior surveys did not identify.

Second, and more importantly, where prior approaches define interpretability needs as a function of stakeholder expertise or role, our framework defines these needs as an independent component of the problem space. As a result, and as the examples above illustrate, our framework helps reveal that interpretability needs can cut across several different stakeholders. For instance, model debugging (O1) is one of the most frequently identified interpretability goals; but, prior work has primarily categorized it as a need machine learning experts (or model builders and developers) have. In contrast, our framework identifies that although certain stakeholders may

not have much formal or instrumental machine learning knowledge, their personal knowledge may be crucial for identifying or fixing model errors. Similarly, while it may have previously been tempting to think that contesting a decision (O7) is a need primarily expressed by decision subjects, our framework highlights that other stakeholders (including lawyers, judges, and activists) may wish to do so as well to affect systematic change.

Finally, in contrast to the uniform treatment of prior interpretability surveys, our framework provides new levels of abstraction for discussing interpretability needs. In doing so, we can distinguish that these needs form a hierarchy: immediate tasks help stakeholders accomplish short-term objectives which, over time, achieve long-term goals. As with other multi-level typologies [41], this structure surfaces the compositionality latent in this space. For instance, as described above, there is a many-to-many relationship between goals, objectives, and tasks: one task may apply to several objectives; many tasks may be required to accomplish a single objective; and, together, they are all necessary to achieve goals. Similarly, our three-level sequence allows for describing interpretability needs as sequences of action. For example, to improve a model (O1), a stakeholder may wish to understand its strengths and limitations (T5) by repeatedly assessing the reliability of individual predictions (T1). Or, a stakeholder's trust in the model (G2) may increase or decrease as a result of better understanding how it works (G1).

While we do not ascribe objectives to specific roles or expertise levels as in prior work, we note that they nevertheless exhibit a latent temporal structure — for example, the need to understand how a model's output should be incorporated into a decision (O3) occurs before someone wishing to contest that decision (O7). However, formalizing this latent chronology is not straightforward since a given objective may (re)occur at several different stages during the ML process. And, there is a risk of

unintentionally recapitulating prior stakeholder categorizations as particular roles or expertise may be implicitly associated with different stages of the ML process.

As a result, the chronology we settle on, shown in Figure 3-1, is more flexible and refers to broad phases of the ML process. Rather than provide a precise ordering, it is meant to lend some helpful structure to the many stakeholder objectives. We indicate the phase(s) in which a particular objective typically occurs, and note that these phases are likely to unfold iteratively. For example, the development and deployment stages may be revisited after observing a system's downstream impact. Furthermore, many different stakeholders may be involved in each phase. For example, beyond engineers with formal ML or data knowledge, downstream users with significant personal knowledge may provide input to the development phase of a particular system if they report bugs or provide feedback that is used to retrain the model. We omit tasks from this chronology to preserve the many-to-many mapping between objectives and tasks.

3.4 Evaluation & Example Applications of the Framework

To assess the implications of our framework, we look to the three powers of interaction models described by Beaudouin-Lafon [27]: the *descriptive* power, or how much coverage the framework achieves over existing points in the problem space; the *evaluative* power, or how well the framework helps us compare two points in the problem space; and, the *generative* power, or how the framework helps us envision new or previously unexplored points in the problem space. In addition to an evaluation of the framework, the evaluative and generative powers also serve as a demonstration

of ways in which the framework can be used.

We find that our framework is able to describe over 50 existing papers on interpretability, and further provides a more granular treatment of relevant stakeholders. We then illustrate how our framework gives us a language with which to more carefully evaluate interpretability systems. Finally, we demonstrate how our framework can be used to generate new combinations of personas and needs, how it may suggest ways of designing future interpretability interfaces, and how it may be turned inwards to facilitate a more reflexive design process.

3.4.1 Descriptive Power

We assess our framework's descriptive power by using it to characterize the users and goals described by existing work on interpretable ML. We collected papers using a mix of explicit keyword searches in academic search engines and libraries (e.g., Google Scholar and arXiv), following the citation graph of collected entries, and by compiling the bibliographies of previous literature surveys [164, 117]. Our final list of papers span several research contribution types [209, 343] including frameworks that define interpretability desiderata or key considerations (e.g., Tomsett *et al.* [323], Arya *et al.* [12], Lipton [220]), evaluations of specific interpretability techniques (e.g., Balog and Radlinski [17], Cheng *et al.* [67], Cai *et al.* [51]) and user/case studies that provide insights into pertinent human factors in interpretability (e.g., Tonekaboni *et al.* [325], Hong *et al.* [166], Liao *et al.* [215]). We excluded any papers that introduced novel interpretability techniques without discussing target users or user-centric considerations (e.g., [361, 359]). Similarly, we excluded papers that included only a passing reference to user characteristics (e.g., "interpretability is important for doctors") without explicitly discussing them. We aimed to collect a representa-

tive sample of current interpretability research directions, going beyond the papers we used to initially develop the framework, and sought out references across different applications, data domains, and computer science disciplines including machine learning, data visualization, human-computer interaction, and scientific computing.

In total, we selected 58 papers, and each paper was coded by at least two authors of this paper. Each coder used the framework to identify instances of stakeholder knowledge types and contexts, as well as goals, objectives, and tasks. When the coders disagreed on a designation for the entry, they discussed the conflicts until there was agreement on the code. Where possible, we collected snippets of the papers corresponding to a description or discussion of stakeholder knowledge or needs. The outcome of this coding process, including snippets[1], is shown in Tables 3.2 and 3.3.

We found that the vocabulary provided by our framework was able to describe stakeholders and needs that appeared in prior work. All knowledge type-context intersections and goal/objective/task categories appeared in more than one paper. The most observed knowledge categories were ML-Instrumental (23 of 58 papers), Data Domain-Formal (19/58), and Data Domain-Instrumental (17/58), which capture substantial technical expertise. The most observed objectives were justifying or explain decisions influenced by model output (O4, 21/58) and debugging and model improvement (O1, 21/58). The most common tasks were understanding factors that influence the model output (T4, 11/58) and detecting mistaken/arbitrary behavior (T2, 8/58). We note, however, that there were some goals or objectives that arose that we were not able to categorize given our current framework, such as persuading the user [249, 293]. We have chosen not to explicitly integrate this into the framework because it was presented as a need imposed upon a stakeholder by, e.g., the company

[1]Quotations in the snippets may be paraphrased, and should be interpreted as describing a common theme in cases where multiple references are grouped together.

deploying a particular system, while the current needs we describe are driven by the stakeholder themselves.

Beyond its comprehensiveness, the framework is often able to add an additional layer of granularity. For example, whereas many papers describe people with various types of milieu knowledge as "lay users," we are able to recognize and tease apart the different types of expertise they possess. In other cases, we are able to provide consistency and draw similarities between concepts that were previously obscured — for example, from looking at Table 3.3, we can see that there are many papers that use different terminology to ultimately refer to the same goal.

In using our framework as a descriptive instrument, we found that users' personal knowledge is often the most challenging to define because of the subtle ways it interacts with its context. For example, personal knowledge in a particular domain may cross over into the milieu in cases where a decision domain intersects with everyday functions in society, like finance ("loan applicants", who learn to interact with banking ecosystems). At the same time, another decision subject such as a medical patient might acquire personal domain knowledge by learning about a medical condition that affects them and relating that knowledge to their own symptoms and experiences. Personal knowledge can be hard to observe by experimenters who are evaluating and developing systems using traditional processes. However, we believe developing methods for eliciting this knowledge is an under-explored opportunity for human-centered design.

Our framework also helps reveal patterns of under- and over-representation of certain stakeholders and needs in current interpretability research. For example, we noticed a relative lack of interpretability methods specifically focused on objectives such as understanding how one's data is being used (O5) and contesting a decision based on the model's output (O7). The papers that did cover these areas were pri-

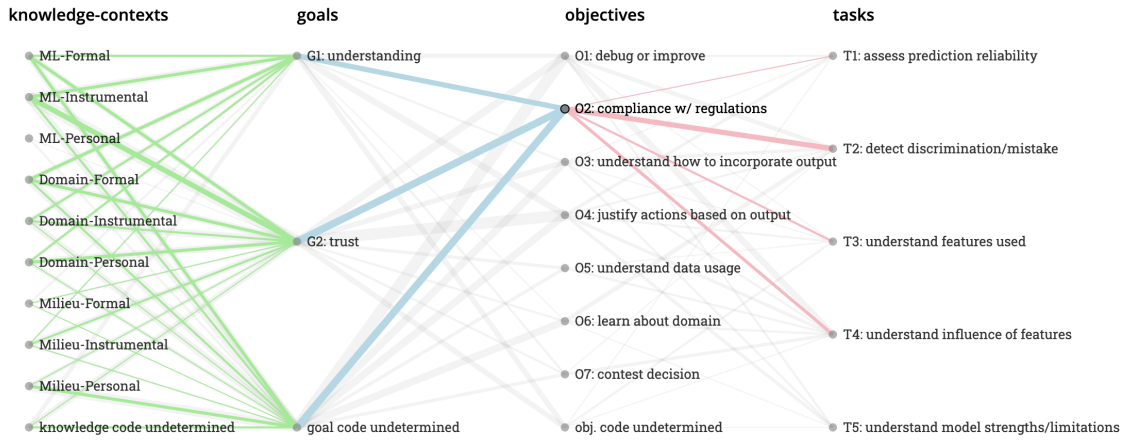


Figure 3-2: A state of an interactive figure that visualizes the results of the analysis of our framework's descriptive power (available at `vis.csail.mit.edu/pubs/beyond-expertise-roles/framework-connections`). We see how the two halves of the framework (knowledge-contexts and goals-objectives-tasks) provide a more granular and composable vocabulary with which to describe 58 papers from the literature on ML interpretability. Light grey links represent the set of all papers, and connect codes that appear together. The width of the link corresponds to the number of papers it represents. We use "code undetermined" to indicate cases where we were not able to code a particular category (e.g., if a paper did not explicitly specify a knowledge-context). In the interactive figure, hovering over a code selects all papers that contain the code, and highlights links to visualize the co-occurrence of other codes (e.g., "O2" shown here).

marily *justifying* these needs from a legal perspective, rather than describing systems or methods to help meet them. Similarly, while users with formal ML and data domain knowledge are mentioned frequently, there is significantly less attention paid to those with formal knowledge in the milieu. These stakeholders, however, are likely to have a deep understanding of the broader sociotechnical context in which systems are deployed. Our framework gives us the vocabulary to consciously recognize these gaps, and subsequently, work towards filling them.

Figure 3-2 uses a parallel coordinates display to summarize the framework's de-

scriptive power, and connect its two halves. Axes correspond to the four components of the framework (knowledge-contexts, goals, objectives, and tasks) and nodes correspond to individual codes used during our qualitative process. Lines connect nodes to represent papers which contain these codes, with line width encoding the number of papers. In the interactive version of the figure (available at vis.csail.mit.edu/pubs/beyond-expertise-roles/framework-connections), hovering over specific nodes selects all papers that contain the code, and highlights links to visualize the co-occurrence of other codes. In doing so, the figure demonstrates the composable, many-to-many nature of our framework.

3.4.2 Evaluative Power

Our framework's evaluative power comes from it's ability to help design more ecologically valid and appropriately-scoped evaluations of interpretability systems/methods.

Similar to prior work [98], we noticed three methods that are primarily used to evaluate interpretability techniques when coding the interpretability literature. The most popular approach does not involve human evaluation. Instead, example outputs generated by the technique are used to illustrate its performance [138, 359, 106, 65, 361, 122, 217] and capabilities including how expressive the technique is [253, 254]. The next-most frequent style of evaluation are user studies conducted with *proxy stakeholders* (e.g., sourced from Amazon Mechanical Turk or a similar online platform) and/or *proxy tasks* (e.g., guessing a model's outcome) [268, 67, 51, 205, 356, 218, 97]. Recent work, however, has demonstrated the fallibility of using proxies—as Buçinca *et al.* [49] describe, proxies induce a different style of cognition, forcing participants to explicitly attend to explanations and the AI rather than implicitly incorporating them as part of the overall process.

The current gold standard evaluative methodologies are application-grounded studies [98] in which domain experts engage with real-world tasks that include interpretable ML. For example, Bussone *et al.* [47], Wang *et al.* [334], and Lundberg *et al.* [223] evaluate interpretability methods by asking healthcare professionals to engage in hypothetical diagnostic scenarios. Although these studies often elicit rich and relevant feedback about a system's real-world implications, they remain relatively rarely used. We posit that this lack of adoption is due, in part, to the difficulty of designing such studies — there is little principled guidance on how to recruit participants, particularly when the real-world situation requires specific types of expertise. Moreover, when comparing interpretability techniques, it can be difficult to design a task that is equitable for the various conditions. And, even when studies are successfully conducted, it can be difficult to understand how the results generalize or inform future work.

We believe our framework's vocabulary begins to make addressing these issues more tractable. In particular, stakeholder knowledge and context offers a more precise way of defining the participant pool, including identifying vectors along which it may be acceptable to introduce proxying. For instance, consider an ML-enabled clinical diagnosis; if it were difficult to recruit a sufficient number of doctors to participate in the study, our framework suggests that residents or medical students might also be viable participants because of their shared formal data domain knowledge (medicine). Similarly, consider evaluating explanations for loan applications; our framework helps identify that a study may not be ecologically valid if participants do not draw from similar pools of personal knowledge of the milieu.

Finally, our three-level typology of goals, objectives, and tasks provides a structure to operationalize comparative studies. For example, to evaluate the relative effectiveness of similar interpretability techniques (e.g., the plethora of saliency and

attribution methods) in a human-centric manner, our framework's *tasks* may be the appropriate level of abstraction to target — they describe operations that can be performed directly and measured through quantitative and qualitative means, and are thus conducive for A/B testing style experiments. On the other hand, for larger-scale interpretability systems, it may be more appropriate to target *objectives* as the specific sequence of operations a participant performs will likely vary significantly between conditions; thus, results will likely be generated qualitatively, through observation and conversation.

3.4.3 Generative Power

Finally, we operationalize our framework to imagine either novel futures or futures underexplored by the existing work.

Generating new personas. Where prior literature has treated "non-experts" (i.e., stakeholders without expertise in either machine learning or the data domain) as a single, homogeneous group, and has designed for them as such, our framework makes explicit how heterogenous this group may be. By introducing the context of the general milieu, and considering how instrumental and personal knowledge may manifest in it, we can generate new stakeholder personas and imagine the implications on interpretability design. For instance, consider everyday people who have some familiarity or exposure to coding, or have tinkered with the Maker movement; we might describe them as having developed mental models for "computational thinking," or instrumental knowledge in the milieu under our framework. As a result of this knowledge, perhaps they would be more amenable to interpretability interfaces that promoted interactive question-answering by manipulating inputs. Similarly, consider someone who has been closely following mainstream media reporting on the

propensity of social media recommendation algorithms to radicalize individuals — our framework would describe them as having rich personal knowledge in the milieu. Perhaps as a result of this knowledge, this person would be initially suspicious of an ML model. In this case, instead of starting with a tabula rasa, perhaps an interpretability interface would be initialized with summaries of the model's strengths and weaknesses (akin to a model card [232]).

Generating new persona-need combinations. Deriving needs from definitions of stakeholders has led to a relatively rigid set of interpretability needs that are recognized. But, by decoupling needs from stakeholder attributes, our framework allows for a much richer intersection of concerns than prior approaches. For example, consider the objective of debugging or improving a model (O1) — prior work has typically viewed this as a need faced by ML experts, and debugging tools are thus built to primarily serve them. Under our framework, we might describe these prior target users as having formal or instrumental ML knowledge; but our framework also exposes other stakeholders who might wish to address this objective as well: people with personal knowledge in the data domain or milieu. Indeed, this aligns with theories of personal and formal knowledge in Participatory Action Research. As Greenwood and Levin say, *"[p]recisely because local stakeholders take action in their own environments, the consequences of errors are both significant to them and often rapidly apparent"* [142]. In contrast, researchers with more formal knowledge may *"rarely know whether they are right or not, as their findings seldom are acted upon and the practical results from their research rarely have direct consequences for them."* Feminist standpoint theory [150] would further posit that what is even considered an "error" or harmful might differ depending on the stakeholder's personal knowledge.

Generating new designs. One might initially consider our framework to be

silent on *how* to design for particular stakeholders — for instance, it does not explicitly prescribe when to use local explanations [278], saliency maps, or feature visualization [253, 254]. However, our more granular definitions of expertise allows us to adapt theories of knowledge development from cognitive science and pedagogy. In particular, the cognitive science literature describes a process of "chunking", where people organize and think about information in terms of high-level concepts (or "chunks") that develop through experience, familiarity, and with increased knowledge [230, 244, 6]. Similarly, the literature on expertise offers several models of decision making that posit two modes: analytic or deliberative thinking that is based in formal or instrumental knowledge, and intuitive thinking that is based on informal or personal knowledge. Thus, when designing interfaces for ML interpretability, designers could begin by first eliciting and characterizing the knowledge their stakeholders have, and the associated cognitive chunks and/or modes of thinking. Standardized instruments — such as the Preference for Intuition and Deliberation (PID) scale [30] — could also be used. These results could then inform what features are used in an explanation (e.g., raw features or higher level combinations of features that align more with the stakeholder's cognitive chunks) or what types of explanations are given (e.g., more intuitive example-based explanations versus more analytical or mechanistic explanations). Prior work by Wang *et al.* [334] provides further guidance on how particular types of explanations can be more/less suited to different modes of reasoning.

Situating stakeholders' knowledge and goals within broader societal power dynamics can also help inform what sorts of interpretability methods do or do not work towards subverting existing hierarchies. Indeed, the literature on expertise from which we derive our framework inextricably links types of knowledge with issues of power. In particular, as Fleck notes, *"the view of knowledge as being disinterested*

or value neutral is idealistic" and *"the possession of formal knowledge confers status and consequently a measure of power or influence within organizations"* [121]. Interpretability can play a key role here, addressing the *"pathology of beneficence"* that Yielder describes [354] — where experts have a tendency to make decisions *for* people rather than allowing them to decide for themselves — and reducing the ability of experts to merely "rent" out their knowledge [202]. However, this work must be conducted carefully for, as Thatcher *et al.* [320] observes, *"[t]he very obscurity of transformation from individual data point to commodified, aggregate big data also masks the asymmetrical power relations between users of technology and the almost exclusively corporate entities which algorithmically collect, link, and analyze the data points of many users."* Take the example of a mortgage applicant living in a redlined neighborhood, who wishes to contest the ML-based decision to reject their application. Their relative lack of power in this situation may be further compounded if they have little formal ML or data domain knowledge. We can begin to see, then, that interpretability methods that put the onus on the individual to change things about themselves in order to receive a better outcome (e.g., "Had your income been $3000 higher, you would have received the loan") help uphold, rather than subvert, existing hierarchies. Interpretability methods that instead shift their gaze upwards and focus on alerting affected stakeholders to potentially discriminatory or arbitrary behaviors by the algorithm might provide much stronger evidence to fight against the reigning power differentials [20].

Generating a more reflexive design process. The scope of interpretable ML should not be imagined by researchers or engineers alone — building interpretability systems that challenge power hinge on the involvement of stakeholders with different goals and knowledge. Indeed, people with formal knowledge (e.g., interpretability researchers, developers and designers, and the institutions within which they work)

are often precisely the ones in positions of power over those with more personal knowledge (often those most directly affected by algorithmic systems). The concept of *interest convergence*, which stems from critical race theory, holds that those in power tend to support goals that serve their own interests [252]. In other words, without actively involving stakeholders whose interests are in opposition to existing power structures, and considering their input crucial, resultant interpretability systems will fit the standards and needs of those in power — for example, executives with a vested interest in maintaining the status quo, or engineers and researchers who might communicate about model decisions in a way that is not understandable to people without formal ML knowledge. Involving stakeholders with different interests first requires *reflexivity*, or explicitly acknowledging what *our own* backgrounds and interests are. However, doing so in the abstract can be difficult. While our framework was primarily designed to describe the external stakeholders of interpretable ML, we believe it can also be turned to focus internally on the participants of the interpretability design process. By using it to describe our knowledge and goals, we can more clearly recognize gaps in our own knowledge and, thus, the additional people we must deliberately include.

3.5 Limitations and Future Work

In this chapter, we present a framework to characterize the stakeholders of interpretable ML, and their needs. Our framework depicts stakeholder expertise as a two-dimensional space that describes the knowledge they possess (formal, instrumental, and personal knowledge), and the contexts in which this knowledge manifests (ML, the data domain, and the milieu). Our framework also details stakeholder needs as a three-level typology of long-term goals (understanding the model, and building

trust in it), shorter-term objectives that build towards these goals (e.g., debugging a model, or contesting a decision), and finally immediate tasks that stakeholders can perform to meet their objectives (e.g., assess prediction reliability, and detect mistakes). In evaluating our framework, we find that it suitably covers a sample of 58 papers on ML interpretability, and its granular structure reveals gaps in the literature. Moreover, while speculative, we believe the framework offers the necessary vocabulary to assist in more precisely comparing and conducting user-focused evaluations of interpretability systems. Finally, we find that the framework offers a richer intersection of stakeholder expertise and needs than prior approaches, and that it can be turned inward to facilitate a more reflexive design process.

Our framework takes the next step in better defining who the users of interpretable ML are, and its limitations point to promising opportunities for future work. In particular, we do not consider our framework to be an exhaustive description of the problem space, but rather a "living" artifact that will grow and adapt as interpretability matures as a research field. For example, we expect new goals, objectives, and tasks to be added to the framework as ML is deployed more deeply in existing domains, and as it reaches new domains. Indeed, when coding the interpretability literature, we found occasional instances of needs that do not precisely fit into our current framework (e.g., persuasion, adoption). But, more evidence is needed to determine at what level of the typology these needs fit, and whether they are specific instances of a more general or fundamental need.

Similarly, while our framework begins to decompose expertise into knowledge and contexts, the models we base it on provide even more granularity. For example, Fleck [121] names several additional types of knowledge including informal knowledge, contingent knowledge, tacit knowledge, and meta-knowledge; and Eraut [112] identifies cultural and tacit knowledge, and the degree to which either have or have

not been codified. Under our framework, these different types all lie within personal knowledge since we did not find sufficient evidence in the interpretability literature to warrant the additional granularity. The milieu context is similarly broad — covering physical, social, and cultural contexts in the literature. As additional work on interpretable ML is conducted, these two broad categories may come under the same pressure we initially identified with prior expertise- and role-based approaches: they may begin to conflate otherwise independent concerns. By identifying recurring instances of these tensions, we can begin to disentangle them and enumerate other knowledge types and contexts that are meaningful for interpretability.

Finally, our framework's model of expertise is grounded only in epistemology, but the literature on expertise also frames expertise as constructed rhetorically. As Johanna Hartelius describes, *"[a] speaker is only able to exercise expertise and enjoy expert status to the extent that she can persuade an audience to grant such things"* [153]. Rhetoric undoubtedly plays an important role in interpretability, and we can see evidence for this in the adjacent domain of data visualization [75]. Researchers have argued that the clean, minimalist aesthetic of modern visualizations lends them an air of authority and certainty [182] that contributes to their *"persuasive and seductive rhetorical force"* [105]. Through close readings of visualizations, researchers have shown how citing sources and representing uncertainty can signal transparency and impartiality [170], and with empirical studies, researchers have demonstrated that even seemingly-innocuous elements like titles can frame or slant reading visualizations [194] and impact trust or recall [195]. How to adapt and replicate these findings for interpretability is a fertile ground for future work, and interpretability poses its own unique considerations. In particular, unlike visualizations, the rhetorical performance of an interpretability interface may sometimes be shared with or mediated by a human (e.g., an "operator" [323] or through reports [166], respectively).

Table 3.2: Knowledge types and contexts for interpretability stakeholders, with examples identified in our literature survey. We found a range of expertise and backgrounds under our framework, highlighting information that might be lost if XAI designers only consider a small set of roles like “ML expert" and “non-expert", as we have widely observed in past work.

	Knowledge		
Context	**Formal**	**Instrumental**	**Personal**
ML	model developers [303], computer science students [339], machine learning scientists and researchers [235, 231, 98, 281], model builders/engineers [37, 284, 32, 274, 323, 353, 117], model analyst [186], general [357] **(15/58)**	model users [303], data scientists/ML practitioners [179, 12, 339], autonomous robot developers [321], data scientists [32, 166, 5, 42], “medical experts’ increasing familiarity with [computer-aided diagnosis] systems" [55], “domain experts who use machine learning for analysis" [235, 117, 325], practitioners [357, 215], optimization expertise [37], students with some ML familiarity [49], greater machine learning community [134], developers/implementers [274, 323, 293, 135, 337] **(23/58)**	“intuition of how the network looks" [357], “[ML] researchers’ intuition of what constitutes a ’good’ explanation" [231] **(2/58)**
Data Domain	biologists [303], robotics [321], scientists [284], doctors [55, 330, 98, 309, 350, 186], “enrolled in law school" [207], judges [19], agronomic engineers [37], regulators [32], HR managers who produce expert estimates [326], energy data operators [31], “business logic" [323], game theorists [337], general [18, 281] **(19/58)**	“model novices" interested in applying ML to specific domains [303], “deep knowledge of the circumstances for employee retention" [12], sign-language learners [261], domain knowledge to verify ML results qualitatively [292], “only specialists in part of the underlying process" [37], internal financial auditors [273], clinicians [55, 293, 98, 325], “increasingly adopt ML for optimizing and producing scientific outcomes" [284], operators [323, 356], peer grading in online education [188], general [326, 166, 67] **(17/58)**	“without accurate mental models, social factors can rationalize suspicious observations [about explanations]" [179], “how well the system’s conceptual model fits their mental model" [68], mental models of the system to generalize the AI behavior [231, 356], patient/client/decision subject [323, 330, 293, 337, 42, 281], “hold preconception of what constitutes useful explanations for decisions" [215], “prognosticating their patient’s condition in their personal experience" [325] **(12/58)**
Milieu	students studying information policy [339], ethicists [274], bodies like institutional review boards or ethics committees [330], “understanding requirements arising from social contexts other than just from usability or human cognitive psychology" [5] **(4/58)**	“community hospital small groups, to academic medical centers" [55], “use AI products in daily life" [235], UX/design practitioners [215, 353], data subject [274, 323], product managers [166], examiners/auditors [323], departments adopting decision-support technologies [325], use of AI in government and industry [42] **(10/58)**	loan applicant [12, 327], “different cultural, demographic or phenotypic groups" [273], recommender system users [363, 196], “actors bring their own points of view and own priorities" [344], “people employ certain biases and social expectations" [231], “anticipating the situated, user-encountered capability of AI is difficult" [353], familiarity with privacy and personal data issues [117], individual fatigue and workflow issues in heathcare [325], general [326, 293, 19] **(13/58)**

Table 3.3: Goals, objectives, and tasks for interpretability stakeholders. Our literature survey identified instances of theoretical and systems work that discuss or address these needs.

Stakeholder Need	References
G1: Understanding the model	"machine models" [234, 339], "understand the agent's behavior and responses enough to participate in the mixed-initiative execution process" [135], "to attain scientific outcomes with ML one wants an understanding" [284], "understand the 'algorithmic decision model'" [67], general [55, 18, 134, 85, 337, 98, 5, 356, 42, 249, 51] **(16/58)**
G2: building trust in the model	mechanisms for steering trust building [303], build appropriate trust [19, 49], mechanistic interpretation needed for trust building [330], trust for tool adoption and continued use [188], ensure that ML models reflect appropriate values [186], general [207, 235, 50, 134, 231, 166, 323, 85, 215, 220, 135, 5, 356, 17, 117, 325, 42, 67, 249, 51] **(26/58)**
O1: Debug or improve a model	model refinement [303, 186], help data experts to tune ML parameters for the data [235, 357, 309], "identify issues with a model and how to fix it" or debug and optimize [166, 164, 337, 249, 287, 292, 321, 339], improve an aspect or part of a system [117], general [68, 284, 32, 274, 7, 215, 281] **(21/58)**
O2: Ensure compliance with standards or regulations	adherence to standards and laws like GDPR and "right to explanation" [273, 134, 7, 98, 117, 281, 196, 287], forensics [323], justify clinical validation of ML in medical studies [330], facilitate monitoring for safety standards [337], general [303, 42, 284, 32, 220, 274] **(17/58)**
O3: Understand how to incorporate the model's output into downstream actions	learn "factors that could be changed to improve their profile for possible approval in the future" [12], learn how to correct actions based on model feedback [261], apply own domain-related decision-making using the XAI or not [49], make better or faster decisions [249], understand impact of prediction on other system components [215], understand how to get a desired outcome [327], understand consequences [281], understand errors for safety-oriented task [98], directing use in patient or medical work practice [325, 309, 350], general [326, 17] **(13/58)**
O4: Justify or explain actions influenced by a model's output.	justify the user's decision-making [303, 12, 325], reason about data outputs [179], explaining findings to collaborators [37], "enables the user to consider contrastive explanations... why one decision was made instead of another" [50], explain causes of an event [231, 337, 5], recommend treatment options to patient [323], "justify the result" [7, 49, 363, 51], general [55, 19, 164, 357, 166, 284, 85] **(21/58)**
O5: Understand how one's data is being used	"disclose what user data is being used in algorithmic decision-making" [235], know how one's data is being used to make decisions about others [323], understand why certain user data is collected [363], general [12, 68, 220] **(6/58)**
O6: Learn about a domain	learn about sign language and how to use it correctly [261], learning about ML [357], explanation "as a vehicle to generate insights about the phenomena described by the data" [166, 220], learn how to solve a task [293], learn game strategy (Go) [287], learn new facts/gain knowledge [7, 98, 281], learn design strategies [31] **(10/58)**
O7: Contest a decision made based on the model's output	"when I see things I don't completely agree with" [55], "present an incontestable subset of reasons to the bank employee" [68], contest a discriminatory decision [327], general [344, 323, 220, 337] **(7/58)**
T1: Assess reliability of a given prediction	identify and explain an outlier [37], increase or decrease trust in the model based on observed accuracy, relative or not to one's own performance [19, 356], "to ensure the scientific value of the outcome" [284], assess the AI's judgment [215] **(5/58)**
T2: Detect mistaken, discriminatory, or arbitrary behavior.	"anticipate ethics-related failures before launch" [273], bias or mistake detection [344, 281, 249, 287], understand skewness and biases in input data [85], find unknown vulnerabilities and flaws [7, 98] **(8/58)**
T3: Understand the extent of the information the model is using	data entanglement [273], be informed when the ML is not suitable for particular systems [284], understand "what the system was sensing to make its inferences" [326, 135, 67, 363] **(6/58)**
T4: Understand the influence of different factors on the model's output.	explore counterfactuals and how changes to data points affect predictions [339], understand model prediction mechanisms [166, 67], "factors influencing their individual decision" [68, 51, 327], "inspect how output changes with instance changes" [215, 220], "how drift in feature distributions would impact model outcomes" [32], "did the factor 'race' influence the outcome of the system" [274], "feedforward can help people understand and predict what is going to happen" [5] **(11/58)**
T5: Understand model strengths and limitations	understand model error from predictions [18], know when to trust the prediction or be skeptical [19, 166], understand limitations [215], "clarity around why the model under-performs" [325] **(5/58)**

Chapter 4

Applying Participatory Methods throughout the ML Lifecycle

This chapter considers the case where we are able to shape the development process for a new system from the start. We ask how, recognizing the many development decisions throughout the ML lifecycle (i.e., as described in Chapter 2), how could we proactively design each step to be suited to specific downstream contexts and stakeholders? We address this question through an in-depth case study of co-designing datasets and machine learning models to support the efforts of activists who collect and monitor data about feminicide — gender-related killings of women and girls. We use intersectional feminist goals and participatory design as a guide to shape each development choice described in Chapter 2, from problem conceptualization to data collection to model evaluation, and highlight several resulting methodological contributions.

4.1 Introduction

Work in data ethics or AI/ML fairness often focuses on "mitigating bias" in harmful systems, building "fair" or "transparent" algorithms, or performing retroactive audits. While these developments are important, they typically locate the source of injustice in individual people or specific technical systems, and solutions that emerge often take the form of "technological Band-Aids" [95, p. 60].

Alternate framings of data and algorithms are emerging that are rooted in considerations of power and justice [87, 88, 95, 28, 246, 44]. These trace the root cause of "biased" systems not to individual programmers or design decisions, but rather, to the deeper structural inequalities in which data-driven systems are embedded. Inspired by this framing, and with a feminist lens, we ask the question posed by D'Ignazio and Klein in *Data Feminism*: "why should we settle for retroactive audits of potentially flawed systems if we can design with a goal of co-liberation from the start?" [95, p. 63]

However, to our knowledge, there are not yet examples of how to apply feminist and participatory methodologies throughout the entire machine learning (ML) life cycle, to conceptualize and design ML tools that center and aim to challenge power inequalities. We target this more prospective goal through a concrete case study that asks how digital, data-driven tools can support the efforts of activists who collect and monitor data on the topic of feminicide (or femicide) — broadly understood as gender-related killings of women and girls. Many of these activist organizations use news articles to find and record instances of feminicide in their region. This process typically relies on using search queries that return a large fraction of irrelevant results, adding to the arduous labor of monitoring this kind of violence [96]. Through co-design sessions with groups, we conceptualized, built, and deployed an ML-based

system to deliver more relevant media results to activists on a regular basis, thus helping to facilitate their monitoring work.

In our initial pilot of the system, we found that it worked well for groups that broadly monitored all feminicides in a given region. However, the results were consistently not relevant for groups who focused on specific, racialized forms of feminicide (e.g., Black women in the US killed by police). In this chapter, we focus on our subsequent efforts to perform iterative data collection, modeling and evaluation steps to build out context-specific models for two organizations that monitor gender-related killing as it intersects with white supremacy, state violence and colonialism. Throughout this process, we take an explicitly feminist approach, both in our overaching process—which we strive to make iterative, reflexive, contextual, and participatory—as well as the technology we build. In particular, we draw on four principles of data feminism that are most salient to our work: *challenge power, embrace pluralism, consider context*, and *rethink binaries and hierarchies*. We describe how these goals shaped each stage of our approach, from problem conceptualization to data collection to model evaluation.

We highlight several methodological contributions, including 1) a data collection process in which we iteratively identify and incorporate context-specific examples that target model weaknesses, 2) annotation and modeling methods that incorporate sociohistorical context and explicitly focus on intersectional identities, and 3) a three-stage evaluation process — with quantitative, qualitative and participatory steps — focused on real-world, context-specific usefulness.

At the same time, we acknowledge the ongoing tensions we grappled with, and areas where our tools currently fall short. In doing so, our contribution is two-fold: we describe our approach to this specific case study in detail, but also aim to provide inspiration for how to mobilize intersectional feminist values in technology more

generally. We conclude with the idea that intersectional feminist and participatory ML is possible, but that the creators of such systems should consider themselves in the humble and bounded role of supporting and sustaining activist efforts to shift power.

4.2 Background and Related Work

4.2.1 Power, Oppression, and Intersectional Feminism

Our work builds on intersectional feminist thought, and stems from the acknowledgement that power is not equally distributed in the world. By *power*, we refer to configurations of structural privilege, in which certain groups experience unearned advantages — i.e., because they control the dominant institutions of law, education and culture [95, 73, p. 24]. *Systems of oppression* arise because of the unequal distribution of power, and involve the systematic mistreatment of certain groups of people by others. There are many dominant and marginalized identities in society, and forms of oppression manifest differently across them. For example, gender oppression takes the form of cissexism, heterosexism and patriarchy, while racial oppression manifests in racism and white supremacy. Other forms of oppression include ableism, colonialism and classism.

These various dimensions of disempowerment converge in complex and unique ways for groups and individuals [79]. The concept of *intersectionality* provides a contrast to single-axis or additive views of how discrimination manifests. Intersectional feminism is grounded in a long history of Black feminist thought stretching back to at least the mid-1800s [245] and into the present day [72], and conceptualized in particular by Kimberlé Crenshaw [78] and the Combahee River Collective

[70]. Intersectionality comprises an analytical framework based on the premise that different systems of power are interdependent, "mutually constructing one another" and producing complex social inequalities that fundamentally shape both individual and group experiences [72, p. 16].

An intersectional view situates all systems of oppression as interlocking axes that construct what Patricia Hill Collins names the *matrix of domination* [73]. The matrix of domination describes four interrelated domains that operate at different scales of granularity, from interpersonal to institutional, shaping society and human actions. These include the structural domain, where oppression is organized and codified in law and policy; the disciplinary domain, where it is enforced through bureaucracy and hierarchy; the hegemonic domain, in which oppressive ideas are circulated through culture and media; and the interpersonal domain, which captures the everyday lived experiences of individuals. Different systems of oppression manifest across each domain to different degrees and generate dynamics of subordination and vulnerability in varying ways. Ultimately, they converge to create very real consequences, such as the invisibility of violence against women of color [79].

Together, intersectionality and the matrix of domination show that there are not clean cut distinctions between victims and oppressors; rather, each individual "derives varying amounts of penalty and privilege from the multiple systems of oppression which frame everyone's lives" [73]. This understanding motivates the goal of *co-liberation*: the idea that dismantling interlocking systems of oppression is necessary for everyone's collective freedom.

4.2.2 Participatory and Feminist ML

While intersectionality and the matrix of domination are *conceptual models* for how inequality is structured and reinforced, expanding participation is often suggested as a *method* for rectifying imbalances of power. Sloane *et al.* [300] survey three main forms that "participation" has taken in ML. Most frequently, much of ML involves *participation as work*, where people (often unknowingly) contribute examples [127] and annotations [257] to large-scale datasets, e.g., through systems that monitor activity [264] or scrape web data [90]. This work is typically unacknowledged and poorly compensated (if at all), and it does not meaningfully integrate user perspectives [29, 140].

More recently, there has been increasing awareness of the importance of more meaningful community and end user participation during the development of machine learning systems [199]. This can take the form of *participation as consultation* [300], where stakeholders are consulted at specific points throughout the development process for need-finding or feedback [225, 54, 39]. While potentially promising, this setup is inherently top-down, "designing *for*" groups rather than committing to their ongoing inclusion [128, 226, 338]. In contrast, *participation as justice* involves more long-term relationships with participants "based on mutual benefit, reciprocity, equity and justice" [300]. These types of approaches focus on "designing *with*" communities to ensure outcomes are valuable to diverse and minoritized stakeholders [152]. Participation as justice — i.e., centering the voices of marginalized groups throughout the whole ML life cycle — is critical if we want to design for the goal of co-liberation.

While such participatory approaches have been more widely explored in some domains (e.g., participatory action research [26], design justice [76], disability justice

[145], environmental justice [10]), they are fairly rare in practice during ML development. Some examples from recent years focus on community-oriented educational materials: for example, the Algorithmic Equity Toolkit is a set of tools for recognizing and understanding algorithmic systems co-designed with community stakeholders [178]. Similarly, "A People's Guide to AI" [255] aims to create accessible educational materials about AI tools and their consequences. Others target specific points during the development process: e.g., the Contextual Analysis of Social Media (CASM) approach is a method for data labeling in which community expertise shapes a team-based annotation process [260]. In our work, we use intersectional feminist thought as a guide for understanding how to incorporate meaningful participation throughout the entire ML life cycle, from problem conceptualization to system evaluation and deployment.

Other work has considered the implications of feminist epistemologies to areas of ML. For example, Hancox-Li and Kumar [146] apply the frameworks of situated knowledge and standpoint theory [149] to understand the values implicit in feature importance methods. Barabas *et al.* [20] explore how the concept of "studying up" [242] could be used to reorient ML research questions to better confront power. Gray and Witt [139] outline a mix of technical, social and cultural interventions that constitute a feminist data ethics of care approach to ML. And Buolamwini and Gebru [46] bring an intersectional lens to model evaluation, considering the performance of facial analysis systems not only across skin tones or across genders, but also across specific intersections of them. Our own work specifically builds on the framework laid out in *Data Feminism*, which introduces seven principles for thinking about and using data that are "informed by direct experience, by a commitment to action, and by intersectional feminist thought" [95].

4.2.3 Feminicide, Counterdata Collection and Media Analysis

The term femicide [271] is broadly understood to mean the gender-related killings of women and girls [123]. Latin American feminists have built on this work and introduced the term *feminicidio (feminicide)*, as a way to capture the systemic nature of this violence and the role of the state in enabling it through either omission, negligence or complicity [204]. Activists in Latin America have also played a pivotal role in bringing worldwide attention to the issue of feminicide through powerful demonstrations and movement-building, and eighteen countries have instituted legislation criminalizing feminicide [61]. However, policies to ensure adequate information collection have not followed, and official government data on gender violence and feminicide remain incomplete, difficult to access, infrequently updated, contested, and underreported due to stigma, victim-blaming, or matters of legal interpretation [38, 129, 181, 290, 333]. Incomplete or inaccurate records of feminicide can be understood as *missing data*, resulting from power imbalances — across all four domains of the matrix of domination — in the collection environment. Missing data masks the systemic nature of this violence, allowing it to go unpunished. In this sense, the lack of information can be interpreted as an active form of "women disempowerment" [203], in the constructed process of gendered delegitimization that results from heteroimposed "patriarchal pacts" [9] and normative violence [48]. "In such a framing, women are set up to be forgettable. Ignorable. Dispensable — from culture, from history, and from data. And so, women become invisible" [80, p. 21]. This is particularly true for groups at the intersection of patriarchy and other forces of domination, like settler colonialism. The movement around Missing and Murdered Indigenous Women, Girls and Two Spirit people (MMIWG2) was founded to challenge the invisibility of this violence.

When the state and its institutions fail to collect important data, civil society organizations increasingly step in to fill these gaps, performing counterdata collection as a way to regain empowerment, legitimization, and visibility [95, 229, 82]. Counterdata science practices mount an explicit challenge to the data practices (collection, analysis, deployment, visualization, ethics, values) of mainstream, well-resourced "counting institutions" such as governments and corporations [94]. As Alice Driver notes in the case of Mexico, "the most accurate records of feminicide are still kept by individuals, researchers, and journalists, rather than by the police or a state or federal institution" [104, p. 7].

In the absence of reliable "official" data sources, anti-feminicide activists in the field have built significant data acquisition pipelines to support creating databases of incidents from media reports. From a data perspective, this work is similar to media analysis projects undertaken in computational social science where informatic tasks include event detection, content extraction, classification, and entity extraction. Platforms such as Media Cloud [283] and GDelt [212] aggregate and collate news stories from the open web to support media research projects. Various projects support extracting and annotating content from news corpora to identify entities, dates, and more [119, 167]. Researchers have combined those systems, and built others, to study the emergence of protest movements, creating automated classification systems to analyze their representation in the news [147]. Others have built systems to automatically detect and extract the names of victims of police killings in the US [180], and analyze shifts in narrative frames employed by the media to discuss them [367].

4.3 Case Study: Data Against Feminicide

The Data Against Feminicide project — co-organized by the Data + Feminism Lab at MIT, Feminicidio Uruguay, and the Latin American Initiative for Open Data (ILDA) — is an initiative intended to support feminicide data activists through knowledge sharing, technology development and community building. Since 2020, the project has conducted semi-structured interviews with 31 monitoring organizations based mainly in the Americas, with a focus on Latin America. Through these interviews, it became clear that most organizations use media articles to identify cases of feminicide in their regions [96]. However, a consistent issue is that many of the articles retrieved via search queries are not relevant. In practice, activists spend much of their time reading articles that are not instances of feminicide (but that might describe other violent or traumatizing events) in order to find the minority that are feminicide, which is both emotionally taxing and time consuming.

Through co-design sessions with groups, we conceptualized, built, and deployed an ML-based system to deliver more relevant media results to activists on a regular basis (referred to from here as Email Alerts). The system uses news content from Media Cloud, an open source platform for analysis of online news [283]. An organization using the Email Alerts system can customize a search query and set of place-based media sources to best suit their project needs. Media Cloud then retrieves matching articles from its continually updated database of global news stories, which are run through a machine learning model we developed that predicts the probability that the article will be relevant to the organization. Articles above a particular probability threshold (which defaults to 0.75) are sorted by the probability of feminicide and delivered in a daily email digest (matching activists' existing workflows) and can also be viewed in an online dashboard.

Our focus in this chapter is on the development and evaluation of the ML models used to filter articles. For our initial prototype, we collected and annotated two datasets of 399 and 424 articles, respectively: the first in English, in collaboration with Women Count USA (a US-based activist group), and the second in Spanish, in collaboration with Feminicidio Uruguay (a Uruguay-based activist group). This data was used to train two language-specific logistic regression models to predict the probability of feminicide from the text of an article. The English and Spanish models achieved 84.8% and 81.6% accuracy, respectively, in 5-fold cross-validation. Further details about data collection, annotation, and model performance for this initial iteration can be found in D'Ignazio *et al.* [110].

Our results with this initial version of the model served as a proof-of-concept that such a system could reduce the burden of labor for activists in this space. In Spring 2021, we ran a two-month pilot with seven groups to gauge if and how it would help in practice. The pilot was run simultaneously in Spanish and English, with four groups based in the United States, one group in Uruguay, and two groups in Argentina. Among the four English groups, we received dramatically different feedback about the model's performance. Women Count USA, which monitors all US feminicides, reported that the results were overall very relevant and useful. Another organization, Black Femicide US, monitors femicides of Black women and reported mixed but still useful results, with around 4 out of every 10 articles the system sent being relevant. Both have continued using the system in their work. However, the system did not source relevant results for two other organizations in particular, each of which monitor specific, intersectional types of feminicide: 1) Sovereign Bodies Institute (SBI), a group that tracks missing and murdered Indigenous women, girls, and two spirit people (MMIWG2) and 2) the African American Policy Forum (AAPF), which monitors police violence against Black women as part of the

#SayHerName campaign. Feedback from these groups consistently showed a lack of relevant articles being returned by the system, despite modifying the search queries to add relevant terms. The groups' frustration could be heard in comments in focus groups and weekly surveys—for example, an activist from SBI wrote, "The majority of articles are not relevant to our focus, which means I'm actually spending more time than usual trawling through potential additions because I'm reviewing so many more news articles than usual." As we reached the conclusion of the pilot, it became apparent that groups dealing with general feminicides (i.e., all women killed in a specific region) were much more satisfied with the tools than groups that monitored more intersectional forms of such violence.

Our participatory development and evaluation processes made it clear that the model needed to be adjusted to better serve projects with more targeted monitoring needs. A commitment to intersectionality means that systems should work not only for the mainstream, majority use case, but also for those on the margins; and it means acknowledging that this might require dedicating additional time and resources to these specific intersectional use cases. With agreement from the two groups, we went through further iterative data collection, modeling and evaluation steps to deploy new models for their specific monitoring needs. The rest of the chapter focuses on motivating and describing this development process, the resulting models, and the underlying theoretical prompts this project creates for those working to create participatory approaches to machine learning informed by intersectional feminism.

4.4 An Intersectional Feminist Approach to ML

Throughout this project, we strive to take an explicitly feminist approach, both in the technology we built and our overarching process. To do so, we build on the

intersectional feminist principles proposed by *Data Feminism* [95]. In this section, we focus on four principles that are most salient to our work: *challenge power, embrace pluralism, consider context, and rethink binaries and hierarchies*. We describe how they shaped our research questions, our approach, and the resultant tools we built. At the same time, we acknowledge the ongoing tensions we grappled with, and areas where our tools currently fall short. Our goal is both to illustrate our approach to this specific case study, as well as provide inspiration for how to mobilize intersectional feminist values in technology more generally.

4.4.1 Challenge Power

Focusing on the problem of gender-related violence might seem like it is inherently challenging power. However, there are many possible directions within this space that could be pursued — not all of which challenge power to the same extent. A deeper examination of how unequal power manifests in each domain of the matrix of domination guides what we choose to build and who we work with. For example, in the structural domain, we understand that data about feminicide is missing in large part because the state and its institutions neglect to collect and report adequate information about it. Activists who collect counterdata challenge and hold these institutions accountable, reclaiming power in the process. With an intersectional lens, we can further understand how unequal power manifests differently for people or groups fighting multiple, intersecting systems of oppression. In the disciplinary domain, for example, victim blaming and a failure to investigate are particularly prominent when the victim is Black or when the case involves police violence [108]. In the hegemonic domain, biased media narratives misgender or ignore trans people, disregard Indigenous identity, or stigmatize and blame sex workers for violence

inflicted on them [346, 306].

Throughout our work, then, rather than support governmental organizations or large international NGOs, we collaborate with and build tools in service of civil society activists. Moreover, to truly challenge power involves not only getting our tools to work for the broadest, majority group, but also for those on the margins who are face multiple intersecting oppressions. Here, we choose to work with a range of organizations monitoring feminicides, including those with specific, intersectional focuses (e.g., Black women in the US killed by the police). And in practice, throughout the development process, the problems we focus on center around getting our system to work equally well for groups monitoring intersectional violence, rather than only improving upon a singular version that primarily benefits the general, majority cases.

4.4.2 Embrace Pluralism

To embrace pluralism means insisting that "the most complete knowledge comes from synthesizing multiple perspectives, with priority given to local, Indigenous, and experiential ways of knowing" [95, p. 125]. Embracing pluralism explicitly calls for the use of participatory methods throughout the ML process, from project inception through deployment, in support of "locally informed, ground-truthed insights that derive from many perspectives."

This principle fundamentally shapes our research process, which is iterative and participatory. The project is co-led by three people with diverse backgrounds and positionalities, including a counterdata activist who helped surface the value of activists' situated knowledge from the start. The ideas we chose to develop were brainstormed in participatory co-design sessions with two activist groups working in different contexts, and piloted with seven organizations across four countries. A

pluralistic process helps enable *conceptual pluralism* — wherein the framing of the problem is cognisant of and affirms the value of multiple perspectives rather than enforcing a single ground truth. For example, while developing our datasets and models, we collected and annotated data in collaboration with each group, with the understanding that a case that was relevant to one group might not be relevant to another, and vice versa. And importantly, our goal is to support the work that activists are already doing — not to replace or override it.

4.4.3 Consider Context

To consider context means acknowledging that "data are not neutral or objective" [95, p. 149]. Rather, data — and missing data — are the product of unequal social relations, and this context is critical for accurate and ethical analysis.

In our project, the importance of context became clear after our first phase of model development, in which we used the same language-specific feminicide-detection model for each of the pilot organizations. While some groups (in particular, those that broadly monitored all types of feminicide) found they were receiving relevant results, others (those that focused on specific, intersectional cases, such as MMIWG2) struggled to find any relevant results at all. If we think about the data (which in our case, are news articles) as being produced within the context of intersecting systems of oppression, we can understand that the way that different forms of violence are reported about (as well as if they are reported about at all) can be vastly different. For example, cases of MMIWG2 are often under-reported; and when they are reported, the articles often omit Indigenous identity or other key information [133]. Cases involving police violence are often written with biased, victim-blaming narratives [108]. In addition, we found that many US-based Indigenous news sources are

published as PDFs, a model of distribution that wasn't readily ingestible by Media Cloud, which was built to support the dominant form of RSS-based syndication and distribution. As we found, by not considering this context, a one-size-fits-all model fails on these types of cases.

In subsequent rounds of development and evaluation, we built out different models for groups that work in different contexts and face different challenges. In part, this involved iterative, context-specific data collection/annotation in collaboration with each group. Our evaluation is also multi-step and contextual, acknowledging that a model that works for one group might not work for another, and that a model that performs well on a test sample might perform differently once deployed in a real-world context.

4.4.4 Rethink Binaries and Hierarchies

Rethinking binaries and hierarchies means "challenging the gender binary, along with other systems of counting and classification that perpetuate oppression" [95, p. 18]. As mentioned previously, we intentionally work not only with groups that view their work as recording "feminicide," but also with those that monitor other types of gendered and racialized violence. We annotate the datasets for each group in accordance with what they consider relevant, rather than an arbitrary binary gender label.

However, our ability to challenge binaries and categorization schemes is an ongoing limitation of this work. While many activist organizations do include murders of trans people in their definition of feminicide, for example, these cases tend to be recorded much less frequently. The reasons for this are complex, but include a lack of legal protections in the structural domain, misgendering and victim-blaming in the disciplinary domain, and media bias in the hegemonic domain. LGBTQ+ ac-

tivists have overcome some of these barriers by relying on community members and ally networks to identify trans violence. News article-based ML classifiers would not be as helpful for this work, and may even perpetuate erasure by learning victims' incorrect genders.

Even beyond gender, choosing categorization schemes to use in each dataset brought up unresolved tensions. For example, AAPF focuses on recording cases of Black women in the US killed in police violence. We annotated the corresponding dataset with "police violence" and "feminicide" labels, but did not include a "race" annotation due to the difficulty of inferring it from most news articles. As a result, ML classifiers trained on this data are not able to capture the specific intersectional cases of interest, and we aren't able to internally evaluate the model's performance on Black women specifically.

4.5 Developing Context-Specific Feminicide Detection Models

In this section, we describe the methods we used to develop and evaluate context-specific models for SBI and AAPF. This consists of four main steps: 1) data collection, where we iteratively collect context-specific examples to target model weaknesses, 2) data annotation, where we re-annotate data with multiple relevant attributes, 3) modeling, where we explore how to build and combine models that center specific intersectional identities, and 4) evaluation, where we define appropriate metrics of success and how to assess them.

4.5.1 Data Collection

For SBI and AAPF, we collected additional data and trained models in an iterative process to target model weaknesses. The data we use are news articles, and a single example is a string containing the article full text.

Round 1.

The initial pilot model was trained with general feminicide cases as positive examples and non-feminicides (usually other crimes) as negative examples. While this model was useful for broadly monitoring feminicide, it was not context-specific, and returned mostly unrelated cases for groups with a specific intersectional focus.

Round 2.

In the subsequent round of context-specific data collection, we asked groups to send us articles they had already collected in their existing databases, which we used as positive examples. In line with *embracing pluralism*, this entailed a shift from a predefined notion of "feminicide" to a framing of predicting "relevant cases" for a particular group. We used a sample of cases from our initial dataset (both general feminicides and non-feminicides) as negative examples. We then trained models using this data, and deployed them to the Email Alerts system, to monitor their performance in a real-world context (i.e., on the thousands of unseen articles pulled in daily from Media Cloud).

While the returned articles were vaguely relevant to the context (e.g., related to police violence), we still found that they were not finding the specific cases of interest. With AAPF, for example, we found that the list of returned articles was often dominated by police violence against Black men or cases where the police were

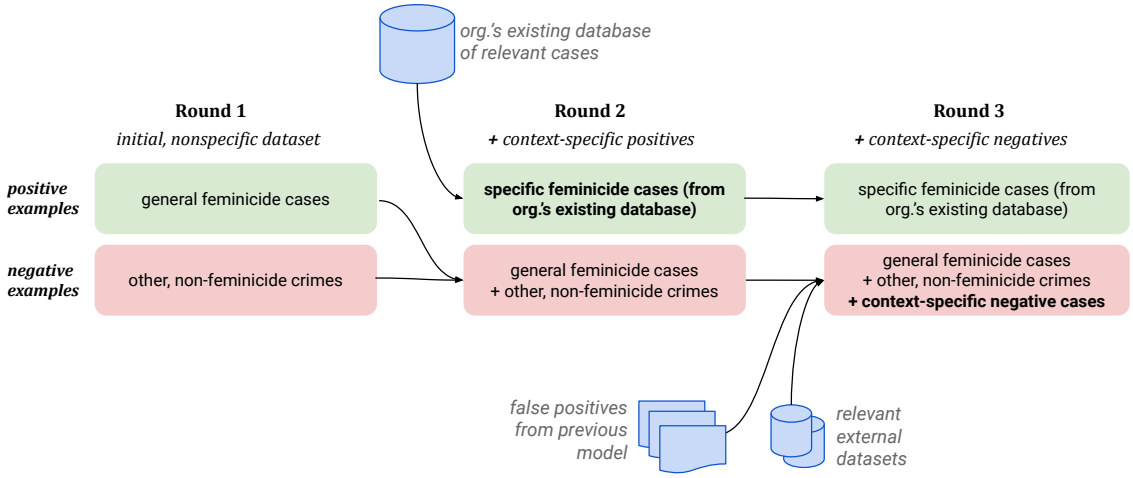


Figure 4-1: Our data collection process involves iteratively collecting context-specific positives (e.g., by sourcing ground-truth articles from organizations' existing databases) and context-specific negatives (e.g., by identifying and collecting types of negative examples close to the decision boundary, for which the model is underspecified).

investigating other violence, both of which are much more common in the media than Black women killed by the police. This shortcoming may be due to an underspecified decision boundary: while the positive examples we curated from the groups reflected specific and intersectional cases of interest, the negative examples were much more general, and did not include many cases close to the decision boundary. In this failure mode, the model may learn that positive cases involve police violence, but nothing drives it to learn the more specific intersectional identities of interest.

For SBI, we noticed a similar pattern — many of the highly-scored articles were generally about MMIWG2 developments (e.g., task forces being formed), rather than cases of a particular victim. Without context-specific negatives (e.g., MMIWG2-related articles that *don't* describe a particular case of feminicide), the model may have learned that any MMIWG2 terms indicate a positive article.

Round 3.

In the next round of data collection, we focused on collecting more tailored *negative* examples, using both select external datasets as well as a sample of the previously unseen false positives returned by the previous model. For AAPF, we focused on collecting articles about police violence against Black men and cases where the police were investigating non-police-related violence. We sourced articles from external datasets such as the Washington Post's *Fatal Force* dataset [267], Guardian's *The Counted* dataset [318], and the *Whose Deaths Matter?* dataset [367], as well as a sample of the new false positives returned by the prior model. For SBI, we collected context-specific negative examples of Indigenous non-feminicides (e.g. missing or murdered Indigenous men, or articles generally related to the MMIWG2 movement but not referencing a specific case) by sampling new false positives.

The data collection process for both organizations highlights the importance of targeted data collection to find negative examples: while activists provided lists of articles that can be used as positives, our work relies on iterative error analysis and data collection to develop a classifier with high specificity.

4.5.2 Data Annotation

Even given more context-specific data, the model may still be limited by the default binary classification setup, in which examples either have a positive or negative label. In the datasets we compiled for AAPF and SBI, a positive label indicates the intersection of multiple identities and/or attributes. For AAPF, for instance, a positive example is one where there was police violence *and* the victim was a Black woman. For SBI, a positive example is one where the victim is Indigenous *and* a woman, girl, or two spirit person. With just a single, positive label, we leave it to

		Feminicide	
		Y	**N**
Police Violence	**Y**	249	168
	N	189	136

Table 4.1: Data breakdown for AAPF.

the model to learn the complex decision boundary delineating examples with these intersecting attributes.

Moreover, treating all negative examples as equally irrelevant misses the ways in which they may actually be similar to the positive examples. For example, for AAPF, a case where police shot a Black man and a case where a white woman was killed by an intimate partner are both given the same negative label. However, the former shares the institutional violence and racism present in the positive cases, while the latter shares the root causes of sexism and patriarchy.

Annotating articles with multiple attributes allows us to build models that explicitly take into account the ways in which different systems of oppression manifest and interplay. For AAPF, we re-annotated each example with two labels: "police violence" and "feminicide". For SBI, we annotated articles with both "MMIWG2-related/Indigenous news" and "feminicide or missing victim". The breakdown of examples (after additional data collection and re-annotation) in the resulting datasets is shown in Tables 4.1 and 4.2. In both cases, the *intersection* of the two new annotations represents a "positive" case. This framing allows us to incorporate this prior knowledge of the domain into the model architecture rather than expecting the model to automatically learn the specific intersection of interest. We expand upon these modeling options in the following section.

		Feminicide or missing victim	
		Y	**N**
MMIWG2-related or Indigenous news	**Y**	217	110
	N	201	151

Table 4.2: Data breakdown for SBI.

4.5.3 Model Development

Given a context-specific set of examples and multiple annotations, we explored a few different modeling approaches to identify the intersectional subgroups of interest. In both datasets, there are two labels: one for whether the article is related to the class of violence we want to monitor (police violence / missing and murdered Indigenous people), and one for whether the victim is a woman or girl (or two spirit person, in the case of SBI). We want to identify cases where both labels are true, i.e. cases of Black police feminicides for AAPF and cases of missing and murdered Indigenous women/girls/two spirit people for SBI.

Featurization.

We tested two strategies to embed article full-texts: term frequency-inverse document frequency (TF-IDF) and TensorFlow (TF) Universal Sentence Encoder [62]. In TF-IDF, each article is converted to a word count vector, with values being normalized based on their frequency across the training dataset. Rare ($<$ 5th percentile frequency) and common ($>$ 95th percentile) words are excluded, along with stop words. Meanwhile, TF's sentence encoder is a Transformer-based model [329] trained on unsupervised text from Wikipedia and news corpuses, and supervised data from

the Stanford NLI (SNLI) corpus. The model outputs general-purpose embeddings with a variety of higher-order text features, and has been shown to perform well on tasks such as identifying semantically similar sentences.

We acknowledge limitations of both these approaches. Since TF-IDF only relies on word counts, it has trouble disambiguating the victim, perpetrator, and other people mentioned in an article. For example, TF-IDF would yield a similar featurization for an article describing a female victim or an article describing a female perpetrator, since both of those articles would typically have high counts of she/her pronouns. Though the sentence encoder can theoretically distinguish these cases by accounting for interactions between words, it may have other flaws: e.g. (1) it is trained on a very general text distribution, so it may ignore or poorly handle the language specific to our intersectional article set; (2) it condenses each article into a 512-dim embedding vector, which may lose relevant semantics and context.

Classifiers.

We used logistic regression (LR) due to its quick training iteration times with our rapidly evolving datasets and limited computational resources.

We tested three different modeling approaches to predict the intersectional labels which combined two annotations. These approaches are summarized below:

1. JOINT: In this approach, we train a single LR on the AND of the two labels. Positives are articles with both labels TRUE, while all others are negatives. Because this approach treats all victims not in the specific category of interest equally as negatives, JOINT is equivalent to a single-label baseline.

2. HYBRID: We train two LRs independently on all articles, one for the feminicide label and one for the police violence / Indigenous-related label. These two

predictors' outputs are then combined into a single intersectional prediction by using the product of their probabilities (multiplying worked better than addition or other weighting schemes).

3. Contextual Hybrid: This model is similar to Hybrid in that we multiply the predictions from two LRs, one for each label. However, the feminicide predictor is made contextual, in that we only train it on the articles where the auxiliary label is TRUE. For example, the contextual model for police feminicides works by training one LR on all articles to identify police violence, and combining it with another LR trained only on the police violence articles to distinguish police feminicides from non-feminicide police violence.

We trained classifiers in Python with the `scikit-learn` package [263]. The LR models use L2-regularization, with the strength optimized via 5-fold cross-validation.

4.5.4 Model Evaluation

Our evaluation methodology consists of three main stages: 1) 5-fold cross-validation on our current datasets; 2) a monitoring phase where team members assess model performance on unseen, possibly out-of-distribution data in the real-world deployment context, and 3) an extended, participatory evaluation with the partner organization. The aim of this multi-level approach is to ensure that the models actually serve activists' needs in deployment, but also to build a degree of confidence in the models before requesting partner feedback, to avoid unintentionally over-burdening them.

In **Stage 1**, we compute internal validation performance by averaging across 5 training-validation splits of the dataset. In **Stage 2**, we deploy models to the Email Alerts system, using the same queries and media source configurations as each

partner organization to reflect the actual deployment context. We then internally monitor the results returned by the system each day for approximately two weeks.

Our quantitative evaluation metrics for Stages 1 and 2 include the Area Under Precision-Recall and Receiver Operating Characteristic curves (AUPRC, AUROC), and Precision@K where K = 50 and 100. Precision@K measures how many of the articles ranked in the top K are indeed positives, where the ideal ratio is 1. This metric is especially relevant to the downstream use case, where we would want to surface as many relevant cases as possible for an activist that only has bandwidth to look at a limited set of articles.

We proceed to **Stage 3** once the results of Stages 1 and 2 indicate an improved set of models. In this stage — which is currently ongoing — activists incorporate the new models into their regular workflows. Each week, we check in to gauge their feedback, both qualitatively (through semi-structured discussion) and quantitatively (through a small set of survey questions focused on result relevance).

4.6 Results

Here, we describe both quantitative and qualitative observations for Stages 1 and 2 of our evaluation. Stage 3 is currently ongoing, and is a longer-term evaluation in which each organization integrates the system into their workflow over a two-month period. While this is a necessary and important part of a feminist and participatory evaluation process, the analysis and results from this stage comprise a separate and significant contribution outside of the scope of this work.

We also note that "success" can manifest in different ways beyond typical metrics — for example, in the extent to which trust is built with partner organizations, power and resources are shared, community is built, or future work is imagined [95].

	AUPRC	*AUROC*	*Precision@50*	*Precision@100*
Joint	0.881	0.913	**1**	0.97
Hybrid	0.902	0.931	0.98	0.96
Contextual Hybrid	**0.921**	**0.944**	0.98	**0.99**

Table 4.3: AAPF model comparison.

	AUPRC	*AUROC*	*Precision@50*	*Precision@100*
Joint	0.894	0.941	**0.98**	0.96
Hybrid	0.913	0.954	0.96	0.95
Contextual Hybrid	**0.92**	**0.959**	**0.98**	**0.97**

Table 4.4: SBI model comparison.

In our project, beyond these three evaluative stages, we were humbled by both AAPF and SBI's willingness to extend their partnership with us, despite the initial tool not meeting their needs. Both participated as panel speakers in a community event our team organized for feminicide activists and spoke to the value of the collaboration [2].

4.6.1 Stage 1

After our additional data collection and annotation steps, we performed an internal quantitative evaluation of the different model architectures for both SBI and AAPF with 5-fold cross-validation.

We found that for the classifiers predicting the auxiliary label (e.g., police violence / Indigenous-related), TF-IDF featurization worked better than the Universal Sentence Encoder embeddings. For example, for AAPF, the police violence predictor was able to identify police violence from words alone: many of these articles

describe the officers involved and police-specific types of force, while they do not include words about domestic violence or other types of civilian assaults. Similarly, for SBI, cases explicitly framed as MMIWG2 typically include the exact phrase "missing and murdered," which can be identified effectively via word counts. Thus, a word count-based featurization (TF-IDF) was sufficient.

The reverse was true for the feminicide predictors, where the Universal Sentence Encoder was more effective than TF-IDF. We speculate that this is due to the relative complexity of this task, which involves disambiguating entities in the text, and the difficulty of using only word counts to so. The sentence embeddings may contain some additional information, e.g., to distinguish the perpetrator's gender from the victim's. There were fewer false positives of female perpetrators killing male victims when we used embeddings, which supports this hypothesis.

Therefore, for the HYBRID and CONTEXTUAL HYBRID models for both organizations, we used TF-IDF featurization for the police violence and Indigenous-related predictors, and sentence embeddings for the feminicide predictors. JOINT only uses one featurization, so we compared TF-IDF and embeddings and found that the latter worked better for both organizations. The Stage 1 results for AAPF and SBI are shown in Tables 4.3 and 4.4 .

For both organizations, CONTEXTUAL HYBRID achieves the highest aggregate metrics (AUROC, AUPRC), as well as the highest Precision@100, which is a particularly relevant metric for an activist's use case (minimizing false positives in a finite number of top-scoring articles). Importantly, both hybrid models performed better than JOINT, which is the baseline in which we train one model on a single label representing the intersectional cases of interest. The improved performance over JOINT highlights that incorporating multiple annotations yields better intersectional performance.

Further, CONTEXTUAL HYBRID's success over base HYBRID underscores the importance of *considering context*. Because the feminicide predictor in the former is trained only on articles related to the context (e.g., only police violence articles), we can interpret it as predicting the probability of feminicide *conditioned* on context. It both reaffirms and is able to utilize our prior knowledge that these intersectional cases manifest and are written about in unique ways.

4.6.2 Stage 2

For the next phase of evaluation, we launched projects on the Email Alerts system with the CONTEXTUAL HYBRID models trained for both AAPF and SBI. For both projects, we found improved performance when compared with the model created before targeted data collection, which returned almost no relevant cases (i.e., only false positives) over the two-month pilot.

AAPF

While the articles returned were overall much more relevant than the previous, general model, many were still false positives, despite few false positives on our internal dataset. Specifically, of the 37 cases returned between 12/07/21 and 12/24/21, 12 described cases of Black women killed by the police. This finding reflects an inherent difficulty in logging feminicides, especially for an intersectional subgroup: though many Black women are unfortunately killed by police, these victims are a small fraction of the overall amount of violence that involves Black victims, women, and/or police. Additionally, Black women victims are systemically under-covered by most media outlets [342]. Compared to our relatively balanced training dataset, real-world news media displays extreme skew towards the "negative" article class, and hence it is

nearly impossible to match internal metrics in deployment settings. While straightforward, this was an humbling realization to frame our expectations during practical evaluation.

We tried to improve our model by learning from this stage. Many of the false positives made logical sense: for example, there were feminicides committed by civilians where the article described police arriving at the scene, or cases of female police officers killing Black male victims (such as female officer Kim Potter killing Daunte Wright, which had an ongoing trial during our observation period). We collected 30 such false positives and added them to our dataset, and retrained the hybrid models. Interestingly, their internal performance decreased, with CONTEXTUAL HYBRID's AUPRC dropping by 6%, implying that these challenge examples were difficult for the model to adapt to. Despite lower metrics, there was some adaptation: when actually deployed, we observed fewer new false positives.

SBI

The returned articles from 12/01/21 to 12/15/21 were overall much more relevant than the previous, general model, with 17 out of 20 related to MMIWG2, and 10 out of 20 describing specific cases of MMIWG2. The false positives that appeared included a few cases where Indigenous men or boys were killed, but female relatives were also involved or quoted in the article. Other false positives included articles where the MMIWG2 movement was referenced outside of the context of a specific feminicide case. As we iteratively improve our models through collecting additional context-specific positives and negatives, we imagine adding a representative set of such false positives to our dataset to target those model weaknesses, mirroring our process for AAPF.

Overall, for both organizations, this stage yielded important insights: (1) During internal validation, a worse-performing model on a more challenging dataset may yield more relevant articles in practice. This connects back to *considering context* in model evaluation as in model development, since a model that appears to perform well in a lab setting may perform worse once deployed in its real-world context, and conversely, a model that performs more poorly in a lab setting may actually yield strong results when deployed *in situ*. (2) Our models have the most trouble disambiguating the roles of different entities in an article, which suggests that we might want to explore other featurization options (such as including hand-crafted linguistic features in addition to a learned sentence embedding [347]) in future work.

4.7 Discussion

In this chapter, we describe the process of building an ML-based system with feminist and participatory methods from the start. While prior approaches to addressing harmful ML systems have focused on *post hoc* technical fixes or considering user input at discrete points in the development process, we instead pose the more forward-facing question of how to conceptualize and design ML systems in support of co-liberation. We consider a concrete case study in which we co-designed datasets and machine learning models to support the efforts of activists who collect and monitor data about feminicide. We focus on our efforts to create ML models to detect instances of feminicide from news media that work well not only for the majority cases, but also for groups that monitor specific, intersectional forms of gender-related violence.

Guided by the framework of data feminism, we demonstrate how intersectional

feminist goals shaped each stage of our process, from problem conceptualization all the way to evaluation and deployment. For example, we highlight how the goal of *challenging power* led us to work with a diverse group of counterdata activists; how the goal of *considering context* motivated our iterative data collection process and CONTEXTUAL HYBRID model architecture; or how the goal of *embracing pluralism* influenced our multi-stage evaluation focused on practical relevance for each group. We found that the resulting models return more relevant results for two intersectional monitoring organizations than a general model that does not take context into account.

Through this process, we distill some practical lessons learned from approaching ML with an intersectional feminist lens. A commitment to intersectionality means that systems should work not only for the mainstream, majority use case, but also for those on the margins. In contrast to dominant values of speed and efficiency, this requires dedicating (possibly a lot of) additional time and resources towards these more specific use cases. Project plans can be developed from the start to anticipate this additional time, reduce the sense of urgency which often leads to overlooking marginal identities, and appreciate the opportunity to build trust and community with impacted groups [174]. In addition to taking longer, a truly participatory process necessarily must be iterative and ongoing. While we discuss the results of a particular set of models here, we also understand that they are imperfect, that the needs of our partner organizations may evolve, and that the data we deal with may also shift substantially as reporting around feminicide changes. We have already begun additional rounds of data collection, development and evaluation, and imagine this as an ongoing process — not in pursuit of a "perfect" model after which we declare the project finished, but towards a sustained, trusted collaboration in which we can continually support activist efforts.

Looking forward, we consider the interplay between generalizability and context-specificity. As part of the overarching Data Against Feminicide project, we are onboarding 20 more monitoring organizations into the Email Alerts system. We imagine that there are likely to be other groups for whom our existing set of models will not work well — in particular, those who focus on subcategories of gender-related violence that sit at the intersection of many forces of domination (e.g., trans violence, feminicides of sex workers, or Indigenous land defenders). The number of relevant cases reported in the news media for these subcategories is also likely to be relatively small compared to the actual numbers of feminicide. While these forms of violence are important to understand on their own, and while we will likely need to build separate, context-specific models for each focus, many aspects of our approach are generalizable. For example, the iterative process of collecting context-specific positives (beginning with cases the groups have already identified) and context-specific negatives (based on iteratively identifying areas for which the model is underspecified) is generalizable to many contexts in which the positive cases of interest comprise a highly-specific type of example.

We also acknowledge the tensions brought up by our current approach. For example, while we annotated AAPF's data with "police violence" and "feminicide", we did not include a race annotation even though their focus is specifically on Black women. While technologies exist to infer race (e.g., from names or photos), they are often ethically fraught. As a result of excluding this information from the dataset, the model cannot learn the specific intersectional identity of interest. In the case of SBI, due to the difficulty of inferring Indigenous identity, we primarily used articles that were explicitly framed as an MMIWG2 case — which means our classifier may miss out on relevant articles that fall outside of this framing. More generally, we see an unavoidable tension in this work between classifying people with rigid cate-

gorization systems (e.g., race, gender) and the inherent fluidity of these categories [291, 183]. Staying with this tension is part of data feminism's commitment to *rethinking binaries and hierarchies*. It is also one of the reasons that our system locates the ultimate decision-making power to determine whether an article is relevant in the human activists themselves.

Chapter 5

Example-Based Interface Modules for Assessing Model Reliability

Chapter 4 considers a scenario where we are able to design each step of the ML lifecycle. However, doing this process well (or updating it, as user needs evolve) can be prohibitively difficult or time-consuming. Existing datasets or pre-trained models are increasingly available. In these cases, we can try to anticipate and prevent downstream harm with deployment tools. These tools aim to provide people in a particular deployment context with the relevant information and agency to judge the reliability and limitations of an existing system for their context. In this chapter, drawing on the framework laid out in Chapter 3, the interface modules we introduce support users with formal and instrumental domain knowledge to intuitively assess prediction reliability.

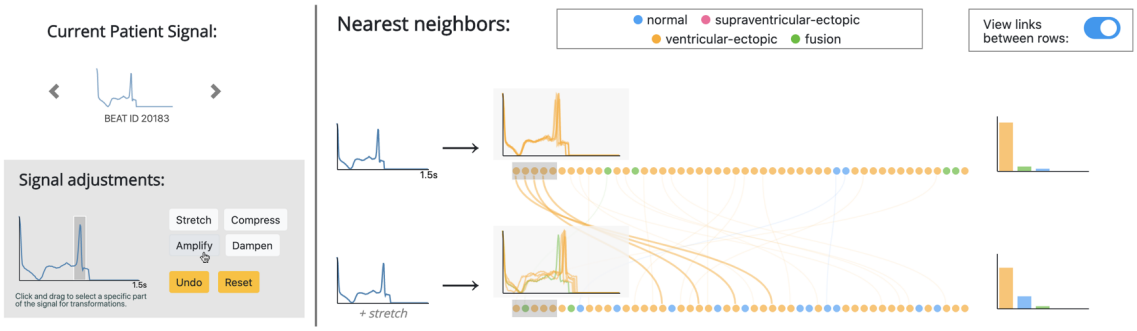


Figure 5-1: An example of the proposed interface for an electrocardiogram (ECG) case study. The output of the machine learning model consists of raw and aggregate information about the input's nearest neighbors. With the editor in the bottom left, the user can apply semantically-meaningful manipulations to the input and see how the output changes.

5.1 Introduction

In this chapter, we introduce two interface modules to facilitate an intuitive assessment of model reliability for users with formal and/or instrumental domain knowledge. First, we use nearest neighbors (NN) to ground the model's output in examples familiar to the user [276]. Alongside the overall distribution of neighbors, a unit visualization depicts individual examples, encoding their class and similarity to the original input according to the model. An interactive overlaid display provides a raw visualization of the examples for more detailed comparison. Second, we introduce an interactive editor for probing the model. Users can apply transformations corresponding to semantically-meaningful perturbations of the data, and see how the model's output changes in response. Using these modules together, users can iteratively build their intuition about the model's strengths and limitations. By interactively examining individual neighbors, they can investigate questions such as whether variation amongst the neighboring examples is expected for the domain, or

if it indicates unreliability; whether the commonalities amongst neighbors align with domain knowledge; or whether these neighbors reveal limitations or biases in the data. Similarly, by interactively modifying the model's input, users can pose and test hypotheses about the model's reasoning, checking that its behavior aligns with domain expectations — for example, ensuring that the model is not overly sensitive to small input modifications that should be class-preserving.

These interface components must be tailored to the model's domain — for instance, different data modalities will require different visualizations of nearest neighbors, and the tools the input editor offers must map to domain-specific operations — but the principles that underlie their design are general-purpose. We briefly illustrate how our interface modules can be instantiated in a diverse range of data domains (including natural language passages on Twitter, and image classification with ImageNet and Quick, Draw!) but devote the bulk of our attention to a medical case study of classifying electrocardiogram (ECG) heartbeats with different types of irregularities.

This case study allows us to perform an application-grounded evaluation [98] with representative real-world decision-makers who have prior knowledge and investment in the domain. We conducted think-aloud studies with 14 physicians, observing the way they interacted with our interface as well as a feature importance baseline [279]. When working with the baseline, participants often rationalized incorrect predictions — for example, back-tracking on their initial assessment and seeking out things in the input that justified the model's incorrect prediction. In contrast, the NN visualizations help participants grasp prediction reliability — for example, by being able to determine whether variations between neighbors was the result of natural ambiguities in ECG data, or whether it reflected the model not learning the right representations for the task. Moreover, by exploring neighbors from different classes,

participants were able to relate the model's uncertainty to clinically-relevant concepts to guide decision-making — for example, pulling out higher-level pathologies that differed amongst neighbors from different classes to understand why the model's prediction would be split between those classes. Finally, participants used the input editor to iteratively form hypotheses about the model's reasoning and test them, using the results to investigate how the model worked and whether its reasoning was clinically sensible.

Our proposed interface modules contribute to the growing work on designing human-centered interfaces for ML systems that highlight both model strengths *and* weaknesses, and that encourage critical engagement with the system. We highlight several important design goals to this end, including grounding visualizations in examples familiar to the user, enabling comparison across examples, and allowing interactive probing of the model. We align visual components and modes of interaction with users' existing conceptual models of the domain, and show that this facilitates more intuitive understanding of the model and its reliability.

5.2 Related Work

5.2.1 Interpretability Methods for Human Understanding

To understand the discord between proposed interpretability methods and their suitability for real-world users, we draw from well-established theories in cognitive psychology that describe how people think about problems and organize information using different "cognitive chunks" [230]. The framework proposed in Chapter 3 lends further insight into how these "chunks" might differ across users with different types of knowledge. For example, a physician with formal domain knowledge might think

about diagnostic decisions in terms of medical concepts that are higher-level than individual features, or relate features to each other in more complex ways than independently ranking them by importance. This idea manifests in theories of HCI stating that effective and engaging interfaces should allow users to view and interact with them in a way that feels *direct* — i.e., the visualizations and interactive mechanisms available to users should align with their cognitive chunks. Specifically, [171] describe "the gulf of execution," arising from a gap between the available mechanisms of an interface and the user's thoughts and goals, and "the gulf of evaluation," arising from a gap between the visual display of an interface and the user's conceptual model of the domain. Our aim is to narrow both of these gaps.

To this end, example-based (also referred to as instance-based) interpretability methods, which produce explanations in terms of other input examples, are of particular interest. Research in cognitive psychology and education supports the idea that people often use past cases to reason about new ones when solving problems [4] and that utilizing examples can help people understand complex concepts, build intuition, and form better mental models [277, 276].

Different types of example-based explanations for ML models have been proposed. Many of these are computed *post hoc*, i.e., they are generated after a prediction is made to try and explain that prediction. For example, counterfactual examples [332, 138] use gradient-based methods to generate the closest example(s) to the input that are predicted to be a different class (defining appropriate measures of "closeness" is an open question). Influence functions [192] try to trace a model's predictions back to the data it was trained on, identifying the examples that were most influential to the prediction. Normative explanations [52] present users with a set of training examples from the predicted class. Xie *et al.* [351] include both counterfactual and normative explanations in the context of radiologic image diagnosis, and find that

providing specific examples can help physicians understand model results. There are limitations of these approaches as well; for example, technical constraints make quickly generating influential examples quite difficult in practice [24, 32], hidden assumptions about actionability in counterfactual explanations can be misleading [23], and normative explanations can be confusing when there is intra-class variation [351].

Others compute example-based explanations by modifying the inference process of a trained model to produce predictions based directly on similar training examples. For example, [59] and [297] use a trained neural network model to improve a KNN classifier, either through using the model to create a weighted similarity function or through computing similarity in the embedding space of the model, respectively. The class label making up the majority of nearest neighbors can be interpreted as the prediction, and the nearest neighbor examples used as an explanation. Recently, [258] extended this methodology to compute neighbors using embeddings from multiple layers of a neural network, demonstrating additional uses for improving the model's robustness and confidence estimates.

In our proposed interface, we compute neighbors using the method of Caruana *et al.* [59]; this could be easily extended to calculate neighbors in a weighted input space [297], or to use embeddings from multiple layers of the neural network [258]. Similarity could also be calculated with other, domain-specific metrics; e.g., Fang *et al.* [114] retrieve similar sensor data examples using a symbolic time series representation. Prior work has focused on developing optimal ways for the trained neural network to inform a KNN classifier, implying that the nearest neighbors would then serve as an explanation. Here, we focus on a relatively unexplored part of this claim, investigating how the resultant output should be presented to the user in an interactive interface to narrow the gulfs of execution and evaluation. We explore a specific

case study to more clearly define the ways in which this type of example-based explanation can improve trust and understanding for users.

5.2.2 Interactivity and Visualization for Interpretability

For interpretability to be useful in practice, effectively communicating information to the user is a critical step. In a literature review of interpretability systems and techniques, Nunes and Jannach [248] found that the vast majority of papers presented explanations in a natural-language-based format (e.g., a list of feature weights). Other types of visualizations include simple charts (e.g., bar plots indicating feature importances) [279] or highlighting sections of the input (e.g., displaying important pixels of an image in a different color or opacity) [308, 206]. With respect to example-based explanations, the visualizations used are often a table of features if the data is tabular [332, 239, 340, 163] or a list of images if the data is image-based [192, 185, 52]. Here, we explore visual encodings that convey more information and allow for more interaction than does merely listing examples.

Other work specifically focuses on visualizations of latent embeddings within a neural network model. Many of these utilize 2 or 3D plots to visualize distance between different examples in the embedding space [221, 35, 155]. Liu et al. [221] additionally visualize examples along 1D vectors corresponding to user-defined concepts, and Boggust et al. [35] provide the ability to compare embeddings of two different models by viewing and interacting with the two plots side-by-side. Particularly relevant to our work, some of the visualizations of text embeddings proposed in [155] aim to display a given word's nearest neighbors in an embedding space. They plot the nearest neighbors as points along a 1D axis that encodes distance, and provide the ability to compare the nearest neighbors across different embeddings.

With respect to interactivity in these interfaces, prior work has primarily studied using human feedback to modify or filter the information that is shown [184, 198, 301, 53]. Here, our goal is instead to provide users with a way to probe the model and test hypotheses about its behavior. The tool described in Wexler *et al.* [340] similarly allows modifying the input to observe how a model's output changes. Unlike our work, it is intended primarily for users with formal ML knowledge.

Like these prior works, our approach aims to facilitate understanding by allowing users to visualize and interact with examples from the data. However, while they are primarily intended for general exploration of what a model has learnt, or for uncovering underlying structure in data, the goal of our interfaces is to help users assess the reliability of predictions on a case-by-case basis.

5.3 Interface Modules for Intuitive Model Assessment

We introduce two kinds of interface components for intuitively assessing the reliability of ML models. In Sec. 5.3.1, we outline the goals that guide our designs. The proposed modules utilize general ideas that can be customized to different domains. We illustrate them primarily with a concrete instantiation of an ECG beat classification task introduced in Section 5.3.2. We then describe the visual components of each module: a display of the model's output in terms of an aggregate and an individual-level view of nearest neighbors (Section 5.3.3), and an editor with which users can interactively modify model inputs and observe how the output changes in response (Sec. 5.3.4). In Section 5.3.5, we walk through specific ways that users can interact with the interface modules to more intuitively assess the model and its

predictions. Finally, in Section 5.3.6, we briefly sketch instantiations of our approach for two other domains.

5.3.1 Design Goals

To facilitate intuitive assessment of model behavior, our overarching goal is to narrow the *gulfs of evaluation* and *execution* for the users of our interfaces [171]. We identify several sub-goals to this end, which motivate our design:

G1. Ground visualizations in examples. To narrow the gulf of evaluation, the visual components of our interface should facilitate reasoning that aligns with users' existing conceptual models. We draw from research suggesting that reasoning through prior examples can aid in problem-solving [4], understanding, and mental model-building over time [277, 276]. For users who are more familiar with the application domain than the mechanisms of ML models, using examples is likely to facilitate more intuitive reasoning than approaches based on model components or individual features (consider reasoning about anatomical structures in an x-ray versus individual pixels). Therefore, we aim to use examples as the building blocks of our visualization.

G2. Facilitate comparisons across examples. To further facilitate interaction aligned with users' existing modes of thinking, we are motivated by literature suggesting that *contrastive* reasoning (i.e., reasoning based on what makes a particular case different than similar cases) is an important way that people understand and explain things [231, 219]. Building on this, we aim to make it straightforward for users to compare specific examples in terms of meaningful high-level concepts in the data, enabling them to build understanding with contrastive reasoning.

G3. Visualize distributions over predicted classes. Often, the output of ML-based systems consists only of a single predicted class, which may convey a false sense of certainty and prompt over-reliance [131, 210]. Conveying model uncertainty can help users align model behavior with their understanding of inherent challenges or ambiguities in the task [54, 324]. Indeed, research on human trust suggests that in addition to conveying assurances of certainty, acknowledging when systems are *un*certain is also an important factor in building effective trust [172]. Providing a probability score along with the prediction is one way to convey uncertainty, though understanding how to interpret abstract probability values is itself challenging for many. Instead, we aim to visualize the output from the model as a distribution over classes at multiple levels of granularity. For example, visualizing an overall probability distribution alongside the specific examples belonging to each class may help users better grasp the sources of the model's (un)certainty and reconcile it with their own understanding of the task.

G4. Enable interactive probing of the model in terms of domain-relevant concepts. Prior work interviewing ML stakeholders has found that one way to build trust is to provide users with ways to confirm that the model is using sensible logic that aligns with their expectations [32, 166, 214, 324, 54]. To facilitate this process, we are motivated by the call to design for "contenstibility," i.e., to make questioning and probing the model an integrated part of the system, rather than an "out-of-band activity" [241, 160]. Interactive capabilities for exploring and querying the model can encourage this kind of engagement — prompting a back-and-forth process where users develop hypotheses and test them, confirming that the model's behavior aligns with their domain knowl-

edge or uncovering unexpected issues. To minimize the gulf of execution, it is also important that users can form such queries in terms of domain-relevant and semantically-meaningful concepts.

5.3.2 ECG Beat Classification Case Study

Our design goals and proposed interfaces are general-purpose and intended to be adapted for different domains. Here, we present a specific case study, classifying electrocardiogram (ECG) beats, to concretely instantiate and evaluate our ideas. This task allows us to perform an application-grounded evaluation of our system using a realistic task that people (i.e., physicians) are familiar with [98, 43]. ECG beat classification, in particular, is an area where machine learning has been widely applied and yielded good performance [288, 366, 176].

The specific task we implement is classifying a single ECG heartbeat into one of four categories: normal, supraventricular ectopic, ventricular ectopic, or fusion. The latter three classes are different types of arrhythmias, or heart rhythm problems. We use a preprocessed version of the MIT-BIH Arrhythmia Dataset [236] available on Kaggle [115]. Each sample in the dataset is an individual heartbeat sampled at a frequency of 125 Hz, and padded to a maximum length of 1.5 seconds. The available dataset contains a fifth class, "unknown," which we exclude here.

We replicate the convolutional neural network (CNN) classification model from Kachuee et al. [176]. We do not use data augmentation since we are interested in seeing whether our visualizations can elucidate that certain classes are underrepresented. The model was trained for ten epochs on the training set (n = 81,123), resulting in a final overall accuracy of 98.3% on the test set (n = 20,284). The breakdown of classes and performances on each is in Table 1.

Class	% of Examples	Test Set Accuracy
Normal	89.3%	99.6%
Supraventricular Ectopic	2.7%	70.5%
Ventricular Ectopic	7.1%	95.7%
Fusion	0.8%	70.4%
Overall	–	98.3%

Table 5.1: Classes used in the ECG beat classification task, along with their distribution in the dataset and the model's test set performance.

5.3.3 Grounding Model Output in Nearest Neighbors

The NN module displays the model's output for a particular example in terms of its nearest neighbors in the data. The nearest neighbors are computed similarly to prior work [258, 297, 59]: Given a neural network model trained to perform the classification task (the *classification model*), we first define an *embedding model*, whose output is the activations of one of the model's hidden layers (see Figure 5-2). We use this to embed all the training examples. Then, for a given new input example, we embed it and return the most similar training examples in this learned representation space.

Computing nearest neighbors in the learned embedding space of the classification model provides the advantage of harnessing the classification model's representational capacity. Since this learned space encodes higher level features relevant to the task, these features are taken into account when calculating similar examples. This step is particularly important to our goal of narrowing the gulf of evaluation [171] since it provides a way for users to understand the model's output in terms of higher-level concepts that align with how they think about the task. The model output can then be visualized in terms of the nearest neighbors.

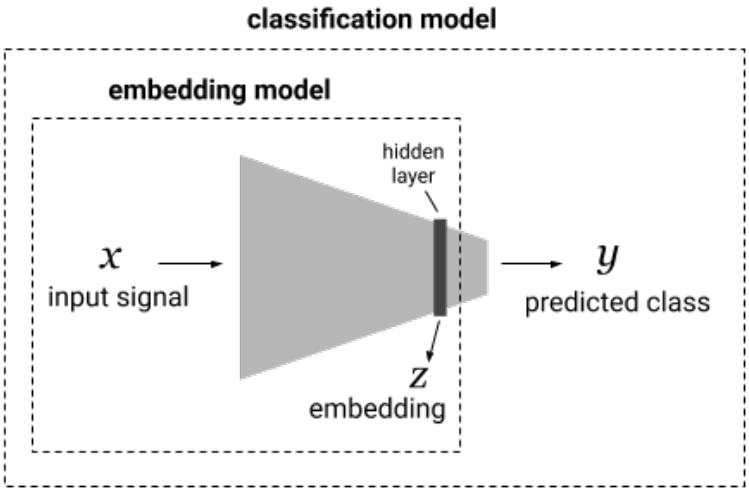


Figure 5-2: To compute nearest neighbors, we extract an embedding model from the original classification model, where the output is a learned representation (i.e., the activation of a hidden layer). We use it to embed the training data examples and rank them by similarity to the input in this learned embedding space, returning the most similar.

Different visual components display the nearest neighbors at varying levels of granularity, which together address our design goals G1, G2 and G3. They include an aggregate view of the neighbors' class labels, a unit visualization of individual neighbors that encodes their class and distance from the input, and a display of the raw input examples associated with each neighbor.

ECG Case Study. For the ECG beat classification task, we use the CNN classification model described in Sec. 5.3.2, and we define the embedding model as the output of the activations from the final hidden layer (a 32-dimensional vector). We use Euclidean distance in this space to rank the embeddings of the training examples by their similarity to a particular input. We retrieve the 50 nearest neighbors for visualization.

Figure 5-3 shows example ECG beats in the interface. Throughout the interface,

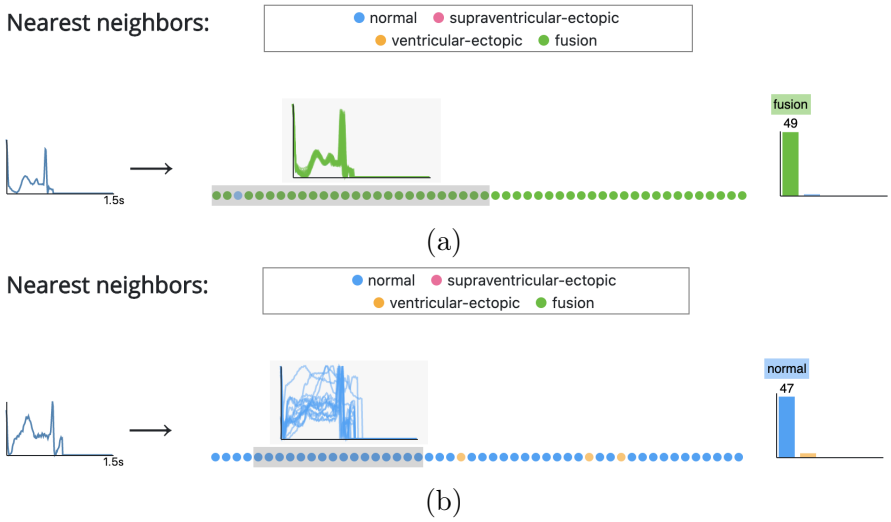


Figure 5-3: Examples of the NN module. On the left is the input signal, and on the right is a histogram of class labels for the 50 nearest neighbors. In the center, each dot represents an individual nearest neighbor, ordered by similarity to the input. The plot above overlays the signals in the selected region. (a) shows an example where the neighbors are very consistent, and (b) shows an example where they are much noisier.

color encodes class labels (e.g., orange waveforms, dots, and bars correspond to ventricular ectopic examples). The aggregate view is a histogram of class labels present in the nearest neighbors, ordered by class frequency to identify the majority class and distribution of other classes. The exact count of each class appears on hover for each bar in the histogram. The unit visualization of individual neighbors is a series of dots arrayed horizontally and ordered by similarity to the input. Users can see, for example, within the nearest neighbors, if certain classes are more similar to the input. When prototyping this component, we also considered designs that encoded the absolute similarity (e.g., placing two neighbors that were more similar nearer to each other). However, we decided against this, since the absolute similarity (i.e., Euclidean distance in the learned embedding space) is not a value that is meaningful or familiar

to the user. Additionally, the distribution of these values is more complicated to visualize, since the distances between neighbors are inconsistent. In our prototypes, for example, there were often clusters of points that densely overlapped and did not facilitate selecting and viewing individual examples.

To visualize the raw input examples, users can brush over specific segments of the ordered dots. The brush is initialized to the first five neighbors, since these represent the most similar examples. Because the ECG data is signal-based, we choose to visualize the neighbors by overlapping signals on a single plot that appears above the brush. This allows users to visually assess consistency amongst the neighbors. If the neighbors are consistent, the overlaid plot will look very similar to a single signal; if they are more varied, the overlaid plot will appear comparably noisy. Outliers are also visible, since they appear as a distinct waveform that does not follow in the same pattern as the other signals. By moving and adjusting the brush to cover specific segments of the neighbors, users can home in on and compare examples from specific classes or individual outliers.

5.3.4 Interactively Editing Model Inputs

To address our final design goal (G4), the editor module allows users to apply meaningful transformations to the input and re-run the modified input through the model to see how the output changes. For example, users can apply transformations that they expect to be class-preserving and check whether the model's output changes drastically.

The available transformations should help narrow the gulf of execution in the interface by providing transformations that align with users' existing ways of thinking about the data and task. For example, in a dataset of photographs, a transformation

that inverts colors is not something that would occur naturally and probably does not reflect users' mental models of the domain. We also would not want to provide transformations like editing individual pixels, which operate at a much lower-level than a person looking at an image would consider. To come up with transformations that are data-specific (meaning they reflect how users think about modifying a specific type of data, like images or ECG signals), relevant to the task (meaning they reflect higher-level factors that users consider important to the task at hand), and aligned with the target users' level of understanding, we emphasize the importance of working with domain experts and other intended end users to design them.

ECG Case Study. For the ECG beat classification task, the editor consists of four transformations, which we arrived at through discussion with a cardiologist: amplify, dampen, stretch, and compress. These transformations can be applied to the entire input signal, or to specific user-defined regions using the brushing functionality. Together, they allow for a large space of possible adjustments to the input signal. There are other options that could be explored here, such as automatically detecting certain important sections of the signal (e.g., "P wave" or "QRS complex") to transform instead of having users select them manually.

Once the transformation has been applied, a new row appears below the original output, displaying the new output. The color encoding as well as highlighting on hover enables tracing how the class distribution changes overall, while links between neighbors that are shared across rows enables tracking how individual examples shift in similarity. The editing toolbar is pictured in Figure 5-4, and an example of the output after several transformations is in Figure 5-5.

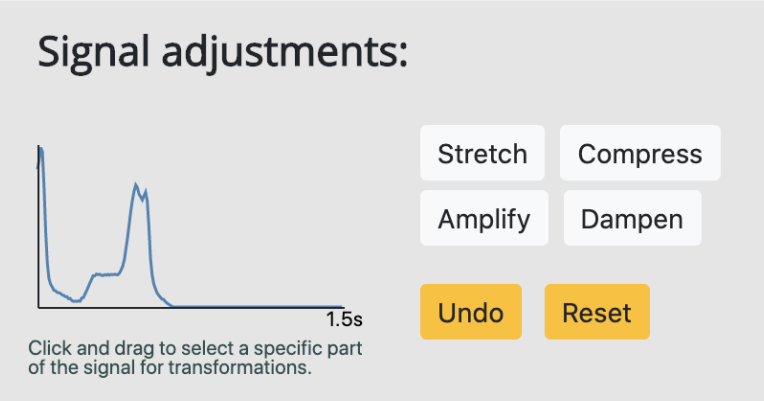


Figure 5-4: The editing toolbar allows users to apply specific transformations or combinations of transformations to the input signal. The transformations can be applied to the entire signal, or to a specific user-selected region. This allows users to select and transform clinically-meaningful segments of the signal (e.g., "stretch the QRS complex").

5.3.5 Enabling an Integrated Workflow

Using the ECG case study, we expand upon several specific ways that a user can interact with the interface modules to assess a model's reliability, understand why it is uncertain, and check whether its reasoning aligns with domain knowledge:

Assessing consistency among nearest neighbors to understand prediction reliability and data limitations

Users can assess the reliability of the prediction in multiple ways. First, the aggregate distribution of class labels can convey the model's uncertainty in the prediction (i.e., the majority class label). For example, if 45 neighbors are normal, this conveys more certainty about the prediction than if only 25 neighbors are normal, and the rest are spread out across other classes.

Second, by viewing the class labels of the unit visualization representing indi-

Figure 5-5: As transformations are applied, new rows appear with the transformed input and corresponding output. Links between each row indicate neighbors that are shared. Links originating from a row's selection are more visible, while the rest are more transparent. Users can get a general sense of how much the nearest neighbors change (by assessing the overall density of links) as well as the specific movements of particular neighbors or sets of neighbors.

vidual neighbors, users can see how similar the neighbors from non-majority classes are to neighbors from the majority class. For example, if there are 40 neighbors labeled normal and 10 neighbors labeled fusion, are those 10 the most similar to the input? Or do they appear closer to the latter end of the nearest neighbors? If the neighbors from the non-majority class are the 10 most similar, this might indicate further unreliability of the prediction.

Third, visualizing the variance or consistency amongst the waveforms themselves can give insight into whether the input example is well-represented in the training data and whether the model is picking up on sensible high-level features common in the neighbors. For example, if the overlaid plot of nearest neighbors shows examples that are very consistent and similar to the input in semantically meaningful ways

(see Figure 5-3a for an example), it implies that the input is well-represented in the training data and that the model is picking up on the right concepts for this input. On the other hand, if the plot of nearest neighbor signals shows examples that are non-overlapping or not similar to the input (see Figure 5-3b for an example), it implies that either examples similar to the input are not well-represented in the training data, or that the model is not learning the right features and therefore not finding those similar examples.

Investigating neighbors from non-majority classes to characterize prediction uncertainty

Typically, a classification model outputs a probability score indicating its certainty. Probability scores can alert the user to some uncertainty in the model, but they don't give the user any additional information to understand *why* the model is uncertain.

In the NN module, one way the model's certainty is conveyed is through the aggregate distribution of class labels. Beyond this, though, the user can further investigate why the model is uncertain by viewing and comparing examples from non-majority classes. Brushing over specific selections of dots representing individual neighbors allows the user to better compare neighbors from different classes. Consider the example in Figure 5-6: 30 of the neighbors have the class label supraventricular ectopic, and 20 have the label normal (these counts are visible upon hover in the aggregate histogram). In Figure 5-6a, brushing over the first 15 neighbors reveals that most of them follow the same general pattern and look similar to the input. The 3 normal neighbors in this selection also seem to follow this pattern — so some of the model's uncertainty is arising from the fact that in the training data, there are normal beats that can look similar to supraventricular beats. In Figure 5-6b, brushing over

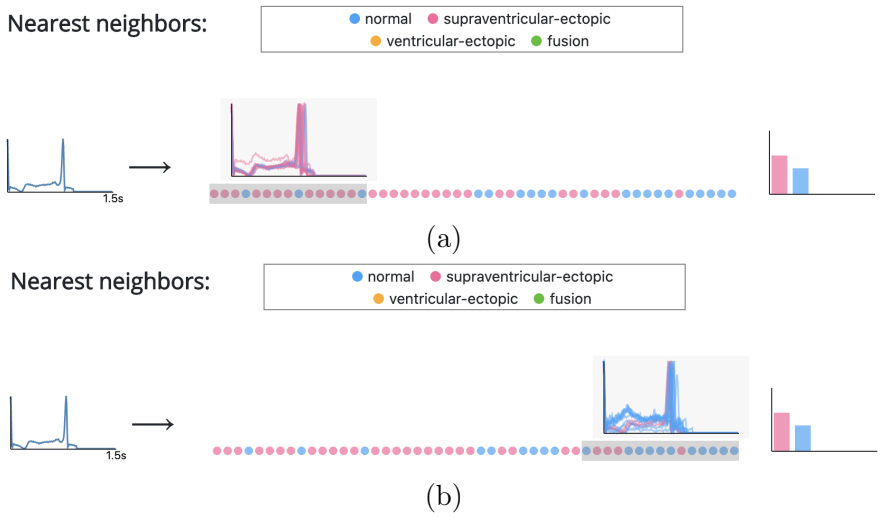


Figure 5-6: The user can home in on different examples to better understand the model's uncertainty. The view of the first 15 neighbors in (a) suggests that some of the model's uncertainty is arising from the fact that normal beats can look similar to supraventricular beats. Viewing the normal neighbors in (b) suggests that another reason for uncertainty is ambiguity around whether the input has a significant T-wave (the spike at the beginning of the signal).

the last 15 neighbors reveals that most of them follow the same general pattern, but have a more elevated T-wave (the spike at the beginning of the signal) than the supraventricular ectopic neighbors. A user might reason, then, that the model is split between supraventricular and normal, and one of the factors driving the uncertainty is whether or not the input has a significant T-wave.

They could then use their domain knowledge to reason about how to proceed. In this example, they might examine the input and decide that the T-wave is significantly depressed, making the input more similar to the supraventricular ectopic examples, and more confidently proceed with supraventricular ectopic as the correct class. Or, they might decide that the different classes present in the neighbors reflect legitimate ambiguities about what the correct beat type is, and choose to consult a

second option or run additional tests.

Comparing examples and labels against domain expectations to prompt critical questioning around the data

If neighboring examples or their labels do not align with the user's expectations, it can prompt questions from the user about the details of the data and how it was collected or labeled, areas that are too often not considered after a model's deployment. Seeing the signals themselves facilitates this type of critical thinking for people who are likely more familiar with the data and what it should look like than they are with concepts like feature weights.

In the ECG case study, for example, the data was annotated by physicians who had access to additional information about the beats preceding and following the input. As a result, there are some examples in the dataset that look extremely similar but are labelled differently (perhaps because of the information available during annotation that the model does not see). In some cases, this leads to nearest neighbors that have different classes but look very similar (see Figure 5-7). Viewing the neighbors for a particular example can prompt questions about how the data was annotated and the subsequent limitations of the model, which would likely not arise if users were not able to view and compare specific similar examples.

Applying input transformations to check if model reasoning aligns with domain knowledge

Checking if the model's reasoning aligns with prior expectations of domain experts is important for building trust, especially in the clinical domain [324, 54]. The editor module allows users to form hypotheses about how particular transformations should

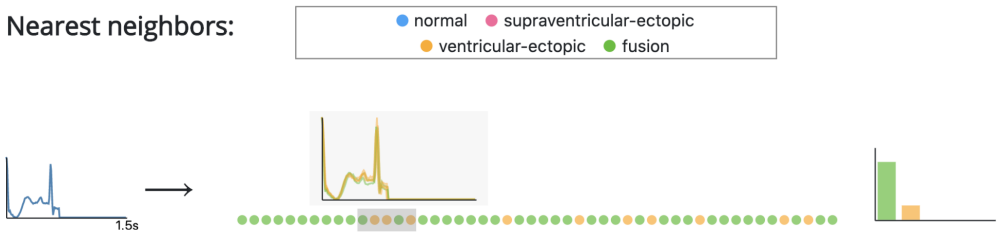


Figure 5-7: An example of neighbors that look similar but have different labels, caused by a difference in the additional information available during annotation versus at test-time. Alerting users to such cases through viewing nearest neighbors can help prompt questions about the data, the annotation process, and limitations of the model.

change the model's output, and build confidence and intuition around the model's reasoning by seeing if these hypotheses hold. For example, the beat in Figure 5-8 is initially classified as supraventricular ectopic. The user might hypothesize that since one indicator of supraventricular ectopic beats is narrowness, and this particular beat is narrow, this is what the model is picking up on. Therefore, stretching the beat should change the model's output, making it shift more towards normal. The user can apply this transformation in the editor to test their hypothesis. In this case, the model's output does change to reflect more normal neighbors, confirming both the original hypothesis and that the model's behavior aligns with the user's expectations from a clinical perspective.

Applying transformations to assess the model's sensitivity to small perturbations

Aside from specific hypotheses about how a particular series of transformations should change the output, a user can gauge the reliability of a particular prediction by performing *ad hoc* sensitivity analyses. If the output changes drastically when the input is slightly tweaked, this can alert users to the fact that the predic-

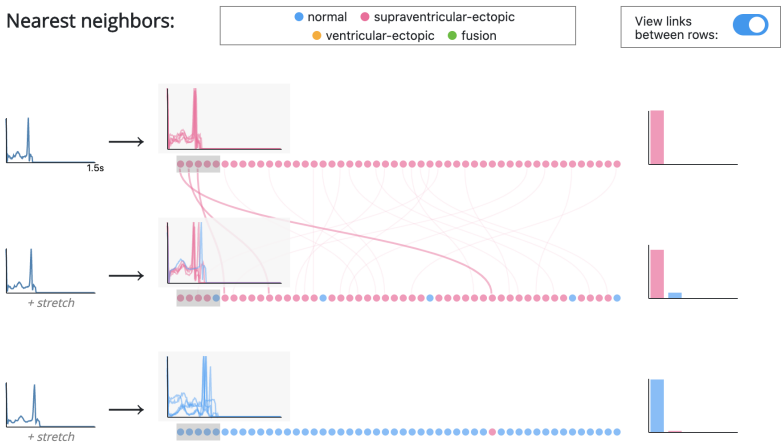


Figure 5-8: An example of using the editor to check if the model's reasoning aligns with domain expectations (i.e., stretching out a supraventricular ectopic beat should shift the prediction towards normal).

tion is precarious and encourage them not to be overly reliant on it. On the other hand, if the output is relatively stable, this can be an additional indicator of model reliability.

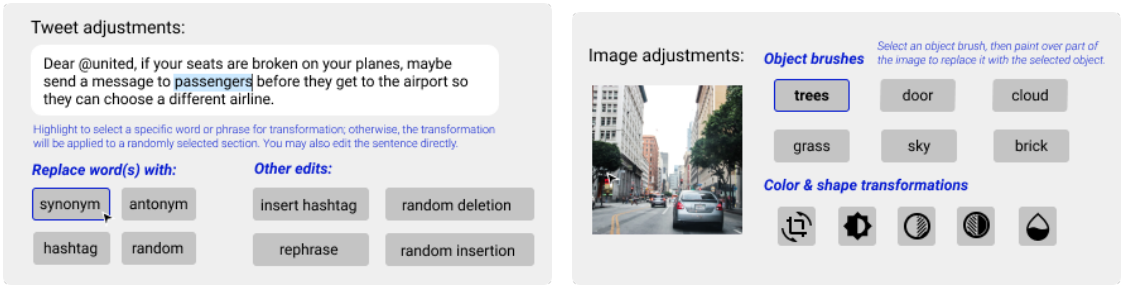
5.3.6 Instantiations for Other Domains

Although we have focused on the ECG case study thus far, our design goals and interface components are general and can be adapted for other domains. To do so, one must identify appropriate domain-specific operations to use in the input editor and approaches to visualize and compare nearest neighbors. Here, we briefly demonstrate how our contributions can be applied to two alternate domains: text passages from Twitter and images from ImageNet [90] and the Quick, Draw! dataset[1].

To identify meaningful transformations for the input editor, we can build on existing work in data augmentation [116, 299] and image generation [25]. For example, for

[1]https://quickdraw.withgoogle.com/

Twitter data, the editor could allow users to edit the text directly, or to apply a range of NLP data augmentations — for example, replacing selected words with synonyms, antonyms, or hashtags. These transformations could be computed using predefined thesauruses, word embedding models, or techniques like back-translation [116]. Depending on the user group, the method of computing augmentations might be predefined, or open to user specification. We show a mockup of an editor for Twitter data in Figure 5-9a. For image data, on the other hand, users might apply traditional affine or color-based transformations (e.g., rotate, crop, saturate) as well as edit meaningful high-level concepts in the example. For instance, Bau et al. [25] show how activating specific sets of neurons in a generative model can allow users to edit an image with object-level control (e.g., realistically replacing a user-specified section of an image with trees). Drawing from their web-based demo[2], we mock up a potential editor for natural images in Figure 5-9b.



(a) (b)

Figure 5-9: Mockups of the editor module for (a) textual data from Twitter, where edits might consist of replace words, rephrasing the example, or random insertions, and (b) natural image data, where edits could include color and shape transformations or object-level painting as in GANpaint [25].

To facilitate comparing NNs and assessing variance, different data modalities require different techniques. We build on insights from Gleicher *et al.* [136], who

[2]http://gandissect.res.ibm.com/ganpaint.html

identify juxtaposition, superposition, and explicit encodings as fundamental buildings blocks used for visual comparison. While our ECG case study primarily uses superposition (i.e., overlaying signals), instantiations of the NN interface for other data modalities might employ different techniques. For image-based data, for example, side-by-side juxtaposition of examples might be better. In Figure 5-10, we show a screenshot of an interactive prototype we built for the Quick, Draw! dataset (consisting of crowdsourced drawings).

Figure 5-10: An interactive prototype of the NN interface for the Quick, Draw! dataset juxtaposes neighbors side-by-side instead of overlaying them. Input editor operations include drawing, erasing or adding shapes. In this figure, for example, we show how using an "erase" tool to remove inner rings from the input image (an onion) changes the neighbors to almost all blueberries instead, suggesting that the model has learned a correlation between inner circles and the onion class.

Other data modalities might combine visual techniques suggested by Gleicher et al. — for instance, an instantiation of our interface with natural language data might employ both juxtaposition (i.e., viewing examples separately) as well as explicit encodings. WordTree [335], for example, visualizes textual data in a tree-like structure that illustrates commonalities amongst sentences as well as areas of high

variance, and Tempura [349] groups sentences with templates that replace specific tokens with abstract linguistic ones. Similarly, Strobelt *et al.* [307] evaluate a palette of techniques for highlighting text which could be adapted to indicate word-level differences between nearest neighbors.

5.4 Evaluative Studies with Medical Professionals

To understand how effectively our interface modules help users build intuition for ML model reliability, we return to our ECG beat classification case study to conduct an application-grounded evaluation [98] with domain experts and a simplified task. We recruited 14 participants through our personal and professional networks: 3 fourth year medical students (P1-P3) and 11 physicians (P4-P14). The studies were certified by our institution as exempt from IRB review under Category 3.

5.4.1 Study Design

In order to study the effect of each of our modules independently, each participant experienced three conditions. The first two conditions were randomly ordered between our NN visualization (without the editor) or a baseline feature-importance visualization, to understand the impact of example-based explanations on building intuition about the ML model. To understand the impact of interactively editing inputs, participants experienced a third condition featuring the NN visualization *with* the input editor. We chose to use feature importance as our baseline since it is a widely researched alternative to example-based explanations [107, 32]. The baseline condition, shown in Figure 5-11, emulates the design of our NN visualization. Feature importance is calculated using LIME [279], a commonly-used open-source method.

LIME results are shown as highlighted regions that overlay the waveform, in line with existing approaches for visualizing ECG feature importance [240, 322]. We plot the feature importance values that are both above the 80th percentile and part of a continuous segment of neighboring important features, to align with physicians' existing ways of thinking about regions of an ECG signal.

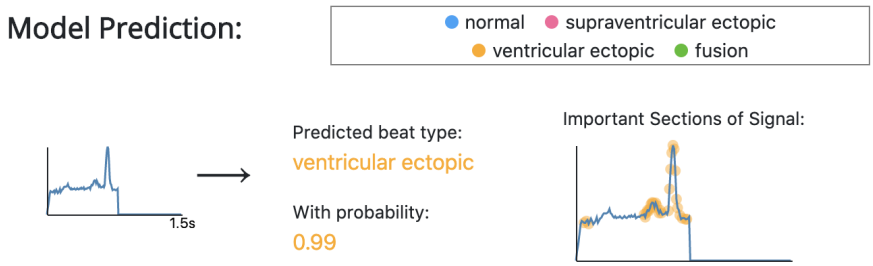


Figure 5-11: The baseline visualization consists of the predicted beat class, the probability with which that class was predicted, and highlighted segments of the beat considered most important for the prediction.

Each condition was pre-populated with 12 input beats chosen from the test set and equally distributed among the four classes. We select beats such that 30% in each condition have incorrect predictions (for the baseline condition, the prediction is the class with highest probability; for the NN condition, the prediction is the class that makes up the majority of the 50 nearest neighbors). These incorrect predictions were aligned with the model's actual performance (e.g., we did not include incorrect predictions for normal beats since there are very few of those; we included more incorrect predictions for supraventricular ectopic since the model's performance for that class is poor).

All studies were conducted using video conferencing. Participants were informed that their participation was voluntary, that they could decline to continue at any point, and that their identities would remain anonymous in any research output. Audio and video was recorded with their consent. The average study length was 52

minutes. Participants were compensated with a $30 gift card.

At the start of each study, participants were told which four categories of beats they would be working with including the granular information about beat types included with the original dataset (e.g., there are multiple pathologies that fall under the umbrella of "ventricular ectopic"). We described that they would see ECG beats one-by-one, along with output from a machine learning model that had high overall performance. Participants were asked to imagine a scenario where their workplace had adopted such a tool for beat classification, and they were both trying to consider the model's output to make the best decision about a particular beat, as well as get a general sense of how the model worked. We introduced each interface as using a separate model to mitigate participants carrying over preconceptions from prior conditions. For each condition, participants were given a brief demo and were then sent a link to open the interface on their computer and asked to share their screen. We prompted them to click through the beats and, for each one, think out loud about how they were coming to a decision about the beat's class, how they were incorporating the model's output, and whether their perceptions about the model changed. At the end of each condition, we debriefed participants with questions about their general impressions of the model's capabilities, the interface, and the strengths and weaknesses of both.

5.4.2 Quantitative Results

We recorded the percent of cases in which participants agreed with the model (versus when they disagreed or were not sure). For cases in which the prediction was correct, the agreement rate was similar across conditions; however, when the prediction was incorrect, we found that participants were less likely to accept the model's prediction

when they were using the NN interface, with or without the input editor (Table 5.2). Often in these cases, they did not explicitly "disagree" with the model, but wanted additional information about the signal and/or patient before committing to an answer. We expand on how our interface prompted these additional considerations in the following section.

Pred. Accuracy	*Baseline*	*NN*	*NN + Editor*
Correct	0.64 (0.2)	0.7 (0.16)	0.67 (0.12)
Incorrect	0.73 (0.23)	0.48 (0.27)	0.5 (0.24)

Table 5.2: The mean agreement rate for correct predictions (8 per condition) and incorrect predictions (4 per condition). The standard deviation across participants is in parentheses.

5.4.3 Qualitative Observations

After conducting the studies, we rewatched all the video recordings and pulled out relevant quotes or actions by participants. We then iteratively annotated and grouped these quotes by themes using a combined inductive and deductive approach [40]. We found that when using our tools, visualizations of neighboring signals allowed participants to reason about the model's output in terms of clinically-meaningful concepts, and examining variation in these signals helped participants to build intuition about prediction reliability. By inspecting the class histogram, ordering of neighbors, and neighboring signals, participants were able to relate the model's uncertainty to relevant challenges of the task. Finally, participants used the editor to confirm if the model's reasoning was sensible and to guide decision-making.

Nearest neighbors enable reasoning with clinically-relevant concepts

Visualizing nearest neighbors enabled participants to reason about the model in terms of clinically-relevant concepts by generalizing and comparing across neighbors. They would often notice a particular morphology present in the neighbors that helped them understand the model's behavior and whether it was clinically sensible. One participant, pointing to a pattern present in all the neighboring signals, said *"Yeah, ventricular. It's this elevation and this space that's making it think ventricular"* [P4]. Another described, *"The model is right — with ventricular ectopic, the QRS spike should be broad, which is present in all the similar examples"* [P13]. Overall, ten participants [P1, P3, P4-P5, P7-9, P12-14] reasoned about the model using high-level clinically-relevant concepts that they observed in the neighbors, such as *"depression in the signal"* [P13], *"slope right after the P-wave"* [P7], *"presence of a T-wave"* [P8], or *"P-R interval"* [P5].

In some cases, participants were unsure why neighbors were considered similar, or disagreed with their class labels. For example, one participant said, *"these [neighbors] are supposed to be ventricular ectopic... I think they're normal. I don't know what to make of this [output]"* [P2]. Such cases may be partly because annotators had access to additional information about surrounding beats during annotation that was not available to study participants. Without this information, it can sometimes be unclear why a beat has the class label that it does. While the model's output was confusing in these cases, visualizing neighbors did prompt additional questions about the data and labeling process. For example, one participant asked, *"Some of these normal ones look like they could be abnormal, so I'd want to know why they were called normal and what that was based on"* [P6]. Another further hypothesized, *"Most likely this data was correctly annotated [...] but it [the model] is not using all*

that information here" [P2].

In contrast, with the baseline condition, participants often had difficulty extracting higher-level, clinically-relevant concepts from the feature importance visualization. For example, echoing a sentiment shared by many, one participant said, *"I don't see how these blue [highlighted] areas are super helpful here... what are they trying to get at?"* [P7]. Another participant, who struggled trying to connect the explanation to the predicted class, said *"I don't understand how they go from this [pointing at highlighted areas] to saying that there's some aspect of a ventricular beat in there"* [P12]. Some others had difficulty figuring out what about the highlighted section was important — for example, one participant asked, *"Why is it highlighted here, is it looking at the height of this, is it looking at width? And why only this part?"* [P1]. In some cases, the highlighted areas *did* align with participants' expectations, though connecting these sections back to the prediction was not straightforward. One participant noted, for example, *"Sometimes it was highlighting things I would also consider, but I still thought its prediction was wrong. I don't have any intuition on that. I guess it's finding some features. I would want to know what those features are, see whether they're useful, if they have any intuitive correlation"* [P2].

Visualizing variation helps assess prediction reliability

All participants said that they did not place as much weight on the model's prediction when there was a lot of variance in the overlaid signals. Participants felt more confident in their answers when the overlaid signals were very consistent and similar. They were also able to distinguish between variation that was acceptable given the task and domain (e.g., *"This input isn't as picture perfect, so it makes sense that the model shows some variation in the overlaid examples"* [P4]) compared to variation

that was an indicator of unreliability (e.g., *"[The model's output] isn't giving me much information right now. If I was given this result I wouldn't just listen to the machine, I would want additional information"* [P4]).

When using the baseline condition, most participants only felt reassured when the predicted probability was very high and the prediction aligned with their own. When this was not the case, we observed that participants had trouble understanding how to incorporate the probability score. As a result, they often rationalized incorrect predictions — even when it went against their initial instincts. For example, one participant saw an abnormal beat, started to say it was abnormal, but then changed her mind after looking at the predicted class, which (incorrectly) was normal: *"I don't think this is normal... well actually seeing that the machine thinks normal... I guess it has a small QRS and the T-wave has a normal slope. Okay, I'll put this in the normal category"* [P7]. Seven participants [P2-4, P7, P9-11] went through similar processes of rationalizing an incorrect prediction after having expressed an inclination towards the correct class.

Even when they did not rationalize an incorrect prediction, participants often struggled with building intuition about the probability score or highlighted sections. For instance, one participant thought out loud, *"I don't know, it seems high probability for a weird looking one like this. And I don't know if it makes sense what it's looking at here and calling important. I'm not confident about this"* [P1]. Similarly, another said *"I'd say this is definitely supraventricular, but the model's not giving it a high probability. I'm really not sure why that would be"* [P11]. Eight participants [P1-2, P5-7, P11, P13-14] expressed similar difficulties in reasoning about the reliability of the prediction in the baseline interface.

Nearest neighbors help characterize uncertainty and incorporate it into decision-making

In the NN visualization, a wide distribution of nearest neighbors classes is one sign of model uncertainty. In such situations, participants consistently homed in on differences using the overlaid plot of waveforms and aligned these differences with clinical concepts. For example, one participant viewed a beat where neighbors were split between supraventricular ectopic and normal, noting *"For supraventricular ectopic one thing you look for is whether or not it has a P-wave. It's unclear in the input. These [brushing over supraventricular ectopic examples] are probably saying it isn't a P-wave. And these [brushing over normal examples] have the P-wave so they're probably saying that the input does also and that's why it should be normal"* [P5].

Similarly, participants often connected the distribution of nearest neighbors to natural ambiguities in the task. For example, one participant noticed some ventricular ectopic beats present in a fusion beat's neighbors — *"Given that fusion is itself a combination of ventricular ectopic and normal, it makes sense that there's uncertainty here, and that there are some yellow [ventricular ectopic] ones that look similar"* [P8]. Rather than distrusting the model, the ability to contextualize its uncertainty helped participants rationalize and move forward with its output. For instance, regarding neighbors split across classes, another participant said *"I would be exactly split like the model is between supraventricular and ventricular ectopic. The fact that the model is also split between those two makes me feel better, and I would do further testing [in person] to differentiate which one it is"* [P4].

Beyond making sense of the presence of multiple classes in the nearest neighbors, participants were also able use this information along with their domain knowledge during decision-making. In many cases, upon viewing neighbors from the different

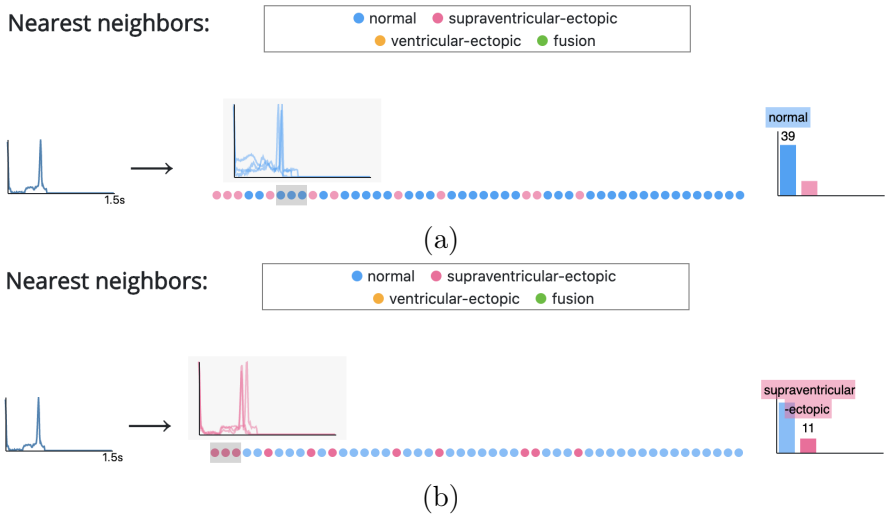


Figure 5-12: For this beat, one participant looked through some of the normal neighbors (a), comparing them to some of the supraventricular ectopic neighbors (b). They reasoned that the normal examples, though they made up the majority of neighbors, were not more similar in clinically-meaningful ways to the input than the supraventricular ectopic examples. As a result, they were able to arrive at the correct classification (supraventricular ectopic).

classes, participants would realize that one of the classes was not actually similar to the input and, as a result, feel more confident in disregarding it. For example, for the beat shown in Figure 5-12, one participant said *"This is supraventricular ectopic. [The model] is calling it normal, but the normal ones don't look so similar. The pink ones [supraventricular ectopic] look more like it because they also don't contain a P-wave"* [P14]. In other words, they were able to relate variation in the neighbors to clinical concepts (normal neighbors with a P-wave, supraventricular ectopic neighbors without), hypothesize why the model is uncertain (it isn't sure whether the input example contains a P-wave), and use their own domain knowledge to determine how to proceed (the input does not actually have a P-wave, so go with supraventricular ectopic). Eight participants went through thought processes to better understand the model's uncertainty and reconcile it with their knowledge of the domain knowledge [P4-8, P10, P13-14].

In contrast, when the model appeared less certain to those using the baseline (i.e., a lower probability score), participants had difficulty reasoning about why. Many said they did not know why the probability was relatively low, or provided explanations based on their own knowledge as opposed to information from the feature importance visualization.

Editing inputs helps check model reasoning

Ten participants used the editor to formulate and test hypotheses about what would happen to the output after applying certain transformations [P4-9, P11-14]. They used this functionality as a way to "sense check" the model's reasoning, and were more confident if it aligned with their expectations (and vice versa). For example, one participant described using the editor to feel more confident in the model's

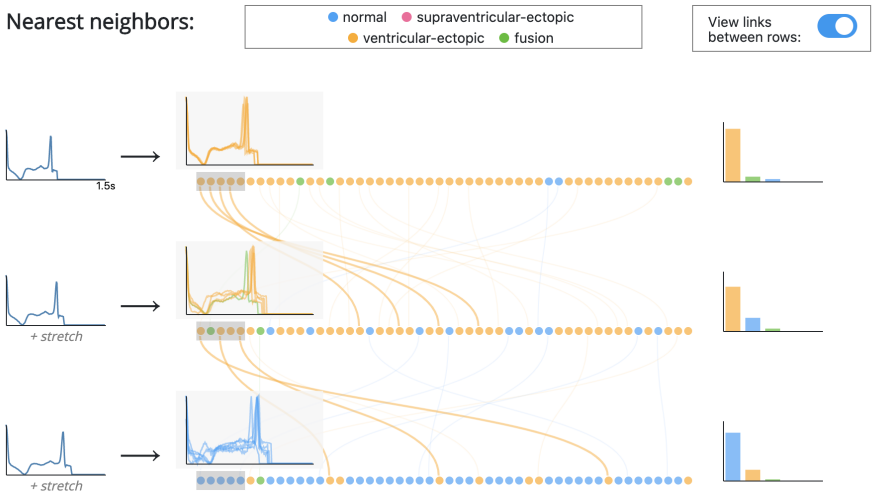


Figure 5-13: One participant hypothesized that the model was picking up on the narrowness of this beat in giving the prediction of ventricular ectopic, and thus stretching it would cause the neighbors to shift towards normal. After applying the stretching transformation, and seeing that the nearest neighbors did change to be more normal, they felt more confident in the model's reasoning for this beat and in classifying it as ventricular ectopic.

prediction for a beat (shown in Figure 5-13), which had mostly ventricular ectopic neighbors: *"I'm not that confident with ventricular ectopic, and this looks almost normal. It's a little narrow, which is partly what ventricular means, so I think that's why this is saying ventricular and if I were to stretch it it would be normal. [Stretches the signal] And that's exactly what happened. That makes me more confident that this is more ventricular ectopic rather than normal. Just because that's exactly what my thought was and that's exactly what happened when I did it"* [P9]. The same participant mentioned later on, *"This is how I think of things. If I can predict what's going to happen I'm more likely to be confident in the decision."*

Sometimes, however, participants applied a transformation but were not able to understand why the nearest neighbors changed as they did, or how to incorporate the observed change into downstream decision-making [P2, P4-5, P8, P10]. This

situation typically occurred when the participant applied a transformation that they expected would shift the neighbors towards one of the non-normal beat classes, but instead skewed the neighbors towards normal — which may be due to the model having learned worse representations for under-represented classes. On one hand, this unexpected behavior prompted participants to rely less on the model's output in these cases — which, since the model is less accurate for these classes, is appropriate. At the same time, however, these instances were not able to offer participants useful insight into the model's reasoning.

In other cases, participants applied several transformations separately to try and gauge the sensitivity of the prediction to small changes, as a way of assessing model reliability [P1, P3, P6-7, P10-11]. Sometimes several small transformations provided positive reinforcement — *"Okay, this makes me more confident. When it's normal, and then you do all these [transformations], I think it should mostly stay normal, which it is. It's consistent so this all makes sense and I feel good with the machine"* [P1]. Other times, these transformations helped alert participants to the model's unreliability — *"Seeing it switch so quickly from supraventricular ectopic to normal does affect my perception of whether it [the model] is good at telling those apart"* [P3].

With respect to the model's behavior more generally, some participants expressed an increased understanding in how the model worked after using the editor and observing what transformations tended to lead to a large change in the output. One participant noted, *"Doing these transformations is making me think about how this program works... I can tell that the narrowness of a beat affects the decision a lot for example"* [P8]. Participants did not typically use the editor when the neighbors were consistent (both in terms of the shape of the signal and their class labels), because they did not feel the need to check the model's reasoning. Other times, they chose not to use the editor because they could not think of a specific hypothesis they wanted

to test — this was particularly true for the participants who were medical students, who often expressed that they "didn't know enough" but that someone with more experience might know what to test.

5.4.4 Study Limitations

With this study, our focus is on evaluating the proposed interpretability and visualization techniques. Thus, our interface is simpler than something that would be used in a clinical setting — for example, in practice, a physician would typically view a strip of beats from multiple leads, rather than one beat in isolation, and often with a grid overlaid to better measure distances. For our purposes, however, these simplifications follow best practices of application-grounded evaluations [98] and would not materially change our qualitative observations about intuition-building and reasoning with high-level concepts. In some cases, these differences in displaying beats made participants more unsure about a beat's classification — however, this limitation applies equally to the baseline condition, so our quantitative observations about relative accuracies continue to hold.

5.5 Discussion and Future Work

In this chapter, we present two interface modules that facilitate intuitive assessment of a machine learning model's reliability. Our work is motivated, in part, by interpretability needs elicited in prior work. For example, studies have found that communicating model limitations and uncertainty is important for building trust [54, 324], but that people have difficulty understanding the meaning of predicted probability scores and incorporating them into decision-making [47]. Other work has described

the importance of users being able to "sense check" a model's decision as a way to build trust [32, 165, 214], but there have been few proposed methods or interfaces for doing so. In response, our interface modules are designed to allow users to interactively probe the model and to reason about its behavior through familiar examples grounded in domain knowledge. Users can explore a given input's nearest neighbors in the training data to better understand if and why the model is uncertain, and what high-level features the model is learning. They can further manipulate the input using domain-specific transformations to test hypotheses about the model's behavior or its sensitivity.

Think-aloud studies with 14 medical practitioners suggest that our interfaces successfully achieve our design goals by helping participants reason about and interact with the model's output in ways that align with their existing conceptual models of the domain. The studies demonstrate how grounding interpretability in real examples, facilitating comparison across them, and visualizing class distributions can help users grasp the model's uncertainty and connect it to relevant challenges of the task. Moreover, by looking at and comparing real examples, users can discover or ask questions about limitations of the data — and doing so does not damage trust, but can play an important role in building it. We also find that our interactive input editor, which offers semantically-meaningful and domain-specific transformations with which to probe the model, provides an effective way for users to sense check the model's reasoning. Participants in our study described the hypotheses they were testing in terms of higher-level features corresponding to their domain knowledge. In contrast, the baseline — which implemented a commonly-used feature importance method [279] — did not facilitate the same sorts of investigation. We found that this baseline interface demanded a large a mental leap from participants in order to understand how the highlighted sections of the waveform contributed to a high/low

predicted probability.

Our results also point to limitations with the current design of our interface components and suggest opportunities for future work. We find that when the nearest neighbor waveforms looked significantly different than expected, participants had difficulty reasoning about why the model thought the neighbors were similar. We posit that part of participants' confusion was caused by the uneven distribution of beat classes in the training data, which affects the quality of nearest neighbors. For example, supraventricular ectopic beats comprise only 2.7% of training examples; thus, the model was neither able to precisely distinguish this beat from others, nor were there sufficient similar examples to fill the list of neighbors. However, this possibility of under-representation in the training data did not occur to participants when seeing low-quality neighbors. Aside from collecting sufficient data to compute better-quality neighbors, this result suggests the need for transparently communicating the model's training data distribution and its implications. If a user is then presented with output where the neighbors do not appear to make sense, they may be better equipped to understand why this might be the case. Indeed, we found that when we described this phenomena to participants after the conclusion of the study, they were able to understand why under-representation would affect the nearest neighbors. Cai *et al.* [54] similarly found the need for an *"AI Primer"* for users to explain, in part, *"AI-specific behavior that may be surprising."* Our observations suggest specific use cases of and types of information to include in such a primer.

In other cases, participants found it difficult to apply transformations using the input editor because the space of possible hypotheses was too open-ended. Here, methods that generate counterfactual examples (i.e., similar example(s) that are classified differently) [332, 138, 238] might provide useful inspiration. These methods automatically generate modified inputs by finding small transformations that

yield different predictions, but because they do not require any user intervention, they can return unrealistic examples that cannot be probed further. However, such methods could usefully bootstrap our input editor. For example, automatically generated examples could help constrain the space of possible hypotheses to only those transformations that cause the greatest change in the model output. Users could then bring their domain knowledge to bear on selecting semantically-meaningful examples to either visualize directly or as a starting point for further transformation.

Chapter 6

Semantically-Grounded, User-Driven Model Testing

In this chapter, we describe a deployment tool that enables context-specific, semantically-meaningful, and user-driven model evaluation. This work shares a similar motivation to Chapter 5: to help users understand model strengths and limitations that are relevant to their context. However, drawing from the terminology in Chapter 3, these tools are designed to facilitate different *tasks*. The interface modules in the prior chapter help users assess the reliability of a given prediction, while the system introduced in this chapter is aimed at evaluating overall model strengths and limitations. In addition, the tools are designed to support different stakeholder groups. Here, responding to a gap revealed in Chapter 3's analysis of existing work, we ask how users with extensive *personal* knowledge can bring that expertise to bear. The system we propose stems from the idea that users with personal knowledge in a particular domain are best suited to evaluate a model for that context, and provides a workflow and interactive user interface to facilitate this process.

6.1 Introduction

There is increasing recognition that evaluations of machine learning (ML) systems should be grounded in *context* [125, 172]. Context is a broad term, and might denote geographies, physical or virtual communities, organizations, institutions, or more. Stakeholders in different contexts have different lived experiences, types of knowledge, and goals [313, 95, 93]; and what constitutes a "model failure" can differ across contexts [151, 201].

Consider the example of content moderation, which we use as a running case study throughout this chapter. Different online communities deal with different types of harassment or trolling, and enforce a wide range of rules/norms [63, 118]. For instance, some subreddits ban talking about specific topics, such as guns or diet advice, while others do not. Specific phrases or emojis might appear to be offensive in one context, but correspond to inside jokes or meanings in another [260]. As automated moderation tools become increasingly available [345, 211, 148], how can the users and/or moderators of an online community understand where a particular system succeeds/fails for them, and assess whether it is suited to their context?

There is currently little infrastructure for users in a particular context to pose or begin to answer these questions. Standard evaluations on static benchmark datasets are often misaligned with real-world deployment contexts, due to issues such as distribution shift [296, 270], shortcut learning [132], underspecification [84], or poor subgroup performance [46]. Evaluations that focus on specific subgroups [232], other metrics [143, 256], or new benchmark datasets [352, 177, 193] are also limited, since they are rigidly defined with respect to predefined subgroup labels or metrics. Some work has incorporated context into model evaluation by compiling context-specific evaluation datasets from scratch with specific groups of users (e.g., Chapter 4), or

proposing complex causal models of societal context from first principles [224]. The resulting evaluations are valuable, but they require significant time, effort, and customization to design (or to update, if/when user needs evolve over time).

To address this gap, we present Kaleidoscope, a workflow and interactive user interface for performing user-driven, context-specific evaluations of ML models. Kaleidoscope leverages users' implicit expectations of "good model behavior" in a given context, and helps them translate these behaviors into explicitly defined tests.

Using Kaleidoscope's iterative workflow, users *identify* important examples using data from their own context, *generalize* them into semantically-meaningful concepts, and *specify and test* model behavior on those concepts. This workflow enables a bottom-up approach, where users can start with a few examples of a particular concept and generalize them into a large, representative *example set* by adding semantically-similar examples retrieved in a learned embedding space. The process is designed to be iterative and exploratory—rather than requiring a precise definition of the concept upfront, its bounds can become more complete and precise as users find and add new examples.

Users can then specify and evaluate model behavior on these example sets by defining and running *tests*. We distill two axes used to specify model behaviors: the behavior type (e.g., specific model outputs, invariances, or shifts) and its granularity (e.g., whether it pertains to a single example set, aggregate comparisons between two example sets, or pairwise comparisons after applying a transformation to each example in an example set). Specifying tests makes desired model behaviors transparent, and running them surfaces insights into model strengths and limitations in terms of domain-relevant concepts. In doing so, these tests can serve to build trust by making anticipated behaviors explicit (i.e., facilitating *contractual trust* [172]).

To evaluate Kaleidoscope, we conduct a two-part evaluation. First, using the

Cognitive Dimensions of Notation heuristic framework [141], we contrast Kaleidoscope's conceptual affordances against template-based and domain specific language (DSL-based) grouping methods for natural language tasks to better understand their tradeoffs. We find that Kaleidoscope results in more semantically-meaningful examples and tests, as opposed to lower-level or syntactically-focused tests. In addition, other methods require formally defining slices of data upfront. Instead, Kaleidoscope allows users to switch between exploratory and confirmatory analyses, clarifying the bounds of their hypotheses and creating slices of data that may have been difficult to precisely define *a priori*.

We also conduct a user study with 13 Reddit users/moderators who used the system to assess two pretrained ML models for content moderation. The iterative process of finding and adding similar examples to build example sets was intuitive and exploratory. Participants typically started with a specific idea of the concept they intended to represent in an example set, but as they found and added examples, this idea sometimes expanded (as they discovered new phrases to search for or types of examples to add), became more precisely defined (as they began to delineate which similar examples did and did not belong), or split into multiple concepts (as they realized implicit subgroups within their initial idea). Resulting example sets represented concepts, drawn from personal experience or specific subreddit rules, that participants considered important (e.g., "LGBT attacks," "colorism," "disrespectful comments"). Each contained diverse examples that would be difficult or impossible to specify via templates or a DSL. Tests built off of these concepts revealed insights into model behavior that helped participants reason about if the model would work well in their context, and how it should be used.

Kaleidoscope contributes to a growing body of work that aims to gives users the agency to probe automated systems. In particular, the system helps users translate

their implicit expectations of model behavior into concrete, domain-relevant tests. Our results indicate that Kaleidoscope facilitates meaningful insights into model behavior, and suggest promising directions for future work in context-grounded model evaluation.

6.2 Related Work

In response to shortcomings of overall evaluation metrics, some work has proposed documenting performance across data subgroups [232, 81, 177, 352]. Typically, these include predefined demographic subgroups such as race, gender, or age. Other work has proposed a range of different evaluation metrics—e.g., notions of fairness, robustness to noise/corruptions, miscalibration, privacy, or the presence of undesirable learned correlations [84, 331, 289, 256, 143, 302, 222, 193].

Importantly, these evaluation paradigms are typically aimed at developers, and rely on several assumptions. First, they assume *a priori* definitions of success. For example, D'Amour *et al.* [84] perform a number of stress tests that measure metrics outside of accuracy. However, these require customized datasets and specific, predefined tasks (e.g., testing a model's robustness to corruptions with ImageNet-C [156]). Second, they assume access to static, labeled subgroups. In many cases, however, the types of examples users in a particular context care about comprise higher-level concepts [230] that are not already labeled in the data (e.g., x-rays with tricky diagnoses [54], aggressive comments [63], arrythmias with broad QRS spikes [314]). Identifying these sets of examples manually is difficult and time-consuming. And finally, they assume that desired model behavior is consistent across different deployment contexts. As is increasingly recognized, though, expectations and norms can differ widely across stakeholders and contexts (e.g., a comment considered aggressive in

one community might be fine in another [260, 201]).

Some recent work has tried to address these issues and perform more context-grounded evaluations of ML systems by designing application-specific evaluation metrics or datasets with specific groups of users [260, 315, 295]. The resulting evaluations are valuable, but their design is highly bespoke. Without a guiding framework or surrounding infrastructure, redesigning this process from scratch in different contexts (or updating it for existing contexts, if user needs evolve over time) requires significant time and effort. Kaleidoscope helps fill this gap, providing a workflow and interactive system that can support context-specific evaluations.

Other work has similarly proposed frameworks for creating custom slices of data for evaluation. Many of these have been proposed for natural language processing (NLP) applications, which we also focus on. For example, Errudite proposes a domain-specific language (DSL) for finding and grouping instances based on linguistic features [348]. Robustness Gym similarly allows users to construct subpopulations based on linguistic features [137]. Checklist enables generating slices of examples using user-defined templates (e.g., `I like {blank}`, where blanks are filled with suggestions from a language model) or transformations (e.g., take an existing set of generated examples and replace proper nouns) [280]. While their goals are related to ours, these systems are designed for developers with technical expertise to identify or generate syntactically-focused groups of examples, and test universally-desirable linguistic capabilities (e.g., "does the model understand negation?") rather than for end users to specify context-specific behavior on semantically-meaningful slices of data (e.g., "does the model flag comments about diet advice?"). Moreover, Kaleidoscope's generalization process enables interative discovery, while other systems typically require users to precisely define the examples of interest upfront. We perform a more detailed comparison between the design affordances of template- and

DSL-based methods versus Kaleidoscope in Section 6.4.

Unlike the deployment tool proposed in Chapter 5, which focuses on helping users assess the reliability of a specific prediction during usage, our work here is motivated by the shortcomings of existing *evaluation* methods. As we describe in Chapter 2, evaluation bias arises when the metrics or data used to assess a model hides harmful effects. We address this issue here by providing ways for downstream users to evaluate the model in terms of relevant and semantically-meaningful concepts for their context. In addition, Kaleidoscope's iterative and exploratory process helps draw out the expertise of users with extensive *personal* domain knowledge (while Chapter 5 focuses on users with formal or instrumental domain knowledge).

6.3 Kaleidoscope

In this section, we describe the steps of Kaleidoscope's iterative workflow and how we instantiated them in an interactive user interface[1]. To make these sections more concrete, we first introduce a running case study that we utilize throughout.

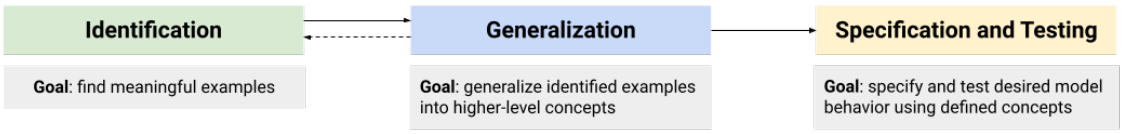


Figure 6-1: Kaleidoscope's workflow consists of identifying meaningful examples, generalizing them into larger, diverse sets representing important concepts, and using these concepts to specify and test model behavior.

[1]Our code for both Kaleidoscope's underlying workflow and the UI is available at https://github.com/harinisuresh/test-cases.

6.3.1 Running case study: Content Moderation

We use a running case study through the rest of the chapter to reason about and instantiate the system with real examples, and illustrate the implications that different contexts can have on model evaluation. In particular, choosing a case study for which ML-based tools are currently being developed and deployed makes these analyses more concrete, allows us to recreate a realistic evaluation by using publicly available models and real-world data, and enables a user study with participants familiar with the domain [99].

Social media platforms and other online forums are an increasingly common venue for discourse, and often, online harassment. A recent Pew Research Center survey found *"41% of Americans have been personally subjected to harassing behavior online, and an even larger share (66%) has witnessed these behaviors directed at others"* [265]. Recent efforts have tried to use technology to help with comment moderation efforts — for example, by building machine learning models to identify posts or comments that violate rules [148, 58, 36, 211]. These moderation systems can be used in a variety of ways, from helping human moderators prioritize what to look at, to allowing readers to filter which comments they see.

Content moderation is a prime example of a domain in which norms (and consequently, desired model behaviors) differ widely across different contexts. For example, Reddit has over 2 million subreddits, each of which has their own set of rules [63, 118]. Even when rules are shared (e.g., "be civil"), the ways in which they are interpreted can vary (e.g., the comment "Thank you for exposing your Jewishness!" has high inter-rater variability for toxicity [319]). Here, we consider the question of how users or moderators of a particular online community can understand the strengths and limitations of an automated moderation system and assess whether it

is suited to their context.

We use Kaleidoscope to look at two publicly available content moderation algorithms: (1) the original Detoxify model released by Unitary (a company that builds moderation tools), trained on Wikipedia comments with crowdsourced toxicity ratings [148]; and, (2) the offensive language identification model released by TweetNLP (an NLP library providing a range of models built with Twitter data), trained on tweets with crowdsourced ratings for offensiveness [58]. We chose the Detoxify model because it is the most highly downloaded comment moderation model (with around 2.08 million downloads) on Huggingface [345], a platform for open source models. We chose the TweetNLP model as a contrast because it is trained on a different data distribution, and we were interested to see if the system could reveal ways in which the two models exhibit different behavior.

Kaleidoscope requires data from which users build sets of examples. Ideally, this should be data sourced from the target deployment context. In the content moderation example, we consider datasets from different subreddits (i.e., each subreddit is a specific context). We create these datasets with comments that are both unmoderated (i.e., still available on Reddit) and moderated (i.e., had been removed by a moderator). We obtained the unmoderated comments by scraping Reddit with the PushShift API [2], and the moderated comments from a dataset collected in prior work [63]. We subsampled each subreddit dataset to 15,000 examples (10,000 unmoderated and 5,000 moderated).

[2]https://reddit-api.readthedocs.io/

6.3.2 Iterative Workflow

Kaleidoscope involves an iterative workflow in which users define meaningful context-relevant concepts and test model behavior on them (Figure 6-1).

Identification

In the identification stage, users identify a few exemplars of a particular concept they wish to define. Users familiar with the deployment context might draw on prior experience to either create the exemplar(s), or query all examples for a particular word, phrase or regular expression and choose from the results. For example, consider a user who wants to test how well the model moderates xenophobic attacks. They might use a particular comment they have seen as an exemplar, or search for all comments containing the word "immigrant," choosing a few that match their intent. The search process might also be more exploratory—for example, our interactive user interface includes a 2D projection of all comments in the dataset, where users can rapidly mouse over areas or clusters to identify different groups of examples. This stage allows users to employ a bottom-up approach—starting with a small number of concrete examples and then iteratively generalizing to a larger set—rather than a top-down one that requires precisely defining the full slice of interest upfront.

Generalization

In the generalization stage, a few examples are expanded into a larger set of examples that represent the higher-level concept. For example, in the identification stage, a user might identify the single comment "immigrants don't belong here," as violating a norm disallowing xenophobic attacks. In the generalization stage, they would expand this comment into a set of different comments from the dataset that capture

the general concept of "xenophobic attacks."

Kaleidoscope enables generalization using iterative content-based retrieval. A user starts with their identified example(s), using them as a seed to search for similar examples. A set of the most similar examples are retrieved using a distance metric in a learned embedding space, and clustered by similarity. Computing distance and retrieving examples in a learned embedding space facilitates finding semantically-related examples (as opposed to generalizing using low-level or syntactic features). Users can then select and add entire clusters or individual examples that fit the desired concept to an *example set*. This process may repeat multiple times, with an expanding set of seed examples.

Facilitating the generalization process is critical, since manually generating a larger set of examples is both time-consuming and difficult. While users might be able to identify an example of an important concept, synthesizing many diverse examples representative of the true data distribution is a much harder cognitive task [144].

The iterative process also allows users to switch between exploratory and confirmatory analyses. A user might start off with an initial idea of a concept ("xenophobic attacks") and a loosely defined mental model of what this concept encompasses. As they iteratively explore similar examples in the dataset, the bounds of this concept might evolve and become more precisely defined, or the concept might split into multiple (e.g., "anti-Semitic attacks" and "anti-Asian attacks"). They might keep adding similar examples, or step back and cast a more exploratory net by searching all examples for a different word or phrase. Switching between these modes allows users to both discover and instantiate a wide range of concepts.

The steps of the workflow need not be linear; during the generalization process for a particular concept, a user might come across distinctly different examples that

become exemplars for other concepts (e.g., while generalizing "xenophobic attacks," they might come across an example mocking someone for being offended—this might then seed a different "insults about being sensitive" example set). While we have been illustrating these steps with the content moderation case study, they only require a meaningful representation space in which to compute distance, and some way to visualize the resulting examples. As a result, the workflow can be applied to different application domains or data modalities by selecting a relevant embedding space, and drawing from data visualization techniques from that domain to display examples.

Specification and Testing

In the specification and testing stage, users specify and examine model behavior on the defined concepts. Specifying desired model behavior serves an important role in transparency and trust. Prior work has formalized human-AI trust as *contractual*—i.e., trust is built on an explicit, context-specific contract that specifies the expected behavior of the systems [172]. The model behaviors defined in the testing stage can serve as part of such a contract. Importantly, these behaviors are built on top of concepts defined in the generalization phase that align with users' existing mental models of the domain.

We distill two distinct axes used to specify model behavior (see Table 6.1). The first axis is the behavior type, which describes desired values or shifts in model outputs. For example, Kaleidoscope provides three *behavior types*: 1) specifying the desired model `output`, 2) specifying a desired `invariance` in model outputs, or 3) specifying a desired `directional` change in model outputs. The first behavior type looks at static model outputs, while the latter two behavior types look at shifts in model outputs.

Behavior Type	Granularity	Definition	Example in words
Output	—	$mean(\{\mathbb{I}[f(A_i) = \hat{y}]\}_{i=0}^{\lvert A\rvert})$	The model should predict that *xenophobic attacks* (A) should be moderated.
Invariance	Concept-level	$mean(\{f(A_i)\}_{i=0}^{\lvert A\rvert}) - mean(\{f(B_i)\}_{i=0}^{\lvert B\rvert}) < e$	The model's predictions should not significantly differ between *xenophobic attacks* (A) and *sexist attacks* (B) .
	Instance-level	$mean(\{f(A_i) - f(t(A_i))\}_{i=0}^{\lvert A\rvert}) < e$	The model's predictions should not change significantly after adding "lol" to each example in *xenophobic attacks* (A) .
Directionality	Concept-level	$mean(\{f(A_i)\}_{i=0}^{\lvert A\rvert}) - mean(\{f(B_i)\}_{i=0}^{\lvert B\rvert}) > e * d$	The model should predict that *xenophobic attacks* (A) are more likely to be moderated than *civil discussion* (B) .
	Instance-level	$mean(\{f(A_i) - f(t(A_i))\}_{i=0}^{\lvert A\rvert}) < e * d$	The model's predicted probability of moderation should increase after replacing "you" with "you prick" in *civil discussion* (A) .

Table 6.1: **Model Behavior Specification.** Output tests check whether the predictions of a model f on example set A align with a desired output $\hat{y}$ (Row 1). Concept-level tests compare two example sets A and B, and check whether the distribution of model predictions significantly differs between the two (Rows 2 and 4). Instance-level tests instead compare example set A, and a user-specified transformation t of A, which is applied to each member of the input example set (Rows 3 and 5). Directionality tests also involve a specified direction $d \in \{-1, 1\}$, indicating whether the difference should be positive or negative. Both invariance tests and directionality tests are governed by a threshold e at which the distributions of model predictions may be deemed significantly different, and can be determined by a statistical test (e.g., a t-test). We also require that the p-value of the statistical test is less than a set threshold.

The second axis, *granularity*, applies to behavior shifts, and describes whether the comparison being made is at the `concept-level` or `instance-level`. `Concept-level` shifts consider two example sets, and ask whether model predictions change on average between them. `Instance-level` shifts ask whether there is an average pairwise change in predictions after applying a transformation to each example in an example set. `Concept-level` shifts are tested via an unpaired t-test, and `instance-level` shifts via a paired t-test.

For instance, an `output` test might specify that the model should flag examples in the "xenophobic comments" example set as moderated. A `concept-level invariance` test might specify that model outputs should not be significantly different between an "anti-Asian comments" example set and an "anti-Semitic comments" example set. And an `instance-level invariance` test might specify that model outputs should not change significantly after replacing "Jewish" with "Asian" for each example is the "anti-Semitic comments" example set.

We include both `instance-level` and `concept-level` shifts, since they play different but important roles and entail different tradeoffs. Prior testing frameworks have primarily examined `instance-level` shifts (assessing if model outputs change after applying a transformation to the data) [280, 348, 331]. `Instance-level` tests are useful because they test hypotheses explicitly by constructing counterfactual examples where the input stays constant outside of a defined transformation. However, these tests might produce out-of-distribution or unrealistic examples. For example, offensive anti-Semitic comments likely look different than offensive anti-Asian comments in complex ways, which would not be accurately captured by simply replacing the word "Jewish" with "Asian."

`Concept-level` tests try to account for this by comparing two realistic, independent distributions of data. With `concept-level` shifts, however, it is difficult

to precisely attribute the cause of a shift in model behavior. For instance, if in the data used to create example sets, anti-Semitic comments are usually much shorter than anti-Asian comments, and we find that the model is more likely to flag them as moderated, it is unclear whether this is because of their content or their length. Findings from `concept-level` tests can still be valuable if the data used to create example sets reflects the actual data distribution and correlations in a particular context, since they provide a lens into the correlations the model would exploit in deployment.

Kaleidoscope provides a selection of model behaviors (e.g., outputs, invariances, directional changes) and transformations for testing `instance-level` shifts (e.g., replacing/adding/deleting words). At the same time, by identifying these higher-level axes and how they fit together, the system is flexible to adding many different types of behaviors and/or transformation functions as they are developed.

6.3.3 Interactive User Interface

We implement an interactive user interface to facilitate Kaleidoscope's iterative workflow, and make the system approachable for users with domain knowledge (but not necessarily programming experience).

The interface consists of three main panes (see "Overall View" in Figure 6-2): identification and generalization primarily happens in the leftmost pane, where users can explore and find examples (B) to add to new or existing example sets (A). Specification and testing happens on the rightmost pane (D), where tests are displayed. The middle pane (C) contains a 2D projection plot where examples can be moused over or selected. Color and shape encodings highlight examples and/or model predictions when an example set or test is expanded.

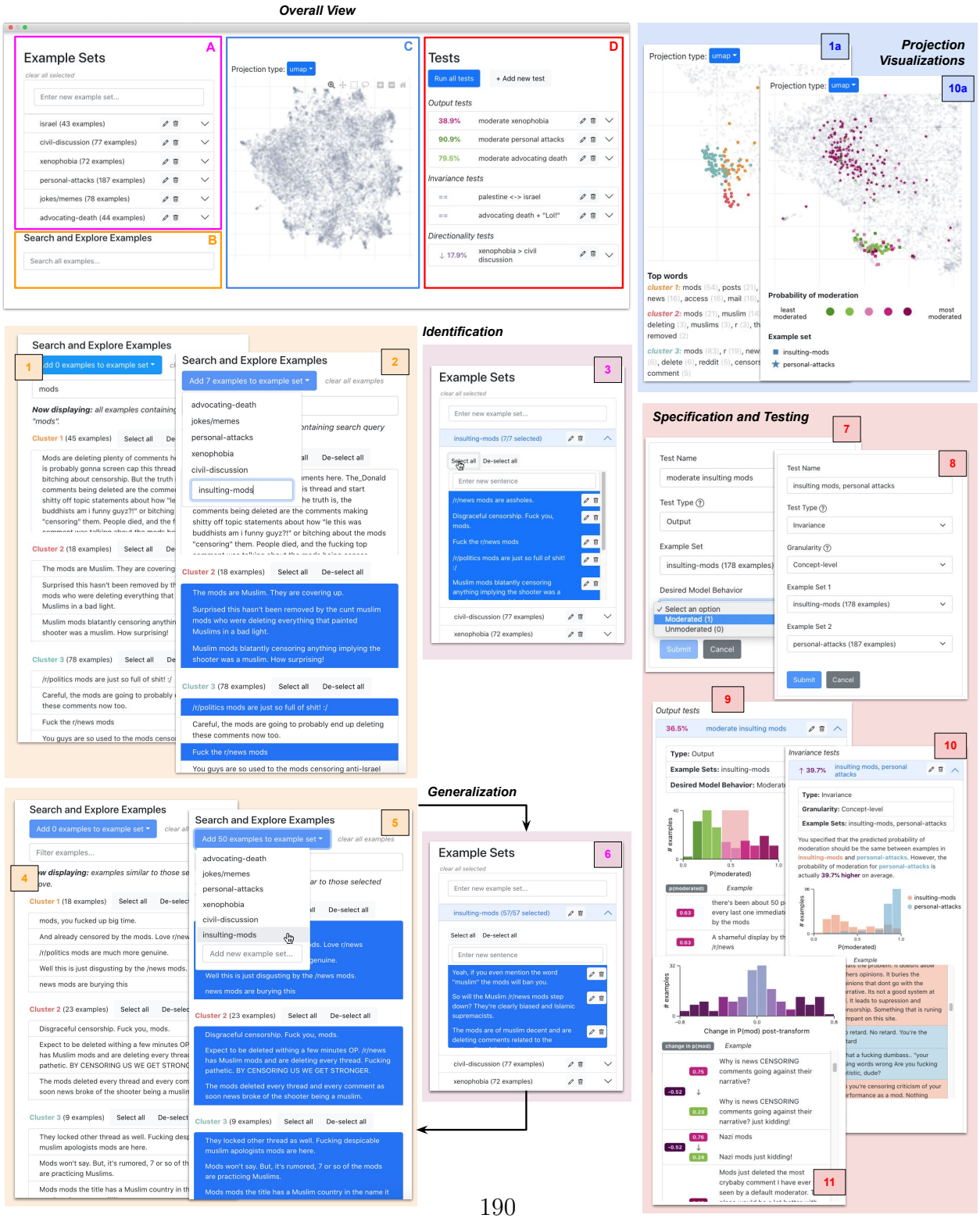


Figure 6-2: Kaleidoscope's interactive user interface, and how different parts of the iterative workflow happen within it. See the text for a step-by-step walk-through.

As an illustrative example, we walk through creating an example set representing criticisms or abuse directed at moderators (a concept that is typically moderated across many different subreddits) using data from r/news. Screenshots from this process are displayed in Figure 6-2, and in the following sections, numbers in parentheses reference specific screenshots.

Identification

To start, a user could either write a seed example (e.g., based on prior experience) or find one in the dataset. To find one, they can use the search bar to search all examples for the word "mods," which they imagine will appear in many relevant examples. In the Search and Explore section, the system returns all comments containing the word "mods" clustered into three groups by similarity **(1)**.

To cluster examples, the system uses K-means clustering in a learned embedding space. The embeddings are computed by the Universal Sentence Encoder [62], a publicly-available transformer-based language model. The points in each cluster are highlighted in the projection plot, and the top words in each appear below (**1a**).

Displaying the result in clusters helps users parse high-level structure in the returned examples. The top words provide an additional summarization of each cluster to help with this sensemaking. Skimming through the examples and top words can help a user get a sense for the types of examples in each cluster: the first with longer rants or discussions about mods, the second with anti-Islamic insults against mods, and the third with more general short, hostile statements against mods. A user's domain knowledge could guide which examples to select to seed an example set, or whether the examples returned comprise distinct enough types that we might actually want to create multiple example sets (e.g., anti-Islamic criticism as well as

general criticism).

In this case, if they believe that examples from the latter two clusters should be treated similarly, they can select the examples from both to seed a new example set named "insulting mods" **(2)**. This example set now appears in the list of example sets, and contains the set of examples they chose.

Generalization

They can now click to select all (or a subset of) these examples **(3)**. This creates the *selected set*, or examples used to retrieve other, semantically similar examples from the data. These similar examples are populated in the Search and Explore section **(4)**.

The system finds similar examples by using Euclidean distance with embeddings from the Universal Sentence Encoder. It computes the mean embedding of all examples in the selected set, and finds the most similar examples to that mean vector. The returned similar examples are also clustered, and a user can follow a similar process as before—getting a high-level sense of each cluster, and then choosing to either add entire clusters, or specific examples, to the example set.

As they add more examples to the "insulting mods" example set **(5)**, these examples are automatically added to the selected set **(6)**. The similar examples continue to update to display ones similar to all of the now-selected examples (repeating **4**, **5** and **6**). Searching by semantic similarity reveals examples that do not necessarily contain the specific search terms we might have thought about. For example, returned similar examples include "I just came from /r/undelete and holy hell is the censorship here is bad," and "So why the fuck would you delete the mass upvoted post that was originally posted?" These are comments implicitly about or addressed

to moderators, that we would not have found with a string or regular expression match.

As we iteratively find and add examples, our mental model of the example set and concept evolves and becomes more precise. Initially, the user may have had the broad idea of capturing insults against mods. Looking at real examples (both individual and higher-level clusters), lends clarity to bounds of this concept (e.g., they chose to focus on shorter, aggressive insults, as opposed to longer discussions) and what it includes (e.g., specific attacks about censorship, anti-Islamic rhetoric, comparisons to other subreddits).

They can also switch between exploration and confirmation. Iteratively finding and adding similar examples is utilizing a particular type of example they have confirmed is relevant. In this process, however, they might see a phrase or word that spurs them to zoom out and return to an exploration stage, searching all examples for a different phrase to see how else it appears in the data. For instance, they might notice the word "censorship" appearing in some of the returned examples, and use that as a search query to search all examples, casting a broader net before drilling down again. Similarly, they could use the projection plot to select the broader area around where points in the example set are located. This populates the Search and Explore section with those examples, and allows them to find examples we might have missed.

As the user continues to iteratively add similar examples, they will typically observe one of two behaviors: 1) *convergence*, where similar examples become increasingly similar, until they are not significantly different from the ones already added, and no longer diversify the example set, or 2) *divergence*, where examples being returned are not actually similar in relevant ways. Which behavior they observe is highly dependent on how well-represented a particular concept is in the dataset

(i.e., if there were very few examples of insults against mods, the retrieved similar examples would quickly become divergent). This is also one of the limitations of our method; while examples are realistic because they are drawn from real data, we also inherit the limitations of that data. We examine this limitation further in the discussion.

In the "insulting mods" example, the user might begin to observe a convergence after 4-5 iterations of adding similar examples, where new retrieved examples seem repetitive, rather than adding additional diversity. At this point, they have 187 examples, and can choose to create a test using this example set.

Specification and Testing

When they click "Add new test," a form appears to specify different axes of desired model behavior. Depending on the behavior type they select (e.g., output, invariance, etc), different parameter options are provided **(7, 8)**.

To test how well the model moderates these examples, they can create an output test, specify the desired behavior as moderated, and run the test. In their header, output tests display the percentage of the examples that have the desired behavior—in this case, we can see that the model only predicts that 34.2% of these comments should be moderated. We also provide a more detailed output when the test pane is expanded **(9)**. This allows users to explore the distribution of predicted probabilities (with a histogram visualization), as well as outputs for individual examples (with an output log). Brushing over the histogram of predicted probabilities filters the examples, so a user can examine, for instance, specifically the ones that were moderated with high probability. Viewing individual examples makes this analysis concrete, allowing users to investigate, for instance, whether the examples with a

higher moderation probability actually are more severe violations than the ones with lower probabilities.

In their headers, `concept-level` shift tests display the mean difference in predicted probability across the two example sets in the test, and `instance-level` shift tests display the mean pairwise difference across an example set pre- and post-transformation. Displaying the actual difference (rather than just pass or fail, for example) provides richer signal into how well or poorly the model does, and helps compare results across tests.

When expanded, `concept-level` tests display a probability histogram and example log containing both example sets in the test encoded via different colors **(10)**. This allows users to compare the overall distributions of predicted probabilities, as well as compare examples from each example set that fall within a given probability window. The expanded view of `instance-level` tests shows a probability histogram of the pairwise differences between the predicted probability of each example pre- and post-transformation. The example log displays each example, its predicted probability before and after the transformation, and the difference in those probabilities **(11)**.

When a test pane is expanded, the projection plot displays the points in the example set(s) being tested. Color encodes predicted probability, and shape encodes example set membership for concept-level tests **(10a)**. This visualization allows users to see if there are high-level patters in the model's predictions (e.g., certain clusters that are highly moderated or not) and drill down into the plot to characterize them.

6.4 Evaluation: Comparing Conceptual Affordances

In this section, we compare Kaleidoscope to other ML model evaluation systems that group examples via template- or DSL-based methods. We focus on Checklist [280] and Errudite [348], respectively, as examples of these alternate classes of evaluation frameworks. To guide our analysis, we draw on the Cognitive Dimensions of Notation framework [141], which includes several axes of comparison (e.g., *hidden dependencies*, *premature commitment*) for systems such as programming languages or visual interfaces.

Using Checklist, users specify a template (e.g., `I really {mask} the flight`) and can fill in the blanks with suggestions from a predefined lexicon or a language model (LM); or, they can apply perturbations to examples from an existing dataset. With Errudite, users write filters using a DSL to query an existing dataset based on linguistic features (e.g., the filter `count(token(x, pattern='PERSON'))` > 2 would return examples where there are more than two `PERSON` entities).

We find that Checklist, Erudite and Kaleidoscope have significant differences in the amount of *premature commitment* and the type of *hard mental operations* required. They also involve different *abstractions* and *hidden dependencies*.

Consider the example of making a test for a content moderation system, to check whether "political comments" are moderated (a subreddit rule in r/funny). We first used Checklist to try and test this behavior. We started by manually defining a custom lexicon of `political_figures` ("Hillary Clinton," "Mitch McConnell," "The President," etc.). We used LM suggestions to fill in the template `{political_figure} is a {mask}`, saving those suggestions into a `descriptive_noun` lexicon. We used LM suggestions again to fill in the template `political_figure is a {mask} {descriptive_noun}`, saving them into an adjective lexicon. We then

used Checklist to generate all combinations of these templates and specified that their label should be moderated, resulting in 30,600 generated examples. We went through a similar process to generate another set of examples of the form `{political_group} has {adjective} views on {political_issue}` and `{political_group} is {descriptive_noun}` where we manually defined sets of words for `political_group` and `political_issue`. This resulted in 8800 generated examples.

Errudite's DSL involves more formal *linguistic abstractions* than Checklist's templates — e.g., specific entity types or part of speech tags. To try find "political comments" with Errudite, we manually created a list of tokens representing political figures, as in the prior example, and wrote a filter to find examples containing those entities (e.g., `has_any(token(x, pattern=‘‘PERSON’’)), [‘‘The President’’, ...])`). This resulted in 124 examples. Some of these examples did not necessarily belong in "political comments" (e.g., "Donald Trump had a cameo? I must have missed it.") but were returned because they contained a matching entity, and there was not a different attribute with which to filter them out. The DSL allows users to compose filters into increasingly complex queries (e.g., constraints on the types of entities, comment lengths, etc), but these are not as useful for our use case, where examples of interest do not group by these linguistic features, but rather, by higher-level semantic features that are difficult to precisely formalize.

With Kaleidoscope, we started by searching for a term we expect to appear in lots of political comments (e.g., "Trump"). The results were grouped into three clusters, each using the word "Trump" in different contexts: the first containing longer discussions about race, the second about the election, and the third with short, aggressive insults. We expect knowledge about the context to guide the breadth/granularity of the example set. Are aggressive insults about political figures distinctly worse than longer discussions, or would all of these political comments be moderated regardless

of tone? This could inform whether to seed one or multiple example sets. Here, we chose a particular set of examples to seed an example set, and then generalized them into a much larger set by searching for similar examples. Looking at semantically similar examples revealed other types of relevant examples that we did not think of from scratch — for example, comments that contained semantically-related phrases or people, such as "make america great again," "gun-owning republican," or "Bernie." The resulting example set contained 272 examples. We also ended up seeding new example sets based on related but distinct types of comments that appeared during generalization (e.g., sets on "detailed political issue discussions" and "insults about being triggered").

For each system, this process of creating sets of examples requires a different set of *hard mental operations* and *hidden dependencies*. With Checklist, we needed to keep track of different sets of words (`political_figures`, `descriptive_nouns`, etc.) and compose them into templates that made sense. It was difficult to generate diverse template formats, and to assess whether we had specified enough of them. While templates are already more abstract than actual examples, Errudite's DSL involves an additional layer of *abstraction*. Because of this, trying to keep track of the mapping from a particular DSL-based filter to the concrete examples specified by it involves several mental jumps. Kaleidoscope does not use linguistic abstractions to define example sets—rather, the example set is simply defined by the concrete examples it contains. Examples are familiar and intuitive to users, and working with them is straightforward. However, templates and DSLs do create a formal definition of the example set; i.e., the dependency between a template or filter and the contents of the resulting example set is *explicit*. In Kaleidoscope, because the example set is less precisely defined, we had to keep track of the types of examples that were included, and continue updating this mental model as we iteratively added

examples.

Checklist and Errudite also require significantly more *premature commitment* than Kaleidoscope. For example, with Checklist, we were able to generate thousands of examples, but they follow a very specific template which we were required to formulate at the beginning. Kaleidoscope's process is more bottom-up—we started with a general idea of a word that would be present in political comments, and seeing the distribution of real examples from this context helped clarify the bounds of our hypothesis. We end up with fewer examples; however, they are more varied, and a more accurate reflection of how political comments actually look in context.

As a result of these differences, the tests we created also tell us different things. With Checklist, we had high confidence in the model's behavior on examples following the specific templates we wrote, but were uncertain about how this might generalize to the natural data distribution. With Errudite, examples are drawn from the real dataset, but the only filter that applied to our use case was a coarse string match that resulted in a skewed sample—returning some irrelevant examples that contain query tokens, and missing a lot of relevant examples that do not. We found that the Checklist tests had 99% and 86% failure rates, and Errudite's test had an 80% failure rate (i.e., saying that most political comments go unmoderated). With Kaleidoscope, our test on political comments had a 66% failure rate. The difference in these results indicates that the model is more likely to moderate political comments from the real data distribution. For example, this might reflect the fact that in this context (the r/funny subreddit), comments about politics are more likely to be aggressive or insulting. If our goal is context-specific evaluation, Kaleidoscope allows us to understand the type of behavior we should anticipate in this particular context.

The design affordances of each tool make them suited to different types of analyses. With Checklist and Errudite, it is easier to test a range of general linguistic

capabilities (e.g., if the model is robust to replacing neutral words with other neutral words, or if it can deal with sentences that have complex linguistic structures). On the other hand, Kaleidoscope is better suited to creating topic-oriented example sets ("aggressive comments", "political comments") and tests that reflect semantically-meaningful goals.

6.5 Evaluation: User Study

To understand how real-world users might use Kaleidoscope's workflow and interface to assess a model's suitability for their specific contexts, we conducted a study with 10 users and 3 moderators from Reddit. The study was certified by our institution as exempt from full IRB approval under category 3 (benign behavioral intervention).

6.5.1 Study Methods

We recruited our participants by posting to r/SampleSize (a subreddit for posting studies), messaging individual moderators on Reddit, and emailing our institutional networks. We filtered participants to those who reported spending 5+ hours on Reddit per week. For each study, we seeded the system with data from a subreddit that the participant was familiar with, and two publicly available moderation models. The system uses the original Detoxify model [148] by default, but we told participants that they could also switch to compare a second model (TweetNLP's offensive language classifier [58]) via the settings tab if they wanted. See Section 6.3.1 for more details on these datasets and models.

Each study lasted between 48 and 62 minutes and participants were paid $20. We spent 15 minutes introducing the project and demonstrating the interface by

creating an example set of insults against moderators, and testing that it should be moderated (similar to the example in Section 6.3.3). We then asked participants to imagine that their subreddit was considering adopting an automated moderation model, and that their goal was to use Kaleidoscope to better understand the strengths and weaknesses of this model and assess if it would be suited to their context. As an initial prompt, we asked them to think about types of comments that came to mind that would be concerning or in violation of subreddit rules, and that they would want to make sure an automated moderation system would know how to deal with. Rather than ask everyone to complete the same tasks, we took this approach to evaluate Kaleidoscope's effectiveness for *context-specific* analysis.

For the majority of the remaining study time, users then continued to use the system independently to create, modify, and explore example sets and tests. As facilitators, we answered simple mechanistic questions about the system and user interface but did not provide further instructions as to what they should test. We ended with a short debrief where we asked how participants felt about the model being used in their context, based on what they had learned about it using Kaleidoscope—e.g., whether they thought the model was well-suited or not, and how they thought it compared to other forms of moderation currently being used.

To analyze the studies, we rewatched all video recordings and extracted quotes or actions that related to how participants approached creating example sets and tests, and/or how they reasoned about or reflected on the model. We iteratively annotated and grouped these into themes — starting with a few a priori hypotheses about expected user behavior (e.g., that they would discover relevant new search terms in the retrieved similar examples), but iterating on and modifying them as we reviewed the data (i.e., a combined inductive and deductive approach [40]). We highlight prominent themes in Sections 6.5.3 - 6.5.6.

6.5.2 Overall usage

Participants created example sets spanning a broad range of topics, based on their personal experience (e.g., offensive examples they had encountered, or posts of theirs that had been moderated) or specific subreddit rules. Examples sets that participants created included "colorism," "self-promotion," "personal attacks," "covert racism," "LGBT attacks," "sexism," "civil discussion about race," and "piracy/torrenting." When specifying tests, participants primarily created output tests (four participants also created tests about behavior shifts).

It took users between 3 and 12 minutes to create an example set, with an average size of 122 examples. When going through the generalization process of building out an example set, participants typically added groups of examples (e.g., an entire cluster of similar examples). They were able to skim the examples and top words to get a high-level sense of an entire cluster, rather than verifying the relevance of each example individually.

6.5.3 Iterative generalization enables discovery

To create example sets, participants typically started with a search query of a term they expected to appear in relevant examples—for example, using the search query "gay" to find examples to seed an "LGBT attacks" example set. They would then perform several rounds of finding and adding similar examples, until they found that new similar examples were either out-of-scope or repetitive. At this point, they often stepped back and diversified by trying a different but related search query (e.g., searching "trans" for "LGBT attacks"), picking some examples, and repeating the generalization process with these new examples as seeds.

They often discovered these additional search queries through noticing words or

phrases in similar examples, and realizing that they might reveal a different subset of the concept at hand. For example, one participant, creating an "attacks against liberals" example set, initially searched for "liberal," and started generalizing based off of some selected examples. One of the returned examples contained the term "SJW," which she recognized as a pertinent term that might appear in a range of other relevant examples, and used it as her next search term. It also prompted other subsequent, related, search terms (e.g., "snowflake"). Other participants discovered different spellings of words (e.g., "p1rate bay") or acronyms they hadn't thought of (e.g., "BLM") that they used as additional search terms. Through this iterative discovery process, participants' mental models of the concepts evolved and expanded, covering additional types of examples they had not initially thought about.

6.5.4 Iterative generalization clarifies mental models of concepts

The generalization process lent clarity to the bounds and contents of the concept in other ways as well—for example, as users delineated which similar examples did or did not belong in the concept, either at the individual example or cluster level. A repeated pattern involved participants noticing implicit subgroups within similar examples and splitting their initial ideal into multiple concepts. For example, one participant initially intended to create a "racism" example set. While looking at retrieved similar examples, she realized that to her, there was a distinction between comments that were outrightly offensive and those that disguised racist sentiments behind lengthy arguments. She ended up creating two example sets representing these two subsets of examples. Another participant started to create a "self-promotion" example set, but noticed several comments returned in the similar

examples that fit into what she called "general spam" rather than specifically self-promotion. She ended up creating an additional "general spam" example set seeded with those examples. These are distinctions that the participants might have had difficulty identifying upfront — but the generalization process helped draw out this implicit knowledge. This might be because viewing data from their context makes reasoning about distinct types of examples familiar and intuitive.

6.5.5 Output tests help reason about context-specific trade-offs

Running and exploring the results of `output` tests helped participants reason about if and how they would use the model in their context. In particular, because tests operate on semantically-meaningful concepts, participants were able to contextualize model behavior in relation to existing moderation methods. For example, a moderator of r/TIFU created an example set representing "fake callouts" (claiming that others' posts/stories are fabricated—these comments are typically removed in that subreddit). They created an output test to specify that these posts should be moderated, and found that the model's performance was 63% (of the 85 examples in the example set, it predicted 65% should be moderated). While compared to typical ML standards, this performance is quite poor, the participant was excited by it: "this could be helpful [...] If it's flagging that much, you know, that's outperforming all the moderators out there and catching stuff they wouldn't." Another participant responded in a similar vein to the model having 80% accuracy on an example set of disrespectful comments: "It's not as accurate as I'd hoped, but I'd still use this model. It's incredibly hard to moderate on my own, and this could be useful, especially if it was used in tandem with a human moderator to filter posts."

In another case, a participant reasoned about suitability across different contexts. He created a "piracy" example set, which he felt were an important type of comment in r/movies that an automated moderation system would need to deal with. However, he found that the model only moderated 8% of examples in that example set. He also created an example set on "personal attacks," and found that the model had 98% performance on it. He subsequently expressed doubt that the model would be suitable for r/movies, but suggested that it might be helpful for another subreddit he moderated, where the bulk of comments that are moderated "are more daft arguments, blatant insults."

Participants also found the more detailed reports from each test useful for understanding model behavior beyond the percentage correctly predicted. In several cases, the model appeared to have subpar performance on a particular output test, and examining the log of individual comments and predictions lent clarity into whether that performance was acceptable or not. For example, in some cases, when participants created output tests specifying that an example set should be moderated, they would look at the specific examples that were not moderated by the model, and find that they were less severe than the other examples—e.g., "These aren't really the worst thing – most of the really bad ones were caught so that's actually useful." Participants felt that an automated moderation system should be used in tandem with a human moderator, so if it was erring on the side of moderating less (and catching the most severe violations), they felt satisfied with its performance. In other cases, examining individual examples and predictions made participants less confident in the model — for example, if the model's decision boundary seemed random, did not agree with participants' prior expectations, or appeared to be reliant on unimportant features.

6.5.6 Testing behavior shifts reveals important model weaknesses

Participants who tested shift behaviors also discovered interesting strengths and limitations about the models that impacted their confidence. One participant, for example, created an example set of "white supremacist dog whistles," and found that adding "thanks for reading" to the end of each comment (via an `instance-level invariance` test) decreased the probability of moderation by 21% for the Detoxify model. The probability of moderation stayed the same using TweetNLP's model, which provided useful insight: "I'd want to look into the second model further, since the first is pretty problematic." Another created an example set of random, benign comments, and found that adding "Yes, I'm gay" to the end of each (also via an `instance-level invariance` test) increased the probability of moderation by 26.2% for the Detoxify model and 52.4% for TweetNLP's model. Together, these tests show these models entail different weaknesses that our system can help characterize.

Others used `concept-level` and `instance-level` shifts together to reveal different things about the model. For example, one participant created example sets representing "homophobic attacks" and "transphobic attacks". They created a `concept-level invariance` test to specify that these two example sets should be treated the same, as well as an `instance-level invariance` test with "homophobic attacks," where they applied a transformation replacing the word "gay" with "trans" in each comment. The `concept-level` test revealed that "transphobic comments" were 25.3% less likely to be moderated than "homophobic comments," while the `instance-level` test reported that predictions were not significantly different after applying the transformation. This difference highlights that the way that "homophobic attacks" and "transphobic attacks" manifest in this context is different, and that simply replacing

the word "gay" with "trans" (while the rest of the comment stays the same) does not fully capture that difference. While the participant considered robustness to switching the attack target (demonstrated by the `instance-level` test) a desirable behavior, they held reservations about the model's performance if deployed, given the subpar performance on the `concept-level` test (which better reflects the real-world distribution of comments).

6.5.7 Limitations

Participants found certain aspects of the system confusing. A common confusion occurred when they observed divergent behavior during generalization, typically due to trying to create a particular example set that was not well represented in the data. For example, one participant tried to create an example set on "positive LGBT discussions," using a dataset from r/funny, but found that the similar examples kept diverging towards negative comments, which were much more present in that context. Others were interested in specific topics (e.g., "China" or "celebrity news") that were not well represented in the data, and thus difficult to represent in example sets. Several participants also brought up that it was difficult to evaluate certain comments without the surrounding context (i.e., what they were written in response to). We chose not to include this context to mimic the way that the models (which do not take context into account) would see examples; but in doing so, participants' experience using the tool felt inconsistent with how they would typically encounter examples.

In addition, the current study design has limitations. We conducted a qualitative observational study so that we could observe participants use and think aloud about the system in real-time. We did not attempt to measure quantitative metrics of trust

or behavior change as we believe that these metrics will only reflect meaningful signal after sustained, engaged use with the system. Finally, while Kaleidoscope's underlying workflow is applicable to different domains and data modalities, our evaluation only focuses on the content moderation case study. Additional studies are needed to understand if our observations generalize to other user groups and application domains.

6.6 Discussion and Future Work

We present Kaleidoscope, an iterative workflow and interactive user interface for user-driven, context-specific model evaluation. Rather than use static tests sets or pre-defined data slices, Kaleidoscope presents an alternative paradigm for model evaluation that allows on-the-ground users to identify examples of important concepts, generalize them into larger, representative sets, and specify and test model behavior with them in semantically-meaningful ways.

Through a comparative evaluation using the Cognitive Dimensions of Notation framework [141], we show how other methods to group examples ask users to define formal definitions of data slices, which requires significant premature commitment and linguistic expertise. Kaleidoscope's generalization process instead enables discovery, and is grounded in real examples. In a study with reddit users/moderators, participants found the interactive process of finding and adding similar examples intuitive, and created a range of example sets populated with diverse examples that would be difficult or impossible to specify via a template or DSL. The resulting example sets reflect semantically-meaningful, context-relevant concepts (e.g., "covert racism" or "LGBT attacks"). Kaleidoscope enables specifying and testing a range of model behaviors using these concepts. Specifying tests makes desired behavior

transparent, and running them reveals relevant insights into model strengths and weaknesses that help users reason about how the model would perform in their context.

We note some of the current limitations of our system, and their implications for future directions. Kaleidoscope trades off precision for flexibility, allowing users to create example sets that are so varied it would be extremely difficult to define them via formal linguistic abstractions (e.g., template or DSL). In doing so, however, it also requires users to keep track of the types of examples they are adding and update their mental model of the example set. Users can assess coverage by observing whether retrieved similar examples are continuing to add diversity to an example set, but this is a heuristic measure (not a guarantee, as in Errudite [348], for example).

We imagine two broad directions for addressing this issue in future work: the first focused on making it easier for users to assess the contents and boundaries of example sets, and the second more computational, focused on methods that facilitate creating example sets with higher coverage. The first direction could draw inspiration from data summarization and visualization [335, 136, 197, 307]—for instance, highlighting distinct exemplars in the set, or visualizing existing or learned features of the examples beyond top words (e.g., length, sentiment, tone). The second direction could explore extensions to our example retrieval method—for instance, rather than finding and returning the most similar examples, we could find and return similar examples that are also different enough from any example already in the set (e.g., drawing from metrics in coverage-based fuzzing [251]), to encourage creating example sets with diverse examples.

Because Kaleidoscope's example sets are grounded in real data, they also inherit the limitations of the dataset used to seed they system. We intend the system to be used to evaluate a model for a particular context, and the dataset used to be from

that context. This helps ensure that users are familiar with the data they see, and that important concepts in that context are likely to appear in the dataset. However, as we found in our user study, sometimes users may want to create example sets that are more hypothetical or less well represented in the natural data distribution. In these cases, similar examples tend to diverge quickly to examples that are not actually relevant to the concept at hand. One possibility to address this issue could be to draw data from other distributions, if we observe that the most similar examples retrieved in the original dataset are further than a specified threshold. For example, a participant in our user study had trouble creating an example set representing "positive LGBT discussions" in the context of data from r/funny, where negative LGBT attacks are much more common. In this case, Kaleidoscope could potentially draw data from a different source where these examples would be more common (e.g., r/LGBT).

The current work also opens up promising future ideas for model development and participatory benchmark creation. We were encouraged by the wide range of different topics, often drawing from their personal experience, that participants in our user study examined. In the future, we imagine Kaleidoscope could be used to facilitate calls for participatory or crowdsourced benchmarks [91]. Kaleidoscope is well-suited to address this need because the system is not *only* exploratory—it allows users to define example sets representing higher-level concepts and specify expected model behavior on them. For example, individuals or groups could specify what kinds of examples they think fit into a particular concept (e.g., "sexist comments") and how they would expect those examples to be treated by a model. These tests could be compiled and used similarly to a benchmark, for evaluating models and their future iterations. This approach acknowledges that "ground truth" is often subjective and dependent on users' contexts and lived experiences [315, 313, 93, 151, 95], and

could help make transparent which people or populations are and are not served by a particular model. Benchmarking methods and datasets drive research agendas and values in ML [101, 91], so this shift has broader implications. Making these processes more participatory shapes future iterations of models and what is considered state-of-the-art, pushing them to prioritize domain knowledge and contextual values [191].

Chapter 7

Conclusion

My view on ML-based systems throughout this dissertation is cautiously optimistic. I believe there is a future in which such systems contribute to justice and equity, but arriving there requires us to deeply consider the broader socio-technical context in which they are embedded. The work presented in this dissertation helps us understand *what* this context encompasses, *why* it is important, and *how* it can shape our approach to developing and deploying societally-beneficial ML technology.

The importance of considering context comes into play from the very beginning of the ML lifecycle. In Chapter 2, we identify distinct choices made at each development step and demonstrate how each can lead to downstream harms in specific deployment contexts. This framework helps illustrate how ML systems and datasets are the result of complex processes driven by human choices and societal norms. It gives us a structure to unpack this context, which may otherwise remain hidden or unquestioned.

In deployment, these systems are not used in isolation; rather, they become embedded into society, interacting with and affecting broad and diverse populations. It

is critical that these stakeholders have the tools, information and agency to understand and hold ML systems accountable. In Chapter 3, we ask who these populations are, and how we can build deployment tools that serve their needs. We develop a graunlar, composable framework that characterizes ML stakeholders in terms of the types of knowledge they possess (e.g., personal knowledge), the context in which it manifests (e.g., the data domain), and the needs they have from the system (e.g., assessing the reliability of a prediction). This framework informs how we might design useful, intuitive deployment tools for diverse stakeholders, and reveals groups who are currently underserved by existing approaches (e.g., people with deep personal knowledge or lived experience).

In Chapter 4, we consider the case where systems can be shaped from the start *with* stakeholders in the target deployment context. We ask how we could proactively design each step of the ML lifecycle described in Chapter 2 to be suited to specific downstream contexts and stakeholders. We also ask how these systems, beyond simply serving their intended context, can actively benefit justice and equity. We explore these questions through an in-depth case study of co-designing datasets and machine learning models to support the efforts of activists who collect and monitor data about feminicide. We focus on the process of building context-specific datasets and models for two activist groups who monitor specific, intersectional types of gender-based violence. To approach the prospective design of the system, we draw from the theoretical framework of intersectional feminism, which provides a conceptual model for how inequality is structured and reinforced. This conceptual models the informs concrete design goals, methods, and choices. Our resulting methodological contributions include an iterative data collection and annotation process that targets model weaknesses, models that explicitly focus on intersectional identities rather than statistical majorities, and a multi-step evaluation process—with quantitative, qualitative and

participatory steps—focused on context-specific relevance.

In Chapters 5 and 6, we focus on a different scenario where we do not have as much control over the development of a system—for example, if an online forum were to adopt a publicly-available, pre-trained content moderation tool. We describe how in these cases, we can design deployment tools that give downstream stakeholders the agency to understand context-relevant strengths and limitations of an ML system.

In Chapter 5, we describe a deployment tool that uses example-based visualizations and an interactive input editor to help users intuitively assess a prediction's reliability. Focusing on users with formal domain knowledge, we align visual components and modes of interaction with users' existing conceptual models of the domain, and show that this facilitates more intuitive understanding of the model and its reliability. In a user study with physicians, for example, we find that our interface modules help physicians reason about model uncertainty in terms of clinically-relevant concepts that are familiar to them.

In Chapter 6, we present Kaleidoscope, a deployment tool that enables user-driven, context-specific model evaluation. The system helps users with personal knowledge of the domain translate implicit expectations of "good model behavior" for their context into explicitly defined tests. Kaleidoscope's iterative workflow enables generalizing from very few concrete examples into a large, diverse set representing an important concept. These example sets can be used to test model outputs or shifts in model behavior in semantically-meaningful ways. Compared to other methods of grouping examples for evaluation, Kaleidoscope's generalization process is more exploratory and flexible — it allows users to define high-level concepts (e.g., "diet advice" or "xenophobic comments") that would be impossible to formalize with template- or DSL-based methods. Specifying tests makes desired model behaviors transparent, and running them surfaces insights into model strengths and limitations

in terms of domain-relevant concepts.

Chapters 5 and 6 are grounded in real-world case studies (ECG beat classification and content moderation, respectively). At the same time, they share the generally-applicable design goal of grounding insights in higher-level, context-relevant concepts that align with target users' existing ways of reasoning. We show how specific techniques—e.g., semantically grouping examples in a learned embedding space, enabling direct manipulation of examples—can help realize this goal.

In summary, the contributions of this thesis include:

- A novel framework of development decisions and their implications throughout the machine learning lifecycle [310]. The framework provides a terminology and conceptual model to tease apart problems that have different underlying sources, which then informs if and when different mitigation techniques are appropriate. The framework provides a granular and comprehensive view on where harm can arise throughout the ML lifecycle, allowing us to recognize sources of harm that may have previously been overlooked.

- A novel framework characterizing the stakeholders of ML systems [313]. The framework highlights stakeholders that may be underserved with existing deployment tools — e.g., users with deep personal knowledge — as well as the heterogeneity of their needs. The framework lends clarity to how we might start to design systems and deployment tools to better serve broader populations.

- A case study of co-designing context-specific datasets and ML models for activists who monitor gender-based violence [315]. This work serves as a first example of translating sustained participation "based on mutual benefit, reciprocity, equity and justice" [300] to the context of ML—from problem conceptualization to dataset collection to system evaluation.

- A system that helps users with formal domain knowledge understand model reliability on a case-by-case basis, via example-based visualizations and an interactive input editor [314]. In contrast to existing methods that describe model behavior in terms of low-level features, the system helps users to reason more intuitively about model relability in terms of high-level, domain-relevant concepts.

- Kaleidoscope, a system that facililtates context-specific, semantically-meaningful and user-driven evaluation [316]. Kaleidoscope allows users to answer questions about model behavior in terms of high-level concepts that would be difficult or impossible to answer with standard evaluation methods.

Together, this work demonstrates many ways in which context-specific considerations and meaningful participation can shape the development and use of ML systems.

7.1 Future Directions

The work presented here motivates several exciting directions for future work, which I discuss below.

7.1.1 Exploring participatory approaches to ML

The work presented in Chapters 4, 5 and 6 shed light on the different ways in which "participation" might manifest in ML. In Chapter 4's case study, we built out context-specific datasets and models manually, with participatory design. In Chapters 5 and 6, we describe systems that facilitate participation during evaluation or deployment.

These systems are broadly applicable, since they can be applied across new contexts to provide insights into existing ML tools.

The latter projects show how custom participatory design may not always be the best-suited approach. In some cases, for example, a generalizeable tool like Kaleidoscope can have more impact, by facilitating participation across many more contexts than would be possible if we tried to design bespoke evaluations for each context individually. At the same time, participatory design in a specific context is a valuable place to start. It builds a deep understanding of real-world problems and potential approaches that stem from the knowledge of people closest to the context. First understanding specific case studies then allows us to draw connections among them, pulling out shared challenges and repeated processes that can inform the design of more generalizeable systems.

While we have started to explore these different approaches in the thesis, there is much work left to understand what participation should look like in practice, when different participatory approaches are appropriate, and how to make these approaches widely used in ML.

For example, which parts of a participatory design process can be generalized, and which parts require human communication and understanding that cannot be automated? In Chapter 4, we developed a participatory data collection process based on iteratively identifying patterns in model errors for different contexts. We repeated the underlying steps of this process with many groups, and we could imagine building out a workflow and user-facing system that facilitates it across new contexts. On the other hand, our method for data annotation is based on the type of violence being described; this requires a socio-historical understanding of the domain and geography of interest that is difficult to automate or generalize. We could imagine both building systems to facilitate the processes that *are* generalizable, as well as

identifying and justifying the aspects that require careful human consideration.

By better understanding questions like this, we can begin to establish a framework for using participatory approaches in ML. This can inform further case studies, which can inform the design of systems motivated by shared needs, and so on. Our hope is that by building up examples, infrastructure, and guidelines around context-specific and participatory approaches, they will become easier to widely adopt across domains.

7.1.2 Collective auditing and community-driven development

How do we extend "participation" in ML from a few users to broad and diverse communities? We can look to instances of community-driven collective action for inspiration. For example, a tool that automatically parsed and pooled Shipt (a grocery delivery service) workers' pay stubs helped uncover systematic underpayment in a new payment algorithm [56]; a browser extension that allowed Youtube users to submit data about their watched videos helped reveal concrete evidence of the harmful effects of Youtube's recommendation algorithm [227]. At the same time, there are still relatively few instances of this sort of collective auditing, compared to the prevalence of ML systems. What makes this hard? How could we support this kind of collective probing and push back on harmful systems? Or, even better: how could we enable communities to collaboratively conceptualize and create the systems *they* believe are important? Future work could characterize the unique challenges that facilitating collective action entails — for example, combining distributed, heterogenous data, enabling community data ownership, or increasing data/ML literacy — and the resulting design implications.

7.1.3 Building societally-beneficial technology by incorporating social science theory

Working towards technology that benefits social justice requires a historical understanding of people and society. However, there is often a disconnect between those implementing systems or developing methods, and those who study theories of justice, oppression, human psychology, et cetera.

Translating social science theory into concrete implications for ML is difficult. It requires a deep understanding of two disparate fields in order to draw connections between abstract theory and specific choices made during ML implementation. In addition, both epistemologically and methodologically, the social sciences and computer science differ widely. While generalization and abstraction are important values in computer science research, social science research deeply values context and specificity. Similarly, while computer science tends towards quantitative methods, social science research is more familiar with mixed methods research (qualitative, quantitative, ethnographic, participatory, et cetera).

While not straightforward to do, there is a lot to be learned from merging these disciplinary boundaries. For example, the approach suggested in Section 7.1.1 of starting with specific, participatory case studies that then inform the design of more generalizeable systems is the result of combining methodological insights from both computer science and social science research. A common theme throughout this dissertation has been drawing from social science theory (e.g., cognitive psychology, pedagogy, feminism) to inspire new research questions, motivate design decisions, or develop methods. There are many opportunities for future work to develop novel approaches to incorporate such theory into ML-based systems. For example, how might frameworks of reparative justice shape the datasets and objective functions

for algorithms informing public policy? How could notions of gender fluidity from queer theory shape the ways we represent and process data about people? A research agenda committed to bridging disciplines is a promising step towards the goal of societally-beneficial, justice-oriented technology.

The work presented here starts to explore this kind of cross-disciplinary work. We develop frameworks that lend clarity to the human choices that shape ML systems and the broad populations that these systems affect. We then demonstrate a human-centered approach to the design of ML systems, describing several ways to incorporate the expertise of downstream users into the development, evaluation and use of these systems. In cases where we are designing a new system, we provide a first case study of using participatory methods to co-design each step of the ML lifecycle in ways that prioritize context-specific and justice-oriented goals. Then, we consider how we might generalize some of these insights to new contexts or cases where an existing dataset or model is being used. We describe the design of two novel deployment tools that help users understand model strengths and weaknesses in ways that are semantically-meaningful and relevant to their context. Together, the work in this dissertation takes a step towards the broader goal of building ML-based systems that are grounded in societal context, incorporate broad and meaningful participation, and benefit justice and equity.

Bibliography

[1] ML Interpretability for Scientific Discovery (MLI4SD) Workshop. https://sites.google.com/view/mli4sd-icml2020/home, 2020. Accessed: 2020-09-16.

[2] Data Against Feminicide 2021. https://datoscontrafeminicidio.net/en/2021-edition/, 2021.

[3] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1): 39–59, 1994.

[4] Agnar Aamodt and Enric Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1):39–59, 1994. ISSN 09217126. doi: 10.3233/AIC-1994-7104. URL https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/AIC-1994-7104.

[5] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y. Lim, and Mohan Kankanhalli. Trends and Trajectories for Explainable, Accountable and Intelligible Systems: An HCI Research Agenda. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–18, Montreal QC, Canada, 2018. ACM Press. ISBN 978-1-4503-5620-6. doi: 10.1145/3173574.3174156. URL http://dl.acm.org/citation.cfm?doid=3173574.3174156.

[6] Ashraf Abdul, Christian von der Weth, Mohan Kankanhalli, and Brian Y Lim. Cogam: Measuring and moderating cognitive load in machine learning model explanations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.

[7] Amina Adadi and Mohammed Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2870052. Conference Name: IEEE Access.

[8] Ali Alkhatib and Michael Bernstein. Street-level algorithms: A theory at the gaps between policy and decisions. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[9] Celia Amorós. Violencia contra las mujeres y pactos patriarcales. 1990.

[10] Isabelle Anguelovski. *Neighborhood as Refuge Community Reconstruction, Place Remaking, and Environmental Justice in the City.* 01 2014. ISBN 9780262322188. doi: 10.7551/mitpress/9780262026925.001.0001.

[11] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *ProPublica, May*, 23, 2016.

[12] Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques. *arXiv:1909.03012 [cs, stat]*, September 2019. URL `http://arxiv.org/abs/1909.03012`. arXiv: 1909.03012.

[13] Robert K Atkinson, Sharon J Derry, Alexander Renkl, and Donald Wortham. Learning from examples: Instructional principles from the worked examples research. *Review of educational research*, 70(2):181–214, 2000.

[14] James Atwood, Yoni Halpern, Pallavi Baljekar, Eric Breck, D Sculley, Pavel Ostyakov, Sergey I Nikolenko, Igor Ivanov, Roman Solovyev, Weimin Wang, *et al.* The inclusive images competition. In *The NeurIPS'18 Competition*, pages 155–186. Springer, 2020.

[15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[16] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in Neural Information Processing Systems*, 32:15479–15488, 2019.

[17] Krisztian Balog and Filip Radlinski. Measuring Recommendation Explanation Quality:The Conflicting Goals of Explanations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*, page 10, Virtual Event, July 2020. ACM, New York, NY, USA.

[18] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter Lasecki, Daniel S Weld, and Eric Horvitz. Beyond Accuracy: The Role of Mental Models in Human-AI Team Performance. page 10.

[19] Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel S. Weld. Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance. *arXiv:2006.14779 [cs]*, June 2020. URL `http://arxiv.org/abs/2006.14779`. arXiv: 2006.14779.

[20] Chelsea Barabas, Colin Doyle, JB Rubinovitz, and Karthik Dinakar. Studying up: reorienting the study of algorithmic fairness around issues of power. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 167–176, 2020.

[21] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Cal. L. Rev.*, 104:671, 2016.

[22] Solon Barocas, Kate Crawford, Aaron Shapiro, and Hanna Wallach. The problem with bias: from allocative to representational harms in machine learning. *Special Interest Group for Computing, Information and Society (SIGCIS)*, 2017.

[23] Solon Barocas, Andrew D. Selbst, and Manish Raghavan. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 80–89, Barcelona Spain, January 2020. ACM. ISBN 978-1-4503-6936-7. doi: 10.1145/3351095.3372830. URL `http://dl.acm.org/doi/10.1145/3351095.3372830`.

[24] Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence Functions in Deep Learning Are Fragile. *arXiv:2006.14651 [cs, stat]*, June 2020. URL `http://arxiv.org/abs/2006.14651`. arXiv: 2006.14651.

[25] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1907375117. URL `https://www.pnas.org/content/early/2020/08/31/1907375117`.

[26] Fran Baum, Colin MacDougall, and Danielle Smith. Participatory action research. *Journal of Epidemiology & Community Health*, 60(10):854–857, 2006. ISSN 0143-005X. doi: 10.1136/jech.2004.028662. URL `https://jech.bmj.com/content/60/10/854`.

[27] Michel Beaudouin-Lafon. Designing interaction, not interfaces. In *Proceedings of the working conference on Advanced visual interfaces*, pages 15–22, 2004.

[28] Ruha Benjamin. Race after technology: Abolitionist tools for the new jim code. *Social Forces*, 2019.

[29] Janine Berg, Marianne Furrer, Ellie Harmon, Uma Rani, and M Six Silberman. Digital labour platforms and the future of work. *Towards decent work in the online world. Genf: International Labour Organization ILO*, 2018.

[30] Cornelia Betsch. Präferenz für intuition und deliberation (pid). *Zeitschrift für Differentielle und Diagnostische Psychologie*, 25(4):179–197, 2004.

[31] Aviruch Bhatia, Vishal Garg, Philip Haves, and Vikram Pudi. Explainable clustering using hyper-rectangles for building energy simulation data. *E&ES*, 238(1):012068, 2019.

[32] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pages 648–657, Barcelona, Spain, January 2020. Association for Computing Machinery. ISBN 978-1-4503-6936-7. doi: 10.1145/3351095.3375624. URL `https://doi.org/10.1145/3351095.3375624`.

[33] Stephen Billett, editor. *Learning Through Practice*. Springer Netherlands, Dordrecht, 2010. ISBN 978-90-481-3938-5 978-90-481-3939-2. doi: 10.1007/978-90-481-3939-2. URL http://link.springer.com/10.1007/978-90-481-3939-2.

[34] Reuben Binns. Algorithmic accountability and public reason. *Philosophy & technology*, 31(4):543–556, 2018.

[35] Angie Boggust, Brandon Carter, and Arvind Satyanarayan. Embedding comparator: Visualizing differences in global structure and local neighborhoods via small multiples, 2019.

[36] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500, 2019.

[37] Nadia Boukhelifa, Anastasia Bezerianos, Ioan Cristian Trelea, Nathalie Méjean Perrot, and Evelyne Lutton. An Exploratory Study on Visual Exploration of Model Simulations by Multiple Types of Experts. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, pages 1–14, Glasgow, Scotland Uk, 2019. ACM Press. ISBN 978-1-4503-5970-2. doi: 10.1145/3290605.3300874. URL http://dl.acm.org/citation.cfm?doid=3290605.3300874.

[38] Artigo 19 Brasil. Dados sobre feminicidio no brasil, Mar 2018. URL https://artigo19.org/wp-content/blogs.dir/24/files/2018/03/Dados-Sobre-Feminic%C3%ADdio-no-Brasil-.pdf.

[39] Tone Bratteteig and Guri Verne. Does ai make pd obsolete? exploring challenges from artificial intelligence to participatory design. In *Proceedings of the 15th Participatory Design Conference: Short Papers, Situated Actions, Workshops and Tutorial - Volume 2*, PDC '18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355742. doi: 10.1145/3210604.3210646. URL https://doi.org/10.1145/3210604.3210646.

[40] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[41] Matthew Brehmer and Tamara Munzner. A multi-level typology of abstract visualization tasks. *IEEE transactions on visualization and computer graphics*, 19(12):2376–2385, 2013.

[42] Andrea Brennen. What Do People Really Want When They Say They Want "Explainable AI?" We Asked 60 Stakeholders. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, pages 1–7, Honolulu, HI, USA, April 2020. Association for Computing Machinery. ISBN 9781450368193. doi: 10.1145/3334480.3383047. URL https://doi.org/10.1145/3334480.3383047.

[43] Zana Buçinca, Phoebe Lin, Krzysztof Z Gajos, and Elena 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.

[44] Joy Buloamwini. *Facing the Coded Gaze with Evocative Audits and Algorithmic Audits*. PhD dissertation, Massachusetts Institute of Technology, 2022.

[45] Peter Buneman, Sanjeev Khanna, and Tan Wang-Chiew. Why and where: A characterization of data provenance. In *International conference on database theory*, pages 316–330. Springer, 2001.

[46] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018.

[47] Adrian Bussone, Simone Stumpf, and Dympna 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, Dallas, TX, USA, October 2015. IEEE. ISBN 978-1-4673-9548-9. doi: 10.1109/ICHI.2015.26. URL http://ieeexplore.ieee.org/document/7349687/.

[48] Judith Butler. *Undoing gender*. Psychology Press, 2004.

[49] Zana Buçinca, Phoebe Lin, Krzysztof Z. Gajos, and Elena L. Glassman. Proxy Tasks and Subjective Measures Can Be Misleading in Evaluating Explainable AI Systems. *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 454–464, March 2020. doi: 10.1145/3377325.3377498. URL http://arxiv.org/abs/2001.08298. arXiv: 2001.08298.

[50] Ruth M. J. Byrne. Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 6276–6282, Macao, China, August 2019. International Joint Conferences on Artificial Intelligence

Organization. ISBN 978-0-9992411-4-1. doi: 10.24963/ijcai.2019/876. URL https://www.ijcai.org/proceedings/2019/876.

[51] Carrie J Cai, Jonas Jongejan, and Jess Holbrook. The effects of example-based explanations in a machine learning interface. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 258–262, 2019.

[52] Carrie J. Cai, Jonas Jongejan, and Jess Holbrook. The effects of example-based explanations in a machine learning interface. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 258–262, Marina del Ray California, March 2019. ACM. ISBN 978-1-4503-6272-6. doi: 10.1145/3301275.3302289. URL https://dl.acm.org/doi/10.1145/3301275.3302289.

[53] Carrie J Cai, Emily Reif, Narayan Hegde, Jason Hipp, Been Kim, Daniel Smilkov, Martin Wattenberg, Fernanda Viegas, Greg S Corrado, Martin 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.

[54] Carrie J Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. "hello ai": Uncovering the onboarding needs of medical practitioners for human-ai collaborative decision-making. *Proceedings of the ACM on Human-computer Interaction*, 3(CSCW):1–24, 2019.

[55] Carrie J. Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. "Hello AI": Uncovering the Onboarding Needs of Medical Practitioners for Human-AI Collaborative Decision-Making. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–24, November 2019. ISSN 2573-0142, 2573-0142. doi: 10.1145/3359206. URL https://dl.acm.org/doi/10.1145/3359206.

[56] Dan Calacci and Alex Pentland. The Shipt Calculator, October 2020. URL https://gigbox.media.mit.edu/posts/posts/tool-the-shipt-calculator/.

[57] Karen L Calderone. The influence of gender on the frequency of pain and sedative medication administered to postoperative patients. *Sex Roles*, 23(11-12):713–725, 1990.

[58] Jose Camacho-Collados, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa-Anke, Fangyu Liu, Eugenio Martínez-Cámara, *et al.* Tweetnlp: Cutting-edge natural language processing for social media. *arXiv preprint arXiv:2206.14774*, 2022.

[59] Rich Caruana, Hooshang Kangarloo, JD Dionisio, Usha Sinha, and David Johnson. Case-based explanation of non-case-based learning methods. In *Proceedings of the AMIA Symposium*, page 212. American Medical Informatics Association, 1999.

[60] Sarah H. Cen and Manish Raghavan. The right to be an exception to a data-driven rule, 2022. URL `https://arxiv.org/abs/2212.13995`.

[61] CEPAL. Al menos 2.795 mujeres fueron víctimas de feminicidio en 23 países de américa latina y el caribe en 2017, Nov 2018. URL `https://www.cepal.org/es/comunicados/cepal-al-menos-2795-mujeres-fueron-victimas-feminicidio-23-paises-america-la`

[62] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, *et al.* Universal sentence encoder for english. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 169–174, 2018.

[63] Eshwar Chandrasekharan, Mattia Samory, Shagun Jhaver, Hunter Charvat, Amy Bruckman, Cliff Lampe, Jacob Eisenstein, and Eric Gilbert. The internet's hidden rules: An empirical study of reddit norm violations at micro, meso, and macro scales. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–25, 2018.

[64] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[65] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. In *Advances in neural information processing systems*, pages 8930–8941, 2019.

[66] Irene Chen, Fredrik D Johansson, and David Sontag. Why is my classifier discriminatory? In *Advances in Neural Information Processing Systems*, pages 3539–3550, 2018.

[67] Hao-Fei Cheng, Ruotong Wang, Zheng Zhang, Fiona O'Connell, Terrance Gray, F Maxwell Harper, and Haiyi Zhu. Explaining decision-making algorithms through ui: Strategies to help non-expert stakeholders. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–12, 2019.

[68] Michael Chromik, Malin Eiband, Sarah Theres Völkel, and Daniel Buschek. Dark Patterns of Explainability, Transparency, and User Control for Intelligent Systems. *Los Angeles*, page 6, 2019.

[69] Danielle Keats Citron and Frank Pasquale. The scored society: Due process for automated predictions. *Wash. L. Rev.*, 89:1, 2014.

[70] Combahee River Collective. *A Black Feminist Statement*. 1977.

[71] Erin Collins. Punishing risk. *Geo. LJ*, 107:57, 2018.

[72] Patricia Hill Collins. *Intersectionality as critical social theory*. Duke University Press, 2019.

[73] P.H. Collins. *Black Feminist Thought: Knowledge, Consciousness, and the Politics of Empowerment*. Perspectives on gender. Routledge, 2000. ISBN 9780415924849.

[74] Sam Corbett-Davies, Sharad Goel, Jamie Morgenstern, and Rachel Cummings. Defining and designing fair algorithms. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, page 705, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450358293. doi: 10.1145/3219166.3277556. URL `https://doi.org/10.1145/3219166.3277556`.

[75] Michael Correll. Ethical dimensions of visualization research. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[76] Sasha Costanza-Chock. *Design justice: Community-led practices to build the worlds we need*. The MIT Press, 2020.

[77] Amanda Cox. Shaping data for news, June 2011. URL `https://vimeo.com/29391942`.

[78] Kimberlé Crenshaw. Demarginalizing the intersection of race and sex: A black feminist critique of antidiscrimination doctrine, feminist theory and antiracist politics. *u. Chi. Legal f.*, page 139, 1989.

[79] Kimberle Crenshaw. Mapping the margins: Intersectionality, identity politics, and violence against women of color. *Stanford Law Review*, 43(6):1241–1299, 1991. ISSN 00389765. URL http://www.jstor.org/stable/1229039.

[80] Caroline Criado-Perez. *Invisible Women: Data Bias in a World Designed for Men*. Abrams Press, 2019.

[81] Anamaria Crisan, Margaret Drouhard, Jesse Vig, and Nazneen Rajani. Interactive model cards: A human-centered approach to model documentation. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 427–439, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533108. URL https://doi.org/10.1145/3531146.3533108.

[82] Morgan Currie, Britt S Paris, Irene Pasquetto, and Jennifer Pierre. The conundrum of police officer-involved homicides: Counter-data in los angeles county. *Big Data & Society*, 3(2):2053951716663566, 2016. doi: 10.1177/2053951716663566. URL https://doi.org/10.1177/2053951716663566.

[83] Gloria Dall'Alba and Jörgen Sandberg. Unveiling professional development: A critical review of stage models. *Review of educational research*, 76(3):383–412, 2006.

[84] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, *et al.* Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

[85] Arun Das and Paul Rad. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. *arXiv:2006.11371 [cs]*, June 2020. URL http://arxiv.org/abs/2006.11371. arXiv: 2006.11371.

[86] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128, 2019.

[87] Lina Dencik, Arne Hintz, and Jonathan Cable. Towards data justice? the ambiguity of anti-surveillance resistance in political activism. *Big Data & Society*, 3(2), 2016.

[88] Lina Dencik, Arne Hintz, Joanna Redden, and Emiliano Treré. Exploring data justice: Conceptions, applications and directions. *Information, Communication & Society*, 22(7):873–881, 2019.

[89] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[90] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[91] Emily Denton, Alex Hanna, Razvan Amironesei, Andrew Smart, Hilary Nicole, and Morgan Klaus Scheuerman. Bringing the people back in: Contesting benchmark machine learning datasets, 2020.

[92] Amit Dhurandhar, Vijay Iyengar, Ronny Luss, and Karthikeyan Shanmugam. A formal framework to characterize interpretability of procedures. *arXiv preprint arXiv:1707.03886*, 2017.

[93] Mark Díaz, Ian Kivlichan, Rachel Rosen, Dylan Baker, Razvan Amironesei, Vinodkumar Prabhakaran, and Emily Denton. Crowdworksheets: Accounting for individual and collective identities underlying crowdsourced dataset annotation. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 2342–2351, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3534647. URL `https://doi.org/10.1145/3531146.3534647`.

[94] Catherine D'Ignazio. *Counting Feminicide: Data Feminism in Action.* MIT Press, 2023.

[95] Catherine D'ignazio and Lauren F Klein. *Data feminism.* MIT press, 2020.

[96] Catherine D'Ignazio, Isadora Cruxên, Ángeles Martinez, Mariel García-Montes, Helena Suárez Val, Silvana Fumega, Harini Suresh, and Wonyoung So. Feminicide & counterdata collection: Activist efforts to monitor and challenge gender-based violence. *In submission*, 2022.

[97] Jonathan Dodge, Q Vera Liao, Yunfeng Zhang, Rachel KE Bellamy, and Casey Dugan. Explaining models: an empirical study of how explanations impact fairness judgment. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 275–285, 2019.

[98] Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv:1702.08608 [cs, stat]*, March 2017. URL `http://arxiv.org/abs/1702.08608`. arXiv: 1702.08608.

[99] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[100] Finale Doshi-Velez, Mason Kortz, Ryan Budish, Chris Bavitz, Sam Gershman, David O'Brien, Stuart Schieber, James Waldo, David Weinberger, and Alexandra Wood. Accountability of ai under the law: The role of explanation. *arXiv preprint arXiv:1711.01134*, 2017.

[101] Ravit Dotan and Smitha Milli. Value-laden disciplinary shifts in machine learning. fat*'20: Proceedings of the 2020 conference on fairness, accountability, and transparency (jan. 2020), 2020.

[102] Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.

[103] Hubert L Dreyfus and Stuart E Dreyfus. The power of human intuition and expertise in the era of the computer. *Mind over machine. Nueva York: The Free Press*, 1986.

[104] Alice Driver. *More or less dead: Feminicide, haunting, and the ethics of representation in Mexico*. University of Arizona Press, 2015.

[105] Johanna Drucker. Humanistic theory and digital scholarship. *Debates in the digital humanities*, 150:85–95, 2012.

[106] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019.

[107] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, December 2019. ISSN 0001-0782, 1557-7317. doi: 10.1145/3359786. URL `https://dl.acm.org/doi/10.1145/3359786`.

[108] Kristin Nicole Dukes and Sarah E Gaither. Black racial stereotypes and victim blaming: Implications for media coverage and criminal proceedings in cases of police violence against racial and ethnic minorities. *Journal of Social Issues*, 73(4):789–807, 2017.

[109] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 119–133, New York, NY, USA, 23–24 Feb 2018. PMLR. URL `http://proceedings.mlr.press/v81/dwork18a.html`.

[110] Catherine D'Ignazio, Helena Suárez Val, Silvana Fumega, Harini Suresh, and Isadora Cruxên. Feminicide & machine learning: detecting gender-based violence to strengthen civil sector activism. *Mechanism Design for Social Good (MD4SG '20)*, 2020.

[111] Danielle Ensign, Sorelle A. Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. Runaway feedback loops in predictive policing. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 160–171, New York, NY, USA, 23–24 Feb 2018. PMLR. URL `http://proceedings.mlr.press/v81/ensign18a.html`.

[112] Michael Eraut. Knowledge, working practices, and learning. In *Learning through practice*, pages 37–58. Springer, 2010.

[113] Virginia Eubanks. *Automating inequality: How high-tech tools profile, police, and punish the poor*. St. Martin's Press, 2018.

[114] Dezhi Fang, Fred Hohman, Peter Polack, Hillol Sarker, Minsuk Kahng, Moushumi Sharmin, Mustafa al'Absi, and Duen Horng Chau. mhealth visual discovery dashboard. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 237–240, 2017.

[115] Shayan Fazeli. *ECG Heartbeat Categorization Dataset*. URL `https://www.kaggle.com/shayanfazeli/heartbeat`.

[116] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*, 2021.

[117] Juliana J. Ferreira and Mateus S. Monteiro. What Are People Doing About XAI User Experience? A Survey on AI Explainability Research and Practice.

In Aaron Marcus and Elizabeth Rosenzweig, editors, *Design, User Experience, and Usability. Design for Contemporary Interactive Environments*, Lecture Notes in Computer Science, pages 56–73, Cham, 2020. Springer International Publishing. ISBN 978-3-030-49760-6. doi: 10.1007/978-3-030-49760-6_4.

[118] Casey Fiesler, Joshua McCann, Kyle Frye, Jed R Brubaker, *et al.* Reddit rules! characterizing an ecosystem of governance. In *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[119] Jenny Rose Finkel, Trond Grenager, and Christopher D Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, 2005.

[120] Jessie Finocchiaro, Roland Maio, Faidra Monachou, Gourab K Patro, Manish Raghavan, Ana-Andreea Stoica, and Stratis Tsirtsis. Bridging machine learning and mechanism design towards algorithmic fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 489–503, 2021.

[121] James Fleck. Expertise: knowledge, power and tradeability. In *Exploring expertise*, pages 143–171. Springer, 1998.

[122] Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[123] Rosa-Linda Fregoso and Cynthia Bejarano. Introduction: A cartography of feminicide in the americas. In *Terrorizing Women*, pages 1–42. Duke University Press, 2010.

[124] William R. Frey, Desmond U. Patton, Michael B. Gaskell, and Kyle A. McGregor. Artificial intelligence and inclusion: Formerly gang-involved youth as domain experts for analyzing unstructured twitter data. *Social Science Computer Review*, 38(1):42–56, 2020. doi: 10.1177/0894439318788314. URL `https://doi.org/10.1177/0894439318788314`.

[125] Friedler, Scheidegger, Venkatasubramanian, Choudhary, Hamilton, and Roth. A comparative study of fairness-enhancing interventions in machine learning. In *ACM Conference on Fairness, Accountability and Transparency (FAT*)*. ACM, 2019. URL `http://arxiv.org/abs/1802.04422`.

[126] Batya Friedman and Helen Nissenbaum. Bias in computer systems. *ACM Transactions on Information Systems (TOIS)*, 14(3):330–347, 1996.

[127] Evgeniy Gabrilovich and Shaul Markovitch. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34:443–498, 2009.

[128] Mariel García-Montes. View source: making secure communication tools more usable. *XRDS: Crossroads, The ACM Magazine for Students*, 26(4):16–19, 2020.

[129] Claudia Garcia-Moreno, Alessandra Guedes, Wendy Knerr, R Jewkes, S Bott, and S Ramsay. Understanding and addressing violence against women. *World Health Organization, Issue brief No. WHO/RHR/12.37)(S. Ramsay, Ed.)*, 2012.

[130] Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644, 2018.

[131] Susanne Gaube, Harini Suresh, Martina Raue, Alexander Merritt, Seth J Berkowitz, Eva Lermer, Joseph F Coughlin, John V Guttag, Errol Colak, and Marzyeh Ghassemi. Do as ai say: susceptibility in deployment of clinical decision-aids. *NPJ digital medicine*, 4(1):1–8, 2021.

[132] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[133] Kristen Gilchrist. "newsworthy" victims? *Feminist Media Studies*, 10(4):373–390, 2010. doi: 10.1080/14680777.2010.514110. URL `https://doi.org/10.1080/14680777.2010.514110`.

[134] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana 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, October 2018. doi: 10.1109/DSAA.2018.00018.

[135] Alyssa Glass, Deborah L. McGuinness, and Michael Wolverton. Toward establishing trust in adaptive agents. In *Proceedings of the 13th international conference on Intelligent user interfaces - IUI '08*, page 227, Gran Canaria, Spain,

2008. ACM Press. ISBN 978-1-59593-987-6. doi: 10.1145/1378773.1378804. URL http://portal.acm.org/citation.cfm?doid=1378773.1378804.

[136] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D Hansen, and Jonathan C Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[137] Karan Goel, Nazneen Rajani, Jesse Vig, Samson Tan, Jason Wu, Stephan Zheng, Caiming Xiong, Mohit Bansal, and Christopher Ré. Robustness gym: Unifying the nlp evaluation landscape. *arXiv preprint arXiv:2101.04840*, 2021.

[138] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual Visual Explanations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, Long Beach, California, USA, June 2019. URL http://proceedings.mlr.press/v97/goyal19a.html. arXiv: 1904.07451.

[139] Joanne Gray and Alice Witt. A feminist data ethics of care for machine learning: The what, why, who and how. *First Monday*, 2021.

[140] Mary L Gray and Siddharth Suri. *Ghost work: How to stop Silicon Valley from building a new global underclass*. Eamon Dolan Books, 2019.

[141] Thomas RG Green. Cognitive dimensions of notations. *People and computers V*, pages 443–460, 1989.

[142] Davydd Greenwood and Morten Levin. *Introduction to Action Research*. SAGE Publications, Inc., 2455 Teller Road, Thousand Oaks California 91320 United States of America, 2007. ISBN 978-1-4129-2597-6 978-1-4129-8461-4. doi: 10.4135/9781412984614. URL http://methods.sagepub.com/book/introduction-to-action-research.

[143] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[144] Frank Haist, Arthur P Shimamura, and Larry R Squire. On the relationship between recall and recognition memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18(4):691, 1992.

[145] Aimi Hamraie and Kelly Fritsch. Crip technoscience manifesto. *Catalyst: Feminism, Theory, Technoscience*, 5(1):1–33, 2019.

[146] Leif Hancox-Li and I Elizabeth Kumar. Epistemic values in feature importance methods: Lessons from feminist epistemology. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 817–826, 2021.

[147] Alex Hanna. Mpeds: Automating the generation of protest event data. 2017.

[148] Laura Hanu and Unitary team. Detoxify. Github. https://github.com/unitaryai/detoxify, 2020.

[149] Donna Haraway. Situated knowledges: The science question in feminism and the privilege of partial perspective. *Feminist Studies*, 14(3):575–599, 1988. ISSN 00463663. URL `http://www.jstor.org/stable/3178066`.

[150] Donna Haraway. Situated knowledges: The science question in feminism and the privilege of partial perspective. *Feminist studies*, 14(3):575–599, 1988.

[151] Donna Haraway. Situated knowledges: The science question in feminism and the privilege of partial perspective. In *Feminist theory reader*, pages 303–310. Routledge, 2020.

[152] Christina Harrington, Sheena Erete, and Anne Marie Piper. Deconstructing community-based collaborative design: Towards more equitable participatory design engagements. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–25, 2019.

[153] Johanna Hartelius. *The Rhetoric of Expertise*. PhD thesis, The University of Texas at Austin, 2008.

[154] Johanna Hartelius. Rhetorics of Expertise. *Social Epistemology*, 25(3):211–215, July 2011. ISSN 0269-1728. doi: 10.1080/02691728.2011.578301. URL `https://doi.org/10.1080/02691728.2011.578301`. Publisher: Routledge _eprint: https://doi.org/10.1080/02691728.2011.578301.

[155] Florian Heimerl and Michael Gleicher. Interactive analysis of word vector embeddings. In *Computer Graphics Forum*, volume 37, pages 253–265. Wiley Online Library, 2018.

[156] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HJz6tiCqYm`.

[157] Matt Henry. Risk assessment: Explained. *The Marshall Project*, 25, 2019.

[158] Sam Hepenstal and David McNeish. Explainable artificial intelligence: What do you need to know? In Dylan D. Schmorrow and Cali M. Fidopiastis, editors, *Augmented Cognition. Theoretical and Technological Approaches*, pages 266–275, Cham, 2020. Springer International Publishing. ISBN 978-3-030-50353-6.

[159] Mireille Hildebrandt. The dawn of a critical transparency right for the profiling era. *Astronomy & Astrophysics - ASTRON ASTROPHYS*, 06 2012. doi: 10.3233/978-1-61499-057-4-41.

[160] Tad Hirsch, Kritzia Merced, Shrikanth Narayanan, Zac E. Imel, and David C. Atkins. Designing contestability: Interaction design, machine learning, and mental health. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, DIS '17, page 95–99, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349222. doi: 10.1145/3064663.3064703. URL `https://doi.org/10.1145/3064663.3064703`.

[161] Kelly M Hoffman, Sophie Trawalter, Jordan R Axt, and M Norman Oliver. Racial bias in pain assessment and treatment recommendations, and false beliefs about biological differences between blacks and whites. *Proceedings of the National Academy of Sciences*, 113(16):4296–4301, 2016.

[162] Diane E Hoffmann and Anita J Tarzian. The girl who cried pain: a bias against women in the treatment of pain. *The Journal of Law, Medicine & Ethics*, 28:13–27, 2001.

[163] Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M. Drucker. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300809. URL `https://doi-org.libproxy.mit.edu/10.1145/3290605.3300809`.

[164] Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2674–2693, August 2019. ISSN 1941-0506. doi: 10.1109/TVCG.2018.2843369.

[165] Sungsoo Ray Hong, Jessica Hullman, and Enrico Bertini. Human factors in model interpretability: Industry practices, challenges, and needs. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–26, 2020.

[166] Sungsoo Ray Hong, Jessica Hullman, and Enrico Bertini. Human Factors in Model Interpretability: Industry Practices, Challenges, and Needs. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–26, May 2020. ISSN 2573-0142, 2573-0142. doi: 10.1145/3392878. URL `https://dl.acm.org/doi/10.1145/3392878`.

[167] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. 2017.

[168] Sara Hooker. Moving beyond "algorithmic bias is a data problem". *Patterns*, 2 (4):100241, 2021.

[169] Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. Characterising bias in compressed models, 2020. URL `https://arxiv.org/abs/2010.03058`.

[170] Jessica Hullman and Nick Diakopoulos. Visualization rhetoric: Framing effects in narrative visualization. *IEEE transactions on visualization and computer graphics*, 17(12):2231–2240, 2011.

[171] Edwin L Hutchins, James D Hollan, and Donald A Norman. Direct manipulation interfaces. *Human–computer interaction*, 1(4):311–338, 1985.

[172] Alon Jacovi, Ana Marasović, Tim Miller, and Yoav 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.

[173] Fei Jiang, Yong Jiang, Hui Zhi, Yi Dong, Hao Li, Sufeng Ma, Yilong Wang, Qiang Dong, Haipeng Shen, and Yongjun Wang. Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology*, 2(4), 2017.

[174] Kenneth Jones and Tema Okun. White supremacy culture. *Dismantling Racism: A Workbook for Social Change*, 2001.

[175] Janet Jull, Audrey Giles, and Ian D. Graham. Community-based participatory research and integrated knowledge translation: advancing the co-creation of knowledge. *Implementation Science*, 12(1):150, December 2017. ISSN 1748-5908. doi: 10.1186/s13012-017-0696-3. URL `https://implementationscience.biomedcentral.com/articles/10.1186/s13012-017-0696-3`.

[176] Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. ECG Heartbeat Classification: A Deep Transferable Representation. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 443–444, New York, NY, June 2018. IEEE. ISBN 978-1-5386-5377-7. doi: 10.1109/ICHI.2018.00092. URL `https://ieeexplore.ieee.org/document/8419425/`.

[177] Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1548–1558, 2021.

[178] Michael Katell, Meg Young, Dharma Dailey, Bernease Herman, Vivian Guetler, Aaron Tam, Corinne Bintz, Daniella Raz, and PM Krafft. Toward situated interventions for algorithmic equity: lessons from the field. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 45–55, 2020.

[179] Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer 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, Honolulu HI USA, April 2020. ACM. ISBN 978-1-4503-6708-0. doi: 10.1145/3313831.3376219. URL `https://dl.acm.org/doi/10.1145/3313831.3376219`.

[180] Katherine A Keith, Abram Handler, Michael Pinkham, Cara Magliozzi, Joshua McDuffie, and Brendan O'Connor. Identifying civilians killed by police with distantly supervised entity-event extraction. *arXiv preprint arXiv:1707.07086*, 2017.

[181] Tamil Kendall. *A Synthesis of Evidence on the Collection and Use of Administrative Data on Violence against Women: Background Paper for the Development of Global Guidance*. UN Women, 2020.

[182] Helen Kennedy, Rosemary Lucy Hill, Giorgia Aiello, and William Allen. The work that visualisation conventions do. *Information, Communication & Society*, 19(6):715–735, 2016.

[183] Os Keyes. The misgendering machines: Trans/hci implications of automatic gender recognition. *Proceedings of the ACM on human-computer interaction*, 2(CSCW):1–22, 2018.

[184] Been Kim. *Interactive and Interpretable Machine Learning Models for Human Machine Collaboration*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 2015.

[185] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! Criticism for Interpretability. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2280–2288. Curran Associates, Inc., 2016. URL `http://papers.nips.cc/paper/6300-examples-are-not-enough-learn-to-criticize-criticism-for-interpretability.pdf`.

[186] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda 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.

[187] Ian M. Kinchin and B. Cabot. Reconsidering the dimensions of expertise: from linear stages towards dual processing. *London Review of Education*, July 2010. ISSN 1474-8460. doi: 10.1080/14748460.2010.487334. URL `https://scienceopen.com/document?vid=7155ea86-ebe9-4f03-9eb8-90305f80b728`.

[188] René F Kizilcec. How much information? effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2390–2395, 2016.

[189] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent Trade-Offs in the Fair Determination of Risk Scores. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 43:1–43:23, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-029-3. doi: 10.4230/

LIPIcs.ITCS.2017.43. URL http://drops.dagstuhl.de/opus/volltexte/2017/8156.

[190] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Algorithmic fairness. In *AEA Papers and Proceedings*, volume 108, pages 22–27, 2018.

[191] Daniel N Kluttz, Nitin Kohli, and Deirdre K Mulligan. Shaping our tools: Contestability as a means to promote responsible algorithmic decision making in the professions. In *Ethics of Data and Analytics*, pages 420–428. Auerbach Publications, 2022.

[192] Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, Sydney, Australia, July 2017. URL http://arxiv.org/abs/1703.04730. arXiv: 1703.04730.

[193] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, *et al.* Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[194] Ha-Kyung Kong, Zhicheng Liu, and Karrie Karahalios. Frames and slants in titles of visualizations on controversial topics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

[195] Ha-Kyung Kong, Zhicheng Liu, and Karrie Karahalios. Trust and recall of information across varying degrees of title-visualization misalignment. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[196] Luciana Monteiro Krebs, Oscar Luis Alvarado Rodriguez, Pierre Dewitte, Jef Ausloos, David Geerts, Laurens Naudts, and Katrien Verbert. Tell me what you know: Gdpr implications on designing transparency and accountability for news recommender systems. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.

[197] Kostiantyn Kucher and Andreas Kerren. Text visualization techniques: Taxonomy, visual survey, and community insights. In *2015 IEEE Pacific visualization symposium (pacificVis)*, pages 117–121. IEEE, 2015.

[198] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of Explanatory Debugging to Personalize Interactive Machine Learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces - IUI '15*, pages 126–137, Atlanta, Georgia, USA, 2015. ACM Press. ISBN 978-1-4503-3306-1. doi: 10.1145/2678025.2701399. URL `http://dl.acm.org/citation.cfm?doid=2678025.2701399`.

[199] Bogdan Kulynych, David Madras, Smitha Milli, Inioluwa Deborah Raji, Angela Zhou, and Richard Zemel. Participatory approaches to machine learning. In *International Conference on Machine Learning Workshop*, 2020.

[200] Bogdan Kulynych, David Madras, Smitha Milli, Inioluwa Deborah Raji, Angela Zhou, and Richard Zemel. Participatory approaches to machine learning. International Conference on Machine Learning Workshop, July 2020.

[201] Deepak Kumar, Patrick Gage Kelley, Sunny Consolvo, Joshua Mason, Elie Bursztein, Zakir Durumeric, Kurt Thomas, and Michael Bailey. Designing toxic content classification for a diversity of perspectives. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*, pages 299–318, 2021.

[202] David F Labaree. On the nature of teaching and teacher education: Difficult practices that look easy. *Journal of teacher education*, 51(3):228–233, 2000.

[203] Marcela Lagarde y de los Ríos. *Género y feminismo: desarrollo humano y democracia*. Siglo XXI Editores México, 1996.

[204] Marcela Lagarde y de los Ríos. Preface: Feminist Keys for Understanding Feminicide: Theoretical, Political, and Legal Construction. In *Terrorizing Women: Feminicide in the Americas*. Duke University Press, 01 2010. ISBN 978-0-8223-4669-2. doi: 10.1215/9780822392644.

[205] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Samuel J. Gershman, and Finale Doshi-Velez. Human Evaluation of Models Built for Interpretability. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 7(1):59–67, October 2019. URL `https://www.aaai.org/ojs/index.php/HCOMP/article/view/5280`. Number: 1.

[206] Vivian Lai and Chenhao 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 - FAT* '19*, pages 29–38, Atlanta, GA, USA, 2019.

ACM Press. ISBN 978-1-4503-6125-5. doi: 10.1145/3287560.3287590. URL http://dl.acm.org/citation.cfm?doid=3287560.3287590.

[207] Himabindu Lakkaraju and Osbert Bastani. "How do I fool you?": Manipulating User Trust via Misleading Black Box Explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, pages 79–85, New York, NY, USA, February 2020. Association for Computing Machinery. ISBN 978-1-4503-7110-0. doi: 10.1145/3375627.3375833. URL https://doi.org/10.1145/3375627.3375833.

[208] Christopher A. Le Dantec and Sarah Fox. Strangers at the Gate: Gaining Access, Building Rapport, and Co-Constructing Community-Based Research. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '15*, pages 1348–1358, Vancouver, BC, Canada, 2015. ACM Press. ISBN 978-1-4503-2922-4. doi: 10.1145/2675133.2675147. URL http://dl.acm.org/citation.cfm?doid=2675133.2675147.

[209] Bongshin Lee, Kate Isaacs, Danielle Albers Szafir, G Elisabeta Marai, Cagatay Turkay, Melanie Tory, Sheelagh Carpendale, and Alex Endert. Broadening intellectual diversity in visualization research papers. *IEEE computer graphics and applications*, 39(4):78–85, 2019.

[210] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1):50–80, 2004.

[211] Alyssa Lees, Vinh Q Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. A new generation of perspective api: Efficient multilingual character-level transformers. *arXiv preprint arXiv:2202.11176*, 2022.

[212] Kalev Leetaru and Philip A Schrodt. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer, 2013.

[213] Min Li, Amy Mickel, and Stanley Taylor. "should this loan be approved or denied?": A large dataset with class assignment guidelines. *Journal of Statistics Education*, 26(1):55–66, 2018. doi: 10.1080/10691898.2018.1434342. URL https://doi.org/10.1080/10691898.2018.1434342.

[214] Q Vera Liao, Daniel Gruen, and Sarah Miller. Questioning the ai: Informing design practices for explainable ai user experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2020.

[215] Q. Vera Liao, Daniel Gruen, and Sarah Miller. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–15, Honolulu, HI, USA, April 2020. Association for Computing Machinery. ISBN 978-1-4503-6708-0. doi: 10.1145/3313831.3376590. URL https://doi.org/10.1145/3313831.3376590.

[216] Brian Y. Lim and Anind K. Dey. Assessing demand for intelligibility in context-aware applications. In *Proceedings of the 11th International Conference on Ubiquitous Computing*, UbiComp '09, page 195–204, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584317. doi: 10.1145/1620545.1620576. URL https://doi-org.libproxy.mit.edu/10.1145/1620545.1620576.

[217] Brian Y. Lim and Anind K. Dey. Toolkit to support intelligibility in context-aware applications. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, page 13–22, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605588438. doi: 10.1145/1864349.1864353. URL https://doi-org.libproxy.mit.edu/10.1145/1864349.1864353.

[218] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 2119–2128, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605582467. doi: 10.1145/1518701.1519023. URL https://doi-org.libproxy.mit.edu/10.1145/1518701.1519023.

[219] Peter Lipton. Contrastive explanation. *Royal Institute of Philosophy Supplements*, 27:247–266, 1990.

[220] Zachary C. Lipton. The mythos of model interpretability. *Communications of the ACM*, 61(10):36–43, September 2018. ISSN 0001-0782, 1557-7317. doi: 10.1145/3233231. URL https://dl.acm.org/doi/10.1145/3233231.

[221] Yang Liu, Eunice Jun, Qisheng Li, and Jeffrey Heer. Latent space cartography: Visual analysis of vector space embeddings. In *Computer Graphics Forum*, volume 38, pages 67–78. Wiley Online Library, 2019.

[222] Yunhui Long, Vincent Bindschaedler, and Carl A Gunter. Towards measuring membership privacy. *arXiv preprint arXiv:1712.09136*, 2017.

[223] Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, *et al.* Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering*, 2(10): 749–760, 2018.

[224] Donald Martin Jr, Vinodkumar Prabhakaran, Jill Kuhlberg, Andrew Smart, and William S Isaac. Extending the machine learning abstraction boundary: A complex systems approach to incorporate societal context. *arXiv preprint arXiv:2006.09663*, 2020.

[225] Donald Martin Jr, Vinodkumar Prabhakaran, Jill Kuhlberg, Andrew Smart, and William S Isaac. Participatory problem formulation for fairer machine learning through community based system dynamics. *International Conference on Learning Representations (ICLR)*, 2020.

[226] Laurenellen McCann. Building technology with, not for communities: An engagement guide for civic tech, Mar 2015. URL `https://medium.com/organizer-sandbox/building-technology-with-not-for-communities-an-engagement-guide-for-civic-t`

[227] Jesse McCrosky and Brandi Geurkink. YouTube Regrets: A crowdsourced investigation into YouTube's recommendation algorithm. Technical report, Mozilla Foundation, July 2021.

[228] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[229] Amanda Meng and Carl DiSalvo. Grassroots resource mobilization through counter-data action. *Big Data & Society*, 5(2):2053951718796862, 2018. doi: 10.1177/2053951718796862. URL `https://doi.org/10.1177/2053951718796862`.

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

[231] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, February 2019. ISSN 00043702. doi: 10.1016/j.artint.2018.07.007. URL `https://linkinghub.elsevier.com/retrieve/pii/S0004370218305988`.

[232] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 220–229, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. doi: 10.1145/3287560.3287596. URL https://doi.org/10.1145/3287560.3287596.

[233] Shakir Mohamed, Marie-Therese Png, and William Isaac. Decolonial AI: Decolonial Theory as Sociotechnical Foresight in Artificial Intelligence. *Philosophy & Technology*, July 2020. ISSN 2210-5433, 2210-5441. doi: 10.1007/s13347-020-00405-8. URL http://arxiv.org/abs/2007.04068. arXiv: 2007.04068.

[234] Sina Mohseni, Jeremy E. Block, and Eric D. Ragan. A Human-Grounded Evaluation Benchmark for Local Explanations of Machine Learning. *arXiv:1801.05075 [cs]*, June 2020. URL http://arxiv.org/abs/1801.05075. arXiv: 1801.05075.

[235] Sina Mohseni, Niloofar Zarei, and Eric D. Ragan. A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems. *arXiv:1811.11839 [cs]*, April 2020. URL http://arxiv.org/abs/1811.11839. arXiv: 1811.11839.

[236] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[237] Jana M Mossey. Defining racial and ethnic disparities in pain management. *Clinical Orthopaedics and Related Research®*, 469(7):1859–1870, 2011.

[238] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.

[239] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*,

pages 607–617, Barcelona Spain, January 2020. ACM. ISBN 978-1-4503-6936-7. doi: 10.1145/3351095.3372850. URL https://dl.acm.org/doi/10.1145/3351095.3372850.

[240] Sajad Mousavi, Fatemeh Afghah, and U Rajendra Acharya. Han-ecg: An interpretable atrial fibrillation detection model using hierarchical attention networks. *arXiv preprint arXiv:2002.05262*, 2020.

[241] Deirdre K Mulligan, Daniel Kluttz, and Nitin Kohli. Shaping our tools: Contestability as a means to promote responsible algorithmic decision making in the professions. *Available at SSRN 3311894*, 2019.

[242] Laura Nader. Up the anthropologist: Perspectives gained from studying up. 1972.

[243] Arvind Narayanan. Fat* tutorial: 21 fairness definitions and their politics. *New York, NY, USA*, 2018.

[244] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682*, 2018.

[245] Jennifer C Nash. *Black feminism reimagined*. Duke University Press, 2018.

[246] Safiya Umoja Noble. Algorithms of oppression. In *Algorithms of Oppression*. New York University Press, 2018.

[247] Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasanakis, *et al.* Bias in data-driven artificial intelligence systems—an introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1356, 2020.

[248] Ingrid Nunes and Dietmar Jannach. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, 27(3–5):393–444, December 2017. ISSN 0924-1868. doi: 10.1007/s11257-017-9195-0. URL https://doi-org.libproxy.mit.edu/10.1007/s11257-017-9195-0.

[249] Ingrid Nunes and Dietmar Jannach. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, 27(3-5):393–444, 2017.

[250] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

[251] Augustus Odena, Catherine Olsson, David Andersen, and Ian Goodfellow. Tensorfuzz: Debugging neural networks with coverage-guided fuzzing. In *International Conference on Machine Learning*, pages 4901–4911. PMLR, 2019.

[252] Ihudiya Finda Ogbonnaya-Ogburu, Angela D.R. Smith, Alexandra To, and Kentaro Toyama. Critical Race Theory for HCI. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–16, Honolulu HI USA, April 2020. ACM. ISBN 978-1-4503-6708-0. doi: 10.1145/3313831.3376392. URL `https://dl.acm.org/doi/10.1145/3313831.3376392`.

[253] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[254] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

[255] Mimi Onuoha and Diana Nucera. A people's guide to ai. 2018.

[256] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

[257] James O'Malley. Captcha if you can: How you've been training ai for years without realizing it. tech radar. *URL https://bit. ly/37H1esA*, 2018.

[258] Nicolas Papernot and Patrick McDaniel. Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning. *arXiv:1803.04765 [cs, stat]*, March 2018. URL `http://arxiv.org/abs/1803.04765`. arXiv: 1803.04765.

[259] Prasanna Parasurama and João Sedoc. Gendered information in resumes and its role in algorithmic and human hiring bias. In *Academy of Management Proceedings*, volume 2022, page 17133. Academy of Management Briarcliff Manor, NY 10510, 2022.

[260] Desmond U. Patton, William R. Frey, Kyle A. McGregor, Fei-Tzin Lee, Kathleen McKeown, and Emanuel Moss. Contextual analysis of social media: The promise and challenge of eliciting context in social media posts with natural language processing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, page 337–342, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371100. doi: 10.1145/3375627.3375841. URL https://doi.org/10.1145/3375627.3375841.

[261] Prajwal Paudyal, Junghyo Lee, Azamat Kamzin, Mohamad Soudki, Ayan Banerjee, and Sandeep KS Gupta. Learn2sign: Explainable ai for sign language learning. In *IUI Workshops*, 2019.

[262] Evan M Peck, Sofia E Ayuso, and Omar El-Etr. Data is personal: Attitudes and perceptions of data visualization in rural pennsylvania. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[263] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, *et al.* Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[264] Claudia Perlich, Brian Dalessandro, Troy Raeder, Ori Stitelman, and Foster Provost. Machine learning for targeted display advertising: Transfer learning in action. *Machine learning*, 95(1):103–127, 2014.

[265] Pew Research Center. Online harassment 2017, 2017. URL https://www.pewresearch.org/internet/2017/07/11/online-harassment-2017/.

[266] Sean M Phelan, Diane J Burgess, Mark W Yeazel, Wendy L Hellerstedt, Joan M Griffin, and Michelle van Ryn. Impact of weight bias and stigma on quality of care and outcomes for patients with obesity. *obesity reviews*, 16(4):319–326, 2015.

[267] Washington Post. Fatal force [database], 2016. URL https://www.washingtonpost.com/graphics/investigations/police-shootings-database/.

[268] Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. Manipulating and Measuring Model Interpretability. *arXiv:1802.07810 [cs]*, November 2019. URL http://arxiv.org/abs/1802.07810. arXiv: 1802.07810.

[269] Alun Preece, Dan Harborne, Dave Braines, Richard Tomsett, and Supriyo Chakraborty. Stakeholders in Explainable AI. *arXiv:1810.00184 [cs]*, September 2018. URL http://arxiv.org/abs/1810.00184. arXiv: 1810.00184.

[270] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

[271] Jill Radford and Diana EH Russell. *Femicide: The politics of woman killing*. Twayne Publishers, 1992.

[272] Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 469–481, 2020.

[273] Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. Closing the ai accountability gap: defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 33–44, 2020.

[274] Gabriëlle Ras, Marcel van Gerven, and Pim Haselager. Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges. In Hugo Jair Escalante, Sergio Escalera, Isabelle Guyon, Xavier Baró, Yağmur Güçlütürk, Umut Güçlü, and Marcel van Gerven, editors, *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, pages 19–36. Springer International Publishing, Cham, 2018. ISBN 978-3-319-98131-4. doi: 10.1007/978-3-319-98131-4_2. URL https://doi.org/10.1007/978-3-319-98131-4_2.

[275] Alexander Renkl. Toward an instructionally oriented theory of example-based learning. *Cognitive science*, 38(1):1–37, 2014.

[276] Alexander Renkl. Toward an Instructionally Oriented Theory of Example-Based Learning. *Cognitive Science*, 38(1):1–37, January 2014. ISSN 03640213. doi: 10.1111/cogs.12086. URL http://doi.wiley.com/10.1111/cogs.12086.

[277] Alexander Renkl, Tatjana Hilbert, and Silke Schworm. Example-Based Learning in Heuristic Domains: A Cognitive Load Theory Account. *Educational Psychology Review*, 21(1):67–78, March 2009. ISSN 1040-726X, 1573-336X. doi: 10.1007/s10648-008-9093-4. URL `http://link.springer.com/10.1007/s10648-008-9093-4`.

[278] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

[279] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, San Francisco California USA, August 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL `https://dl.acm.org/doi/10.1145/2939672.2939778`.

[280] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, 2020.

[281] Mireia Ribera and Agata Lapedriza. Can we do better explanations? a proposal of user-centered explainable ai. In *IUI Workshops*, 2019.

[282] Bernhard Rieder and Yarden Skop. The fabrics of machine moderation: Studying the technical, normative, and organizational structure of perspective api. *Big Data & Society*, 8(2):20539517211046181, 2021.

[283] Hal Roberts, Rahul Bhargava, Linas Valiukas, Dennis Jen, Momin M Malik, Cindy Sherman Bishop, Emily B Ndulue, Aashka Dave, Justin Clark, Bruce Etling, *et al.* Media cloud: Massive open source collection of global news on the open web. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 1034–1045, 2021.

[284] Ribana Roscher, Bastian Bohn, Marco F. Duarte, and Jochen Garcke. Explainable Machine Learning for Scientific Insights and Discoveries. *IEEE Access*, 8: 42200–42216, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2976199.

[285] Hee Jung Ryu, Hartwig Adam, and Margaret Mitchell. Inclusivefacenet: Improving face attribute detection with race and gender diversity. In *Workshop on*

Fairness, Accountability, and Transparency in Machine Learning (FAT/ML), 2018.

[286] Steven L Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data mining and knowledge discovery*, 1(3):317–328, 1997.

[287] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[288] Giovanna Sannino and Giuseppe De Pietro. A deep learning approach for ecg-based heartbeat classification for arrhythmia detection. *Future Generation Computer Systems*, 86:446–455, 2018.

[289] Suchi Saria and Adarsh Subbaswamy. Tutorial: safe and reliable machine learning. *arXiv preprint arXiv:1904.07204*, 2019.

[290] Camilo Bernal Sarmiento, Miguel Lorente Acosta, Françoise Roth, and Margarita Zambrano. Latin american model protocol for the investigation of gender-related killings of women (femicide/feminicide). *New York: United Nations High Commissioner for Human Rights (OHCHR) and UN Women*, 2014.

[291] Morgan Klaus Scheuerman, Kandrea Wade, Caitlin Lustig, and Jed R Brubaker. How we've taught algorithms to see identity: Constructing race and gender in image databases for facial analysis. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–35, 2020.

[292] Udo Schlegel, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, and Daniel A. Keim. Towards a Rigorous Evaluation of XAI Methods on Time Series. *arXiv:1909.07082 [cs]*, September 2019. URL `http://arxiv.org/abs/1909.07082`. arXiv: 1909.07082.

[293] Johannes Schneider and Joshua Handali. PERSONALIZED EXPLANATION FOR MACHINE LEARNING: A CONCEPTUALIZATION. In *Proceedings of the 27th European Conference on Information Systems (ECIS)*, Stockholm & Uppsala, Sweden, June 2019. URL `https://aisel.aisnet.org/ecis2019_rp/171`.

[294] Andrew D Selbst, Danah Boyd, Sorelle A Friedler, Suresh Venkatasubramanian, and Janet Vertesi. Fairness and abstraction in sociotechnical systems. In

Proceedings of the Conference on Fairness, Accountability, and Transparency, pages 59–68. ACM, 2019.

[295] Mark Sendak, Madeleine Clare Elish, Michael Gao, Joseph Futoma, William Ratliff, Marshall Nichols, Armando Bedoya, Suresh Balu, and Cara O'Brien. " the human body is a black box" supporting clinical decision-making with deep learning. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 99–109, 2020.

[296] Shreya Shankar, Yoni Halpern, Eric Breck, James Atwood, Jimbo Wilson, and D. Sculley. No classification without representation: Assessing geodiversity issues in open data sets for the developing world. In *NIPS 2017 workshop: Machine Learning for the Developing World*, 2017.

[297] Chung Kwan Shin and Sang Chan Park. Memory and neural network based expert system. *Expert Systems with Applications*, 16(2):145–155, 1999.

[298] Ben Shneiderman. Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human–Computer Interaction*, 36(6): 495–504, 2020.

[299] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[300] M Sloane, E Moss, O Awomolo, and L Forlano. Participation is not a design fix for machine learning. *Participatory Approaches to Machine Learning*, 2020.

[301] Kacper Sokol and Peter Flach. One explanation does not fit all. *KI-Künstliche Intelligenz*, pages 1–16, 2020.

[302] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2615–2632, 2021.

[303] Thilo Spinner, Udo Schlegel, Hanna Schäfer, and Mennatallah El-Assady. explAIner: A Visual Analytics Framework for Interactive and Explainable Machine Learning. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1064–1074, January 2020. ISSN 1941-0506. doi: 10.1109/TVCG.2019.2934629. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[304] Clay Spinuzzi. The methodology of participatory design. *Technical communication*, 52(2):163–174, 2005.

[305] Megan Stevenson. Assessing risk assessment in action. *Minn. L. Rev.*, 103:303, 2018.

[306] Susan Strega, Caitlin Janzen, Jeannie Morgan, Leslie Brown, Robina Thomas, and Jeannine Carriere. Never innocent victims: Street sex workers in canadian print media. *Violence against women*, 20(1):6–25, 2014.

[307] Hendrik Strobelt, Daniela Oelke, Bum Chul Kwon, Tobias Schreck, and Hanspeter Pfister. Guidelines for effective usage of text highlighting techniques. *IEEE transactions on visualization and computer graphics*, 22(1):489–498, 2015.

[308] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the Impact of Feature Attribution Baselines. *Distill*, 5(1):10.23915/distill.00022, January 2020. ISSN 2476-0757. doi: 10.23915/distill.00022. URL `https://distill.pub/2020/attribution-baselines`.

[309] Mukund Sundararajan, Jinhua Xu, Ankur Taly, Rory Sayres, and Amir Najmi. Exploring principled visualizations for deep network attributions. In *IUI Workshops*, volume 4, 2019.

[310] Harini Suresh and John 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. 2021.

[311] Harini Suresh, Jen J. Gong, and John V. Guttag. Learning tasks for multitask learning: Heterogenous patient populations in the icu. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2018.

[312] Harini Suresh, Natalie Lao, and Ilaria Liccardi. Misplaced trust: Measuring the interference of machine learning in human decision-making. In Emilio Ferrara, Pauline Leonard, and Wendy Hall, editors, *WebSci '20: 12th ACM Conference on Web Science, Southampton, UK, July 6-10, 2020*, pages 315–324. ACM, 2020. doi: 10.1145/3394231.3397922. URL `https://doi.org/10.1145/3394231.3397922`.

[313] Harini Suresh, Steven R. Gomez, Kevin K. Nam, and Arvind Satyanarayan. Beyond expertise and roles: A framework to characterize the stakeholders of interpretable machine learning and their needs. CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. doi: 10.1145/3411764.3445088. URL https://doi.org/10.1145/3411764.3445088.

[314] Harini Suresh, Kathleen M Lewis, John Guttag, and Arvind Satyanarayan. Intuitively assessing ml model reliability through example-based explanations and editing model inputs. In *27th International Conference on Intelligent User Interfaces*, pages 767–781, 2022.

[315] Harini Suresh, Rajiv Movva, Amelia Lee Dogan, Rahul Bhargava, Isadora Cruxên, Ángeles Martinez Cuba, Guilia Taurino, Wonyoung So, and Catherine D'Ignazio. Towards intersectional feminist and participatory ml: A case study in supporting feminicide counterdata collection. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 667–678, 2022.

[316] Harini Suresh, Divya Shanmugam, Tiffany Chen, Annie Bryan, John V. Guttag, and Arvind Satyanarayan. Kaleidoscope: Semantically-meaningful, context-grounded ml model testing. CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.

[317] Reed T Sutton, David Pincock, Daniel C Baumgart, Daniel C Sadowski, Richard N Fedorak, and Karen I Kroeker. An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ digital medicine*, 3(1):1–10, 2020.

[318] Jon Swaine, Oliver Laughland, Jamiles Lartey, and Ciara McCarthy. The counted: people killed by police in the us. 2016.

[319] Nithum Thain, Lucas Dixon, and Ellery Wulczyn. Wikipedia Talk Labels: Toxicity. 2 2017. doi: 10.6084/m9.figshare.4563973.v2. URL https://figshare.com/articles/dataset/Wikipedia_Talk_Labels_Toxicity/4563973.

[320] Jim Thatcher, David O'Sullivan, and Dillon Mahmoudi. Data colonialism through accumulation by dispossession: New metaphors for daily data. *Environment and Planning D: Society and Space*, 34(6):990–1006, December 2016. ISSN 0263-7758, 1472-3433. doi: 10.1177/0263775816633195. URL http://journals.sagepub.com/doi/10.1177/0263775816633195.

[321] Andreas Theodorou, Robert H Wortham, and Joanna J Bryson. Designing and implementing transparency for real time inspection of autonomous robots. *Connection Science*, 29(3):230–241, 2017.

[322] Geoffrey H Tison, Jeffrey Zhang, Francesca N Delling, and Rahul C Deo. Automated and interpretable patient ecg profiles for disease detection, tracking, and discovery. *Circulation: Cardiovascular Quality and Outcomes*, 12(9):e005289, 2019.

[323] Richard Tomsett, Dave Braines, Dan Harborne, Alun Preece, and Supriyo Chakraborty. Interpretable to Whom? A Role-based Model for Analyzing Interpretable Machine Learning Systems. In *arXiv:1806.07552 [cs]*, June 2018. URL `http://arxiv.org/abs/1806.07552`. arXiv: 1806.07552.

[324] Sana Tonekaboni, Shalmali Joshi, Melissa D McCradden, and Anna Goldenberg. What clinicians want: Contextualizing explainable machine learning for clinical end use. In *Machine Learning for Healthcare Conference*, pages 359–380, 2019.

[325] Sana Tonekaboni, Shalmali Joshi, Melissa D. McCradden, and Anna Goldenberg. What Clinicians Want: Contextualizing Explainable Machine Learning for Clinical End Use. In *Machine Learning for Healthcare Conference*, pages 359–380, October 2019. URL `http://proceedings.mlr.press/v106/tonekaboni19a.html`. ISSN: 1938-7228 Section: Machine Learning.

[326] Joe Tullio, Anind K. Dey, Jason Chalecki, and James Fogarty. How it works: a field study of non-technical users interacting with an intelligent system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 31–40, San Jose, California, USA, April 2007. Association for Computing Machinery. ISBN 9781595935939. doi: 10.1145/1240624.1240630. URL `https://doi.org/10.1145/1240624.1240630`.

[327] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 10–19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. doi: 10.1145/3287560.3287566. URL `https://doi-org.libproxy.mit.edu/10.1145/3287560.3287566`.

[328] Kristen Vaccaro, Karrie Karahalios, Deirdre K Mulligan, Daniel Kluttz, and Tad Hirsch. Contestability in algorithmic systems. In *Conference compan-*

ion publication of the 2019 on computer supported cooperative work and social computing, pages 523–527, 2019.

[329] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[330] Effy Vayena, Alessandro Blasimme, and I. Glenn Cohen. Machine learning in medicine: Addressing ethical challenges. *PLOS Medicine*, 15(11):e1002689, November 2018. ISSN 1549-1676. doi: 10.1371/journal.pmed.1002689. URL `https://dx.plos.org/10.1371/journal.pmed.1002689`.

[331] Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *arXiv preprint arXiv:2106.00545*, 2021.

[332] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2):841–887, March 2018. URL `http://arxiv.org/abs/1711.00399`. arXiv: 1711.00399.

[333] Sandra Walklate, Kate Fitz-Gibbon, Jude McCulloch, and JaneMaree Maher. *Towards a global femicide index: Counting the costs*. Routledge, 2019.

[334] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y Lim. Designing theory-driven user-centric explainable ai. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–15, 2019.

[335] Martin Wattenberg and Fernanda B Viégas. The word tree, an interactive visual concordance. *IEEE transactions on visualization and computer graphics*, 14(6):1221–1228, 2008.

[336] Ryan Weber. Review of The Rhetoric of Expertise. *Rhetoric and Public Affairs*, 15(1):193–196, 2012. ISSN 1094-8392. URL `https://www.jstor.org/stable/41955617`. Publisher: Michigan State University Press.

[337] Adrian Weller. Transparency: Motivations and Challenges. *arXiv:1708.01870 [cs]*, August 2019. URL `http://arxiv.org/abs/1708.01870`. arXiv: 1708.01870.

[338] David Werner. *Nothing About Us Without Us: Developing Innovative Technologies For, By, and With Disabled Persons*. Healthwrights, 1998.

[339] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65, 2020.

[340] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics*, 26(1):56–65, 2019.

[341] Robin Williams, Wendy Faulkner, and James Fleck, editors. *Exploring Expertise*. Palgrave Macmillan UK, London, 1998. ISBN 978-1-349-13695-7 978-1-349-13693-3. doi: 10.1007/978-1-349-13693-3. URL `http://link.springer.com/10.1007/978-1-349-13693-3`.

[342] Sherri Williams. # sayhername: Using digital activism to document violence against black women. *Feminist media studies*, 16(5):922–925, 2016.

[343] Jacob O Wobbrock and Julie A Kientz. Research contributions in human-computer interaction. *interactions*, 23(3):38–44, 2016.

[344] Christine T. Wolf. Explainability scenarios: towards scenario-based XAI design. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 252–257, Marina del Ray California, March 2019. ACM. ISBN 978-1-4503-6272-6. doi: 10.1145/3301275.3302317. URL `https://dl.acm.org/doi/10.1145/3301275.3302317`.

[345] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, *et al.* Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[346] Frank Wood, April Carrillo, and Elizabeth Monk-Turner. Visibly unknown: Media depiction of murdered transgender women of color. *Race and Justice*, page 2153368719886343, 2019.

[347] Minghao Wu, Fei Liu, and Trevor Cohn. Evaluating the utility of hand-crafted features in sequence labelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2850–2856, 2018.

[348] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel 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.

[349] Tongshuang Wu, Kanit Wongsuphasawat, Donghao Ren, Kayur Patel, and Chris DuBois. Tempura: Query analysis with structural templates. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[350] Yao Xie, Ge Gao, and Xiang 'Anthony' Chen. Outlining the design space of explainable intelligent systems for medical diagnosis, 2019.

[351] Yao Xie, Melody Chen, David Kao, Ge Gao, and Xiang'Anthony' Chen. Chexplain: Enabling physicians to explore and understand data-driven, ai-enabled medical imaging analysis. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[352] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 547–558, 2020.

[353] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. Re-examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–13, Honolulu, HI, USA, April 2020. Association for Computing Machinery. ISBN 978-1-4503-6708-0. doi: 10.1145/3313831.3376301. URL `https://doi.org/10.1145/3313831.3376301`.

[354] Jill Yielder. *Professional Expertise: A Model for Integration and Change*. Thesis, ResearchSpace@Auckland, 2001. URL `https://researchspace.auckland.ac.nz/handle/2292/2340`. Accepted: 2008-01-30T02:04:57Z.

[355] Jill Yielder. An integrated model of professional expertise and its implications for higher education. *International Journal of Lifelong Education*, 23(1):60–80, January 2004. ISSN 0260-1370, 1464-519X. doi: 10.1080/0260137032000172060. URL `http://www.tandfonline.com/doi/abs/10.1080/0260137032000172060`.

[356] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. Understanding the Effect of Accuracy on Trust in Machine Learning Models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI

'19, pages 1–12, Glasgow, Scotland Uk, May 2019. Association for Computing Machinery. ISBN 978-1-4503-5970-2. doi: 10.1145/3290605.3300509. URL https://doi.org/10.1145/3290605.3300509.

[357] Rulei Yu and Lei Shi. A user-based taxonomy for deep learning visualization. *Visual Informatics*, 2(3):147–154, September 2018. ISSN 2468502X. doi: 10.1016/j.visinf.2018.09.001. URL https://linkinghub.elsevier.com/retrieve/pii/S2468502X1830038X.

[358] Tal Z Zarsky. Transparent predictions. *U. Ill. L. Rev.*, page 1503, 2013.

[359] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8689, pages 818–833. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10589-5 978-3-319-10590-1. doi: 10.1007/978-3-319-10590-1_53. URL http://link.springer.com/10.1007/978-3-319-10590-1_53. Series Title: Lecture Notes in Computer Science.

[360] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

[361] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[362] Yunfeng Zhang, Q Vera Liao, and Rachel KE 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*, pages 295–305, 2020.

[363] Ruijing Zhao, Izak Benbasat, and Hasan Cavusoglu. Transparency in advice-giving systems: A framework and a research model for transparency provision. In *IUI Workshops*, 2019.

[364] Miri Zilka, Holli Sargeant, and Adrian Weller. Transparency, governance and regulation of algorithmic tools deployed in the criminal justice system: a UK case study. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '22. ACM, jul 2022. doi: 10.1145/3514094.3534200. URL https://doi.org/10.1145%2F3514094.3534200.

[365] Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. Deepred–rule extraction from deep neural networks. In *International Conference on Discovery Science*, pages 457–473. Springer, 2016.

[366] Muhammad Zubair, Jinsul Kim, and Changwoo Yoon. An automated ecg beat classification system using convolutional neural networks. In *2016 6th international conference on IT convergence and security (ICITCS)*, pages 1–5. IEEE, 2016.

[367] Ethan Zuckerman, J Nathan Matias, Rahul Bhargava, Fernando Bermejo, and Allan Ko. Whose death matters? a quantitative analysis of media attention to deaths of black americans in police confrontations, 2013–2016. *International Journal of Communication*, 13:27, 2019.